UNIVERSIDAD POLITÉCNICA DE MADRID

Escuela Técnica Superior de Ingenieros en Telecomunicación

Development methodology for semantic and distributed systems, with
emphasis in Ambient Assisted Living Domain

Tesis Doctoral

Alejandro Martín Medrano Gil
Ingeniero en Informática

2019

Departamento de Tecnología Fotónica

Escuela Técnica Superior de Ingenieros en Telecomunicación

Development methodology for semantic and distributed systems, with
emphasis in Ambient Assisted Living Domain

Autor:

Alejandro Martín Medrano Gil

Ingeniero en Informática

Director:

María Teresa Arredondo Waldmeyer

2019

Tribunal nombrado por el Magnífico y Excelentísimo Sr. Rector de la Universidad Politécnica de Madrid, el día .. de Octubre de 2019 .

Presidente	Javier Uceda	Catedrático de Universidad Doctor Ingeniero Industrial E.T.S.I. Industriales Universidad Politécnica de Madrid
Vocales	Carlos Enrique Palau Salvador	Catedrático de Universidad Doctor Ingeniero de Telecomunicación E.T.S.I. Telecomunicación Universitat Politècnica de València
	Jordi Aguiló Llobet	Catedrático de Universidad Doctor en Física Escuela de Ingeniería Universidad Autónoma de Barcelona
	Reiner Wichert	Managing Director Doctor en Ingeniería en Informática National Research Center for Applied Cybersecurity (CRISP) Technical University of Darmstadt
Secretaria	María Fernanda Cabrera Umpiérrez	Prof. Contratada Doctora Doctor Ingeniero Biomédico E.T.S.I. Telecomunicación Universidad Politécnica de Madrid
Suplentes	Alejandro Rodríguez Ascaso	Profesor contratado doctor Doctor Ingeniero de Telecomunicación E.T.S.I. Informática Universidad Nacional de Educación a Distancia
	Constantina Álvarez Peña	Profesor Titular de Escuela Universitaria Doctora en Ingeniería Escuela de Ingeniería Informática Universidad de Oviedo

Realizado el acto de defensa y lectura de la tesis el día .. de Noviembre de 2019 en Madrid, acuerda otorgarle la calificación de:

El Presidente

La Secretaria

Los Vocales

Abstract

Semantic technologies are used to share data, link data, and perform high level abstraction and reasoning over data. Distributed computer systems take advantage of networks to split the computing, storage, sensing, and networking workloads over different nodes. Semantic and Distributed systems are systems which combine both paradigms, enabling ubiquitous computing and semantic interoperability between heterogeneous components.

Ambient Assisted Living (AAL) is a technological paradigm where the user becomes the centre of the system, and the system is the environment the user lives in. AAL not only defines heterogeneous environments, services and technologies, it is also must address the complete spectrum of possible users. One particular area AAL is used extensively is in Active and Healthy Ageing, or the application of technologies to facilitate the ageing process of individuals, their relatives, as well as positively impacting society as a whole.

In this context a model for AAL oriented Semantic and Distributed systems is proposed, examining 9 simultaneous perspectives for the analysis and design of said systems considering all complex scenarios they may entail. In order to validate Semantic and Distributed Systems, from the multiple perspectives of the different stakeholders involved in AAL, as well as itemizing concrete technological values of the system a value oriented multi-dimension evaluation methodology is proposed. This methodology is used to determine and analyse the required values from AAL semantic and distributed platforms. A cocreation methodology is proposed for the overall development of semantic and distributed systems, combining both the proposed model and methodology.

The model validated by applying it to 3 distinct and unique AAL related concept projects, showing the extents the model could be used for. The AAL platform values are evaluated and analysed thoroughly through the participation of up to 60 individuals, covering all involved stakeholders. The model, and AAL platform values, are analysed to produce a set of improvements over existing AAL platform (universAAL IoT), this process is used as validation of the fitness of the proposed model and methodology; as well as the validation of the AAL platform itself. The Cocreation Methodology is reviewed for its application of 2 large scale projects, analysing the concrete experiences, results, and lessons learned in each one.

Resumen

Las tecnologías semánticas se utilizan para compartir datos, vincular datos y realizar abstracciones y razonamientos de alto nivel sobre los mismos datos. Los sistemas informáticos distribuidos aprovechan las redes de computadoras para dividir las cargas de trabajo de computación, almacenamiento, detección y red entre diferentes nodos. Los sistemas semánticos y distribuidos son sistemas que combinan ambos paradigmas, permitiendo la computación ubicua y la interoperabilidad semántica entre componentes heterogéneos.

Ambient Assisted Living (AAL) es un paradigma tecnológico en el que el usuario se convierte en el centro del sistema, y el sistema es el entorno en el que vive el usuario. AAL no solo define entornos, servicios y tecnologías heterogéneas, sino que también debe abordar el espectro completo de posibles usuarios. Un área particular que AAL se usa ampliamente es el Envejecimiento Activo y Saludable (AHA), o la aplicación de tecnologías para facilitar el proceso de envejecimiento con salud y bienestar de las personas, sus familiares produciendo un impacto positivo en la sociedad en su conjunto.

En este contexto, se propone un modelo para sistemas semánticos y distribuidos orientados a AAL, que examinan 9 perspectivas simultáneas para el análisis y diseño de dichos sistemas considerando todos los escenarios complejos que puedan originarse. Para validar los sistemas semánticos y distribuidos desde las múltiples perspectivas de los diferentes actores involucrados en AAL, así como para detallar los valores tecnológicos concretos del sistema, se propone una metodología de evaluación multidimensional orientada al valor. Esta metodología se utiliza para determinar y analizar los valores requeridos de las plataformas semánticas y distribuidas en AAL. Se propone una metodología de cocreación para el desarrollo general de sistemas semánticos y distribuidos, que combina el modelo propuesto y la metodología propuesta.

El modelo se validó aplicándolo a tres proyectos conceptuales relacionados con AAL, distintos y únicos, que muestran ejemplos de la extensión de aplicación del modelo. Los valores de la plataforma AAL se evalúan y analizan exhaustivamente a través de la participación de hasta 60 personas, que cubren a todos los roles involucrados en el AAL. El modelo y los valores de la plataforma AAL se analizan para producir un conjunto de mejoras sobre una plataforma AAL existente (universAAL IoT), este proceso se utiliza como validación de la idoneidad del modelo y la metodología propuestos; así como la validación de la plataforma AAL en sí. La Metodología de Creación se revisa para su aplicación en dos proyectos de despliegue a gran escala, analizando las experiencias concretas, los resultados y las lecciones aprendidas en cada uno.

Contents

1	Introduction	1
1.1	Background and Motivation	1
1.1.1	Socio-Economic Background	2
1.1.2	Ambient Assisted Living	3
1.1.3	Technological Background	4
1.2	Problem Statement	8
1.3	Research Questions	9
1.3.1	Semantic and Distributed System Model questions	9
1.3.2	Evaluation questions	10
1.3.3	Methodology Questions	11
1.3.4	Technology questions	12
1.4	Hypotheses	13
1.5	Objectives	14
2	State of the Art	17
2.1	Semantic System Models	17
2.1.1	Ontology design	17
2.1.2	Technological approach	19
2.2	User Interaction, Accessibility and User Experience for AAL	20
2.2.1	survey of existing generalized solutions	21
2.2.2	Ongoing Challenges	22
2.3	Distributed System Models	23
2.3.1	Distributed Services	23
2.3.2	Complex Event Processing	24
2.3.3	Cloud Model	25
2.3.4	Peer to Peer Model	26
2.3.5	Internet of Things	29
2.3.6	Edge Computing	30
2.4	Virtualization and Containerization	31
2.5	Security and Privacy	32
2.6	System Evaluation methodologies	34
2.7	System Development	35
2.7.1	Waterfall	35
2.7.2	Spiral	35
2.7.3	Agile	36

2.7.4	Continuous Integration	36
3	Methods and Materials	37
3.1	universAAL IoT	37
3.1.1	universAAL Terms	37
3.1.2	universAAL Architecture	39
3.2	Security Assessment, Impact and Mitigation Methodology	44
3.2.1	Asset identification and description	45
3.2.2	Threat identification: STRIDE	46
3.2.3	Threat impact assessment: DREAD	47
3.2.4	Risk Management and mitigation	49
3.3	Open Platform Ecosystem Assessment (OPEA) Framework	49
3.4	Overall Research Methodology	51
4	Results	54
4.1	Model for Semantic and Distributed Systems	54
4.1.1	Semantic Model	55
4.1.2	Context Model	60
4.1.3	Service Model	61
4.1.4	User Interaction Model	63
4.1.5	Member Model	66
4.1.6	Component model	67
4.1.7	Distribution Model	68
4.1.8	Deployment Model	73
4.1.9	Security and Privacy Model	77
4.2	Value Oriented Multi-Dimension Evaluation Methodology: The Show- case Methodology	81
4.2.1	Definition of the showcases and tool-set	81
4.2.2	Pre-self-evaluation of Showcases	83
4.2.3	Demonstration of the showcase	84
4.2.4	Video recording	84
4.2.5	Value Assessment	86
4.3	Cocreation Methodology	87
4.3.1	Scenario Analysis	89
4.3.2	Adaptation	91
4.3.3	Assessment	92
4.4	AAL platform Showcases	94
4.4.1	Resource Sharing	97
4.4.2	Plug and Play	103
4.4.3	Advanced Distribution	111
4.4.4	Scalability	116
4.4.5	Evolution	122
4.4.6	Administration	132
4.4.7	Integration with legacy systems	139

4.4.8	Services Integration	144
4.4.9	Security & Privacy	151
4.4.10	Service Transferability	156
4.4.11	Advanced User Interaction	163
4.4.12	Personalized Content	168
4.4.13	Ambient Intelligence	174
4.4.14	Enhanced Market communication and distribution	179
5	Validation and Analysis	184
5.1	Model Validation	184
5.1.1	Full Edge SDS implementation: Indoor Localization benchmarking tool	184
5.1.2	Hybrid Cloud-Edge: Passage	190
5.1.3	Federative Blockchain over P2P: H-Chain	195
5.2	Showcase evaluation	204
5.2.1	Demonstration of Showcases	206
5.2.2	Value of Showcases	212
5.2.3	Reflection on showcase evaluation	232
5.2.4	Conclusion: Key Value Propositions	232
5.3	Methodology Validation	240
5.3.1	The adaptation Process in ReAAL	240
5.3.2	ACTIVAGE Madrid Deployment site case	249
5.4	Bringing Technology beyond the state of the art	259
5.4.1	universAAL IoT enhancement with multi-cloud and Multi-tenancy	259
5.4.2	universAAL IoT enhancement with multi-platform capabilities, the REST API	263
5.4.3	universAAL IoT enhancement with semantic and distributed configuration framework	266
5.4.4	universAAL IoT enhancement with Automatic Semantic Representation	268
5.4.5	User Experience improvements under universAAL IoT	272
5.4.6	universAAL IoT improvements in Security and Privacy	282
6	Conclusions and Future Work	291
6.1	Discussion on Model for Semantic Distributed Systems	291
6.2	Discussion on Showcases	294
6.3	Discussion on Cocreation Methodology	297
6.4	Research Questions	299
6.4.1	Semantic and Distributed System Model questions	299
6.4.2	Evaluation questions	301
6.4.3	Methodology questions	302
6.4.4	Technology questions	303
6.5	Hypothesis Validation	304
6.6	Objectives Review	307

6.7 Future Work	308
Bibliography	310

CHAPTER 1

INTRODUCTION

Semantic and Distributed systems are systems that use both semantic information technologies, for the description and processing of information, as well as distributed communication technologies for propagation and parallelization of information and processes. The *Ambient Assisted Living (AAL)* is the domain of services and technologies applied to facilitating every day life, particularly of elderly people.

This introductory chapter will review the background and motivation for the work. Exploring the current socio economic problems of individual and population ageing, specific services arise to address these issues. One of the specific solutions addressing, among others, these problems is a paradigm called *Ambient Assisted Living (AAL)* for which the approach is embedding technology in living environments. The *Active and Health Ageing (AHA)* domain, is a more human centred approach under the same philosophy of helping people on their ageing process through technology and other methods. The commonly used technological approaches will be presented, as these technologies will be the basis of the proposed methodology of this work. The study will be driven by the motivation and through a set of research questions, formalized in Hypotheses and overall objectives.

1.1. BACKGROUND AND MOTIVATION

Technology has always been used to improve some human process, from the wheel to the computer, the need to more efficiently transport things, or calculate things, while lowering the effort required by the human operator is what is motivating humans to improve and develop technology.

One key human process is living, keeping the basic human body needs of being fed, clean, clothed, sheltered, and healthy in a secure way may not be as trivial as initially thought. Without these the body will rapidly deteriorate and die; therefore the most basic instincts are to accomplish these basic needs. However each of the necessities not only requires physical effort, time and planning are key part of it too. Humans have created complex social networks allowing for more efficient attainment of the basic necessities, at least more efficiently than one single individual could do on their own. Technology has been also an important part of this optimization

process by developing: machines, chemicals and manipulating biomass to grow food faster and more nutritious; water delivery and sewage systems to take water to where it is needed; machines and materials to create clothing more affordable and insulating; methods, materials, and machinery to make stronger, more resilient, and affordable buildings; knowledge to apply to the better fight against diseases.

In the 21st century humanity is still learning what we can do with the latest revolution on information and communication technologies. Why not apply these technologies to improve the life of everyone by making their basic needs fulfilled even more efficiently than before? Specifically why not apply these technologies to help those with higher risk of exclusion, those for who accomplishing basic needs, such as elderly and disabled people, requires more effort?

1.1.1. SOCIO-ECONOMIC BACKGROUND

These advancements, directly or indirectly, have produced the effect where populations around the world are rapidly ageing. Projections indicate that the older population (over 65 years) will grow from the current 11% up to 22% average in 2050, with regions ramping up to 40%. By 2050, 82 countries are projected to have at least 20% of their population over 65 years old, compared to 13 countries in 2018 [1]. Forecasts suggest that the EU Silver Economy is increasing approximately 5% per year and will reach the amount of 5.7 Trillion € in 2025 [2]. Population +80 will rise from 1% to 4% (400M) of the world's population by 2050 [3, 4]. In average, older people will live between 10-12 years of progressive physical and cognitive deterioration, with and without a chronic disease. Parallel to the ageing, there is systematic evidence that frailty syndrome, associated with physical, functional and cognitive decline, mental diseases like dementia and Alzheimer, as well as social isolation are health ailments that trigger more dependencies and comorbidities [5]. The impact of ageing is projected to be a significant challenge in almost all Member States.

Overall in the EU, the total cost of ageing (public spending on pensions, health care, long-term care, education and unemployment benefits), is expected to increase by 1.7 percentage points to 26.7% of GDP between 2016 and 2070. Long-term care and health care costs are expected to contribute the most to the rise in age-related spending [6, 7] and thus, it will have wide-ranging implications in the sustainability of the current social and health systems models. This has led many governments to search and invest in sustainable solutions that help slowing down the progression of these conditions allowing older people staying longer in smart living environments that evolve along with them as they age. The European Commission has invested over 12 Billion € [8, 9, 10] into research and innovation projects (e.g. FP7, AAL, Horizon 2020, and more), to respond to the ageing and health challenge through technology innovation in the sector (demand pull approach), as well as in innovation partnership promotion and support for Active and Healthy Ageing (AHA) enabling adoption and market scale up throughout the regions of Europe. Although there are currently 74 Reference Sites 3 running stable projects in Europe assisting an important older population, this market sector remains predominantly hyper-

fragmented, limiting the capacity of growth and rising cost containment [5, 6]. This situation demands that the barriers to innovation be removed. Only if this occurs will Europe’s industry leaders, entrepreneurs and SMEs drive economic growth.

1.1.2. AMBIENT ASSISTED LIVING

AAL is a technological paradigm where the user becomes the centre of the system, and the system is the environment the user lives in. This environment may be the home, the office, transportation vehicles and stations; public spaces such as streets or shopping centres. Each environment will have different capabilities depending on the embedded systems it has, and how these systems are interconnected. The challenge is then to offer services to the user in all these environments to enhance the users’ lives, where each individual user perceives her/himself as the center of the system.

The ambient can support many users, independently of their association with the system, their gender, their physical or mental capabilities, or age. Thus AAL not only defines heterogeneous environments, with heterogeneous services and technologies, it is also must address the complete spectrum of users. For each interaction between users and systems, the combination of context, services, technology, and user profile is unique, and thus the system becomes extremely complex to design and implement.

One approach is to centre the design on the most challenging users, solving the problem for these users will already cover most of the other users as well. Therefore AAL has been centred on the elderly and ageing population.

1.1.2.1. ACTIVE AND HEALTHY AGEING

Thanks to the advances and achievements in medical science and technologies, life expectancy, as well as quality of life have been increasing worldwide during the last decades; and it is expected to increment further in the near future [11]. However, the latest global economic crisis and the increased proportion of the ageing population are showing that current health and social care systems are not adequate to guarantee physical, mental and social well-being of their citizens, from childhood to late adulthood, the latter being the most challenging and most demanded. In this context, the World Health Organization has defined Active and Healthy Ageing (AHA) as a process that includes challenges but also opportunities for health maintenance and social involvement of senior citizens, to preserve their quality of life as they age [12, 13, 14].

In Europe, the European Commission (EC) launched the European Innovation Partnership (EIP) on AHA initiative in 2011 [15], to accelerate the uptake of innovative solutions from health and social care systems. This initiative aims to implement research and innovation involving all relevant actors at different levels, both European, national and regional, with the aim of extending the good health of Europeans in two years of life by 2020 [16]. Within the EIP-AHA initiative, a conceptual def-

inition of active and healthy ageing has been proposed, which includes different characteristics such as physical and cognitive functionality, well-being, and social participation of the citizenship. This conceptual framework has been proposed to allow comparisons of interventions, and other initiatives [12] carried out within the different European regions.

1.1.3. TECHNOLOGICAL BACKGROUND

The technological support these smart environments need depend fundamentally on the capability that these systems have to represent the world, as well as conveying this representation to other systems which are interoperating with them. The solution are semantic systems[17], because they have complex, yet interoperable information representation standards; which are also distributed, as they can offer the services through multiple and heterogeneous systems.

In the dawn of electronic computing, the systems where used as single purpose machines, and even though they used the same underlying principles, machines could not interoperate, exchanging information or processes was impossible. This trend kept going, even throughout the development of multi-purpose and personal computers, although certain standards started to appear, the industry was driven by closed formats, believing this would improve sales of the particular product. Even with the appearance of networking, Internet and the world wide web, where every computer could now connect potentially to the internet, allowing each of them to interoperate with any other; the industry is working out how the legacy philosophy, and systems, can now solve true interoperability problem.

1.1.3.1. SEMANTICS AND ONTOLOGIES

There are many solutions to the interoperability problem, ironically, not interoperable between them. In the hyperconnected world of today, services, webs, and apps need to talk seamlessly with one another, the industry is tending to use string based syntaxes rather than standardised binary representations. However even these syntaxes fall short many times, particularly when sharing data and services across different providers, services, and domains[18].

In the hyperconnected world where any piece of data can be generated and shared, as well as data about the data it self (e.g.: when was the data produced and by whom) called metadata; it might be not surprising at all to see how all data has value. Data is a renewable resource, data is all around and it is continuously being generated, in the context of data being an abstract concept, the data needs to be measured, collected; Example healthy people walk daily, and thus produce a number of steps daily, this number exists whether it is collected or not. In many cases, at least the most common, this is a low cost operation specially if there is already the infrastructure to collect data (for example a smartphone already has connectivity and sensors thus adding a pedometer can add a new stream of data to the already present streams of data from GPS, gyroscope and other sensors). Once

data is gathered is has some value, generally to the entity who has gathered this data, for example a person may use a pedometer to gather the steps (s)he makes daily, the value of this data is for the person. Potential value can arise from data, data by itself is valuable; in the example, the user now knows how many steps (s)he does daily, and intuitively knows how many they should make. Other values can be added to data, services can emerge from the presence of data; in the previous example a service to motivate the user to comply with their objectives. Data can be shared and thus create marketplaces for data; following the example the user might want to share her/his data with some entity, in exchange they get better estimates for the daily steps (s)he has to do. Data can be computed, more value can arise from merging and comparing data in what is now known as Big Data; In the example above the estimates can be even more precise when correlated with age, and Body Mass Index (BMI). At this level, even more services can be added, increasing further the value; in the example now new forms of motivational service can be offered like ranking globally or locally or against age and BMI using all the aggregated data from many users. Finally at this stage, data has an impact value, in the example users collecting automatically their walking habits and improving them has an impact on their health, this impact scaled by the number of users, means that there is also a societal impact on the healthcare system.

Data needs to have a structure, this structure is what it gives the numeric values any meaning, thus standardising the meta-model providing structure to data is a good idea, this way not only data, but the structure of the data (and metadata) can be shared and potentially made interoperable. This is where the Semantic Web[19] makes its appearance. The semantic web is a concept where the data is linked into a web of data, self-contained, and self-explanatory; sometimes referred as the Web 3.0. The semantic web is also an extension of the World Wide Web standards by the World Wide Web Consortium (W3C) promoting standards, data formats, exchange protocols, and a whole stack to enable data to be shared and reused across application, provider, community and domain borders.

The most notable standard in the semantic web is the Resource Description Framework (RDF)[20]. RDF is a simple language describing data structures with the resource, as in a web object, as the centre concept. RDF is also used to describe complex data in form of graphs of linked data following the structures defined.

However RDF is not the only standard in the semantic web stack, there are many other which are used. For example Uniform Resource Identifier (URI), is a standard schema for referral of resources, thus it is used as pointers in RDF. The essential text encoding standard used is unicode or UTF8 as the framework needs to be robust enough to encode characters in any human language regardless of the international script used. International Resource Identifier (IRI) is an URI allowing for the encode of unicode characters, thus making it international.

RDF draws many concepts from XML (eXtensible Markup Language), however XML is more of a serialization standard, this is how to encode the RDF graphs into a text based exchange format. There are other serialization standards for RDF such as Turtle, N-triples, N-quads, and JSON-LD.

Figure 1.1: The semantic web stack (Source: https://en.wikipedia.org/wiki/Semantic_Web_Stack).

On top of RDF further structures can be defined, most prominent is RDF Schema (RDFS). This vocabulary extends RDF to allow the definition and description of Classes, Properties for RDF resources to follow. Simple restrictions such as sub-class or sub-property allow the description of hierarchical semantic knowledge bases.

The Semantic Web needs a querying language, this is standardised in SPARQL. SPARQL characterizes for being reminiscent of SQL, but the queries are at the same time human readable and self explanatory; this partly thanks the structures defined in OWL, RDFS and other taxonomies. With SPARQL simple questions, listings, or complete RDF graphs can be retrieved from triple stores; and even it has distributed powers as the engines (in different machines) could potentially connect to solve a requested query.

The Ontology Web Language (OWL) is a further extension of RDF and RDFS, with more precise definitions. Some examples include the concept of cardinality, defining how many values can a property have; class relations (such as disjointness), property definitions (such as transitivity). OWL definitions draw heavily from Set theory and other mathematical fields allowing it to be coherent and complete.

Up to this point standards are used to describe data structures, however data may have embedded logic. To describe first order logic on data structures and data instances Rule Languages can be defined on top of RDF, RDFS, and even OWL.

The stack is incomplete, the upper layers, as the writing of this document, do

not have well accepted standards. These layers would be comprised of components such as reasoning engines, which provide some unifying logic to the data.

1.1.3.2. INTERNET OF THINGS

The internet of things (IoT) is a system of interrelated “things” that can be assigned with unique identifiers and have the ability to transfer data over a network autonomously[21]. A “thing” is any physical object, or organism, which is in it self, or is enhanced by a digital system enabling it with processing, networking, and optionally, storing, sensing and actuating capabilities. In the IoT each “thing” can provide, or consume information and services from one another, in the same way humans use internet to provide and consume services.

IoT is by definition a complex distributed system, from the definition, design and implementation of the nodes enabling “things” to connect to internet, to the development and provision of networks capable of connecting large number of devices, providing adequate communications to each of them. IoT follows a similar approach as any other system based in internet. There can be client and servers, in this context the cloud concept has a place in IoT. In fact “things” may be capable to connect directly to cloud systems, or maybe they need an intermediary such as a gateway, which manages local connection to the devices on one side, and internet connectivity on the other. Other distribution models are acceptable in IoT, such as Peer 2 Peer (P2P) backed by Blockchain.

The paradigm is also moving towards semantic technologies. The need to manage highly heterogeneous environment (i.e.: each node in the IoT can potentially be from a different manufacturer, or have different configuration), as well as providing solution to manage the wealth of data that is being captured processed and stored. Ontologies such as SSN[22] and SOSA[23] are examples of how semantic technologies can provide some standardization of data and processes, like data acquisition, to be able to share across systems and domains.

Even though these standards exists there is currently an explosion of IoT platforms claiming to be the solution for IoT not using any semantics at all. The climate in the industry is that of a gladiator match to see who will “get it all”, which platform will be the new “TCP/IP” stack for IoT, the others must just die. This is not going to happen, none of the IoT platforms provide a complete solution, nor they need to. When these platforms use semantic technologies, they can exchange data and functionality with other platforms, not competing but cooperating, taking advantage of the strengths of each other.

In the AAL domain IoT provides innovative solutions to promote more efficient approaches; it has also gained a great momentum as a key enabling technology for a wide range of applications and particularly help older adults live independent lives [24]. The European Commission aims at demonstrating that IoT is a key enabler for the successful deployment of Smart Living solutions; positively impacting Active and Healthy Ageing (AHA) [25].

Some of IoT’s main characteristics such as Ubiquitous Connectivity, Pervasive

Computing, and Ambient Intelligence are all integral in AAL environments. Services like monitoring health, delivering wellness interventions, or providing alert and information notifications are all examples of how IoT is involved in the promotion of AAL [26]. However IoT in AAL specifically addresses particular challenges, concretely individual and environmental factors in both indoor and outdoor environments. Individual factors are those that contribute to prevent cognitive and physical decline [27]. Environmental factors are those related to daily living activities, such as housing [28], transport [29], mobility [30], or leisure [31].

1.2. PROBLEM STATEMENT

So far it has been clear that there is in deed a need for technology interventions to mitigate the social and personal effects of the ageing of the population. The main problem is related with the technology developed for this purpose. On one side it needs to be distributed, as in the age of internet we have the need to connect our devices, data and services to be consumed in an ubiquitous manner. On the other it needs a common language to make the natively complex domain and environment understandable, not only for the involved systems but also for the humans involved in developing, installing, maintaining and using these systems.

Whenever designing or analysing an ICT system there are common topics to address, these topics may vary in function to the paradigms and concrete technologies used. In this context an abstract model of the technology is needed to help address the individual challenges, not only of distribution and semantics on their own, but when combined. Is precisely in the combination of both paradigms where the model is most needed, as it is adding complexity to the system and prone to confusion and thus inefficiencies on the design process which can later show in the performance of the system it self.

AAL in it self is a challenging field, from an engineering perspective the system needs to be extremely reliable, and provide the desired effects, both individually and socially. Following clinical practices, Randomized Control trials might be an option to validate technologies used in health and AAL in general, however these might not be feasible, in which case the alternative might be observational studies, regardless of the strategy it is important to consider the study design[32]. The technology validating studies in this area need to carefully compensate for all these variables, as well as other hidden variables in the experiment. As there is always the good intention of showing how the system has a positive impact on the lives of the population, but in most cases this impact may not be directly traced to the system it self, unrelated factors such as better than expected surgery intervention may actually be the main contributors. The only way to control for these hidden variables is to either improve the contextual information in the research (by for example adding more questions to standardized questionnaires), or to largely increase the number of participants in the study, both of which may not be feasible, or atleast to the extend required to positively remove undesired variables from the experiment. This is why a validation

methodology is needed, a methodology which focuses on the technology usage and value it provides to the users, rather than the effects the users may perceive. This approach will be limited in the purpose of demonstrating the usefulness of the technology, specially in terms of quality of life and other societal factors, however it will be useful to itemize the concrete factors helping the technology achieve these grand objectives.

Overall development of SDS for AAL is a complex endeavour, not only technologically is challenging, validating is also challenging, additionally the key aspect of AAL service is the diversity and number of actors involved in the system, all of which need to engage with it in order to properly provide its utility. In order to improve the acceptance, and usefulness of the system a development methodology is needed. This methodology should consider user needs, as well as other stakeholders involved in the delivery of care and related services. The resulting system will then perform as close to expected by all stakeholders. Although the use of this methodology does not assure these results, it needs to provide all the tools and principles to follow in order to do so. The previous models and methodologies will be an integral part of the development methodology, analysing designing and evaluating the end results.

A final problem to be addressed is the lack of existing technologies, or rather functionalities which existing technologies miss in order to fully provide all the flexibility proposed in the model vision of SDS for AAL. Even though there are many semantic technologies, and even more distribution platforms helping in the development of ICT systems, there are very little technologies addressing both areas, and even these need to increase in functionality in order to provide the required value expected from a platform enabling the development of SDS AAL applications and services.

1.3. RESEARCH QUESTIONS

As any good research, this work will be guided by a set of questions which define the main hypothesis and objectives of the research. Following are the guiding research questions categorized into topics.

1.3.1. SEMANTIC AND DISTRIBUTED SYSTEM MODEL QUESTIONS

SDS are quite complex, there are many dimensions to the problem from the data models, to the security employed. A common model to analyse and design SDS is needed, and this model must be capable of providing solutions to resolve:

1. How the semantics are used?
2. What resources are going to be represented?
3. What services are going to be provided and used?

4. How it is going to interact with the users?
5. How its components interact?
6. How are its components built?
7. What are the distributed resources, and how do they interconnect?
8. How are its components installed and configured throughout the nodes?
9. How the whole system secures and protects data, itself and its stakeholders?

Can this model be used in purely Edge Computing environments?

Given all the complexity of SDS and the model for them, is it possible to model a purely Edge computing system. validation of the model in closed environments such as that of Edge computing is required, because Edge computing is one of the central paradigms of SDS in AAL.

Can this model be used for IoT, hybrid Edge-Cloud environments?

The model should also be able to model IoT based systems, where the system contains interconnected and smart devices, local smart environments and cloud support all coordinated to offer AAL services. The model will have to be validated against IoT environments as IoT will be increasingly relevant to AAL domain.

Can this model be used in Blockchain based environments?

The model should also be robust enough to analyse and develop blockchain based systems. The semantic web is readily prepared for P2P direct sharing of linked information and services, but to do so securely a form of distributed ledger needs to be used. This concept could help develop new innovative AAL and health services disrupting any current paradigm of system implementation for these domains.

1.3.2. EVALUATION QUESTIONS

Evaluating ICT systems is not trivial, because there need to be a common framework for this evaluation. This framework is not always possible, particularly with SDS where even though there are ontology validation methodologies, and distribution metrics, These may vary according to the objectives of the system it self.

System should also be validated not only from the technical perspective, but from the impact perspective. In this case one solution might be to perform pilot runs guided by a scientific protocol, typically employing control population and a population with intervention (i.e: people using AAL solutions). This method is slow expensive, and also falls in the same category as for technological validations, results from different systems can not be directly compared. Additionally it is very difficult to design and implement a truly scientific experiment, as there are too many

unknown variables, particularly in the field of AAL we cannot know if for example the users have reduced their visits to the doctor because of the system, or because other external factors, even compensating with control population and with pre- and post-evaluation data.

Thus an evaluation methodology is proposed to look at the impact, concretely from the value perspective, which would account for all stakeholders' value interests, as well as from a multi-disciplinary approach linking technology and impact. This methodology, inspired by the semantic web, promotes the definition of generic study cases which can be applied to different domains, and solutions at different levels (i.e: platform or application level), and still throw easily comparable results.

What is the technical value and impact of AAL Semantic and Distributed System platforms?

Given the proposed methodology, the interest is to determine the main study cases for extracting value from AAL platforms. These study cases will be examined, and the methodology conducted, to provide tools to detect and highlight individual value propositions from each of the study cases. All the value will be analysed from the perspective of different stakeholders, as well as from technical and economical disciplines.

1.3.3. METHODOLOGY QUESTIONS

A general methodology specific for AAL Semantic and Distributed Systems is needed. This methodology needs to be able to account for user needs as well as all the stakeholders involved in the system. Particularly because in AAL domain the technological solutions are often supported or supporting customer services generally involving human resources. Because there are also many existing systems, not only in the field of AAL, but in other relevant domains, the methodology emphasises the need to reuse existing component, thus the focus is placed on integration and interoperability.

Can this methodology be applied to the development of any generic AAL SDS?

The methodology will be applied in a large scale pilot project devoted to create standard AAL services and applications within an open AAL platform. This execution will be reported in general, not for each individual service. Regardless it will report on the differences found for each of the services, helping develop guidelines and lessons learned for future adaptations.

Can this methodology be applied to the development of IoT based AHA solutions?

The general methodology will be applied to another large scale pilot project, this time focusing on a local ecosystem of services based on IoT philosophy. This analysis will focus on the execution of the methodology in concrete scenarios, validating also how it can fit for different purposes and backgrounds.

1.3.4. TECHNOLOGY QUESTIONS

Semantic and distributed systems is in it self a technological paradigm, composed of many intricate technological advancements, methodologies and materials. However there are particular questions which need to be addressed in order to ensure the proposed model (see 1.3.1) has the underlying technological infrastructure.

How can heterogeneous systems interconnect including all the associated semantics of their data collection, processing and displaying?

Interoperability is the key enabler for any system connected to the internet. Enabling interoperability could be as simple as following simple syntactic rules to offer and consume information and services. However when the syntaxes fall short, for example interoperating between different syntax languages and structures, then semantic interoperability is needed. Enabling semantic interoperability through heterogeneous systems needs to combine the best of both worlds: simplicity and robustness.

How can semantic technologies be enhanced by advanced cloud distributed systems, maintaining privacy between the different user side sub-networks?

Cloud systems are ruling the technosphere, everything from data storage, to data analytics and even the most trivial services reside in cloud operators. However, most of these systems do not interoperate, there might be no need to do so, there is economic and organizational culture explicitly avoiding it. In fact there are also strong technological reasons to keep systems and services isolated: the elevated complexity of dealing with data sets which need to be kept isolated. Data sets from one user, must not interfere with data sets or even services provided to another user. These systems can interoperate, not only enabling multi-cloud operations, where each user can spark the cooperation of multiple clouds to cooperate to resolve the user's request; but also maintaining the users' privacy through the employment of multi-tenancy mechanisms, improving the performance of the systems involved without leaking information from one user to another.

How can AAL semantic and distributed systems include advanced user interaction mechanisms?

AAL users may not be technology savvy, yet they need to access information and services through their smart environment and devices. User interaction is crucial for users to trust and understand the system, otherwise they will reject it, and thus the system objectives will be nullified. A robust user interaction mechanism needs to be implemented, not only capable of understanding the semantics of the content, but being able to adapt to contextual situations, distribute the interaction across different interfaces in the smart environments, using different channels to best approach the user and fulfil the interaction needs.

How can legacy systems be more easily connected with semantic systems? how can non-semantic systems be enhanced with semantic capabilities?

Users demand more complex services, as well as the capability of creating their own services from existing services. For example creating custom supporting systems, which integrate with the already contracted, or owned services, producing better tailored behaviours without the need of adding any redundant subsystem. Because current systems are not semantic in nature, this poses a challenge for which semantic systems would otherwise be almost trivial.

How can AAL semantic and distributed systems include strong cybersecurity and privacy protection mechanisms?

AAL deals with many data, particularly sensitive data such as medical information, or transportation habits, in the wrong hands this information can be used to target the user in terrible malicious ways. Thus it is imperative for SDS to include robust implementation of cybersecurity and privacy protection mechanisms. Any entity (user, software or hardware component) accessing the system should be identified and authenticated in a trusted manner. The system should enforce that any asset, whether it is data, services, or physical devices, can only be accessed in the appropriate manner by authorised entities, and that we can trust the the authorisation of these entities. Any accessed asset should not be corrupted, any authorised access should trust that the asset state is not be compromised. Entities accessing the assets should be always informed about the security operations taking place, so that no unauthorised transaction is made without the explicit consent of the entity. All communications (including disk operations) must be cyphered, so that unauthorised entities cannot access the assets in transit or at rest; yet the assets should always be available to the authorised entities. Finally all cybersecurity operations should be logged as to detect or identify possible misuse or failure of the security mechanisms. These are the common security properties known as: Authentication, Authorisation, Integrity, Non-repudiation, Confidentiality, Availability, and Audit.

1.4. HYPOTHESES

This section will enunciate the hypotheses, formalising the research to be carried out through this document.

Hypothesis 1: *Combining distributed systems and semantic systems combine the benefits as well as the complexity of both domains.*

Both Distributed systems and Semantic systems are burdened with many challenges on their own. SDS will face the combination of both, in addition to the complexity of merging them. Determining whether the complexity is compounded not only justifies the need for an explicit methodology for SDS, but also will highlight the challenges to be addressed.

Hypothesis 2: *A model for SDS, contemplating all the dimensions of the complexity within, can be formalised and validated against different concept SDS covering the majority of cases in AAL.*

A model will be developed to help analyse and design existing and new SDS. The model will address the most critical aspects of any SDS, providing methodological approaches for said modelling, as well as insight into potential solutions and patterns. The model will be tested against advanced concept SDS in the AAL domain.

Hypothesis 3: *A multi-dimensional approach evaluation methodology for Semantic and Distributed systems can be formalised, and validated against AAL domain SDS.*

Existing methodologies used to determine the value of SDS are do not take a multi-dimensional approach. A multidimensional approach to technological validation methodology will be proposed, the methodology will focus on the technological value for the different stakeholders. Showing the results can be comparable between SDS validations.

Hypothesis 4: *A unified methodology to properly develop Semantic and Distributed Systems for AAL, which is valuable for all stakeholders involved can be formalised, and validated against real large scale piloted SDS.*

A complete methodology from conception, through development, to evaluation will be proposed. This methodology will include the needs from the users as well as the assets the different stakeholders will contribute, and their benefits. The methodology will profit from the model described in Hypothesis 2, to be able to enlarge existing systems by including semantics and distribution in a cohesive way. The methodology will also promote the usage of the methodology described in Hypothesis 3 to aide in the assessment of the value the solution has for the different stakeholders, and drive the next iterations of the process.

Hypothesis 5: *The model for SDS can be used to complete existing Platforms offering all the advantages in a reusable way.*

Current SDS and supporting platforms do not provide a complete solution for providing all the features needed by the complete model in Hypothesis 2. The most advanced AAL SDS Platform will be brought beyond the state of the art by providing needed implementations of specific components providing the missing features.

1.5. OBJECTIVES

The stated hypothesis can be transformed into objectives, which direct the vision, mission and value of this whole document and, of course, the work it portrays.

Objective 1: To determine the challenges for Semantic and Distributed systems, in particular for AAL domain, and address them technologically.

The first milestone to achieve is to examine the questions related to the current limitations of SDS not only from technical perspective but from other important viewpoints for a large scale uptake of the paradigm, such as economical aspects, as well as educational. Determining the challenges of SDS will help define better models, address the missing components, create more reliable evaluation methodologies, and generally create a methodology which takes not only the advantages but leverages the impediments.

Objective 2: To develop, a model for analysing, adapting and implementing Semantic and Distributed Systems viewed from the different dimensions of the system.

A model will be proposed to be able to identify the different subsystems of SDS, analyse their place and function according the different dimensions of the problem. The same model can be then used to iterate and propose new subsystems, as well as identify the required subsystems and functionalities of the overall system.

Sub-objective 2.1: To validate the Semantic and Distributed System proposed model against three advanced projects.

The model to analyse and develop SDS must be validated, to do this three different projects will be modelled using the model. These projects will emphasise different parts of the model, covering overall the whole model casuistic. Because on one side a project which highlights all dimensions might be too complex, on the other the more projects the model is tested on, the better validation of the applicability of the model to the diversity of SDS in AAL. The selected projects will represent different scenarios and objectives within the AAL domain.

Objective 3: To develop a generalized multi-dimension methodology to determine the value for Semantic and Distributed Systems.

Scientifically the methods to evaluate information systems are limited to the technical prowess the particular software or hardware system has. Another approach is to interview users to gauge their perceived value from the system, this method often throws good results due to the participants willingness, and good faith. Finally another approach, particularly in eHealth and AAL domains, is to define an experiment with a control group which does not use the technology and the experimental group that does. This method appears to be the most scientifically, however it only validates the results obtained, and it must be very carefully designed to avoid any biases, it also has to involve many users to assure repeatability, however even this might be undermined by the fact that each person is unique and the system may not have been involved in the quality of life at all.

Thus a generalised value centred evaluation methodology needs to be provided, not only linking the technical capabilities but also analysing what these capabilities mean to the different stakeholders in the system.

Sub-objective 3.1: To determine de potential value for Semantic and Distributed Systems in AAL and analyse roadmap to market.

Applying the aforementioned methodology is important to validate it, and showcase how it works. This methodology will be applied to SDS open platform for the AAL domain, this will highlight also values for SDS in general. This analysis will map the appreciation for each potential value for different stakeholders in the AAL domain such as Service Providers, Technology Providers, Lawmakers, and End Users. The results will report which concrete values might be key, overall for all stakeholders, which will in turn determine where the efforts must be invested, not only in SDS platforms, but also in AAL services.

Objective 4: To develop a methodology for analysing, adapting and assessing Semantic and Distributed systems focused on AAL.

As the title of this document implies, one of the main objectives is to create a methodological approach to the development of SDS. As such the proposed methodology will include all phases of development, from ideation to validation; trying always to profit from existing and established systems. One component the methodology must include is the model for SDSs, encompassing all aspects of the SDS to develop as well as examining aspects which may be more challenging to account for, particularly in the AAL domain. The methodology should emphasise the challenges in the AAL domain, particularly the need to address user needs, as well as incorporate all relevant stakeholders in all the phases of the development.

Sub-objective 4.1: To demonstrate the development methodology on concrete Semantic and Distributed Systems in the AAL domain.

Different propositions will be suggested, these are SDSs systems in the AAL domain which will be modelled using the proposed methodology. To validate the model it is important that the propositions involve different aspects and architectures as to cover the whole model. Thus SDS projects will be analysed in the areas of edge computing, hybrid edge-cloud and blockchain, all in the AAL and eHealth domains. An additional example of the full methodology will be provided for cloud oriented SDS in the AHA domain.

Objective 5: To improve the state of the art of Semantic and Distributed AAL technology in order to fit the proposed model.

Given the model, and following the methodology, there are existing SDS technologies or platforms which provide most of the functionalities needed to implement any SDS, however these platforms, will need to be improved to include all the functionalities declared by the model. These implementations will be presented as validation for the proposed model, explaining why they are needed and how it fits in the overall picture portrayed by the proposed model.

CHAPTER 2

STATE OF THE ART

Proposing new methodological approach to the development of SDS in the AAL domain is not a trivial nor isolated endeavour. When designing methodologies and methods, it is always recommended to address the current issues, through the existing efforts in the scientific community. This section will review existing models and methodologies on semantic and distributed systems as well as evaluation and software development methodologies.

2.1. SEMANTIC SYSTEM MODELS

Extending the semantic system introduction provided in section 1.1.3.1, we will need to explore the state of the art in regards to both the methodologies to design ontologies and semantic models, as well as the available technologies for the implementation for the semantic web.

2.1.1. ONTOLOGY DESIGN

Once the modelling language is understood, the next challenge is to model a particular domain using said language. This is not trivial, the quality and usefulness of a model may vary[33]. Thus there is a research area devoted to find the best methodologies, strategies, algorithms and philosophies to approach ontological development[34].

There is many research into the problem of modelling a domain. As such there are many different approaches to modelling, one example is the usage of ontology design patterns[35, 36]. There are many patterns all described through a Generic Use Case (GUC), this GUC attempts to answer recurrent questions about the target domain and its subcomponents, such as:

- Who does *what*, *when* and *where*?
- Which objects *take part in* a certain event?
- What are the *parts* of something?

- What is an object *made of*?
- What is the *place* of something?
- What is the *time* frame of something?
- What *technique, method, practice* is being used?
- Which *tasks* should be executed in order to achieve a certain goal?
- Does this behaviour *conform* to a certain rule?
- What is the *function* of that artefact?
- How is that object *built*?
- What is the *design* of that artefact?
- How did that phenomenon *happen*?
- What is your *role* in that transaction?
- What that information *is about*?
- How is it *realized*?
- What is the degree of confidence that you give to this axiom?

The design patterns are collected in the ontology design pattern wiki page, an initiative of the European research project NeOn[37]. The design patterns are classified into general types, even in this form one can already approach the design problem methodologically.

Structural Defining architectural or logical structures within the target domain. Identifying classes of concepts and their structural relation between them, such as being part of, or made from.

Correspondence Defining alignments of concepts through equivalences, disjoints, or other description language operations. This allows the later re-engineering of the model, or the inclusion of other domains through referral as well as adding complex relations.

Content Identifying and defining content, container relationships between classes of concepts. Typical relations of *is_part_of* or *contains* may be found here, but more subtle content relations may be identified, such as what goes where.

Reasoning Defining logical rules and behaviours within the target domain.

Presentation Defining how the ontology itself, or the particular concepts would be presented. This covers naming conventions and the use of annotations for proper human interpretation.

Lexico-syntactic Identifying vocabularies and relations of natural language syntax. These vocabularies may form taxonomies for later generalization and use for description of other concepts in the domain.

Ontology design may also be validated, in this regard there are also many quality indicators and methodologies to evaluate how well the resulting ontology models the domain[38, 39, 40].

2.1.2. TECHNOLOGICAL APPROACH

As for technological frameworks, there are many capable of representing RDF and other Semantic standards such as OWL on many syntactic representations (as per standard). There are even some extensible frameworks which allow for inference and even semantic rule support.

Because the field of semantic web is relatively young, and because it is relatively unused; but most importantly because it promotes the use of standards, there are not as many alternative implementations as there are for user interface design and implementation[41].

OWLAPI ¹ a Java API and reference implementation for creating, manipulating and serialising OWL Ontologies. It can parse RDF/XML, OWL/XML, OWL functional Syntax, Turtle, KRSS, OBO, RIO and JSON-LD. There are many pluggable and embedded reasoners.

Apache Jena ² A free and open source Java framework for building Semantic Web and Linked Data applications. Offers RDF graph manipulation, a triple store, and OWL semantic reasoning.

Eclipse RDF4J ³ an open source Java framework for processing RDF data[42]. This includes parsing, storing, inferencing and querying of/over such data.

librdf ⁴ a set of free software C libraries to provide support for RDF and through Bindings it can also be used in PHP, Perl, Python and Ruby; includes RDF graph manipulation, alternative graph storage and SPARQL querying, several parsers such as RDF/XML, N-triples and Turtle.

rdf-ext ⁵ RDF library for NodeJS and the Browsers' javascript engine; includes JSON-LD, RDF/XML and Turtle parsers, and many alternative store implementations.

¹<http://owlcs.github.io/owlapi/>

²<https://jena.apache.org/>

³<https://rdf4j.eclipse.org/>

⁴<http://librdf.org/>

⁵<https://github.com/bergos/rdf-ext>

2.2. USER INTERACTION, ACCESSIBILITY AND USER EXPERIENCE FOR AAL

User Interface (UI) design is a complex multidisciplinary activity. It deals with interrelated aspects, which need to be very carefully thought out including, interaction, navigation, notification considering the ICT behaviour in general. Poorly planned UI can lead to inefficient resource consumption which in turn can lead to reduction of user quality-of-experience (QoE).

Elderly and/or disabled people, are particularly vulnerable to inefficient UIs. This demographic has the same need to access information as everybody else (main topic under the e-Inclusion area of research), however they have different system requirements, including UI design.

Many of existing interfaces does not adequately fulfil accessibility requirements of users suffering from visual, hearing or any other impairment; and in particular elderly users, who in addition may have cultural and cognitive barriers towards technology. In fact it is rather common for individuals to suffer from multiple impairment[43], this fact adds complexity to UI design in AAL.

One of solutions is to use multimodal interfaces that can address many of the target group requirements in a modular fashion. According to Oviatt [44], "multimodal interfaces process two or more combined user input modes (such as speech, pen, touch, manual gesture, gaze, and head and body movements) in a coordinated manner with multimedia system output". Complementary means of interaction employing different communication channels between a user and an AAL system greatly improves system's usability, while covering any particular accessibility case.

The interaction with technology has been evolving, as illustrated on figure 2.1, from *one-to-one* towards *many-to-many* interaction relationship, resulting in the apex of the IoT era where multiple users and multitude of devices interact in a dynamically changing environments. In this context multimodality has become increasingly important enabled by technological advancements in interaction possibilities. Multimodal UIs [45] comprising different complementing types of input methods (e.g. natural language and gestures) support the trend of ubiquitous interaction and e-Accessibility very well.

UIs are one of key factors that enable elderly users, users with disabilities, or even any user, to accept technological aid offered to them. Therefore, UIs must comply with following requirements:

Effectiveness accuracy and completeness of interaction.

Efficiency user can finish a task with a minimum effort.

Understandability provides information in a format easy understandable for users.

Learnability, intuitive speeds up a learning process and reduce training effort time.

Figure 2.1: Evolution of the interaction over the years

Unambiguity avoiding so-called “human errors” because users do not have to deal with inconsistencies or ambiguities.

Acceptance user attitude towards a used ICT system has a high influence on.

Sales well-designed UI can bring a competitive edge to an ICT system.

2.2.1. SURVEY OF EXISTING GENERALIZED SOLUTIONS

A survey of user interaction platforms, frameworks, and projects has been performed[46]. Since considered solutions had different scopes and goals, their architectures and user interaction frameworks were different but always focused towards responding to end user needs as much as possible. The main elements this study has discovered across these solutions are:

pluggability The idea that UIs should be easily changeable (pluggable) without (or with minimal) changes in the ICT system application layer[47]. It is not supported by most of the platforms, separation of presentation and application logic is a precondition for pluggable UIs, and it was addressed only in a small number of solutions.

UI Model In order to achieve good UI adaptation underlying user models and contextual awareness are essential. Some solutions implemented the model using semantic technologies. The key models to be considered are the user, the environment of the interaction, and the System itself. These models have been extensively used, and are now deep wells of data for any ICT system, furthermore this has created new opportunities in the field of interaction.

Adaptation Addressing heterogeneous demographic of target users requires some adaptation of the solutions. The adaptation is divided into 4 stages: initiation, proposal, decision, and execution. A system which performs the four stages autonomously is a self-adaptive system. Having in mind that targeted users are

often very heterogeneous (e.g., have different demographic, cultural, knowledge backgrounds, as well as desires, preferences, impairments or needs) it is clear that *one-user-interface-fits-all* is very difficult, if not impossible, to achieve. A degree of adaptation depends on a number of adaptation stages performed by a user.

2.2.2. ONGOING CHALLENGES

Designing generic solutions to create proper user interfaces, is still an open challenge, in particular designing for elderly and disabled users is a big challenge. Amongst the main factors that need to be carefully considered are age, gender and education, as well as social and cultural background. However there are other technological challenges which are not yet definitely addressed by any solution.

Support for multimodality is one of the key requirements for ubiquitous home systems. Once properly addressed application developers will be relieved from dealing with different end users, devices and different interfaces.

Intelligent UIs need to deal with the increasing amount of information arriving from ever increasing number of applications. One of the key principles when dealing with artificial intelligence, is that the user should always be in control, the system must be predictable and transparent. Yet it can still adapt and learn how to best meet user's expectations.

User modelling for UI adaptation are powerful tools to help make systems more accessible[48]. Since the target user group requirements change in time, the system needs to continuously and dynamically adapt.

Standardization and guidelines provide the foundation for widespread usage of all these ideas. There are quite a few studies that deal with older population and UIs [49, 50]. Since a standard approach for interaction for elderly is not widely applied, acceptance and use of standards may be more important rather than their availability.

Trust and acceptability . There are ethical as well as trust concerns, that need to be addressed properly. Fear of technology can lead to rejection of new solutions. Therefore, trust is of crucial importance for acceptance of such solutions.

In this context it is important to mention initiatives such as the Global Public Inclusive Infrastructure[51] (GPII). GPII is an infrastructure helping end users identify their solutions, particularly those solutions applicable for the type, degree or combination of their disability. These solutions share a common cloud based service, enabling them to self configure to the user's profile. GPII also provides support for developers to build adaptive solutions through the centralization of information,

components, and tools. GPII is addressing the above challenges through a standardization process of the user models such as User Profile Management (ETSI EG 202 325) and User Profile Preferences and Information (ETSI ES 202 746).

2.3. DISTRIBUTED SYSTEM MODELS

Since the dawn of computer networking there is always the drive to distribute the computing, storage, and networking loads. Thus since the very early innovators there has been plenty of research on this domain. This section will not attempt to cover the whole wealth of knowledge and know-how, but instead will look into the different initiatives and concepts regarding computing distribution which will be relevant for the discourse of this document.

A worthy notice is the distinction between distribution and parallelization. Parallelization is the process of splitting the workloads algorithmically so they can be executed simultaneously. Distribution is the process of splitting workloads between different nodes, so that they can be processed in different environments. As a result distribution may result in parallelization, but parallelization does not require distribution. In fact parallelization of certain workloads is a research area in its own right, there are many problems stemming from how the algorithms work, and in fact there are certain approaches which will also be useful when distributing the workloads. Apache Hadoop⁶ is a framework that allows for the distributed processing of large data sets across clusters of computers using Map-Reduce(-Merge) programming paradigm[52], essentially approaching parallelization from a conceptual perspective, which also allows for distribution of computing workloads.

2.3.1. DISTRIBUTED SERVICES

One basic concept of programming is the function structure, inheriting its meaning from the mathematical definition of function, where it is a relation between sets, this relation can be defined algorithmically, and here is where a programming function is born. These functions are logically operations to transform one set of inputs into a set of outputs (typically just one output). This concept encourages the structuration of code into logical and architectural blocks which not only makes it easier to develop and maintain code, it helps reuse it too.

If these functions need to be offered through a network from one node to another, then they are called services. The use of services in a distributed environment adds several layers of complexity, for example the syntax of the services, the languages the producer and consumer use need to be compatible; the logic operands and referral to the service must be unique, specially when there is more than one service; and finally when there is a full ecosystem of services, a consumer might need to discern between all the producers of services and the services them selves.

⁶<https://hadoop.apache.org/>

In this regard many technological solutions have been put forward, from middleware approaches such as rOSGi[53]. However the rise of the world wide web has put forward several standards based on the HyperText Transfer Protocol[54] (HTTP) used for machine to machine communication, all these are known as web services. Within the classification of web services there are different syntaxes used like XML⁷ and JSON[55]. Web services may be asynchronous, AJAX is one standard used to handle this type of communication[56]. They may have states, as in the state the session might be in after one or several calls to other services. Alternatively they can be stateless, meaning that any possible required state is transferred to the consumer/client who then needs to manage this state for the next service invocation. This paradigm is the one most used in today's web services and it is called REST[57], for Representational State Transfer; These services also tend to use JSON as base syntax. REST purist would argue REST operations are self explanatory, however description of REST APIs has been standardised by OpenAPI⁸ de-facto specification.

Traditionally web services were described and invoked through XML syntax. Web Services Description Language[58] (WSDL) was used for clients to identify the available services in an endpoint (server plus path within it); where as Simple Object Access Protocol (SOAP) was an XML formalization for the protocol of message exchange between parties. The use of WSDL is interesting as this had multiple purposes other than defining the protocol for each service, it could be used to create service agnostic libraries which could adapt to the given service on runtime, service composition and data flows could be described, and service discovery protocols could be enacted. All these latter functions are not so standardized in the modern REST philosophy.

In reaction to this the Microservice architecture has been proposed[59]. Microservices are a software development paradigm (a variant of the service-oriented architecture[60] (SOA)) that structures an application as a collection of loosely coupled services. Each service is a fine grained operation using lightweight protocols. This style has also promoted the need for virtualization of applications, also known as *containerization*, see section 2.4.

2.3.2. COMPLEX EVENT PROCESSING

Contrary to services, event driven processing relies on the occurrence of events. Events are simply things that happen, and can be reported to a processor using some kind of protocol, for example a service.

The most basic form of event processing is just delivering the events to subscribers who have declared their interest in such an event. In this context the service in charge of delivering events to subscribers is called broker. A broker may be a Complex Event Processor (CEP), which can potentially perform some advanced operations over the events[61], such as:

⁷<https://www.w3.org/TR/REC-xml/>

⁸previously known as Swagger. <https://swagger.io/docs/specification/about/>

Event-pattern detection finding patterns in the streams of events, being capable of detecting previously occurring repetitions of events, and reporting as patterns.

Event abstraction when a pattern is detected it can be abstracted into a higher level event which can also be delivered as any other event.

Event filtering detecting noise, or not delivering events to subscribers who are not interested.

Event mapping, aggregation and transformation Events of the same type and content can be aggregated, instead of delivering every repeated instance, the broker just delivers the first and aggregates the rest repetitions.

Modelling event hierarchies events can be classified into taxonomies or hierarchies, and these might be useful for the other operations.

Detecting relationships such as causality, membership or timing between events.

One noteworthy implementation of a CEP is the FIWARE's Orion context broker⁹. Orion is a broker of context information using NGSIv2¹⁰ protocol (defined in FIWARE platform¹¹) for the subscription, publication and delivery of context information.

2.3.3. CLOUD MODEL

Cloud computing is a term referring to the availability of computer resources over the internet, without direct and active management by the user of said resources[62]. Typically resources are storage and computing resources, but networks can also be offered in cloud computing platforms. The infrastructure of cloud computing is built on the concept of virtualization (see section 2.4).

In the context of cloud models, the terminology tends to describe everything as a service. Most of the time as the type of service described in section 2.3.1, others as a service in the context of an economic transaction. There are many concepts which can, and are offered as a service, however the most common cloud services are:

Infrastructure as a Service (IaaS) Offering services to control low level operation over the infrastructure, typically management of the virtualized components.

Platform as a Service (PaaS) Services for developers to build their applications without the need to deal with platform deployment and maintenance.

⁹<https://fiware-orion.rtd.io/>

¹⁰<https://fiware.github.io/specifications/ngsiv2/stable/>

¹¹<https://www.fiware.org/>

Service as a Service (SaaS) Complete applications are deployed as services.

many separated services have been ported to the cloud, given the promises of the cloud computing approach. The AAL domain is no different, semantic query and data storage to support mobile devices has been proposed [63]. Another example is the usage of the cloud to store patients health care records and medical images[64], as well as emergency medical services in the cloud[65].

The Knowledge-Aware and Service-Oriented (KASO) Middleware connects small devices with a cloud service, however, multi-tenancy aspect is not considered. Multi-tenancy is the capability of independently serving different clients, or tenants, as if they where the exclusive usage of the cloud resources. This aspect is crucial for cloud applications as it helps their development, and naturally is a feature cloud platforms will offer. However multi-tenancy does not grantee the advanced functionality required for smart environments.

So far, it was not investigated to have a complex AAL platform running in the cloud that is capable of handling multi-tenancy where tenants may be other instances of the AAL platform.

2.3.4. PEER TO PEER MODEL

The Peer to Peer (P2P) model defines network operations where there is only one hierarchical level, every member of the network is equally privileged, and equipotent participant of the system (hence Peers). Each node is capable of issuing as well as receiving requests. All of it makes it a very contrasting model to the client-server model.

Since Napster¹², the proliferation of P2P for file sharing has risen until today, networks like eDonkey¹³, Torrent¹⁴, etc. The P2P model has been used for other domains, the most interesting of which are crypto-currencies like bitcoin¹⁵, based on a logical layer build on top of P2P called Blockchain (see 2.3.4.1). Advocates of the P2P model, will argue these model promotes freedom. Some networks or applications are designed explicitly along this dogma; for example the onion network¹⁶ allows users to browse the web anonymously, or FireChat¹⁷ and Netsukuku¹⁸ create government-control-free networks parallel to internet.

There are many types of P2P networks, but the most common is the distributed hash table model. In This model the core service offered by the network is a recalling service, by which given a hash of a document that document is recalled from all nodes that can provide it.

¹²<https://en.wikipedia.org/wiki/Napster>

¹³https://en.wikipedia.org/wiki/EDonkey_network

¹⁴<http://www.bittorrent.com/>

¹⁵<https://bitcoin.org/>

¹⁶<https://www.torproject.org/>

¹⁷<http://opengarden.com/>

¹⁸<https://github.com/Netsukuku/netsukuku>

Figure 2.2: Example of file distribution in a traditional P2P network. Solid arrows represent requests for a resource, dashed arrows represent the attainment of the resource. Each node has different completeness of the resource, and thus each node has different capability to provide said resource.

Technically the P2P model, although more challenging to implement, offers many advantages to the traditional cloud model:

Scalability, the distributed system can work with as little as 1 node (also known as offline), and adding nodes does not require a great increase of computation, network, nor storage resources on each individual peer; yet the total capabilities of the system grow with every new member.

Integrity, any attacker must scale the attack to almost as many nodes as the network itself, in many cases making it unfeasible to successfully bringing down the system, or extracting or corrupting information. e.g. in figure 2.2, the document must be deleted from all nodes, if any single node loses the document, it only needs to request transference from the network.

Safety, typically the resources, information, and services, are fragmented and replicated amongst many nodes; if any node were to fail, the system is capable of continuing operation uninterrupted by redistributing resources and load.

Resilience, The data and services may still be available even when one, or several, nodes fail or go offline. The remaining cluster may complete the request from said file, if the set of distributed capabilities fragments is equal to the original ones. e.g. in figure 2.2, if node α disconnects, the transfer to δ can be completed only if the sum of fragments in β and γ are equal to the whole document in α .

Resource optimization, Resource optimization, especially compared with client-server model, there is a higher throughput in a P2P network as all the communication links are used, instead of saturating the server's communications. e.g. in figure 2.2, node γ connects both to α and β optimizing outgoing traffic from

α . Computation and storage resources, can also be optimized by distributing the load amongst the nodes in the network.

Privacy, as many resources can be handled in a deferred manner (e.g. in figure 2.2, node δ does not need to connect directly to the source of the document, α), the recipient of a request may not be able to determine the requester's identity. Other techniques, such as content-based addressing, help maintain nodes independent from the content.

The challenge of implementing P2P networks, not only lies on the network difficulties, like increased collective network load, or overcoming firewalls (to enable clients behind firewalls to receive asynchronous communication); there is also a problem known as the Byzantine generals' problem[66]. This definition includes the possibility of "traitor" nodes, or malicious nodes which appear to be legal members of the network but behave differently. To solve this issue, P2P systems must always assume that other peers are malicious, thus making what would otherwise be simple transactions, very complex. Not only the nodes need to protect their operations from untrusted peers, the problem becomes significantly harder when a group of peers (typically the whole networks) need to consolidate a common status, or operation; this is called consensus.

2.3.4.1. BLOCKCHAIN

One way to achieve this consensus is using blockchain method. The blockchain is a system architecture by which a ledger (a type of transactional database) is maintained in a distributed manner using an underlying P2P network[67]. Technically the blockchain is formed by blocks of information, which are linked together. This linkage is performed using cryptography, using a type of hashing algorithm creates a block identifier; the hash includes the reference to the identifier of the previous block in the chain, hence the blockchain. The consensus across all nodes in the network is reached by ensuring 2 simple rules, there is only one link for each block, and the longest chain of blocks is the master chain. This resolves any conflicts when adding a new block to the chain, which may be complex depending on the algorithm chosen, giving time for the whole system to receive the changes and stabilize. Any blocks which is not part of the master chain is rehashed with the end of the master chain as the previous reference. See Figure 2.3 for a representation of a simple example.

The blockchain builds a trust infrastructure where the most important feature is the invariability of the data, as changing any block in the chain will require changing all the subsequent blocks, or creating a longer chain than the master chain. This could ever been achieved either by collusion of the majority of the nodes, or exponential effort relative to the length of the sub-chain to change. The natural consequence of this invariability is non-repudiation, the inability of any entity to deny their operations or data recorded in the ledger. To compensate this invariability, the database can be defined as an incremental or transactional system, thus modifications and deletions do not erase previous data, yet previous data, or versions of

Figure 2.3: Simple Blockchain, the green chain is the master chain, all nodes arrive at this consensus despite having different candidates.

the whole database will still be available to the system.

Smart Contracts are computer scripts which can define a contract between two entities, and even ensure their compliance automatically. In some blockchain systems, the Smart Contract is the information recorded in the ledger. Smart Contracts can be as simple as transactions, or as complex as tokenization of physical objects (including buildings and land), sometimes including scripts with different purposes like ensuring the tokenization rules are obeyed. In the last decade there has been an incremental use of blockchain technologies. This trend has been fuelled by the interest in cryptocurrencies, since these typically rely on a blockchain ledger, sometimes extended by smart contracts. A good example is the Ethereum project¹⁹, this system provides a platform where distributed applications can be developed; one of such applications is the Ethereum cryptocurrency, which is used mainly to reward miners (the entities investing computer resources in maintaining the blockchain). Yet other distributed applications can be developed on top of this system taking advantage of the distributed nature of the platform. Another similar example is Hyperledger²⁰, although this platform is more oriented towards applications, and system customization.

2.3.5. INTERNET OF THINGS

Internet of Things or IoT[21], is new technological paradigm, by which all devices interconnect to share data, services, and resources offering services for humans. This new paradigm advocates for intelligent autonomous devices which can perform limited tasks, but can accomplish more ambitious objectives when working together.

¹⁹<https://www.ethereum.org/>

²⁰<https://www.hyperledger.org>

Human intervention is typically not required, or at least in any individual device, the whole system works as a pervasive, ubiquitous system from the perspective of the user.

From Framing[68] to autonomous vehicles[30], smart cities[69] and of course health[26] and elderly care[24], IoT offers new approaches to system design. The implementation of IoT is still challenging, from many perspectives. Networks need to offer connectivity at high bitrates, with low latency, and low power consumption, of course wirelessly over vast coverage areas at low costs to fulfil the IoT demands, thus it is not surprising to find next generation 5G[70] specifications scrambling to define standards for IoT networking. Electronics need to be also advanced, not only connecting to next generation networks, but also being capable of high computation, large storage, and embedding new sensors, once again, with low power consumption, and at an affordable price for scaling purposes; technological providers are envisioning this as a key market in the future for prototyping (with single board computers like the raspberry pi²¹, or beaglebone²²; or microcontroller development boards such as the Arduino²³ and its variants or the esp8266 family²⁴) and production ready hardware. Finally software is the most challenging area for IoT, not only it needs to handle the new networks, and the advanced hardware, it needs to deal with the heterogeneity of networks, sensors and devices on top of handling the services and applications it needs to provide for IoT users; this is as much as a challenge as it is an opportunity and so is reflected by how the number software frameworks for IoT have exponentially grown in the last years proposing alternative approaches and paradigms.

This challenge should be handled by standardization body, avoiding the extreme competition and chaos, but even then many bodies are competing to provide a uniform solution. On the mean time the industry is relying on trusted protocols such as http, CoAP[71], REST services, and cloud based architecture; however IoT should not be limited to these technologies. IoT stack not only includes the application layer protocols, other layers defined in the TCP/IP or OSI[72] network models are also considered for standard IoT implementations, again based on existing standards, for example Alljoyn²⁵ for discovery.

2.3.6. EDGE COMPUTING

In contrast to cloud computing, Edge computing is the vision of bringing the computer resources nearer to wherever they are needed[73, 74]. This concept has many benefits, since the services are managed locally there is no need to share data with anyone, not even the service provider, increasing cybersecurity and privacy; the network load is reduced, to the point where services can still operate offline.

²¹<https://www.raspberrypi.org/>

²²<https://beagleboard.org/bone>

²³<https://www.arduino.cc/>

²⁴<https://www.espressif.com/en/products/hardware/esp8266ex/overview>

²⁵<https://openconnectivity.org/developer/reference-implementation/alljoyn>

Edge computing is an area of research which intersects with IoT, however Edge computing may not involve internet at all. In fact the coined term for systems which mostly operate on the edge but interact with and act like cloud providers is Fog computing. This paradigm is the intersection between Edge and Cloud, and a better approximation of IoT for Edge.

The challenges for Edge computing are very similar to those for IoT, except for those regarding network requisites. And thus many IoT platforms consider edge operations, or Edge platforms turn IoT platforms. However it is extremely interesting to observe how semantics plays crucial role also in the Edge paradigm. This is the case of Solid²⁶, a platform lead by Prof. Tim Berners-Lee (the proposer of the world wide web), which focuses on providing Web 3.0 paradigm implementation. In this paradigm the data belongs to the user, and the applications run locally over this data, this makes reusing data for different applications transparent. For this the data should be modelled using semantic technologies, so that applications can make sense of it without external intervention.

2.4. VIRTUALIZATION AND CONTAINERIZATION

Virtualization is the process of providing virtual computing resources, they act as physical computer resources, in fact they are typically supported by actual hardware, but they are compartmentalized from the other virtualized resources on the same hardware, and are managed through software. The most common virtualization are virtual machines, they are virtualized full computers[75]. These computers act as normal computers, with assigned CPUs, memory, hard disk and other peripherals. Virtual machines run on a host computer, this host has to be at least as powerful as the virtual machine provisioned, and in fact it typically provisions multiple virtual machines. The software managing Virtual Machines is called hypervisor, since it works at an even lower level than the operating systems in the virtual machines.

Virtualization may also occur at network level[76]. The most common form of network virtualization is Virtual Private Network (VPN), which allows for the deployment of network overlays over existing networks. Normally they are used to secure connections for clients, when deployed the client is effectively inside a local area network, which may be physically in a remote place and other secure mechanism may apply to it.

The virtualization concept can also apply to software. In this case the term is Containerization, because the software is contained inside another management software called container. Some examples include OSGi²⁷ (an open specification for a Java module container based on service oriented architecture), but the most common in the industry is Docker²⁸. Docker is a container of applications, the model defines

²⁶[urlhttps://solid.mit.edu/](https://solid.mit.edu/)

²⁷<https://www.osgi.org/>

²⁸<https://www.docker.com/>

applications as images, which are stackable, the images provide the initial software resources as well as the descriptions on how to start and run the application[77]. Once a Docker image is running is referred as a container, because there may be more than one container running the same image. A container may have persistence added through volumes connected to mount points in the image, they may also have network configuration, such as which ports are accessible. Containers are much more lightweight than virtual machines, they contain a single ready to operate application, which can also be managed providing redundancy, security, and, specially for web applications, extreme flexibility like configuring endpoints, or stacking applications to cooperate.

Virtualization also helps organize configuration. For example systems such as kubernetes²⁹ or docker swarm³⁰ help configure containerized applications, specially in distributed environments ensuring reliability across distributed nodes. Another example is the OSGi's configuration service, which helps centralize and manage the configuration for all bundles, not only does it provide parameter based configuration to bundles, but the specification defines configuration factories, which creates instances of objects each with independent configuration.

2.5. SECURITY AND PRIVACY

In an information system, the main Cybersecurity objectives are the following [78, 79] (often referred to as the “CIA triad”):

- Confidentiality: no improper disclosure of information.
- Integrity: no improper modification of information (alteration, deletion or creation).
- Availability: no improper impairment of functionality.

In order to reach these objectives, the main typical cybersecurity services or requirements are the following:

- Authorization: the definition of rules on who is allowed to read, modify or delete which information.
- User and entity authenticity: the assurance that the other party is the intended communication peer, no “man-in-the-middle” scenario.
- Integrity (Data and service Authenticity): the data are not altered during transmission (accidentally or intentionally).
- Confidentiality: the exchanged data cannot be overheard or made available to the third party.

²⁹<https://kubernetes.io/>

³⁰<https://docs.docker.com/engine/swarm/>

- Timeliness and validity of the data (e.g. protection against message replay).
- Non-repudiation of the transaction.

Privacy is defined as the quality or state of being apart from company or observation. In terms of data privacy this means that individuals should have the ability to keep their data secluded. Data privacy plays an important role in society, as thanks to it individuals are free to express themselves free and uncoerced. Because Data can be linked and processed, it means that not only personal data may be under privacy concern, but any data emanating from an individual should be protected as it could be de-anonymized, or identified.

From Solve, a taxonomy of privacy has been defined:

Information collection the act of collecting data is in it self harmful to privacy, the data might be revealing, or even metadata can contribute to reconstruct what needs to be keep private. Collection data includes *Surveillance* and *Interrogation*.

Information processing how the data is processed needs to be monitored too. Some techniques to protect data include *Aggregation*, *De-identification*, *Security*, avoiding *Secondary use*, and *Exclusion*.

Information dissemination is when privacy is violated, through the disclosure of information, partial or total. Examples of how information can be disseminated and undermining privacy are *Breach of confidentiality*, *Disclosure*, *Exposure*, *Increased accessibility*, *Blackmail*, *Appropriation*, and *Distortion*.

Invasion Privacy can be explicitly targeted through *Intrusion* and *Decisional interference*.

Many countries have their own legislation on privacy protection, most notably the GDPR. All these legislations revolve around core principles of privacy protection:

- Lawfulness, fairness and transparency
- Purpose limitation
- Data minimisation
- Accuracy
- Storage limitation
- Integrity and confidentiality (security)
- Accountability

Cybersecurity and privacy protection is a serious business, not only being a huge research field in it self, but having support from authorities, legislation, organizational and technical standardization bodies, as well as being an extremely competitive market.

2.6. SYSTEM EVALUATION METHODOLOGIES

Evaluating information systems may be very complex, specially when considering the perspectives of evaluation the study must produce. As a first approach the simplest form of validation is checking whether the system complies with its specifications. For this typically technical tests, for unit operations, and integrated operations are performed. An additional approach is to measure how effective the system is, for which it is necessary to ponder the resources the system consumes, and whether these are optimal[80]. The next step would be to analyse the production capacity, in the AAL domain this might be tricky as the production of the system is highly abstract such as improving quality of life.

An evaluation may have to split the study into different dimensions, for example: measuring how much effort and costs does the system demands, as well as how much economic benefit it may produce. Equally the evaluation may look into technical aspects as discussed before, or the quality of the documentation and support.

Evaluation methodologies should also consider the target entities for which the evaluation is useful[81]. These entities may be recognised as the AAL Stakeholders, and in deed each one of them expects to get different value from the system.

Figure 2.4: The Bain & Co. Value pyramid.

The very concept of value is still very abstract, this is why a good analysis of what

value is, is needed. Bain & Company have developed the Bain Value Pyramid[82] (see figure 2.4), which creates a categorization of value for any commercial product. This categorization of value is compatible with AAL values; considering functional values are provided by the supporting system and technologies; Emotional value categories are provided by the AAL services; Life changing value category can be considered as the the personal impact AAL may have on individuals; and finally Social Impact is the whole purpose of AAL: addressing ageing population, extending life span, reducing burdens on society and changing the culture around ageing.

2.7. SYSTEM DEVELOPMENT

Overall system development methodologies are essential for any system development. Over the years there have been many methodologies proposed, however there are 3 basic ones into which they are typically classified.

2.7.1. WATERFALL

Considered the quintessential software development method, the waterfall model defines a rigid linear model of sequential phases[83]. These phases typically are requirement extraction, system design, implementation, testing and maintenance. For the process to continue the current phase must be completed, only when the phase fails the whole process is allowed to backtrack.

Waterfall model is easy to understand and manage, after all there is only one single track and a single phase for each project. The waterfall model is recommended for projects which have clear objectives, stable requirements, and are to be developed in one iteration in a predictable time scope. On the other hand, the rigid structure of the waterfall model type development methodologies, make it slow and costly compared to other methodologies.

2.7.2. SPIRAL

When combining the waterfall methodology with other rapid prototyping methodologies, the spiral model arises[84]. The key improvement is the iterative risk analysis, allowing for the growth of larger projects. By breaking the project into smaller segments, and adding flexibility the risks of the overall project are minimized. The breakdown needs to consider the core functionalities which need to be produced first, then focus may shift to the extra and dependent functionalities adding value on each iteration. The nature of the growth life cycle makes extending the project easier, as well as adapting to new and evolving requirements.

Each iteration follows a similar approach as in the waterfall model, a linear sequence of steps each part should follow. The 4 basic steps, represented in quadrants are: Objectives review, Risk assessment, Develop and validate; and finally Plan next iteration.

2.7.3. AGILE

Agile development methodologies[85] emphasise the need to adapt on each of the steps, and generally focus on the decision processes involved with adding new functionality. Every new functionality is broken down in to smaller increments to achieve the functionality, which are added to what is referred as the backlog, where it can then be assigned to a single developer to manage. All backlog increments constitute more or less the same amount of effort and time, which may depend on the project management. There are many different forms of the agile development method, including scrum, crystal, extreme programming (XP), and feature-driven development (FDD).

Agile development allows the product to be released in iterations, allowing for more efficient validation, testing and fixing; specially when these tasks can be pipelined. Another great advantage of agile development is that early working prototypes are encouraged, ensuring the maturity of the system follows the desired path from the beginning. Agile development techniques rely on high synchronization and organization of the involved teams, and project management should coordinate carefully the activities.

2.7.4. CONTINUOUS INTEGRATION

Sometimes referred as DevOps[86], is not just a development methodology but also a set of practices extending all the way to the organizational culture of the development team[87, 88]. This organizational change, enhances collaboration between the different people involved in the different phases of development, specifically as development, validation and maintenance.

Continuous integration builds on agile ideals, extending, optimizing and even automatizing life cycle stages for each increment. The idea is that for every increment developed a process is triggered by which the increment is tested and validated, and then deployed for full integration testing. When the increment is approved it can then be released into production, which means DevOps enable low time-to-market developments; but at the same time assuring the quality and stability of each release. This process produces high frequency of releases and updates, which in some cases may be decremental to users, or even impractical for certain environments and industries.

CHAPTER 3

METHODS AND MATERIALS

In this chapter Methods and materials on which, the rest of the rest of the research will be fundamentally based upon.

3.1. UNIVERSAAL IOT

universAAL IoT[89] is a mature open platform for the integration of open distributed systems of systems. Semantic interoperability is the main tool to define its service-oriented architecture. In this way it provides an open horizontal service integration layer at the highest abstraction layer, also applied to the whole stack, which makes it a suitable solution for many of current system issues such as vendor lock in or incompatible protocols.

The history of this platform can be traced back over 15 years from general conceptual work in German research projects EMBASSI and DynAmITE. A proof of concept was developed in the European Framework Programme 6 research project PERSONA. Finally consolidating and providing first tooling in FP7 universAAL[90]. The European Competitiveness and innovation framework programme (CIP) funded the ReAAL project, which provided stress testing in real life and has led to the creation of the initial universAAL IoT true ecosystem.

universAAL IoT is distributed with the Apache Software License 2.0, available under <https://github.com/universAAL/>.

3.1.1. UNIVERSAAL TERMS

universAAL being a novel approach, it introduces many new concepts which need deep understanding when considering complexity of certain aspects of the platform and the solutions it provides. These concepts are going to be important for the modelling section 4.1, as they are the basis for the rationale for most of the models presented there. Here are the most important concepts introduced in universAAL IoT platform:

uSpace Or universAAL Space, is smart environment (in the sense of a physical

space, such as a home, a car, an office, a supermarket, a hospital, an airport, etc.) with defined boundaries of hardware, software, and human resources (in terms of, e.g., access control, multimodal user interfaces, etc.), which is capable of responding to the needs of its users in an adaptive way by providing relevant assistance. In certain ways a uSpace models an ELE, or effectively any edge computing environment. The universAAL platform builds an uSpace with the union set of locally connected uNodes, together with the set of connected devices to all of these uNodes. uSpaces may be extended remotely, creating hierarchies and larger virtual uSpaces serving multiple users.

uNode An uNode is an instance of the universAAL middleware running. There could be more than one uNodes running on a single connected computer. Each uNode is a container of not just the middleware components but also application and platform components, as well as device connectors.

AAL service Refers to the physical action of helping someone, typically offered (and charged) by service providers such as health insurers, day care centres, news outlets, financial entities, etc. An AAL service may or may not be supported by technology and/or human resources.

AAL Application Software that implements certain use cases involved in the provision of AAL Services. It may consist of several software modules and / or utilize special-purpose devices (e.g., sensors, actuators, medical devices, etc.).

AAL Platform Refers to the software that supports the creation of integrated AAL systems by providing a logical layer that facilitates the integration of interoperable AAL applications as well as a standard set of capabilities commonly needed by AAL applications. The included level of abstraction may include uSpace, uNode, AAL Application, and even AAL Service.

universAAL IoT Middleware The core part of universAAL IoT which provides the logical horizontal integration layer for creating integrated systems across distributed components and subsystems. It hides distribution and possible heterogeneity of the components and acts as a broker between the communicating parties. This way, it facilitates integration and communication and effectively provides for interoperability among components that may have been developed widely independently from each other. The kind of interoperability enabled by the universAAL Middleware is called “semantic interoperability” because it eliminates syntactical dependencies among the communicating components and reduces their dependencies to shared domain models or ontologies.

universAAL IoT Buses Part of the universAAL Middleware with the ultimate integration and communication logic that implements the semantic brokerage mechanisms and handles the flow of messages among the communicating parties. For each type of communication, there is a specific bus with its own

model, strategy and match-making: (1) the Context Bus for event-based communication based on the subscribe / publish pattern, (2) the Service Bus for the general call-based communication based on the request / response pattern, (3) the User Interaction Bus (UI Bus) for the specific call-based communication when utilizing interaction services, and (4) the Control Bus for the specific call-based communication with regard to configuration management. It is these Buses to which applications connect in the process of integration.

universAAL IoT Manager A Manager is a software component that runs on top of the universAAL Middleware as part of the platform itself. universAAL is distributed with a set of such manager components that are necessary for its proper operation or provide relevant basic services and / or events to the other managers and / or to the application layer.

3.1.2. UNIVERSAAL ARCHITECTURE

The middleware at its core understands data, and defines functionality, exclusively through the definition of ontologies[91]. Along with the middleware, a set of ontologies is offered, primarily covering the domains of Smart Environments and Ambient Assisted Living (AAL). However the applicability of universAAL IoT does not limit to these domains; as its semantic nature enables it to cover the desired new application domain just by adding the necessary ontologies.

The universAAL middleware is an enabler and broker for the exchange of messages among autonomous components. It is comprised of

Container interface enabling portability, configuration, and logging among other technical features,

Data Representation and Serialization for unified messaging using the machine-readable RDF/Turtle standard for exchanging both models and data instances,

Node Discovery and Messaging to find and message with other universAAL IoT capable nodes in the same local area network, and

semantic brokerage a set of virtual communication buses managing context information, semantic service , and user interaction exchanges between nodes and components.

The objective is to hide distribution and heterogeneity within the device ensemble and facilitate the integration of components through communication at semantic level.

Along with the middleware a set of platform components are delivered. These components are mostly optional, but help implement common use cases in AAL such as storing, reasoning, service composition, connectivity, user interaction extensions, security functions, device protocol interconnections, etc..

The universAAL IoT community also provides a set of tools to ease the development, deployment and maintenance of unversAAL IoT deployments and applications.

3.1.2.1. SEMANTIC DISCOVERY AND INTEROPERABILITY

Discovery in universAAL IoT has two distinct levels. The first discovery level is related to the mutual discovery of middleware instances. This discovery is targeted towards uNodes, or instances, and is based on network broadcast; concretely, universAAL uses java SLP framework. Once a node is discovered, the peering between instances checks compatibility and authorization. When the communication is established a logical secure network for message distribution, called a uSpace is created. This logical network can be implemented in a modular way however the only current implementation is based on the jGroups framework. The whole layer is based on well-defined APIs, and the middleware is highly modular so different solutions can be plugged, and configured in diverse combinations.

The second level of discovery is the semantic discovery provided by the middleware buses. Components connect to the buses describing their capabilities and requirements semantically. The brokerage, or routing of messages, is performed over semantically formulated content; it is based on a semantic matchmaking, which effectively discovers the proper provider (of functionality) or consumer (of data) which best fits the semantic description of the requested action.

This bus abstraction is the primary concept behind the paradigm of semantic interoperability as realized by universAAL IoT. In this paradigm developers use the provided semantic data structures and protocols of a bus to interact with them. This is a huge innovation from the convention where semantic interoperability is just linking data and enabling semantic search. Consequently, the new paradigm of semantic interoperability is not limited to finding “objects” (via semantic quering) to later call their functions based on an a-priori agreed API; this two steps can be performed simultaneously.

This new concept causes all dependencies between the participants in a communication to be shifted to models shared by them (shared ontologies or compatible ones with an existing mapping between them). This reduces all syntactical dependencies in the communication to the usage of a single standard API of the brokerage mechanism implemented by the middleware buses.

Therefore semantic interoperability is defined as the ability of two communicating entities to share data and functionality at a semantic level, without necessarily syntactical dependencies (such as common APIs). Specific use case dependencies can be resolved through common (or specific) shared ontologies which are pluggable into the system.

3.1.2.2. SERVICE BROKER AND ORCHESTRATOR

The Service Bus is responsible for request / response (call-based) pattern management of services. This bus enables the provision and consumption of semantic

services similar to a service oriented architecture. The concept of “service result”, i.e. *some info returned*, or *an effect achieved* or a combination of them the the core understanding for the registration of service provision and the request of services (using the construct called “ServiceRequest”). In service provision registration construct (i.e. “Service Profile”), components can formulate which service results can be achieved by the target service. Similarly requesters formulate the expected service results in their service requests. Through semantic matchmaking, the service bus as the broker between these two type of entities tries to find out which registered processes should be initiated in order to get the expected service results described in the request. If there is a match, then the bus asks the provider to initiate the underlying service, maps the response back to the structures usable by the requester, and finally returns the adapted response to the original requester. If more than one provider is matched and executed (which could happen under certain circumstances), the responses would be merged into a single response. One thing that should be specifically noted, is how the service request never makes explicit reference to the provider, this independence is the key for lowering the syntactical dependency between both, only semantics are used to match requesters and providers.

Registered processes may be parametrized, achieving a range of different results depending on the described parameters. The semantic matchmaking process of the service bus can find out which of the information items in the request should be passed as parameter in order to get the expected result.

One of the most prominent platform components dedicated in the service bus is the service orchestration, which taking javascript language can compose different services as one, not only capable of defining sequence of service requests, but also conditional execution flows to produce the final service results.

3.1.2.3. CONTEXT BROKER, CONTEXT HISTORY ENTREPÔT, AND SEMANTIC REASONING

Context is any information reflecting the state of any parameter inside the target environments. Context awareness, the capability of reading context, is a key aspect of intelligent environments. Context can be provided by an actuator who knows its own state; estimated by a sensor; derived by a reasoner; or even explicitly provided by a human user. In In unviersAAL IoT, Context awareness is achieved through ontology-based data sharing managed by the “context broker”.

The context bus is this context broker, is the bus is structured in a publish/subscribe pattern, also known as the event-based communication. Messages accepted by the context bus are context events (sent by context publishers) and are filtered for every subscriber given their context event patterns subscription parameters. Each context event represents a change in the uSpace context. Inspired by the RDF model, the change is described by the subject, or the object being subject to the change, the predicate, or which aspect of the state is changing, and the object, i.e. the new value for the given aspect of the state in the subject. Each context event has additional metadata, such as the provider info (details about the component producing this

information, which may be different from the subject), a timestamp, and a possible expiration stamp.

With context event patterns, context subscribers can specify the semantics of context events in which they are interested. The semantics are described using OWL restrictions concepts, this is defining specific conditions on any predicate of the context event graph. The semantic matchmaking on the context bus is simply checking if a given context event is an instance of restrictions specified by the context event patterns. The context event will be forwarded only to those components for which the event is indeed an instance, this is how context information is distributed within the uSpace.

The Context History Entrepôt (CHE), a platform component, guaranties the persistence of context. CHE receives all context events by registering as a subscriber with a broad enough restriction. It stores all context events received in an RDF triple store, which applies the ontological inference mechanisms. This inference may produces new knowledge based on known context, which makes the CHE also a reasoner. The CHE also connects to the service bus, allowing components to access current and historical states, as well as performing SPARQL queries for statistical analysis.

Specific reasoners can be added, they can subscribe, like the CHE does, to all events and produce new contextual “reasoned” events. One of such reasoners provided as a platform components is using the drools rule engine¹ to provide applications with the possibility of defining rules in the drools language similar to java. In fact these rules may produce new context events that are published to the context bus, make a service request to the service bus, or both. A second reasoner, the Situation Reasoner uses SPARQL “construct” queries for inferring new facts, which are then published to the context bus.

3.1.2.4. HUMAN-ENVIRONMENT INTERACTION

For many years the way ICT systems interact with humans has been called human-computer. This includes all interaction channels connected to one single computing device, which are typically abstracted as user interfaces, usually abbreviated as UI. universAAL IoT proposes a shift in this paradigm fundamentally targeting human-environment interaction paradigm. This paradigm depicts interaction channels distributed in the environment, thus not necessarily connected to one single computing device, see section 2.2. The user interaction framework is the main feature by which this paradigm is achieved.

Note that when the UI abbreviation is used it is meant for user interaction and not user interface, the distinction is subtle but reflects the paradigm change proposition. The thought pattern needs to change from single interfaces to environment interaction. This abbreviation will hold true for all this section, and following sections, and specially for all topics related to universAAL IoT and its component names,

¹<https://www.drools.org/>

The UI framework defines two distinct layers for the brokerage mechanism. The first layer, consisting of all components which may initiate or react to human interactions, is called the application layer. The second layer, the presentation layer, describes how the interaction is delivered or collected from the user, it is formed by a special type of component called “UI Handlers” who’s expertise is utilizing specific interaction channels.

The brokerage mechanism between the application and presentation layers handles the UI requests of applications assigning them to appropriate UI handlers in a situation-aware way. For this brokerage specialized components are used such as:

UI Bus the virtual communication channel which received UI requests from the application layer and takes care of the whole coordination of the interaction enacting protocols between all the rest of the components.

UI Model required for specifying interaction requests in a modality neutral way and without any layout bias, this model is inspired by the “Form Controls” of the W3C specifications for XForms[92].

Resource Manager is responsible for storing alternative presentation resources like visual icons and signs for different display sizes and their acoustic equivalents allowing the presentation layer to retrieve these resources when needed.

Dialog Manager coordinates interactions for every individual user in the uSpace, provides the following features:

1. manages system menus and standard dialogs which are not provided by any other application in the application layer
2. assists the UI Bus by incorporating the user context and preferences, which will be used in analysing the interaction situations,
3. implements a persistent mechanism for user-specific dialog management,
4. handles user instructions that are not in the context of a running dialog.

The UI framework is proposed to ISO/IEC as a Publically Available Specification (PAS) with the number 62883:20144. The typical UI use case starts with applications building their UI requests, consisting of the dialog to be performed (using the UI Model), the addressed user, as well as the language, privacy level and priority of the dialog, all in a declarative way. This UI Request is issued to the UI bus by the application, which will receive the response, or other status changes, via an asynchronous notification. The first step the UI bus performs is to delegate the UI request to the Dialog Manager, which will enrich the request with the user context, such as the user’s current location and his/her interaction preferences and capabilities. When the Dialog Manager decides the user should receive the dialog, typically if there are no other pending dialogs, it will notify the UI bus with the request to be handled, this request is now extended with all the knowledge the Dialog Manager has about the user. A matchmaking process is initiated by the

UI bus, using the enriched UI request against the registered UI handler profiles; This matchmaking will try to find the most appropriate UI handler for handling the UI request, taking into account different criteria like proximity to the user to the physical influence of the UI handler, the supported languages, and the privacy of the managed channels compared to the privacy required by the contents of the UI request. When the user answers the dialog, both the Dialog manager and the application are notified, the Dialog Manager un-queues the dialog as it is finalized, the application receives the UI response with the desired user input.

This protocols handles all possibilities from the origins of the UI request being remote, in terms of node the application is executing in, to fail conditions, such as when UI handlers stop working. It is also able to handle multi-modal automatically. Additionally, it provides for dynamical, loss-less change of interaction channels with users. If the user’s context changes in any way which makes the currently selected UI handler irrelevant (e.g. change of user location, or the entrance of other people in the same location as the user), the bus will request to cut the execution of the current dialog and return all data collected from the user so far. With the new information, provided by the Dialog Manager, a new more relevant UI handler is selected to better match for the new situation. Scenarios, such as “follow me” where the dialog follows the user from one room to another one, are realized seamlessly.

The most interesting value for application providers is that the UI framework offers all these advanced features easily. Any application developed on top of the UI framework, is relatively easy to implement, but has all the intelligence power of the human-environment paradigm, including multi-modality, context awareness, adaptation, personalization and all of the scenarios described above are transparently managed for this layer. Equally development of UI Handlers in the presentation layer allow for pluggable components, and thus third parties could specialize on the development of expert systems managing complex communication channels, independently of the applications who will use it.

3.2. SECURITY ASSESSMENT, IMPACT AND MITIGATION METHODOLOGY

Security is as complex as it is critical for any interconnected digital asset manager. Private companies [93], public bodies [94, 95] and standardization/harmonization institutes (e.g. RFC 2196 Site Security Handbook) have published recommendations aiming at improving the quality and consistency of the cybersecurity across interconnected systems. Such recommendations are addressed towards system managers, organization officers, service providers, infrastructure owners, product manufacturers, developers, and end users. Security experts argue that every security measure, mechanism and algorithm must rely on publicly available specifications.

Generally all these sets of recommendations present diverse aspects, but generally converge on similar rules and recommendations. This is, recommendations insist on the fact that cybersecurity should be a continuous process, necessarily

self-improvement procedure should be embedded, and continuous evaluation of the cybersecurity threats and measures should be exercised. Self-auditing and internal expertise are strongly required, but generally this is not enough. Expert companies offer services to independently analyse the cybersecurity mechanisms of the system. These auditing typically includes cybersecurity standards, such as ISO/IEC 27001 and 27002, the NIST Cybersecurity Framework.

When analysing cybersecurity on any system the first question is “what is it that needs protection?”. Indeed the answer is an asset, and identification of said assets might not be as trivial as it looks. The next question given an asset is “How can it be compromised?”, creating a list of potential threats, and evaluating the likelihood of each attack is necessary. “What are the consequences if the asset is compromised?” is the next question, analysing the scenario where cybersecurity fails may have different impact, the seriousness of this impact should be categorized. Finally Risk analysis can be performed, when we know what needs protection, how likely is it to be attacked and how harmful the results of that attack may be, we can consider if the risk is manageable, or not, and if not which countermeasures should be placed (which may vary in costs, thus optimizing costs per risk).

3.2.1. ASSET IDENTIFICATION AND DESCRIPTION

An asset is anything (and anyone) which needs to be protected from a threat. The following categorization of asset types would help, via examples, identify assets through categories, but please note this list is not complete.

Data Any piece of data running in the system. Although simple enough to visualize, further analysis will determine more data categories, for example:

- The Data models describing how the data is encoded and interpreted.
- Data describing all entities producing or consuming data (Identifiers and attributes of individuals, stakeholders, sensors).
- Data flows and how the data is transmitted between entities.
- Data that are monitored and analysed by the system in order to ensure the expected service (raw measurements, processed data elements).
- Decisions of the system that influence the subject’s environment (guidance or prescriptions for individuals, environmental instructions for smart sensors, configuration instructions for devices). Management data relate to system operation. They include, for example:
- Procedure, action plan descriptions (definition of all the planned actions in case of occurrence of an extreme event).
- Data storage organization definition (for example, a Grading Table, Detail Description predefines categories for data storage, such as Medical information, Medical report, Wellness information, Service, etc.).

- Access Rights Table, defining the access rights for each stakeholder profile.
- Transaction registers, logging the History of all operated transactions (communication channel, data, data user, time, etc.).
- Cryptographic material that may include log-in credentials, cryptographic secrets for authentication and encryption, root-of-trust information (e.g. trusted PKI public key), public key certificates, etc.

Services internal and external services and Inter Process Communications (IPC) should be considered as assets. Replacement or removal of services could potential lead to integrity and availability issues. Considerations on who should access each service, and with which privilege (as well as how these are managed) need to be part of the analysis as well.

Hardware Component for example, typical hardware assets are data storage units, processing units, power management blocks, sensing and actuating blocks as well as all device interfaces (e.g. I/O, JTAG ports, etc.) and device casing. Maturity and configuration must be assessed, as hardware vulnerabilities are difficult to identify and protect against.

Software Components and configuration information must be analysed at all levels: OS, firmware, high-level application container, libraries and applications. Boot mechanisms (such as signed binaries) and system configuration (such as trusted Certificate Authorities) at all levels also need particular protection and are included in the assets list.

Communication channels between any component must always be considered for a potential attack vector. This is particularly true for wireless physical connections (WiFi, BLE, Z-wave; Zigbee, etc.). Increasingly the use of Cloud services, means that the connections rely on internet connectivity, which means that the information will be travelling through untrusted, unmanaged third-party networks, and thus special care should be considered. In these cases it is also particularly important to consider the connection it self as an asset, for example mobile internet providers (2G,3G,4G or even 5G) may not provide complete coverage, thus the system might lose the required connectivity, setting the scene for a potential system vulnerability.

These categories and may be even extended to abstract concepts like intellectual property, or brand reputation; or to physical objects such as people, or buildings, among other.

3.2.2. THREAT IDENTIFICATION: STRIDE

The STRIDE (see table 3.1) threat classification model was developed by Microsoft [96], and helps answering the question “what can go wrong in the system?”. This classification just maps possible attack vectors to the security properties.

Threat Concerned	Security Property
S poofing	Authentication
T ampering	Integrity
R epudiation	Non-repudiation
I nformation disclosure	Confidentiality
D enial of service	Availability
E levation of privilege	Authorization

Table 3.1: STRIDE Threat Classification

The main idea is to analyse each asset for each threat in STRIDE. Limiting the scope of the analysis to just one asset and one threat type, it is then possible to identify individual attack vectors and methods, as well as vulnerabilities in the asset protection mechanisms.

3.2.3. THREAT IMPACT ASSESSMENT: DREAD

The DREAD method is an evaluation of the vulnerability of the asset. To perform this evaluation, we need to ask: “What will be the impact of the identified STRIDE threats on the assets?”. Assessing the impact as a whole might be challenging, this is why DREAD proposes a standard set of individualized impacts. To rate the impact of the identified STRIDE threats is assessed through the lens of these impacts, and ranked (typically from 1 to 3, 1 being the least impactful). In Table 3.2, the individual DREAD impacts are listed, identified and an initial raking is demonstrated.

Table 3.2: DREAD ranking definition.

Impact	Impact question	Impact evaluation
D amage poten- tial	How great can be the damage?	1. Leaking trivial information 2. Leaking sensitive information 3. Can subvert the security system

Reproducibility	How easy to reproduce?	<ol style="list-style-type: none"> 1. Very difficult to reproduce, even with knowledge of the security threat 2. Can be reproduced, but only with a timing window and a particular situation 3. Can be reproduced every time and does not require any particular situation
Exploitability	How easy to realize this threat?	<ol style="list-style-type: none"> 1. Requires an extremely skilled person and in-depth knowledge every time to exploit 2. A skilled programmer could make the attack, then repeat the steps 3. A novice programmer could make the attack in a short time
Affected users	How many users are affected?	<ol style="list-style-type: none"> 1. Very small % of users, obscure feature; affects anonymous users 2. Some users, non-default configuration 3. All users, default configuration, key customer

Discoverability	How easy to find this vulnerability?	<ol style="list-style-type: none"> 1. The bug is obscure, and it is unlikely that users will work out damage potential 2. located in a seldom-used part, and only a few users should come across it 3. The vulnerability is located in the most commonly feature and is very noticeable
-----------------	--------------------------------------	--

3.2.4. RISK MANAGEMENT AND MITIGATION

Pondering Thread likelihood versus Thread impact produces a Risk index per threat per asset. This risk index might be calculated via simple multiplication, high probability and high risk produce high risk index. This index can be used to determine the minimum risk requirement, as a threshold on what risks will be accepted, and which need to be mitigated. Another option is to transfer the risk to a third party to manage the corresponding threat.

Common Risk mitigation technologies, such as Cybersecurity measures or Cybersecurity controls, can be considered for the threats who's risk has been deemed to be mitigated. These would include typical cybersecurity tools and systems, or recommendations by foundations, best practice guides, or standard bodies could be used as guidelines, providing lists of Cybersecurity requirements depending on risk level [16, 17].

3.3. OPEN PLATFORM ECOSYSTEM ASSESSMENT (OPEA) FRAMEWORK

the Open Platform Ecosystem Assessment framework consists of the OPEA-conceptual model, the OPEA-indicator model and the OPEA evaluation strategy.

The OPEA-framework is one of the main results of the ReAAL project, using resources and experience with in it, complemented with a literature search, expert consultation and validation. The OPEA conceptual model is the theoretical base for the evaluation. Based on the general model for telemedicine assessment[97] (MAST), it was thoroughly revised to apply it to the AAL domain, and open platforms. Because of this adaptation the technical and economic domains are more elaborated. In order to make it practical for evaluation the Information Systems Success Model[98] (ISS) is used. The ISS is a model that draws causal relationships

between the integral quality of an information system, its use and its experienced benefits. The ISS model has been an inspiration for definition of relevant indicators for the evaluation of open AAL platforms.

Indicators are mapped in OPEA to a three-dimensional model, this assists evaluators in the construction of all indicators for the evaluation, as well as understand the relationship between them. The first axis depicts the value network, these are the main stakeholders: the AAL platform provider, AAL application provider, Health Service or Social Service provider, the informal carers, assisted persons and society as a whole; each have their own value objectives for deploying AAL services on an open platform. The second axis marks the assessment domains of the evaluation: Assistance services, Technical aspects of the application and platform, User perceptions, Outcomes, Economic aspects, Organisational aspects, Contextual aspects, and showcases. The third axis relates to the three levels of assessing the AAL ecosystem: the platform, application, and service level. Where the platform enables applications to be developed; these then facilitate the delivery of health or social care services to organisations in the industry, and potentially directly to end users.

The final list of indicators were developed by a process of literature review (e.g. the MAST model itself, the ISS model and models for technology acceptance), analysis of the pilot concepts and discussions in the evaluation team, and between the ReAAL partners.

The OPEA evaluation design used the conceptual model and indicator model for the actual evaluation strategy. The evaluation strategy takes a double approach: the evaluation of the pilots and the evaluation of the showcases that demonstrate the value of open platforms. Each pilot is involved in one or more showcases. The pilot evaluation consists of two phases: phase 1 evaluates the universalisation and testing; phase 2 evaluates deployment and operation. The data collected in the pilot and showcase evaluation is combined and complemented in the final step of the evaluation: the ReAAL impact evaluation. This final step is meant to validate ReAAL's results (displaying the findings about the value of open platforms), and draw scenarios about the impact for the AAL market.

Both the pilot, showcase and impact evaluation use multi-method designs with qualitative and quantitative data. The pilot evaluation has a before-after design. The following data collection tools are used: questionnaires, focus group interviews, individual interviews, templates, and blogs. In addition the materials other are used as a source, for example test reports, and operation reports. The showcase evaluation is designed as a technical demonstration and value assessment from the perspective of different stakeholders. The impact evaluation uses all previously collected data, and adds dedicated focus groups of, questionnaires to stakeholders, and discussions with experts.

3.4. OVERALL RESEARCH METHODOLOGY

Once reviewed the state of the art, and armed with some materials and methods, the overall research methodology is planned. This process consists in defining which steps need to be taken in order to accomplish the given objectives (see section 1.5). The overall research methodology is important to contextualize the research and provide repeatability to the research process. The context provided by the main research projects contributing to this research, helps understand not only the process, but the structure and chronology of the research.

universAAL FP7-ICT-2009-4 ² is a research project born from the need for an unified approach to AAL service development and deployment. Up to the conception of the project, there where already many research areas which produced solutions for the AAL domain, however these solutions where not coalescing into a product which could easily make all this solutions usable by the AAL community. One of the main objectives of the project was to consolidate all this knowledge into a single platform, which would offer the proper interoperability not only to gather these advanced technologies, but to be able to build AAL services easily on top of the offered components. This produced both a reference architecture and a reference implementation of said architecture, which is fully explained in section 3.1; additionally developer supporting tools where provided, as well as an initial validation of the system. Because the availability of this implementation would be the first open specification for technologically supported AAL services, it looked into creating the first AAL market by also providing a market place for universAAL enabled applications, devices, and services.

ReAAL CIP-ICT-PSP-2012-6 ³ is a large scale pilot project carried out in the framework of the Competitiveness and Innovation Programme (CIP) of the European Commission. REAAL was set out to demonstrate the advantages of using open and flexible ICT solutions, namely AAL platforms, as basis for flexible and personalised delivery of a range of services, such covering the different perspectives of the problem - societal, organisational and economic, needed for AAL. The AAL platform selected was unviuersAAL IoT, the main result from the universAAL research project, and the consortium as a whole was able to provide services to 7000 users over seven countries on top of this platform.

ACTIVAGE H2020-IOT-2016 ⁴ is a Large Scale Pilot Project with the objective of building the first open IoT ecosystem for AHA. Solutions are being tested around 9 Reference Use Cases (RUCs): activities of daily living, integrated care, monitoring outdoor, emergency triggers, promoting exercise for

²<https://cordis.europa.eu/project/rcn/93776/factsheet/en>

³<https://cordis.europa.eu/project/rcn/191949/factsheet/en>

⁴<https://cordis.europa.eu/project/rcn/206513/factsheet/en>

fall prevention, preventing cognitive decline, preventing social isolation, offering safety and comfort, and support for transportation.

The different solutions will be implemented across 9 *Deployment Sites* distributed in Spain, France, Italy, Germany, Greece, Finland and United Kingdom. Each site will carry out a subset of use cases, evaluating the actual value they provide to the market and end users.

Interoperability is a key aspect, not only for the systems, but for the ecosystem too, as it allows for easier deployment and operation of IoT-based AHA solutions. Thus one of the main outcomes will be a secure IoT interoperable layer (for integrating applications and platforms at different levels), as well as service layer (offering support for a marketplace, and application support tools).

The research was complemented by other AAL-related projects, however all had very clear vision for:

1. Interoperability between services and applications.
2. Evaluation of the AAL solutions and systems.
3. The creation of a cohesive ecosystem of AAL service and technology providers.

In all cases the interoperability solution selected was Semantic Interoperability. However it is very clear that this paradigm is not very well understood, or popular amongst the population of technology providers, including developers. These projects provide a phenomenal perspective on the challenges semantic technologies face, as they bring together researchers on interoperability and market grade developers of AAL solutions. Thus for objective 1 the methodology is straight forward. On one side perform state of the art survey on SDS, and on the other compare the main requirements developers have in their typical solution development. This comparison is complemented with direct interviews, as well as questionnaires. The overall impact of the challenges for SDS in AAL, is always provided by the evaluation and market uptake tasks where through methodologies like the OPEA framework (see section 3.3), other perspectives can provide insight on the general problem of technology application to AAL services; and precisely determine which are the challenges, opportunities and values in the whole ecosystem that is AAL.

These challenges typically show that there is a big gap in the understanding of semantic technologies, and how semantic interoperability can be used. Thus objective 2 is about bridging this gap. In order to achieve this, the initial process is to survey existing architectures, and the state of the art in semantic interoperability. In this sense the universAAL research project is a prime example, as it consolidated many technologies. In fact the reference architecture of universAAL is the initial input for the SDS model development. This model is iterated several times, applying the knowledge gathered from the results of objective 1, enhancing the model by applying new technologies, paradigms and state of the art methods; and finally the

model is refined to ensure the internal coherence and responsibilities of each subpart are clear. The last iteration of the model is presented in section 4.1.

Each iteration of the SDS model is validated, this corresponds to sub-objective 2.1. These validations are generally aligned with the research project, however sometimes interesting external proposals can be modelled with the SDS model. The most interesting cases are described in section 5.1.

Objective 3 relates to a validation methodology oriented towards value. As result of objective 1, it was determined that one of the main challenges was to explain and demonstrate value of SDS. It is precisely the results of objective 1 the initial condition for the development of the validation methodology. This set of requirements, is used to complement the OPEA framework (see section 3.3) to provide the required itemized value for SDS. The methodology is presented in section 4.2; which is executed in the context of sub-objective 3.1 for AAL platform value, resulting in the set of showcases presented in section 4.4, which in turn are validated results of which can be seen in section 5.2.

The overall development methodology, main result of this work, and target of objective 4 follows a similar pattern as the rest of the objectives. This is the initial methodology is determined through the research projects, then it is evolved iteratively through the experience obtained to produce a more robust methodology. In this case the methodology is linked to the global research project methodology, where several SDS are analysed, adapted and assessed. However because of this, the methodology can be unique to the project, so the main challenge to accomplish this objective is not so much to overview the evolution the methodology, as it is to extract the most interesting elements of each experience, always pondered with the overall results of the project. The main result is explained in section 4.3, and it is validated, in the context of sub-objective 4.1 through the application in different research projects in section 5.3.

Finally through all the iterations of challenges, models, and methodologies, there are concrete technological advancements in the form of implementations of the universAAL IoT platform. This is how the objective 5 is executed. Generally speaking, each new improvement is the result of evaluating both the evolution of the model, i.e. the result of objective 2, and the need to provide value, i.e. the result of objective 3. Each improvement, because it is not a fully fledged SDS, is developed with a lightweight version of the SDS development methodology, i.e. the result of objective 4. In this way each improvement, once commissioned (as explained before, due to the need of completing the model implementation, or to provide value), has performed a state of the art analysis, identifying the most compelling technologies to use. Then the adaptation process of either these technologies to universAAL, or universAAL to the state of the art paradigms is performed. Finally each improvement is assessed using different methodologies (generally not the value oriented methodology), before it is incorporated and released as part of the universAAL code base.

CHAPTER 4

RESULTS

The main Results of the research are illustrated in this section. A model for a methodological approach to analysing and designing SDS is presented, it defines 8 different perspectives to address during the modelling process, each compartmentalising the complexity of SDS. A value oriented, multi-dimensional methodology is presented. This methodology is used to find the value in many dimensions the different stakeholders of the system will place on the technical features of the SDS. With this information further refinements of the SDS are possible, as well as serving as a cross system comparison framework. A cocreation methodology is presented, this methodology is used to contextualize the use of other methodologies in the overall SDS development lifecycle. This methodology emphasises the need to address user requirements, as well as cocreating with the rest of stakeholders involved in an AAL service supported by SDS. Finally a Set of predefined showcases for AAL Platforms is presented, this is the result of application of the first step of the multi-dimensional evaluation methodology.

4.1. MODEL FOR SEMANTIC AND DISTRIBUTED SYSTEMS

The model of SDS is a proposed tool to enable system designers and developers to have a common understanding of the complexities of SDS. The model as defined here, can be used as a meta-model to specify concrete SDS models; it is also part methodological approach to SDS analysis and design.

This methodology is divided into several sub-models, each providing an explicit view on different complexities of SDS, particularly SDS in AAL. These sub-models will affect one another, thus is important to get acquainted with the sub-models first before proceeding to develop any model. In fact the sub-models should be considered for the analysis or design methodology for any given SDS; which may vary greatly between SDS, as there might be different needs for each sub-model, some systems may not require a particular sub-model, other may require several iterations. The following sub-models are presented in an initial order for the first iteration.

This model is the result of years of experience in analysing and designing SDS in AAL. Keen readers will observe this model has many similarities to the reference architecture proposed by universAAL IoT. Indeed this platform has been the prominent source for inspiration, as it is a platform to develop SDS particularly devised for AAL domain. However the proposed model has been conceived as more abstract enabling it to be more extensive in scope and application. Through this exercise it will be proven this model is applicable to other SDS not based on the universAAL IoT platform.

4.1.1. SEMANTIC MODEL

The semantic model is the selection, and/or definition of the ontologies of the solution. Typically ontologies will rely on one another, in what is called the semantic web. The best practice approach is to have the system import existing, preferably standard ontologies which map the operative domain, the concrete domain of knowledge the system is working with; or ontologies which will help construct the operative domain from tangential and/or overlapping domains. It is important to map these relations, as they form the knowledge base of the system, determining what can be represented and processed.

The complete semantic web may not be needed for a given system, as sometimes an ontology is imported only to represent a small fraction of its modelled domain, sometimes this may be as little as one class. Even when many ontologies are loaded, sometimes the operative domain is not represented. Thus the first use of the semantic model is to analyse the missing representations of the model. This analysis may not be as trivial as it appears, particularly because the initial set to analyse is the complement to the union of all imported ontologies, which is the open world, and by definition infinite.

An issue arising from using numerous ontologies is that they will tend to overlap, the more ontologies there are, the higher probability of this happening. This is because designers of ontologies sometime need a common concept, and instead of finding it in the repository of ontologies, they include it directly in the ontology itself. This might be counter productive, and against best design practice: reuse as many existing ontologies as possible for the given domain. However semantic systems provide mechanisms to overcome this issue, particularly there are axioms which can be used to assert to individuals are the same, properties and classes are equivalent; not to mention that the minimal differences can be considered through other more common mechanisms such as class sub-class relations. Sometimes these mechanisms are provided by the ontology it self, in which case it is denominated self-mapping ontology, or by a third-party or extra ontology, a mapping ontology.

When dealing with the need to create an explicit ontology to cover the operative domain there are already many good practices collected in the body of engineering known as knowledge engineering, see also section 2.1.1. This subject is far too broad to be completely covered by this document, however it is profitable to know something about the processes[99, 17, 100], design patterns[101, 102, 103, 104, 35,

Figure 4.1: An example of the visualization of the semantic model. Two operative domains are shown, with some standard ontologies like HL7, FOAF, SOSA and SSN, and other more specific ontologies both for health and home automation. There is a mapping ontology between home and SSN ontologies.

36], and evaluation techniques[105, 33] regarding ontology development.

Regardless, the semantic model must be an important view of the design of any semantic system, and in particular Semantic and distributed system. To help understand the model this visualization technique is proposed, an example of which is provided in figure 4.1. Ontologies as some authors do, will be represented with triangles (with one of its nodes pointing upwards), with their reduced form name in the middle. When ontologies directly import other ontologies, will be represented with arrows. More abstract ontologies, which typically are the imported ontologies, but don't have to, will be placed on top of the diagram, whereas the more specific ontologies will be placed towards the bottom of the diagram. When two (or more) ontologies' domain, or concepts, overlap this is represented by their triangles overlapping. If there exists some mapping axioms, whether as a self-mapping ontology, or as a mapping ontology, will be represented with a different color. A self mapping ontology will be identified as the ontology which overlaps and imports the extended ontology. The operative domain, may be represented as a dotted area, encompassing all related ontologies. When more than one operative domain is represented in the same diagram, they must be distinguishable, either by colour or name. This is useful for when representing the operative domain of the different components (see 4.1.6), to understand how they relate and how they overlap, helping for optimizations in the deployment model (see 4.1.8); knowing which ontologies need to be loaded where (and when), as to minimize the reasoning to be performed on each node[106].

This visualization will help visualize the missing aspects of the operative domain(s), and therefore identify the needed representations. The procedure to follow

when a missing representation is identified is:

1. research for existing ontologies. Even though existing ontologies might already be researched, and included in the model, these can be used to research 2nd degree ontologies related to the already identified ontologies, which may provide better presentation for the operative domain. This process may be repeated until the operative domain is represented, or no more existing ontologies can be found.
2. In case no existing ontologies can be found, create a new ontology, with extension oriented patterns, i.e. based on the existing ontologies included, extend and reuse those concepts to prove the needed semantic support for the operative domain.
3. In case the existing ontologies do not provide any base to the operative domain modelling needs, then develop an explicit ontology for the operative domain. This should be the last resort, but still many best practice approaches should be applied, like making it as abstract as possible, so it covers not only the details of the problem the system at hand will solve, but so it can be reused by other systems dealing with the same operative domain.

In the AAL domain many recurring knowledge domains, in some cases represented through (semi)standard ontologies, thus these should be considered as the initial set of ontologies included in the semantic model.

Health In AAL one of the most common objectives is to maintain users health, and for this some kind of representation of the user's health is required. There are many existing and standard Health ontologies, in fact this is the most prolific fields as of number of ontologies, not just for historical reasons but practical reasons too.

Health Level-7 or HL7¹ refers to a set of international standards for transfer of clinical and administrative data between software applications used by various healthcare providers. The application layer in the OSI Model[72], is "layer 7" hence the name. The HL7 standards are produced by the Health Level Seven International, an international standards organization, and are adopted by other standards issuing bodies. This makes HL7 an outstanding example in the Health industry to achieve semantic interoperability through a common representation of data.

Sensor and Actuator Smart environments like those proposed by the AAL vision, need to sense the environment, as well as act upon it. A robust conceptualization of what a sensor is, its context (like the processes involved, or the features of interest), the type of measurements it can take and the current measurement or observation need to be embedded into the model. A similar level of

¹<http://www.hl7.org/>

detail is also required for actuators, which instead of sensing the world they change some state of it.

Semantic Sensor Network ontology or SSN[22] is a wide spread ontology that does precisely this; as well as its core, the minimal set of concepts, which is published under the name of *Sensor, Observation, Sample, and Actuator* or SOSA ontology[23].

Units of measurement It might seem trivial to have any data in a given unit of measurement. In fact, sometimes the unit of measurement is implied, and therefore not coded into the model. However this is part of the semantics associated with the data, and thus it should be explicitly coded, either in the ontology model, by statically stating all instances of a particulate numbered property are always in the given units; or by adding a property to the measurement, which allows for different components to provide and interpret the data in different units.

Although it may not be a very tangential domain to AAL, the Semantic Web for Earth and Environmental Terminology (SWEET)² provides some modules including upper ontologies, as well as units of measurement.

Specific devices and Wearables In AAL, and as an extension to the sensor and actuator models, there is also a need to know the type of sensors and actuators being used. Where the Sensor and Actuator model provide a common process to represent information and processes, the subclasses of sensors and actuators are also relevant for systems to be able to make sense of the world or to even self configure given the capabilities the environment has. Because this list of subtypes might be too large, specially if we include the embedded sensors in wearable devices, or smartphones, we will consider it different from the Sensor and Actuator model.

SAREF, or Smart Appliances REference ontology[107] is designed for household and home appliances in residential buildings, especially for the purpose of energy management, and it extends the SSN/SOSA model. However for AAL it typically needs to be extended.

Outdoor Location Many AAL services apply on the user when moving around the city, examples may include public transportation system helpers, or safety zones for prone to be lost people. Thus a robust model for outdoor location is needed. Although this model might be dismissed as the World Geodetic System 1984 (WGS84) representation system (the system used to represent global coordinates by for example GPS systems), more advanced representations might be possible, particularly in cities representing streets, and quarters, as well as other physical places to get other services. Some systems, such as GeoSPARQL[108] extend the RDF representation to allow for spatial querying of data.

²<http://sweetontology.net>

Indoor Location Equally as important as outdoor location, so is indoor location. Most implemented smart environments are indoors, for example at home, or office. These locations provide crucial semantic, and contextual information to the system, and thus needs to be properly modelled. Because indoor location systems are not as advanced as outdoor, possibly because of the added technological challenges and lack of demand; there are not so many indoor location standard representations (other than extensions of WSG84, or simple Cartesian system of coordinates) means that there are neither standard indoor location ontologies (aside from universAAL IoT Location ontology).

Time Not only it is important to properly represent time, and time zones in AAL; it is also important to represent repetitions in time (for example for pill dispenser or exercise reminder services). Date and time (with time zone) can be trivially represented by using XML date Time which is also one of the default datatypes of RDF. There is also the OWL time ontology extending time concepts like relations between instants, and alternative calendars (sort of units of measurement for dates). Time synchronization will become a crucial issue in distributed systems, and as such it requires the best semantic representation of it.

User model in AAL many semantic statements are about the user. Not only the user profile should be properly defined, including possible sub-profiles for preferences, security; the different types of users are semantically relevant in many services. The most standard user model is the FOAF (Friend of a Friend) ontology³, which models persons, their activities and their relations to other people and objects; it is in fact one of the basic ontologies used as example in many semantic and ontology modelling courses.

The user model should also include a model for user activities, since many services rely on this information, and even the smart system it self will react to the current activity of the user.

Another recurring representation needed for AAL services is Rule representation, and reasoning. When the data is represented semantically, simple if-then-else rules can be very powerful, and indeed some solutions could be defined by just a set of these rules. Although reasoning with rules is right at the boundary of the semantic model, as the semantic model only cares about the data representation, and not how the system reacts to the data instances; because some semantic reasoners do indeed utilize standard semantic rule languages to perform inferences, rules should be part of the semantic model as well. When the SDS is very rule oriented, it is recommended to have its own rule model, which methodologically is developed very similar to the semantic model, and it is a layer on top of the semantic model.

³<http://xmlns.com/foaf/spec/>

4.1.2. CONTEXT MODEL

In abstract terms the Context is operative state of the world. This means that the Context contains everything and anything that can be depicted about the world which is relevant to the system or component. The operative state can be defined as all the instances of concepts declared in the operative domain of the semantic model. Therefore the context model is something that is restricted by the examined system or component. Because the state of the world is only relevant to a single instant in time, the Context is dynamic and changes with time.

Because the Context is based on the Semantic model, any piece of information required by the context model needs to be present in the semantic model. In contrast to the semantic model the context model is not about how to represent information, but about the information represented relevant to system. The Context is essential for any AAL system, the context is where the end user “lives”, and where the system operates.

The context may, or may not have storage support, this means that the semantic instances may not exist anywhere. This is because the subsystems which provide the Context may work in many different ways. These systems may react only to changes, announcing the new status only (Active Context Providers). They may only provide the state on demand, when queried (Polling Context Providers). They may determine the state through additional reasoning from other contextual elements (Reasoned Context Providers). The systems may be envisioned, but not deployed (see deploy model 4.1.8). Other components may just be context concentrators, providing a single point of query for the last known world state (Context Hub). Thus all the components which provide Context will need to be represented in the Context Model.

The Context Model, not only defines what information is there, and how it is collected and processed; it represents the workflows of how the state is distributed. Context information is consumed by members of the system, in order to do their operations, and while these operations may not be part of the Context model, the fact that the member requires the Context to operate is.

Graphically, the Context Model can be represented as a flowchart where the Active and Polling Context providers are the origin of the flows, and Context Consumers (and Context Hubs) are at the end. Context Hubs and Polling Context Providers, have representations of their on demand services, as well as the members who call these services, and optionally the conditions under which these services are called. Each Contextual element could be represented as a different flow of information.

Although not restricted to it, the most common Context Model in AAL is related to the Ambient Intelligence Context of the user’s home. In this model we could see all the smart environment sensors as Context Providers, and the applications as the Context Consumers. This way we could have simple Contextual Elements, which could be stated as semantic statements such as, *UserX is physically located in the kitchen; the stove in the kitchen is switched on; UserX performed the activity*

Cooking. Where the sources for the first is a RFID sensor in the kitchen, a connected stove, and a simple rule based reasoner. The consumers would be the reasoner (for the first 2 Contextual Elements), and an application with the purpose of measuring the time the user spends daily for each activity.

4.1.3. SERVICE MODEL

In the AAL economy all systems are attached to some sort of service. This services may be traditional Tele-assistance services, which involve human resources to be executed, or simple information services like weather services providing valuable information to the user (some times depending on the context of the user, in the example of the weather service an important context is where the user is, or where are they heading).

There are different levels of services. Services described above correspond to a more economic definition of services. However Services can also be defined from a technological view throwing definitions like those provided by Service Oriented Architecture (SOA)[60] or micro services Architectures[59]. Whichever level of service, they too need to be covered by the semantic model. Both the description of the service, and the results need to be modelled. Designers of the system should be able to determine usable services, as well as the expected results of these.

When the services are semantically defined, then it is possible to have semantic interoperability[109]. Typically services are defined using contracts, in which all the preconditions, operations, and end results; with optionally exceptions, special cases to consider and subcontracts explicitly declared. In technological services, these contracts are quite strict, requiring inputs and outputs to be of a specific type, as well as in an specific order, all tied to a univocally named function which is called using a specific language; this is syntactical interoperability. In contrast to syntactic interoperability, semantic interoperability extends syntactic interoperability by allowing flexibility in all of the aspects of the service execution. This is done through a broker which maps the request with the supplier(s) dynamically as needed, therefore:

- operands (input and outputs) do not need to be a specific type, as long they can be semantically mapped through the semantic reasoner,
- the order of the operands is not strict, as long as the set of the required operands stated in the supplier, is a subset of the operands in the request,
- the service function does not need to be known by the requester, as it too can be mapped semantically through the use of semantic reasoner,
- the language used to invoke the service can differ from the language used to define the service execution,
- because the service requested is matched against the supply of services there are a set of additional benefits:

- there can be more than one supplier for the same request (similar to economics's concept of competition)
- both the request and supply can be more specific as the preconditions (such as contextual elements), operations (how the service is executed), end results (desired status), special cases, exceptions, as well as other explicit restrictions can be part of the matchmaking process.
- a service request can be issued without knowing whether there is a supplier fulfilling it at all.

Semantic interoperability brings services to a human level of operations, in the case of SOA, the services behave more like contracts. When dealing with Semantic services, it is useful to think in terms of a marketplace, where if there is enough demand services will emerge, and if there is very high demand more than one supplier will offer the service. Therefore when defining services, both from a requester and from a supplier perspective it is important to define clearly the contract. The requester will typically need to specify more explicitly the required service, as to minimize effort, duplicities, and probable special cases consequence from abstract services dealing with more concrete services. On the other hand service suppliers will want to define, and implement, their services as broadly as possible, making their service as reusable as possible. Special cases can, and should, be defined in more explicit services, after all in the marketplace of services nothing impedes suppliers from having more than one service, this unloads the generic service from any specifics and the client will not notice the difference as the special case will be selected only when the request and supply are matched.

Another advantage of semantic interoperability is service composition. Given a complex request it could automatically be broken down into simpler requests, matching simpler services. Or a supplied service can be defined by simple rules and a set of sub-contracts or sub-requests. Either way semantic services have a better chance to be automatically composed.

Purely syntactical services can be included in the semantic interoperability picture, for this a special gateway adding the semantic layer to the syntactic services, usually this requires manual definitions. In fact this is how most of the semantic service frameworks work, by enabling existing services to be matched through the semantics of their registered profiles to requests.

The Service Model should include all the services in the system, whether syntactic or semantic, as well as the semantic gateways. Including all services may not be feasible, specially if the system is connected to the internet where there are vast number of services available, the model may be cropped to focus only on the operative domain. It should also include the full contracts of said services, specifying all the specifics of the services involved in the system. The service model includes internal and external services, this will be specially relevant for the distribution model (see 4.1.7). The service model notes services offered by the system (as potential external services for other systems), as well as the services consumed by the system (even those self-fulfilled the system both offers and consumes service). It should also

include the service compositions and dependencies, when possible, providing a clear workflow of services.

The objectives of the service model are:

1. to provide an overview of the (relevant) service marketplace for the given system,
2. to provide an implementation plan (through service dependencies, for testing and validation)
3. to identify any potential semantic gateway needed (transformation of syntactical services to semantic services)
4. to identify any potential service needed,
5. to identify any potential service which can be provided,
6. to identify any potential service redundancies,
7. to identify any potential service compositions,

4.1.4. USER INTERACTION MODEL

The UI allows any system to coherently interact with the end user. In AAL this is particularly important, as well as challenging. The user needs to understand the need for the system, if the interaction fails to inform the user, the user will deem the system as useless, not perceiving the benefits they get from it. On the other hand if the system is too verbose, the user will perceive the system as too intrusive and cumbersome. Additionally interaction mechanisms with AAL users, particularly elderly need to be specially adapted, as in most cases these users have a particular set of restrictions compared to other users, such as poor technological savviness, cognitive decline, or visual and hearing disabilities.

Modalities are the interaction channels the system uses to interact with the user. Typically this may be through a screen, or through audio, or a combination of both. For an AAL system it is important to have multi-modality interaction, as this way if a communication fails to be internalized by the user, another modality could be used. Each modality of the system has intrinsic properties, as it is not the same to interact with the same application through a computer terminal, a smart phone or a smart TV.

Of course modalities may not be compatible, for example images can not be rendered through auditory modalities, this is usually solved by adding content description to the image, making it accessible. This fact is particularly relevant when considering the design of an application's UI. The UI design must be able to be rendered in screens in many sizes, as well as automatically translated to other modalities for end user consumption. Thus through the UI model the analyst or designer would be able to detect when a particular user interaction cannot be fulfilled because of

the content and potential modality incompatibility. To aide in this exercise it is also recommended to look in to the wealth of research in the Accessibility domain.

Generally interaction methods can be either pull or push. Pull type interactions are when the user intentionally requests some interaction, an example of this would be traditional web page surfing, the content is not shown to the user until the user request it by clicking on a link. Push type interactions are more becoming common, this is when the system takes initiative; for example to notify or alert the user about some circumstance (e.g. Push notifications). This is an important distinction that needs to be considered in the UI model, as the handling mechanisms as well as the planning are not the same.

In this context the UI may be seen as a special kind of service where the consumer (for pull) or suppliers (for push) of the service are the users them selves. However it should not be confused with the service model, service model may interact implicitly with the user, or their context, whereas the UI model is used to define explicit interactions with the users.

Another important distinction between UI and service model, is that typically UI is designed as a loop. When the user is presented with an interaction opportunity there are several options the user may decide, for example not only pressing buttons would potentially initiate other interaction opportunities, but the user may also decide to ignore the interaction opportunity altogether. These interaction flows should be represented in the UI model, showing all the options the user has when presented with each singular interaction opportunity, showing the complete graph with all its loops. There is much research into how to develop these UI workflows, the focus of this document is not so much on how to develop these interaction flows, but to use the results of these designs to analyse potential problems and provide more tailored solutions for AAL SDS.

Finally The most important difference with the service model is that the interactions with humans tend to be in many different languages (as opposed to services using machine languages to interoperate). Most applications today are multilingual, which mean they have the capability to switch languages, depending on the context, typically provided by an operating system. This is a laborious task, as it needs to be performed manually (people tend not to know more than 2-3 languages), and in most cases human languages are highly contextual, meaning the information has to be displayed in a particular way (e.g. syntactic order) which may differ from other languages. In AAL multi-language applications is a huge challenge, setting aside market size and demographics, some AAL applications such as cognitive test or rehabilitation programs rely on language. Not only affecting the language skills of the user, but also using these skills to access other cognitive abilities to be tested or trained. These type of applications cannot be directly translated.

The UI model until now, looks only into the traditional theory of Human-Machine interaction. Adding semantics to the User Interaction Model brings a plethora of features, such as:

contextual awareness is defined as the system's capability of knowing the Context of the user (see 4.1.2). Knowing the context allows the system to be

weary about when, how, and where is the best way to present information to the user. Some cases include, only showing dialogues to the user when the user is home, switching modalities when some are in use (e.g. using audio to remind users of something when they are watching TV)

privacy As extension of the contextual awareness, privacy can also be taken into account to determine when and how to show sensitive information to the user. For example, the system might determine it is best to use smartphone screen when the context indicates the user is not alone, or the smart TV when (s)he is alone in a room.

declarative interaction When the internal information representation of the system is semantic, then its structure is declared in an ontology, to show this data to the user it is only needed to declare how this data is presented. This could also be done through a special UI ontology helping map models to renders, for which it is imperative that the displayed information is also part of the semantic model (see 4.1.1). Declarative interaction is not only useful for displaying information to the user, it is also useful to allow the user to directly edit the information. It is the triad of the Model View Controller, where the Model is the data used by the system, the View is how this information shown to the user (through UI ontology), and the Controller is how to edit the Model from the given input of the user (which is mapped directly back to the model).

specifics of the modality There is always the possibility to define metadata which refers only to a specific kind of modality, it is possible to add annotations describing how the different pieces of information should be rendered in a particular modality. This means the application could indicate for example which color to use or how to group graphical elements assigned to the information when using a graphical based modality; or which pitch of voice when using a voice type modality.

distributed interaction Although hinted through out this section, the modality renderers will be distinct components of the system, and as such they can reside in different nodes in the distributed system, thus it is possible to interact with the user in different places in different modalities. Depending on the deployment model (see 4.1.8), several instances of the same renderer can be installed in different nodes, then the system will be able to migrate interactions from one renderer to another according to the context change of, for example, the user changing location, effectively implementing a “follow me” experience from the user’s perspective. However it is important that the system works regardless of the distribution, for example in GPII if the interaction node cannot connect to internet, then the self configuration functionality is lost, but this should not affect the fact that the user can still interact with the system.

4.1.5. MEMBER MODEL

A member is an implementation of a single provider or consumer of Context, Services, or UI. Therefore the members are the operational blocks performing the interaction within and between systems, these can take the form of end points, clients, Inter Process Communication like message queues, or just a piece of software like an object implementing an interface. The Member model is used to map how these blocks logically interact with one another, using the previous models as guidelines of what kind of interaction these members can do. Essentially the members compose what would be considered in syntactic interoperability the API (Application Programming Interface).

Even though a member is defined as having a single operation, sometimes members can perform more than one operation. This is only possible when two or more operations of the same kind are merged into a single more generic or abstract one. For example given context consumer members for *kitchen lights*, *kitchen presence sensor*, and *smart stove*; could be merged into a single context consumer member for *kitchen context*; 3 members (and potentially more) have been merged into a single abstract member how is performing the same type of operation (context consumption).

Equally a member could be split into two, or more, members to accommodate for more specific operations. Take as example the case expressed in the service model where specific cases can be implemented as an additional service provision member. The member and component model should also help take the merging and splitting decisions.

In distributed environments members can be connected using different strategies either coordinated or non-coordinated; and this will be relevant in the deployment model (see 4.1.8). Coordinated strategies require a node that will perform special actions that other nodes cannot perform. Thus typically for coordinated bus strategies a fixed, powerful, high bandwidth node must be present, since most part of the extra network traffic will flow to and from this node. The need for a coordinated strategy is typically based on the need for a centralization of resources, i.e. a place where information and services are maintained for the whole system to use. The centralization of resources is typically done through one special member, the coordinator member, which handles its special role differently to other members.

The advantages of a non-coordinated strategies are that the nodes, and members, implement the same basic strategy. The network traffic is not concentrated in a single node, and therefore the optimization of the network resources might be better. Furthermore, there is no single point of failure. For coordinated buses the bus strategy must take care of the possibility of coordinator failure, whereas for non-coordinated bus strategies the failure of any single node will not affect the performance of the system.

Members have another property according to the intrinsic type of messages, there are mainly two types: called-based, and event-based members. Event-based use mainly event messages delivery of which does not require a response. Typically

these messages are useful to broadcast information (such as context see 4.1.2), mark an occurrence in time, synchronize peers, among others.

Call-based imply that the procedures are initiated by a request message which are followed by response messages. This paradigm is used when requesting services (see 4.1.3). A particularity of the call paradigm is that the calls (request and response procedures) can either be synchronous or asynchronous. Both cases require the member issuing the request to be specifically designed for the synchronization method used.

In case of synchronous callers the execution is paused right after issuing the request awaiting for the response. Special attention has to be placed on the cases where the response will never be received. For example when premature termination of any of the acting members, inhibits reception of the response once the request is issued.

Asynchronous callers typically implement a callback that is called by the bus strategy when the response is received. For the callers have to register with the request, so the caller associated to a response (linked to a request) can be located, and then called back, when the response is available. If there is no response then the member does not remain in a stand still, it may just have internal data that is not updated, but its main processes continue running.

4.1.6. COMPONENT MODEL

Components are sets of members, with internal logic, providing a coherent minimal functionality to the system, and therefore they are essentially the building blocks for the system as a whole. Components are the logical blocks conforming the business logic of the system. Components use their members to interact, or manage interactions, with other components.

The component model should be used as discussion on what a component is, particularly the definition of *coherent minimal functionality* can be disputed at different levels of modelling of the system, as well as the details of the component itself. This component model the analyst has an additional tool to make decisions on how the different components of the system are divided. How the components are defined depend in great measure of the Deployment Model (see 4.1.8), thus it is important to keep the distribution and deployment models in mind as we build the member and component model.

The representation of the Member and Component Model can be based on the UML component diagram, where every UML component is a component as defined here, and their interfaces are the members. There are, however differences that need to be noted:

- each member can connect to more than one other member, essentially n:m relations are not only possible, but probable. Bus-like connections will appear.
- each member is semantic in nature, thus the connections are not static connections, they are possible matchmaking opportunities.

4.1.7. DISTRIBUTION MODEL

The distribution model describes the nodes (where components may be deployed) and the relation each node has with every other node. This model looks into each node's capabilities, including computing power, storage, and network capabilities.

Network capabilities have special relevance as there are some special cases to consider on SDS in the AAL domain. Particularly untrustworthy networks, or unstable connectivity are prevailing in AAL, on one side elderly users tend not to have need for internet connectivity (although this is changing rapidly, extreme cases still exist) and therefore smart environments do not have this support. On the other Elderly people tend to live in remote and unpopulated areas, thus once again internet connectivity may not be as stable or as capable as needed for the system.

Processing and storing capabilities may also be relevant, for example training of Machine Learning algorithms such as neural networks, requires intensive parallel processing which may not be available on energy efficient devices such as smartphones. Similarly storage may be critical for some applications, and equally these capabilities need to be balanced and shared.

Nodes may also have other relevant properties, such as mobility. Smartphones can accompany the user in their daily activities providing context, services and UI where ever the user is. Nodes may be important because of the unique context, service or UI feature each can provide. Access to the context, particularly to the user's context is an extremely valuable property for nodes in SDS in AAL.

The distribution model should be used as a tool to consider all the technological context and the capabilities each node has. All these considerations will be relevant to the overall distributed system as well as the application. The representation of the distribution model may be formalized in many ways, for example network diagrams already include nodes, network segments and network devices. The model can be very complex, stemming from the fact that networks can get very complex, for example Virtual Private Networks can play important role/solution in the design of the system; similarly hardware virtualization may also play some role. The representation of the distribution model may also benefit from the explicit inclusion of the users' context.

In fact some systems may be designed to explicitly include more than one user's context, this feature is typically denominated multi-tenancy, where the user plays the role of a tenant. In a rented condominium analogy each user, as a tenant, occupies a unit which are independent apart from some common structures. In AAL systems multi-tenancy is desired when the provided service can be replicated and scaled, but also it can share common infrastructure, which in turn make the service more efficient and affordable.

There are typical node connection patterns, or topologies, which can be found in most distribution models:

Singleton is a single node, not connected with any other node for the bulk of the offered services (it can be connected for diagnosis, updates, or for other non-service related tasks). This pattern might be at first too archaic, and fallen

out of use in a world where everything is connected; but in fact this pattern is statistically most probable. This is because single use devices such as traditional clocks, electronic weight scale, non-connected smart TVs, electronics in static bicycles and exercise machines fall under this category: they offer one or more services without connecting with other devices. In fact this pattern should always be considered for high availability services, networks and network adapters may fail at any moment; or as stated before, there might not even be a network to connect to; in all these cases the node will fall to this pattern and it should still be able to operate and offer the service (or part of it).

Edge pattern is similar to singleton, but instead of a single node, there are several nodes connected in an isolated network. The same connectivity exceptions apply. This local network might be used to distribute sensing, processing, storing, interaction, and actuation tasks around the different nodes, but generally all nodes act as a single entity offering the service. Edge computing is a rising paradigm which offers several advantages over other pattern, for example being isolated attack vectors are reduced, also data ownership is clearer and data is not shared with any other party. Typically in this configuration the system is installed in the edge to collect data and offer services locally, thus a system is replicated but the data is never shared, a key concept in the design of these kind of distributed systems. In practicality Edge networks are in deed connected to other systems such as cloud services, but the power of the Edge nodes allows to offload networking and processing needs from the remote services; the data shared is typically aggregated and/or anonymised to maintain the philosophy of Edge privacy.

Cloud pattern refers to delivering hosted services over the internet (see 2.3.3). The key aspects are the service, which can be in principle anything, and internet, which implies that most certainly the user will need a connected device (client node) with internet connectivity to receive the service. The Everything-as-a-Service model of cloud computing defines three different levels of abstraction: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS).

Almost by definition AAL cloud systems must be multi-tenant, as it is rare to find private cloud system deployment (IaaS) per user, probably because this is too expensive, and not very efficient for AAL applications; although in some special cases it may be considered, see virtualization below. Multi-tenancy could be achieved through PaaS, where a platform could be instantiated for every application, or set of applications, ensuring the context of the user is maintained across these applications, even when these are offered as individual SaaS (see figure 4.2). In some cases the relationship between the application and the platform is too strong, or the platform it self can not be instantiated; in this case a solution could be to bundle the applications together with the

platform as SaaS. This is not optimal, on one side SaaS should be in the context of each app, this philosophy will lead to replication of the platform components for each app; on the other, the platform is contextualized to the application and not the tenant, thus sharing tenant information across different SaaS instances of applications can get complex.

Figure 4.2: Everything-as-a-Service: cloud-computing architecture and its association with AAL platforms and AAL services; AAL platform can be either part of SaaS (left) or PaaS (right)

A property Cloud systems have is that the user context can be extended, as the service is accessible through internet, any client node can be connected to it, in fact multiple client nodes can connect simultaneously. If all these nodes report to the same user, this means that the effective user context can extend across internet to wherever the nodes are. For example a user can access their services from home, the office, or on the go, and even at all those places simultaneously by using the appropriate client nodes.

Peer 2 Peer (P2P) is a connectivity pattern where every node has equal privileges, and the sensing, processing, storing, interaction, and actuation tasks may also be distributed amongst any node (see section 2.3.4). Although typical P2P are defined as nodes being equal, true P2P manages heterogeneity thus nodes may have different capabilities. P2P may be seen as a generalization of the above topologies, where it can be a singleton (when the whole accessible P2P network is just one node), or it can grow locally (similar to the Edge model), and it may also spread across internet offering distributed services (sharing some properties with the cloud pattern).

Even though theoretically P2P may be more efficient and robust connectivity pattern, it is far more complex to implement. One challenge is when reaching global consensus amongst all nodes, to resolve conflicting contexts reported by different nodes (for example, one node might state *UserX is at home* and another *UserX is at the Office*). This fact also puts the whole system under threat from possible attackers, exposing to what is known as the byzantine generals problem, where the system can be compromised by traitors. One technique to solve this issues is Blockchain, producing a ledger of linked statements (such as transactions) which every node can rely, produce and agree; as editing the ledger is difficult for attackers, as the linkage of statements is

typically resource consuming and handled by other nodes other than those producing the statements.

Hybrid pattern is just a mixture of the above patterns, when either refers to a node, this node could potentially be expanded to other connectivity pattern. For example we may have many edge networks connected in a P2P pattern; or edge networks which act as a whole as a client node for a Cloud; or a cloud service which is hosted in a P2P distributed network.

As seen all these patterns produce a unique set of properties which affect the handling and representation of the user context. Thus the distribution model is an important tool to analyse the form these users' contexts have and apply accordingly to the design of the system.

As well as topologies, it is also interesting to look into node patterns, as these too are commonly used sharing many features:

Personal Computer (PC) a desktop or laptop computer may be used to offer services. Increasingly these computers have become more powerful including for example graphic cards capable of training and executing complex neural network algorithms, as well as cheap mass local storage.

Single Board Computer (SBC) in contrast to PCs, SBCs are designed to be power saving devices. Not as powerful but still very capable computers, these are very commonly used in AAL to be continuously operational for monitoring sensors, and continuous connectivity. They may also have interactive properties, having simple control screens, or even connected to TVs as Set-Top-Boxes. They are also cheap, so it might not be uncommon to find these as part of Edge networks, P2P networks, or as cloud clients.

Mobile and wearable are a category of nodes which have the property of being portable. They typically go along wherever the user goes, although in AAL this might not always be the case, as elderly people tend to forget to take the device with them; thus the system may have to compensate for this fact. These nodes offer a unique view on the context, as they can not only track the user in their outdoor movements, they can also perform roaming. Roaming is the capability to switch between networks, in the distribution model this means that for example a node can connect to distinct Edge networks depending on the availability.

Virtualization Nodes and networks may be virtualized. Virtualization is the dissociation between the hardware and software, transforming what otherwise be hardware into a software representation of it. This means that the host (the actual hardware supporting the virtual component) must have enough resources to simulate the virtual system. Virtualization is also useful to isolate operations in a shared host. Common virtualization of nodes is used when the hardware required is very specialised and needs to be shared, for example

graphic cards to execute machine learning algorithms can be shared amongst many users through the virtualization of the node. Virtualization also applies to networks, effectively creating overlaying networks which can span across many host networks, which adds another level of complexity the distribution model must contemplate.

A special case of virtualization is Proxying, the proxied component is typically unaltered, which means it can be physical, however it virtualization, the proxy, is a dummy component which interacts directly with it. This means that other components may use the proxy as if it were the proxied component with negligible performance downgrade. This would still be true when the proxy and the component are distributed, remote operation of the component is performed by the proxy, giving the rest of the system the appearance the component is local. This is a very common technique in distributed systems, and thus it should be considered when designing the deployment model (see section 4.1.8).

4.1.7.1. INTERNET OF THINGS DISTRIBUTION MODEL

Internet of Things is an emerging paradigm, which addresses specifically distribution problem. As such it proposes an instance of the distribution model for IoT applications. This model can essentially be broken down into 3 main layers, corresponding with some of the connection/node patterns described above:

Device is the node which actually makes the sensing, or actuation. Typically these nodes are power efficient and connect to a gateway, but others can connect (through WiFi or wireless internet connections such as GPRS, UMTS, or 5G) to cloud services.

Gateway is an optional local concentrator of devices, acting as a local coordinator for edge operations, as a simple packet commutator to get information to and from the cloud, or both.

Cloud is an optional global concentrator of devices and/or services.

As it is this simple categorization allows to build many of the existing IoT Distribution models. Such as the typical example in IoT for AHA where wearable, and other home devices, connect to a smartphone gateway, which in turn resends the information to cloud services which can do their business logic, like Machine Learning, and return back to the gateway with results, or to other devices for other interventions, such as from formal caregivers. Other distribution models include, the devices which connect directly to 3rd party service providers, and then show how this information is retrieved either by cloud system or gateway system. It can even show how devices, gateways can create Edge computing networks, and even how multi-cloud operations are composed.

There will be however important missing components in the model. For this reason it is proposed to extend the IoT standard distribution model for AAL with 3 more “interfacing” layers:

middleware , protocols, or IoT platforms are typically used to interconnect devices to gateways, gateways with clouds, clouds with clouds, and any other combination, furthermore these platforms typically offer common components to address common IoT problems, and these components may play an important part of the system as a whole.

Application is a layer which indicates the software component of the service, the business logic of the system.

User is extremely important in AAL to note with which node each user is interacting, and which application is providing support to this interaction, this not only helps define the interaction model, it also incorporates, at least partially, the view of the user’s context.

Hardware	Middleware	Application	User	
				Device
				Gateway
				Cloud

Figure 4.3: Template for Extended IoT Distribution model

Together with the existing hardware layer, representing the actual hardware being used, rows in figure 4.3 is one possible representation of the extended IoT distribution model. In each cell, the relevant component (hardware, middleware, application or user) using or being used for each of the 3 IoT layers would be represented. Then the connections between components would be represented as annotated lines.

4.1.8. DEPLOYMENT MODEL

The deployment model describes how components, and their members, are, or may, be deployed. Components may be deployed in principle on any node. However the deployment model refines the restrictions on whether a software component can or should be installed on a particular node. Primarily access to the context should

drive this discussion, however other factors, such as those discussed in the distribution model, like connectivity, power efficiency, computation power and storage capabilities may play also a role.

In distributed environments, software components maybe installed more than once, either on the same node or on different nodes, to achieve reliability, and efficiency features. In this context there are many rules to apply when dealing with multi-instance components, and some considerations such as the following should be considered:

- what would be the resources (hardware resource like: CPU, memory, storage; and network resources like: ports, bandwidth, ping, etc.) or node Capabilities (such as presence of a particular device, or access to particular context) required for a node to accept the software component.
- how many instances are required to be running at all times?
- are there an optional amount of instances which can be additionally deployed when needed?
- are there any special conditions that should trigger when a software component is started or terminated?

These may apply to per node, per network (cluster), or at a global system level. Particularly there may be exclusion rules forbidding running a component in a node if it is already running in another node.

Distributed environments should be considered as are ever changing, nodes can join and leave at any point in time. The Deployment model should consider these special situations too. The system should consider an internal accounting of the nodes, components, and the reachable members, so when a node (or component) leaves (or is uninstalled) this registry is to be updated, in some cases even specific actions need to be played, so unreachable members do not cause the system to fail when invoked. This problem is addressed as bus consistency, meaning that the knowledge of the bus strategy and the actual status of the AAL Space are synchronized. For coordinated member strategies, this problem is critical, especially in the cases of being the coordinator member joining or leaving the system.

There are tools which already automatize the deployment problem for containerized systems such as kubernetes ⁴, however event this advanced system needs to be supplied with a deployment model for the components to be properly managed.

The deployment model may be used as as tool to define which parts of the semantic model are deployed where, and in fact in SDS the deployment model should include this too. The complete set of the semantic model may not needed in all nodes, as long as the union of the ontologies deployed on each node matches the complete semantic, and each model only contains the ontologies it needs it should be fine. This also helps optimize the main drawbacks of semantic systems, which

⁴<https://kubernetes.io>

are heavy on process as an inference process needs to take place for almost every operation, thus lowering the number of ontologies in each node, should make this process faster.

Another important topic to be incorporated, and resulting from, the deployment model is the deployment strategies. Particularly how the next versions of the software going to be deployed, and accessed by the different users. AAL services typically do not have much tolerance to downtime, for example panic button services must always be available; let alone to discover bugs in production. In AAL typical deployment strategies ⁵ are:

Recreate Existing is terminated then the new version is rolled out. Only recommended in case of extended pre-production tests are carried out, assuring the new version is production ready.

Ramped (also known as rolling-update or incremental) the new version is slowly rolled out replacing existing version.

Blue/Green The new and existing version are deployed, after pertinent pre-production tests the traffic is switched entirely to the new version.

Canary New version is released only for a subset of users, when fully tested, proceeds to a full roll-out.

A/B testing New version is released to a subset of users under specific conditions.

Shadow New version receives real-world traffic alongside the existing version. Responses can be compared, and if they are the same (or fixed in case new version fixes responses) then terminate existing version.

4.1.8.1. DISCUSSION ON SAAS MATURITY MODEL

When designing and deploying components it is important to realize that many components will share code, and/or data. In the context of the deployment model, this will definitely play a role in the decisions and resulting deployment model, and not only for cloud systems.

Microsoft Developer Network has defined a maturity model for SaaS ⁶ with four levels of maturity (see figure 4.4). This maturity model describes how applications can offer their functions to multiple users, as a service, and particularly how the code and data sharing conditions apply.

Level 1, ad hoc/custom each requester has access to its own version of the component. Each instance of the component is independent from other instances, managing its own data.

⁵<https://thenewstack.io/deployment-strategies/>

⁶<http://msdn.microsoft.com/en-us/library/aa479069.aspx>

Figure 4.4: SaaS maturity model. Credit [110].

Level 2, configurable all requesters have access their individual instance of the component, but the code of this instance is the same for all tenants; and therefore the component must be configurable for requester-specific operation. Data is managed by the individual instances, but behaviour is configured statically for each, which differs from level 1, where the instances are not configured by have custom code for each instance.

Level 3, configurable, multi-tenant-efficient all requesters have access to the same instance of the component that needs to be able to distinguish between different requesters dynamically. The main difference with level 2 is that the shared instance is capable of dynamically adjusting configuration for each requester keeping different silos for the data for each requester independent, or even sharing data when appropriate.

Level 4, scalable, configurable, multi-tenant-efficient adds a load balancer to increase the scalability by having a certain number of tenants for each level 3 instance.

The four levels of maturity form a continuum with the first level having *isolated* code and data, and the third level having *shared* code and data, the forth being a scalable version of the third. In terms of privacy, the more code and data is shared for tending for a requested, the higher the risk for data to be disclosed between tenants. However, the more isolated the constituents are, the higher the demand on memory and processing power will be.

The first level is defined by completely isolating code and data for each tenant. This increases demand on maintenance e.g. setting up the instances is not trivial, specifically when a new tenant connects, how and who will manage it?. Another important case can be risen for managing design and implementation error, bugs, if a bug is found, how will all instances be corrected? Additionally, resources cannot be

shared between different instances of the component, thus they need to be duplicated increasing the costs.

The second level improves by sharing the code for all tenants, but it needs to be configurable as to handle the needs of different tenants. Because there is still one instance per tenant, the requirements on the infrastructure may be high, as the resources to run each instance need to be replicated for each. Data may not be shared between instances, thus other important resources are not shared, still being costly.

The third level only runs one instance of each component. The components themselves need to be able to handle multiple tenants, i.e. they are tenant-aware. In this case, security and privacy is partially delegated to the component, however this enables the component to share data between tenants if this is part of the business logic of the component. Because only specific situations are isolated, this level is already optimized for mult-tenant applications in terms of resource sharing and underlying costs.

The fourth level adds a load balancer for scaling up to very high numbers of tenants. Multiple level 3 (equivalent) instances are deployed with the load balancer distributing the tenants between the different instances. How the balancing is performed depends on the application different strategies can be implemented, e.g. according to authentication during connection establishment. Having multiple instances again increases the effort of configuration, and makes sharing information between instances more complex, but may be needed to allow scalability to large number of tenants.

4.1.9. SECURITY AND PRIVACY MODEL

The security and Privacy (S&P) model, although left for last, is actually one of the most critical views of the system. In particular the domain of AAL, privacy is essential, as the domain systematically deals with sensitive information such as medical records, geopositioning of the user, and timestamped activities the user performs. Privacy is protected mainly through cybersecurity mechanisms, although other explicit data protection mechanisms such as (pseudo anonymization).

Through this model what the system needs to protect, as well as the potential attacks on them, how to manage risks, and how the system will protect them will be clarified. S&P is one of those research domains which has already many literature, and methodology written about, thus this is not an exercise on how to best apply S&P methods, but how to methodically include those methods in SDS analysis and design. Having said this, we propose the following general methodology approach:

1. Identify the Assets, what the system must protect. For this we could rely on the other models above, each looking a different kinds of assets such as general data, contextual data, services, data rendered for the user, the subcomponents of the system, the nodes and networks, as well as the binaries themselves. Assets may not only be objects in cyberspace (as those mentioned above),

these objects might be used to protect real assets such as a bicycle behind a bicycle lock, or even more abstract such as health.

2. Analyse the Risks. Once the Assets are identified, it is easier to think about how potential attackers might get to break the protection. Methodologies such as STRIDE (see section 2.5)) may be used.
3. Risk management. Once the Risks have been identified, we need to assess how likely and how dangerous each risk is, a prioritization of risks needs to be performed. This prioritization is not to leave risks out, once a risk is identified it should always be addressed, the prioritization is to resolve any possible conflicts in the implementation of the protections. Other methodologies to classify and manage risks may be applied such as the DREAD method.
4. Security measures implementation. These are the actual implemented features to protect the assets. Each measure should be mapped to the assets they protect as well as the risks they try to mitigate. Note that Security measures are assets them selves and thus should also be analysed for vulnerabilities.

The Security and privacy model is something that permeates the other models, as depending on how they are build they may contribute more or less to S&P protection, and S&P model decisions might affect the other models as well. The most sensitive model is the distribution model, it is essential to understand the networks, as they are typically carrying the assets, and data flows, as they are typically the assets to protect. Thus to provide the necessary security elements in the appropriate places, the distribution model helps identify not only the assets, but the appropriate places to deploy security measures. Some times the model it self offers privacy solutions, this is the case of Edge computing model, if the boundaries, and components of the system are properly configured, the edge operations can hold any information of the user knowing it will never leave the secure space is the edge network.

Another example is the semantic model, as it may include rules for who and how can any entity access the different assets, as the assets are described in the semantic model. There are in fact some security ontologies which expressly model different security features such as access control, anonymization strategies, and encryption. This supports further protection of different assets such as context information, services, or UI expressed in the other data related models. Specifically the analysis should always ask who can this information/feature, and how can they access it (with which privileges, read, write, ...). Particularly UI needs to examine the context, as specially for privacy protection the context of the interaction with the user is critical, things like how might also be watching/listening? is the user being impersonated? has the user left?

On the other hand, maybe users do want to be impersonated. In AAL domain, users are not technologically savvy, and feel unsecure around technologies which makes users seek assistance. This assistance may be formal or informal caretakers, however when they need to access the system they will need to be authenticated

and authorised. To do so, one common practice is just to share the credentials with the assistant, and this is not good practice as then the assistant can maliciously or unintentionally access all the private information and services of the user. One solution is the concept of delegation, the user delegates only a specific group of access rights to the assistant, preferably with time limitations. This allows the assistant to perform the appropriate operations, and only those, with their own credentials. It also aides Formal cares with responsibility over may users, with delegation they don't need to create a workflow for each user in order to access the information, they can have their own workflow accessing all the users' information at once.

S&P policies are typically addressed from different perspectives, thus a functional approach to identifying S&P blocks is proposed:

Policy Database (PDB) the block in charge of storing and maintaining the S&P policies, this functional block should be particularly protected.

Policy Administration Point (PAP) This block is in charge of creating, updating, deleting and managing the policies on the PDB, generally an administrator is interacting with this block, but users should also have their own PAP basically to enact their GDPR rights (see 4.1.9.1).

Policy Decision Point (PDP) This block evaluates and issues authorization decisions based on the policies in the PDB and from the user, module, node, device or service (entity) requesting any action.

Policy Enforcement Point (PEP) This block is in charge of intercepting all entities' access request to an asset and enforces PDP's decision.

Policy Information Point (PIP) This block will provide external information to a PDB, PAP, or PDP, such as external LDAP attribute information.

Any of these blocks may be implemented in many ways, particularly they can be distributed or centralised, each with its own problems and advantages. At least identifying these functional blocks in the member and component model as well as in the distribution and deployment models will help implement, or ensure the security measures are implemented despite flexibility in the deployment.

Finally, the system should be auditable. All transactions and operations should be securely logged, as to be able to detect possible attacks (even those not analysed), and correct the system when possible.

4.1.9.1. DIRECTIVE COMPLIANCE

When dealing with data, and particularly with sensitive data, many nations have created a legal framework to protect those parties involve as well as to resolve any disputes. In Europe, the European Parliament and Council of the European Union have defined and passed The General Data Protection Regulation (EU) 2016/679 (GDPR). Because any company who wishes to operate in Europe needs to comply

with this regulation (among others); and because implementing it for Europe alone may not be economically viable, or in other words once implemented in Europe it can be applied globally; this regulation has become the de-facto rule-of-law to comply with globally.

When analysing the GDPR from the system perspective, i.e. those technical requirements imposed on the system, the following requirements arise:

the right to be informed so long terms and conditions, of the user's consent should be provided in a human-readable information, articles 6 and 7.

Transparency and modalities the information to the user must be concise, transparent, intelligible and easily accessible form, using clear and plain language, article 12.

the right of access the citizen should be able to see all the data that each of his providers have about him, article 15.

the right to rectification that is the citizen can get his personal data fixed, article 16.

the right to erasure that is to be deleted from the system, article 17.

the right to restriction of processing that means the provider will keep the user's data, but cannot use it for other purposes as stated in the consent without further consent of him, article 18.

the right to data portability the citizen is able to export her/his data in a machine-readable format, article 20.

integrity and confidentiality providing a highly secure environment with additional measures to guarantee that the data has not been inappropriately modified, such measures include pseudo-anonymising personal data or encrypting data at rest, article 25.

Auditing Records of processing activities must be maintained, including purposes of the processing, categories involved, etc., article 30.

Breach notification the supervisory authority within 72 hours after becoming aware of the data breach to make the report to the supervisory authority, article 33, and in some cases the user, article 34.

data minimization providers should not collect more data than necessary.

As seen the system should preferably include some form of consent management, not only to present to users with a consent form (which is curated to be transparent and informative to the concrete consent the user is agreeing), but it should also allow the user to enact all of their rights in an automatized way. In fact this consent

management should be part of the S&P policies, ensuring the system is always synchronized with the consent the user has agreed with.

The other requisites are common sense, like: anonymization, encryption of data, audit log (to detect breaches), data portability (native in SDS, by the semantic nature of the data).

4.2. VALUE ORIENTED MULTI-DIMENSION EVALUATION METHODOLOGY: THE SHOWCASE METHODOLOGY

There was the need for a specialised tool to describe concrete values of open platforms and solutions. Showcases are the concept represents generic features that can be translated to actual value for each of the stakeholders involved.

The showcase tool-set can be used for many other purposes other than evaluation. For platform developers, showcases are monitored closely as to make sure the platform actually provides the features necessary to enable such showcases. For marketing and communication teams, the detailed analysis of open platforms through showcases offers a simplified, value oriented view of both the open platform and the applications, and thus showcases can be used to structure dissemination and marketing material. As for evaluation teams, showcases offers, not only the possibility to assess the socio-economic impact of systems in an itemized way, it also provides a common framework to evaluate and demonstrate the full application portfolios, despite their diversity.

Under certain conditions this multi-functionality of the showcases tool may be counterproductive. The perspectives applied when using the showcases tool-set have, in some cases, intrinsic conflict of interests. One particular case is when using the showcases for both dissemination and demonstration; if the demonstration is not very refined, then the tendency is to overcompensate for dissemination perspective, thus the tool loses its value as demonstration tool. Therefore it is important to be mindful of the multiple purposes of the tool, and properly balance its use.

4.2.1. DEFINITION OF THE SHOWCASES AND TOOL-SET

Because showcases share many similarities with use cases, technical experts can create an initial a list of showcases. This first iteration could be based on groupings of the platform features. It can be extended with other potential features platform developers are aware are needed as well as features found in other common open platforms. Additionally this list can be extended by grouping of common use cases, as well as prominent Business Cases, and other inputs. Another source of inspiration is the Bain Value pyramid (see figure 2.4, in section 2.6), the first round of showcases will be representing functional value, later iterations of the showcases will escalate

the pyramid; this is of course challenging, as doing so becomes less technical and more abstract, and thus it is also expected to find less showcases in the upper stairs of the pyramid.

For each showcase a simple description is provided. This simple description typically starts with an abstract feature, which could be implemented through technology. To evolve it into a detailed description, technical, evaluation, business experts as well as other stakeholder representatives would argue the value for technology provider, service provider, end user and on a societal level the stakeholders that stimulate its use and/or pay for it (governments, policy makers, insurance companies). The detailed description also included this vision each stakeholder has, as well as the boundaries, relation between showcases and finally some examples.

Along with the description of the showcases, assessment tools for each showcase are created. The objective of these tools is to evaluate a given system against the showcase, not only determining whether the showcase is indeed implemented by the system, but to gauge the value obtained through its implementation, as well as the relevance of said value. These include a series of scripts, generally meant to be executed by human beings. The scripts are categorized in different levels according to the requisites to perform the test:

Analytical requires an analysis of the system or the code by experts; Looking at the SDS model, code, architecture, ontology, provided documentation, etc. The scripts will explain a rationale experts should follow to analyse the compliance with the showcase.

Technical Procedure defines a process to test a system for the showcase in laboratory, test, or controlled environment, typically further technical analysis is required, for example analysing logs; or changing the running environment. The script will explain how to collect data, and provide guidelines for the process of analysing said data. These scripts should be executed as part of the laboratory tests assessment, before larger scale deployment of the system.

Operational the test has to be performed at operation phase, these tests usually affect real users. These scripts are meant for pilots to execute as part of their operations, gathering data. To reinforce the need for pilots to keep track of these scripts, a data collection template is provided. The template tailored a set of data to be delivered per showcase; operational pilots have to do similar actions as described by the operational scripts to obtain the data. This tool also aimed to gather more economical data on each showcase as to evaluate the value on real life scenarios for each showcase.

Programmatical Procedure will define requirements for an executable module to be implemented, and the steps to use it to test for the showcase. The implementation will not be provided, since in some cases the module itself may have dependencies to the specific case to be tested. As such these requirements may be transformed into a set of unit and integration tests which would throw

the system into specific conditions where the showcase is manifested. The objective of this category of tests is to automatize the demonstration and data collection processes, whenever human intervention is not primarily required.

These scripts, probe and test the system searching for the particularities that each showcase would manifest.

Sometimes this manifestation is expressed in the form of Key Performance Indicators (KPI) which help quantify the evaluation of a system using the given showcase. The tool includes the definition of data that can be gathered, as well as how it is gathered. These KPI are also commented as part of the assessment tool, specially useful to help a multi-disciplinary team understand the implications of each KPI with regards to the showcase. These KPI typically focus on economic implications of each showcase for Service or Technology providers; as well as some general statistics of the showcase itself.

Another developed tool was a set of questions to be asked through questionnaires or focus groups to specific target stakeholders. This tool's main objective is to determine the potential value any system implementing the showcase could achieve; as opposed to the obtained value of a given system implementing the showcase (measured through the previous tool).

A list of showcases can be tailored to domains, platforms and even applications. Showcases may be relevant horizontally, for more generic or enabler showcases; they may also be shared by different domains, whenever they are relevant. Equally they may be implemented by different platforms and/or applications. This is the result of showcases being value oriented.

Showcases may be conceptually very similar to Use Cases[111], however the key aspect of showcases is that they are meant to highlight value (similar to the Bain & Co. Value pyramid, see section 2.6); whereas the concept of use case is oriented to conceptualize the system behaviour. Their similarities help understand the place of showcases, and even share methodologies, regardless it is remarkable how showcases and use cases complement each other.

4.2.2. PRE-SELF-EVALUATION OF SHOWCASES

Depending on the length of the showcase list, the evaluation undertaking might be overwhelming, thus to help plan the resources for evaluation, as well as establish a baseline for perceived value, it is proposed that the first step of the evaluation process is a pre-self-evaluation. This step is optional, and it is left to the evaluator to judge the benefits given the required effort investment.

Up to now, the system provider may not have been involved in the evaluation process, in fact it would help reduce the probability for any conflict of interest if they did not. The pre-self-evaluation process is performed by the system provider, and subjectively measures the confidence the system provider has on the system implementing each of the showcases.

This can be simplified just by stating which of the tests in the showcase definitions is the system a priori (without executing any of them) can pass. The results

are composed as a percentage of the possible tests over the total tests, grouped by showcase. These results can be presented as diagram, which may depend on the evaluation team preferences, the number of showcases, number of self-pre-evaluations, etc. Alternatives include, but are not limited to, bar chart, spider graphs, or shade graphics (the shade depends on percentage ranges).

4.2.3. DEMONSTRATION OF THE SHOWCASE

The demonstration of the showcases is performed via the execution of the scripts defined earlier. This demonstration can be used either for the next step, video recording, or for direct demonstration during the value assessment. If there is no demonstration capability of the system, then this step may be skipped.

Results of demonstration evaluation can be shown using the diagram specifications as expressed in section 4.2.2; where the percentage used is the number of tests passed over the tests performed. This will help compare results from both evaluations, and therefore also evaluate the quality of each showcase definition.

The data set collected for each showcase, as defined in the showcase full description, can also be partly used as weak indicator for compliance with the showcase. Given that providing data for a specific showcase does not imply that the showcase itself was implemented. It does, however, indicate that there is at least some interest in this showcase being implemented. The fact that there is some interest may have had a consequence that the showcase is actually implemented by the application, but this needs to be cross-validated using other information.

Other specific demonstrators can be developed to display the features of the system, These are typically desired for dissemination events. These demonstrators can be designed with particular showcases in mind, so they could be demonstrated for attendees in these events. The showcases are highlighted in the presentation.

In some cases these demonstrators can be built from different modules from different providers to build a completely new system. Thus these ad-hoc systems constitute hard evidence for the implementation of certain showcases which focus at ecosystem level (value beyond a single system, or where several systems cooperate).

4.2.4. VIDEO RECORDING

Showcases, and the testing scripts, constitute a great tool for the creation of dissemination material, particularly well structured video recordings. These recordings may not only be used as dissemination materials, they can be also part of the evaluation process, proving the means for evaluation teams to verify the actual implementation of the showcases, with out the need to travelling on site. Additionally they can also serve similar to interview or focus group results, as these sessions might also be recorded serving the dual purpose of providing more insight to the evaluation team and as dissemination material. Finally the video materials can be used as part of future focus groups and interviews, being a didactic and enjoyable way to explain the showcases more efficiently and thus obtain more accurate results.

Video recording may be a very resource and effort consuming task, and although video recording is not technically required for showcase evaluation, the remainder of this section will provide guidelines to follow in order to incorporate this process in the evaluation workflow, while also balancing its use as evaluation and dissemination tool. These guidelines may be implemented or adapted as it may fit the particular case of application.

The videos for each showcase are produced following the same generic script: an introductory animation of the showcase, up to 3 minutes; followed by one or two examples from different pilots (local ecosystem of service providers, technology providers, users, and maybe policy makers). Each example, up to 3 minutes each, would introduce the pilot, demonstrate the showcase within the pilot, and show some extracts from the interviews with their stakeholders (related to the particular showcase). The interviews, recorded either in English or local language subtitled in English, also help gather more information about the value on each showcase.

In order to enforce the video recording task, and to produce homogeneous video material; a team of people need to be appointed to coordinate and overview the task of video recording, and interface between the pilots, and the video edition, to produce material with specific objectives. The video material tasks need to be carefully planned. In order to minimize the load on pilots, each would only be assigned a subset of showcases to demonstrate through video, but the union of all pilot demonstrators will complete the list of showcases defined. This process can leave pilots without the ability to demonstrate their full set of showcase implementation. In this case this may not be the best process to consider for demonstration purposes.

The coordination team is responsible to create the material necessary for the introductory part, which will be animated. The coordination team will produce scripts and storyboards of how the animation should be. The production team can then translate the script and story boards to a comprehensible video for everyone, following a global style for all showcases.

Raw Video material needs to be reviewed several times in order to meet the video production team's standards. Describing these standards the production team may produce a style guide, to help pilots without recording experience to produce better video material. The raw video material produced will be in general mediocre, according to the standards, specially if this material is not recorded by professionals nor with the proper equipment and methods (also described in the style guide). However the video production team should manage to select the best material in order to produce professional grade results, or provide improvement recommendations for re-recording sessions.

The most important consideration is to stick to time constrain for each video, at around 10 minutes total. There are several reasons for this, first short videos take far less resources to edit and render. The review of the material for these videos is faster. The material production will be focused on providing most relevant material for the given time. And finally, short videos will help audience understand and keep engaged when presented.

While reviewing raw material, the quality objectives may shift towards improving

the end product (the video for a focus group, or dissemination video) rather than collecting video evidence of showcases, applications, and users using the systems. This may be exacerbated partly because of the difficulty, in some cases legal impossibility, to include users in recording events; and partly because of the resources needed to record video evidence may be underestimated.

4.2.5. VALUE ASSESSMENT

Once the showcase has proven to be implemented, either through specific demonstrator or through video evidence, the final step is to assess the value for all stakeholders. The potential value has to be checked, while demonstrating in real life the key selling points of the evaluated systems. This assessment is performed through a combination of focus groups, interviews and data collection related to showcases.

Stakeholders from the region; for example other vendors, policymakers of the municipality/region, etc., are invited in order to perform a focus group.

The sessions may last between one and two hours, within which, there is only room for discussing two to four showcases. Each of the focus group sessions may follow standard focus group methodologies. As a focus group methodology example, the attendants may be presented with the produced videos, or a presentation (if video is not available) for a showcase. After which they fill in an assessment form, where they rate each showcase on a scale of 1 to 5 from different stakeholder perspectives: assisted person, technology provider, service provider. Furthermore, they rate if in their opinion the showcase would have a positive or negative impact on technological aspects, organizational aspects, end user experience, ethical aspects. After the ratings have been performed, the moderator asked the participants to comment on their ratings and discuss any related point. The attendants should sign a consent form, so that their quotes can be used for the evaluation, and quoted, anonymously, on reports.

It is important to have a good balance between the stakeholders. Sometimes the discussions may derive out of scope, for example more related to the application shown in the video than to the showcase itself, however this may not be discouraged as it can also show the interest of stakeholders in particularities related to the showcase, or even detect new showcases for future iterations.

Produced videos are excellent for focus groups, not only they highlight the main concept for each showcase, they provide concrete examples, and promote discussions about the value declared by stakeholders in the videos. This material not only makes it easier for attendees to understand, as they can identify with stakeholders in the video, they can also connect the concepts to their local ecosystem. All while maintaining homogeneous experience for all focus group participants independently of where internationally the focus group is taking place. Videos may be (re)subtitled to local language, however because the focus group moderator understands the showcases and are native to the local language, the videos may be used as presentation material for the moderator to introduce or clarify in native language the concepts.

4.3. COCREATION METHODOLOGY

As the general development methodology, a cocreation framework is proposed to gather user requirements in collaboration with all stakeholders, including end users. The Logical Framework Approach [112] has been used as main inspiration for this methodology. Widely used in cooperation and cocreation projects, as it highlights the analysis and diagnosis of the community and their experience [113]. It has been used with the objectives of identifying the basic information from stakeholders in order to collect their needs and preferences [114] and explicitly address them when designing or implementing a ICT system. The Logical Framework is heavily focused on the user perspective [25]. The following elements (as depicted in figure 4.5) are to be determined through this process:

User needs The user goals, expectations and aspirations which a given product or service must satisfy to be considered right outcome.

Use Cases In inline with the definition of UML Use Cases[111], the set of possible sequences of interactions between the system(s) and the user(s); in a particular environment in order to achieve a particular goal.

Solution a combination of AAL services with technologies which support or enable the services.

Stakeholders actors who are providing a contribution to, or receiving a retribution from the solution. In AAL these these typically are:

Service Provider the entity which offers one or more AAL services, which may or may not be supported technologically.

Formal Caregivers typically employed by service providers, are the professionals how deliver the AAL service.

Technology Provider hardware or software producers, who offer technological support to AAL service provision.

Assisted Person the end users of the service, can also include Informal Caregivers like family and friends.

Policy Maker Governmental entity how creates laws and/or ensures laws are not broken, they typically have political objectives which align with other stakeholders.

The objective of the methodology is to further process the elements from a technological development perspective. It consists of three phases: analysis, adaptation, and assessment. During each phase, the elements of the cocreation framework are reviewed, refined and redefined, where specific values are eventually extracted and enhanced. This methodology is designed according to the experience gathered from previous initiatives and projects [115, 116], then expanded with related literature,

Figure 4.5: The cocreation framework, the building block of the iterative approach to derive the solution.

Figure 4.6: Overview of activities and tasks executed during the cocreation framework and their dependencies.

an analysis of the state of the art and finally consolidated through interviews and focus groups with end users and stakeholders.

Although initially conceived as a simplified waterfall model approach [83], the process necessarily needs integrated feedback between the different tasks and phases. Thus it resembles a more dynamical spiral model[84], or even more released based (non-continuously integrated) agile development model [85]. The execution is not a rigid linear execution; in fact it defines different break points where going back to previous steps is necessary. If, for any reason, one step is executed before the planned one, e.g., a meeting with a stakeholder can take place before and/or after presenting the solution. This might be useful when the process detects potential problems before the overall solution is deployed.

Each of the three phases is composed of different tasks and activities at the level of the element of the cocreation framework, as shown in figure 4.6.

4.3.1. SCENARIO ANALYSIS

During the analysis phase, services and solutions available in the market are compared with what is required to satisfy the users' actual needs. The objective is to produce a comprehensive plan of execution to deploy a new and/or improved solution.

4.3.1.1. REQUIREMENT EXTRACTION

This task focused on the user needs, analyses the requirements for the adapted solution, and the focus of these requirements derived mainly from the validation with users.

Traditional tools such as interviews, focus groups, and questionnaires may be expanded with technological demos, mock-ups, or previous iteration of the solution. These methods should be carefully prepared[117]; for example, careful attention must be placed to avoid positive reinforcement of validators, as they tend to agree with any proposition. Asking detailed questions, trick questions, describing wrong scenarios, or present faulty systems might help validators engage, provide true input and avoid bias from the interviewers.

The objective is to define and map user needs and expectations with values, attributes, and requirements. The function of values is to answer the question on how the final solution, and the paradigm it is based on, actually improves the lives of the end users, as well as benefit other stakeholders. Although at this stage showcases (see 4.2) might not be yet defined, this tool could be used as inspiration for value extraction from stakeholders. By pulling showcases from previous iterations, or experience, driving the conversation in terms of value the solution might provide a reference framework from which to approach value extraction. Alternatively, following the showcase definition recommendations could help arrive at value driven exchanges with stakeholders.

One of the main uses for the results produced by this task is to guide the activities in the assessment phase. By defining the main evaluation dimensions and define research questions to be answered.

4.3.1.2. USE CASE DEFINITION

This task is the first task related to the Use Case element of the cocreation framework. Formalizes the conclusions of the analysis phase in a set of Use Cases (UCs) [111, 118].

There are different levels of formalization, and there can be some abstract and generic UCs, which describe how the solutions works rather than why the solution is needed (these would be showcases4.2). Specific UCs also need to be provided for the AAL service definition and system development tasks. These UCs will formalize the interaction between AAL services and the end solution.

While defining the UCs, a revision of the requirements might be required, as specification of UCs typically pose new questions that were not considered during

that task. Any new UC added to the final set will always trigger an analysis of existing resources, to search for state of the art technologies and concepts which may help in the implementation or refinement of said UC. UCs might also be handled as part of available resources, and these can be reused directly or adapted to the requirements.

4.3.1.3. ANALYSIS OF AVAILABLE RESOURCES

This task bridges UC element to the solution aspects of the cocreation framework. It is a crucial task, as it will inevitably help reduce the costs of adapting and developing a solution, by anticipating potential conceptual and technological barriers, or, in most cases, taking advantage of existing developments as a stepping stone for innovation.

The objective is to research the market for all the resources needed for the solution, and these might include complete solutions, running services, existing use cases, business models, technologies (such as libraries, frameworks, and platforms), and even competitor solutions and valuable team experience.

The Model for Semantic and Distributed Systems (see 4.1) could be used as an analytical tool. On one side it could be used to analyse the existing technologies, helping identify the strengths and pitfalls for each. As well as understanding how these technologies can be put together. Finally, if the model is used in this phase, it can be used as initial design for the adaptation phase, where it shows the implementation required.

4.3.1.4. NEGOTIATION

This task is necessary to avoid interoperability problems between AAL services and the technological solution. The cocreation framework requires all stakeholders to participate in the process. This will on one hand improves the solution, as only certain stakeholders have the know-how and experience to implement specific, or parts of, UCs; as well as organizational, legal and economic constraints of each of the stakeholders. On the other hand, it will help to avoid design mistakes. For example, a solution could be designed assuming some service was provided by some stakeholder, but then it is discovered that the service is unavailable or provided in a different manner.

This task can benefit from the methodology explained for the development of showcases (see 4.2). Having a value oriented discussion, helps all the parties involved understand the value other stakeholders put on the different aspects of the final solution. This focus on value, not only will help streamline the negotiation between stakeholders, it also critical for all stakeholders to full understand the final solution, defining a vision for it, and the specific role each stakeholder has in the achievement of said vision.

4.3.2. ADAPTATION

During the adaptation phase, the required technologies and AAL services as defined by the analysis phase are selected, reconfigured, implemented (when needed) and deployed. The objective is to create a solution which can be evaluated in the next phase. This phase is the most relevant phase when the cocreation methodology is iterated, as in second and subsequent iterations this phase will have the results from the previous iteration's assessment phase not only in the form of new analysis results, but as applied experience.

4.3.2.1. AAL SERVICE DEFINITION

Centred on the AAL services involved in the solution, the definition of AAL services is derived from the UCs and it is required to better analyse the different aspects both the technology and the AAL service provision which needs to be fulfilled. The main objective is to coordinate the integration between technologies and the AAL services, as well as (re)defining the provided AAL services.

Typically AAL services are already defined, trusted and are being provided in real life environments. In these cases it might be difficult to change, or even define new AAL services, in these circumstances. However by carefully negotiating the improvements and added values the technology might add to these AAL services, it is possible to adapt these not to better fit the technology, but to better take advantage of it, making the overall provision more efficient through different metrics.

Some times during this process relevant requirements that are not defined are discovered; in this case it is advisable to review the requirements using the methodologies employed in the previous phase.

4.3.2.2. SYSTEM DEVELOPMENT

Focusing on the technological part of the solution, this task encompasses all the technology development process. Any system development methodologies may be used, however it is better to use methodologies based on UCs and requirements, as those have been defined in the previous phase.

The model for SDS (see section 4.1) was specifically developed as a methodology to fulfil this task, as it specifically analyses and designs the critical aspects of the system. This task should consider topics such as:

- interoperability between components,
- data, service and interaction flows,
- modularization and re-usability of the system,
- type and hardware requirements the system will demand,
- How the system will be deployed,
- security and privacy aspects of the whole system and solution.

4.3.2.3. AGREEMENT

An agreement from the different stakeholders has to be reached. The terms of the AAL services provided, the system execution, the required adjustments, and any legal and economic aspect needs to be ratified.

In contrast with the negotiation task of the previous phase, where the objective is to help in the design of the solution, this task is meant to plan and implement the organizational structures and processes needed from the different stakeholders to enable the end solution to work seamlessly.

In concrete this task is the organizational aspect of the implementation and execution of the AAL services and system as defined in this phase. In case organisational, legal and/or economic barriers are encountered through the agreement process, these should be relayed to the other tasks in this phase to be able to refine the whole adaptation to these barriers. Hypothetically these issues could be transformed into requirements, and be handed to previous phases, however during the negotiation tasks considerations for mayor barriers should have already been analysed.

4.3.3. ASSESSMENT

During the assessment phase, the solution is inspected in order to see whether it fits the user needs and requirements. The evaluation is carried out along different axes such as technological, economic, organizational, human and social factors.

The assessment is guided by the definition of an evaluation framework, which in turn define indicators and how to collect them. These indicators will help answer, among other, the following questions:

- Does it fulfil the requirement?
- How good does it fulfil the requirements?
- How secure is it?
- Are there any improvements that can be made?
- What are the expertise the involved personnel needs?
- Can the solution be divided into smaller sub-solutions? (allowing for personalization of solutions)
- How easy is it to install and maintain?
- Is the solution stable (e.g. indicators such as mean time between failure)?
- How much does it cost?
- Can it be exported for other stakeholders?

- Where are the liabilities of the system? Does it comply with laws and standards?
- Can the solution be extended to solve other problems?

Careful readers might have noticed that these questions map to the cocreation framework's building blocks; and indeed this is one of the many methodologies which can be used to derive the questions to be answered, and thus help define indicators.

4.3.3.1. EURISTIC EVALUATION

The objective is to validate the solution and ensure it will satisfy the user needs in an efficient and economically viable way. In the first iteration, heuristic evaluations such as usability tests, interviews, and focus groups are enough. However subsequent iterations should include extensive validation determining the limits of the solution as well as rigorous testing before real life operations begin, determining not only whether user needs have been accomplished, but also how much does it cost to operate the solution and whether there are other potential business cases and solutions to be developed.

4.3.3.2. SHOWCASE EVALUATION

In order to determine the potential value of the solution, the Showcase Methodology (see section 4.2) can be used. The approach to to define the solution in terms of value, i.e. showcases. This analysis will be useful to determine the provided value as well as the actual value and even the potential value of the solution (for future iterations of the methodology).

This first step of the Showcase Methodology: defining the solution showcases, already provides good insight on the solution itself, by defining a framework which shows the values provided by the solution. Definition of showcases can be based on the existing AAL platform showcases, presented in section 4.4; extended with individual showcases provided by the solution. This step also provides some indicators to be used by the other tasks in the assessment phase.

The second step: pre-self-evaluation of the showcases, is used as validation of the showcase list. However the results of this step show what the stakeholders, concretely the stakeholders involved in the offering of the solution, most appreciate of their own solution. These results can be used to contextualize other results in the Assessment phase, or even answer evaluation questions about stakeholder expectations.

Third and fourth steps: demonstration and video recording, may be optional steps, however these steps can be very interesting as preparation of the materials needed for the final step.

The final step: value assessment, is where the actual, and even potential values are discovered. The showcases are presented to consumer stakeholders, which evaluate the system in abstract terms, this is interesting as it is not required for them to interact with the solution itself, and thus they are not evaluating the features of the solution, but the value the solution provides.

4.3.3.3. COMPONENT EVALUATION

In every technological endeavour testing is an essential part of the development process. As such individual technologies must be reliable enough to be used in their target environment, typically real life environments.

Component validation is the process of testing the individual components of the solution, certifying they each comply with the specifications, and ensuring each can perform as expected. Individual unit tests is one way to test components, particularly software components. Other types of components of the solution, such as hardware and even AAL services could be individually tested for compliance with their own specifications.

4.3.3.4. INTEGRATION EVALUATION

When designing and running evaluations, a small scale test must be included, not only to test integrated components, but also to test them in a controlled real life analogue environment. A Living Lab[116] is a perfect example of this environment, where the technologies and services can be executed in a controlled manner, as well as being the perfect environment for monitoring and dissecting the behaviour and the interaction of the users towards the solution.

During integration evaluation, the probability of finding issues and bugs in the system, AAL services, and integration between both is very high; thus this task must report to the other tasks and phases in order to improve the solution.

4.3.3.5. PROCUREMENT AND COST ANALYSIS

This is the process by which the stakeholders prepare their structures and finances to provide the solution in production, i.e. Real life environment. This may include comparison of supplier budgets and the negotiations with suppliers for acquiring the different required resources, materials, hardware and software, as well as hiring staff, training, and setting up the legal processes (consent forms, service contracts, etc.).

4.4. AAL PLATFORM SHOWCASES

A list of 14 showcases was carefully curated with general and common AAL problems in mind. The list might include other showcases from more general domains, but we deemed necessary to focus only on AAL specific showcases. Some of the selected showcases can also be applicable in generic domains, but are also important in AAL applications. An example of showcase that is being discarded is “system branding” which allows service providers to customize the system to appear as a self-produced service; this showcase might have had some value in AAL but we deemed it marginal compared to other showcases, thus it was discarded. This implies the list of showcases is not complete, and many more showcases can be defined.

Figure 4.7: Showcases for open platforms, placed in OPEA-AAL conceptual model

Administration The ease of administrating deployments, how easy it is to configure, install, and maintain single deployments.

Administration strategies might include remote systems management, simplification of recurrent tasks, powerful tools that allow management of extreme cases, process streamlining.

Advanced Distribution The capability of the system distributing its resources on different computing nodes. Distributing tasks and information through different networks to achieve a common goal.

Resources may be distributed in a local space, when there several computing nodes in the same space, or across internet, which is typically called cloud scenario.

Advanced User Interaction The user is able to easily interact with the system, in many ways, even involving different devices. The system is able to adapt the interaction to user preferences and the environment context.

Ambient Intelligence In smart environments, equipped with sensors and devices, the system is capable of reacting to the user needs at each moment as well as proactively engaging in conditioning the environment for possible future needs of the user.

This typically involves reasoner technologies that gather all contextual information from sensors to command actuators accordingly.

Enhanced Market Communication and Distribution Typically a digital store constitutes a market that enables solution providers to sell applications, and or content to the demand side.

This showcase argues additional features digital markets should have, for example representation of all the roles in the AAL domain like Technology providers (building the systems), Service providers (providing services and content) and consumers. All should be brought together in order to improve technology, services and quality of life.

Evolution The capability of the system to be enhanced with new services and adapt to new situations.

New services can be offered in an active deployment, without the need of replacing the old system. And systems can adapt to new situations in their deployments (for example transitioning from offering the service to a single user to multiple users in the same deployment).

Integration with Legacy Systems When building new systems there are always subsystems that are reliable and mature, that might very interesting to interface with in order to achieve the required functionality.

Not always new technology is friendly with old technology, thus this showcase ensures open platforms have methods to integrate other technologies.

Personalized Content ICT systems typically offer the user a set of content for him/her as part of the offered service. Not every content is optimized for every user, thus there is the need to personalize content.

The showcase ensures the content is personalized, but not costly for the system development, system exploitation, service provider nor the user him/herself.

Plug and Play Economic competition ensures that different vendors can produce different devices that accomplish the same task. Thus when the task is needed the system needs one of these devices.

The showcase ensures the system is not vendor locked in (exclusively compatible with a specific brand or model), allowing technology providers, service providers, as well as end users to choose the devices they deem necessary.

Resource Sharing Systems tend to use the same type of resources, all hardware, procedures, and information. When two or more systems are put together, to optimize costs it is expected that all systems share the resources.

This showcase ensures that this is feasible at all stages of development; and that demand side is not forced to acquire duplicated resources.

Scalability Systems have to be able to be deployed in many sites.

This showcase ensures there are easy mechanisms that allow for easy installation of large number of deployments with minimal costs. This allows the savings to be translated to the demand side too.

Security and Privacy ICT systems manage many types of information, much of which might be sensitive information from the user.

This showcase ensures sensible data is protected via easy to use and robust mechanisms offered by open platforms.

Service Integration With large ecosystems of services, it is expected that some will overlap.

This showcase ensures different services can coexist both collaboratively and competitively over information and capabilities; allowing for new and advanced features to arise from these interactions, at the same time lowering integration costs.

Service Transferability ICT systems can be deployed anywhere that meet the technical requirements, but in certain occasions there are other limitations that prevents this from happening.

This showcase ensures that constraints like language, culture, legislation, or locally offered services can be easily resolved without costly adaptations.

4.4.1. RESOURCE SHARING

Shared resources refer to the different AAL application or services which are able to share hardware, data and network resources thus allowing users to operate as if each application has its own resources. In the field of Ambient Assisted living, the most used shared resources are hardware and data resources. As main advantage, resource sharing reduces the amount of needed hardware to implement AAL services. From the other side, without resource sharing, each AAL application and service is built with separate platform. The above defined show case is implemented when

different applications share the same resources (e.g. two applications using the same presence sensor).

4.4.1.1. TECHNOLOGY PROVIDER'S VIEW

From a technology provider perspective, the current feature will offer them the opportunity to exploit their hardware by several services in parallel. This will motivate them to break down the chain hardware-software, thus providing their technology in more standard and generic way to be later on used by several application and services.

4.4.1.2. SERVICE PROVIDER'S VIEW

From a service provider view point, the “**Cross-application resource and capability sharing**” will open the door for several improvement and advantages. In fact, from hardware resource sharing point of view, this will avoid *Vendor Lock*, where the provided application will no more depend on specific device type or manufacture. From a data sharing perspective, the above highlighted capability will allow the application to be personalized through sharing relevant data and information with the other application regarding the environment, the user profile, preference... Finally, this will allow the services providers to directly interact with the other available services...

4.4.1.3. ASSISTED PERSON'S AND CAREGIVER'S VIEW

- From the assisted person perspective: The show case will offer him the opportunity to profit from more AAL services without being obliged to pay more for extra hardware, this will held to a more simple and non-complex hardware environment in the end user location. Sharing data with other application and service will allow to the end user to benefit from a more adequate and friendly user environment where the application can be automatically adapted to his/her need.
- From a Caregiver perspective: sharing resources will allow him to have a more flexible working environment, where other software different to the owned one by the user to be used to get the same kind of information (distant Blood pressure for example)

4.4.1.4. GOVERNMENT'S VIEW

From the Government perspective, the resource sharing, meaning while cutting the edge application-resources will be an entry for reducing the vendor lock. In parallel, it will increase the installation and the exploitation flexibility of AAL technology. Definitely, this will make a step forward toward facilitating scaling up the AAL culture in Europe.

4.4.1.5. RELATIONSHIP WITH OTHER SHOWCASES

4.4.1.5.1. COLLABORATION

- Scalability, sharing resources in individual deployments makes scalability costs more accessible.
- Service Transferability, services able to share resources, are better fitted to be transferred to other deployments (especially when they combine Plug and play showcase).
- Ambient Intelligence, sharing information is the basis of Ambient Intelligence paradigm.

4.4.1.5.2. NOT TO BE CONFUSED WITH

- Plug and Play, Sharing resources is when a resource is used by different applications; while PnP is when an application uses different (alternative) resources.

4.4.1.6. EXAMPLE USE CASES

4.4.1.6.1. HOME NODE A user at home may have several applications like Cardiac monitoring, Diabetes self-management, Nutritional counselor, Weight monitoring. Evidently each application will use different set of sensors (or data produced by the same sensor), these sets may intersect but those sensors do not need to be replicated per application. This way the user only needs one instance of each of the sensors e.g. heart rate, blood pressure, weight, blood glucose making a total of 4 sensors for all applications (even if the applications are provided by different vendors).

4.4.1.6.2. CLOUD A Service Provider is providing a service through a web application portal. When the same Service Provider plans on extending the service by providing a second application, this second application will be capable of sharing some resources with the first, the most important is the user authentication mechanism, so that users do not require re-registration. Other resources the application might share are reasoner rules and engine, and context information.

4.4.1.7. EVALUATION OF RESOURCE SHARING SHOWCASE

Procedure type	☒ Analytical ☐ Operational ☐ Technical ☐ Programatical
Procedure general description	
Determine if Resources are being or can be shared among different applications, by looking at all the available documentation of the application.	

Procedure script	
<ol style="list-style-type: none"> 1. List all available documentation. This may include, but not limited to: user manual, installer manual, design documentation, adaptation schema, issue report system, help fora, developer's documentation, Code comments, Commercial brochures, etc. All of these may be in different formats: web page, paper. 2. Go through all the available documentation looking for: <ol style="list-style-type: none"> a. The Ontologies used by the application, and the application components, are standard and/or generic enough to allow components to share resources. b. Looking at the Dependency graph (extracted from code or deduced from design documentation) there should be converging dependencies to the same components. Analyse is these components to see if they are actually being shared: <ol style="list-style-type: none"> 1) If candidate shared component is an ontology look if it has services, then look for service calls for these services in dependant modules. 2) If candidate shared component is an ontology, look for references to instances of its concepts specially look for cases where there more subscribers than publishers. c. Use of standard platform components is high indicative of resource sharing. All platform components are susceptible of sharing resources. d. Components that make available resources for other modules to use simultaneously thus enabling resource sharing. An indicator might be generic service profiles being published. 	
Success cases	
In step 2 some solid evidence indicating that at least 2 different components sharing resources (using the same services, hardware, etc. . .) is found.	

Procedure type	☐ Analytical ☐ Operational ☒ Technical ☐ Programatical
Procedure general description	
Share hardware resources thus allowing users to operate as if each application has its own resources	

Procedure script
<ol style="list-style-type: none"> 1. Identify at least 2 applications/modules/components that use the same device type; and a Hardware device that is compatible with both applications 2. Run all applications/modules/components, making sure they are connected to the same hardware abstraction module, and that the hardware device is responding / connecting with such hardware abstraction. 3. For each application/module/component test a workflow that will make use of the hardware device.
Success cases
<p>In step 3 the same hardware device is used for all applications/modules/components. Depending on the device type there might be different indicators. If the hardware device has to be explicitly summoned, then in step 3 the experience will be evident as each application will make the device work. In case the information is reused, i.e. the device is only read every time there is a status change, the experience in step 3 will be more integrated (make sure all applications get the same information, and react accordingly when the hardware state changes).</p>

Procedure type	☐ Analytical ☐ Operational ☐ Technical ☒ Programatical
Procedure general description	
Create a module that uses existing data, thus demonstrating such data is shared.	
Procedure script	
<ol style="list-style-type: none"> 1. Create a new module which subscribes to existing events, or calls existing service profiles; thus making use of already existing information from other functionalities as part of the existent system. Make sure this module is able to show to the operator that in fact is using information from the existing system. For example by launching a GUI screen that shows the actual information that is gathering. 2. Test the Perform the following show case demonstration task: <ol style="list-style-type: none"> a. Install/add the module developed in step 1 to the existent ones b. The installed bundle must be able to access to the information though the subscriber or service call mechanisms. c. The created module should perform the targeted task taking the shared information as a main input. 	

Success cases
On step 2.c. it is clearly shown the same information that is running through the system can be used by the developed module. By being able to perform these basic operations, the service provider\end users are assured to be able to make use of the shared information.

Procedure type	☐ Analytical ☒ Operational ☐ Technical ☐ Programatical
Procedure general description	
Demonstrate resource sharing capability in real life environments, by having several applications access the same hardware resources in deployment sites.	
Procedure script	
<ol style="list-style-type: none"> 1. Identify at least 2 applications that use the same device type; and a Hardware device that is compatible with both applications. 2. Deploy all applications and hardware devices. <ol style="list-style-type: none"> a. (alternative) the deployment can be done all at once b. (alternative) additional applications can be added to existing deployments with compatible hardware devices 3. Users should be able to benefit from all applications, where both are using seamlessly the same hardware resources. Monitor user issue reports, have questionnaires or focus groups for any indication that may show users are experiencing problems with applications not sharing resources correctly. 	
Success cases	
<p>In step 3 it can be proven that users are using all applications, and these applications are sharing the information from the hardware devices.</p> <p>Issues are expected; they should not affect more than 50% of the users involved in this procedure, and/or their severity must be low.</p>	

Indicator	Service provider focus group
Value centred in	☐ Technological Provider ☒ Service Provider ☐ Assisted Person ☐ Caregiver ☐ Government

Description
Questions: <ul style="list-style-type: none"> ■ Are the shared data/resources capabilities of the platform convenient? ■ How do you think the resources sharing capabilities of the platform will optimize resources and economize the cost when you want to add new functionalities?

Indicator	End user focus group
Value centred in	☐ Technological Provider ☐ Service Provider ☒ Assisted Person ☐ Caregiver ☐ Government
Description	
Questions: <ul style="list-style-type: none"> ■ Have you ever had to buy a new device to perform new functionalities that might be performed with an existing one? ■ How often do you think you would need to make use of shared resources 	

4.4.1.7.1. TESTING IF AN APPLICATION/DEPLOYMENT COMPLIES WITH THE SHOWCASE

4.4.2. PLUG AND PLAY

The “Plug and Play” showcase allows declining a fundamental characteristic of the platform: with PnP there is not “vendor lock-in”, so the end-user may be able to be independent on a vendor for HW products or services. The most important aspect is that the resources are interchangeable, so it is possible to buy similar services from a different vendors without incurring significant costs and risks. Another specification of PnP is the detection and the configuration of hardware devices with little or no user involvement. In fact the platform provides PnP of AAL applications and services, so the concept can be seen in an extended version. The main advantage is related to interoperability feature and adaptability feature; the platform permits with little effort the adaptability of application and services even if changes in hardware occurs. It is important to note that the previous definition should be contextualized on the basis of whole system complexity. For example, for a smartphone application PnP showcase should be read as no user involvement, whereas in the case of an integrated complex home automation system the PnP definition may refer to little as “necessary” involvement of technicians.

4.4.2.1. TECHNOLOGY PROVIDER’S VIEW

The ability of the platform to provide PnP both for hardware and services/applications increases the number of technology providers that can show interest in the platform although officially they are not involved in the project. The new vision of a PnP facilitates participation of providers that have developed applications in other open platform. Another advantage is that the provider will not incur in high costs for training of resources to devote to new technologies for AAL platforms.

4.4.2.2. SERVICE PROVIDER’S VIEW

From a service provider point of view, the PnP showcase will increase for example the capability to choose from different "default" hardware sensors for the provisioning of the services; this allows for service provider (SP) a cost optimization. Another interesting aspect is that SP may offer entirely different packaged solutions for a specific service, for example different software/hardware sets that may help SP to build service category business model (the more the client pays the more he gets).

4.4.2.3. ASSISTED PERSON’S AND CAREGIVER’S VIEW

In this context, the end users may have an indirect benefit; certainly there will be cost reduction for the following two reasons:

- installation work by personnel not highly qualified;
- ability to switch between different hardware or software solutions (e.g. an AAL service or application that can be exploited by different brands of specific product/sensor)

4.4.2.4. GOVERNMENT’S VIEW

From the Government perspective, Plug and Play showcase plays an important role because increasing in interoperability and adaptability of the solutions will facilitate the spread of a new concept concerning the use of AAL technology, increasing in very short periods of time the knowledge about the use of "open" solutions.

4.4.2.5. RELATIONSHIP WITH OTHER SHOWCASES

4.4.2.5.1. COLLABORATION

- Services Integration, increases the service flexibility, so it is possible to have a greater interchangeability between resources.
- The change of hardware/software resource involves some sense of Evolution.
- Administration. For some PnP devices some configuration aspects are necessary, so administration fills the gap.

4.4.2.5.2. NOT TO BE CONFUSED WITH

- Resource Sharing. Capability sharing enable to share the resource while PnP enhances the capability of changing the resource.

4.4.2.6. EXAMPLE USE CASES

4.4.2.6.1. HOME NODE An actuator for opening/closing windows (and the application for its management), produced by the vendor X is installed in the user environment. After a period of use, the device breaks and must be replaced by an actuator produced by the vendor Y; the new actuator is of a different brand but, thanks to PnP feature, the user can continue to use the application with which he had become familiar.

4.4.2.6.2. CLOUD Cloud-based architecture is directly related to PnP showcase. The “Plug&Play Cloud architecture” extends the techniques of modular architecture; for example you can replace, add or remove application or services without affecting the rest of the cloud structure. Further, extending this definition to be able to add, remove or replace modules (involved, for example, in the functioning of a specific hardware device) at “runtime” rather than trying them together at compile time gives us the adaptability and extensibility feature for every kind of customization.

4.4.2.7. EVALUATION OF PLUG AND PLAY SHOWCASE

Procedure type	☒ Analytical ☐ Operational ☐ Technical ☐ Programatical
Procedure general description	
Find among all the available documentation of the application, if processes, methods, practices or operations to perform plug and play are included	

Procedure script
<ol style="list-style-type: none"> 1. List all available documentation. This may include, but not limited to: user manual, installer manual, design documentation, adaptation schema, issue report system, help fora, developer's documentation, Code comments, Commercial brochures, etc. All of these may be in different formats: web page, paper. 2. Go through all the available documentation looking for: <ol style="list-style-type: none"> a. The Ontologies abstracting hardware are standard. These ontologies are either the official platform ontologies (look for them in the dependencies of the code) or internationally consolidated ones. b. The Ontologies used by the application, and the application components, allow for third party extension. Other developers are capable of extending or mapping the concepts to fit other applications/purposes. c. Hardware devices are abstracted. There is no explicit reference to concrete brands/models or protocols used to interact with the devices are in a separated module outside the application's business logic. d. Explicit mention of different compatible devices⁷ on especially indicative in user, installation or operation manuals. Typically if the service is offered with different devices, the manuals for the different devices are included in the user manual. Descriptions of workflows may change if the workflow is tightly bound to the device used; sometimes all workflows are described for the different devices. e. Mention of different hardware protocols. As long as these protocols comply with 2.c. . Compatible devices may use different protocols.
Success cases
In step 2 some solid evidence for at least 2 different devices the application works.

Procedure type	☐ Analytical ☐ Operational ☒ Technical ☐ Programatical
Procedure general description Testing the proper operation of an application even in case of device change (the specific device must be manufactured by another vendor).	

⁷Compatible devices are devices that have similar functionalities, i.e. they belong to the same device type. Compatible devices not necessarily mean they use the same protocol.

Procedure script
<ol style="list-style-type: none"> 1. analyse and annotate the functionality/services that the application offers 2. Using a device of vendor X: <ol style="list-style-type: none"> a. Open de application, and test the functionality discovered in step 1. b. Annotate the functionality/services that work properly c. Fill in Tested/supported sensors Indicator table with new data point for the device being tested. 3. Connect another compatible device of another vendor, repeat 2
Success cases
In step 2.b. the functionality/services that work properly, for at least 2 tested devices, must be the same as the available functionality/services analysed in step 1.

Procedure type	☐ Analytical ☒ Operational ☐ Technical ☐ Programatical
Procedure general description Demonstration of the capability of plug and play in real life environments and evaluation of the effort required (human resources, update in software, physical changes...) to perform a component switch.	

Procedure script	
<p>Operation must be planned to be able to demonstrate plug and play as hardware substitution is necessary. There must be also the possibility to have enough devices of at least 2 vendors to be able to switch them.</p> <p>Another option could be to perform the device switch because the initial device deployed in a site has failed, and can be replaced by another device model.</p> <ol style="list-style-type: none"> 1. Make sure the deployment sites are operational with current hardware. Or has worked in the past (in case of replace due to fail). 2. Replace the installed hardware in deployment sites for a compatible device. Make sure to update the indicator Tested/supported sensors with adding to number of users. 3. Annotate the mean estimated relative cost of the switch operation per deployment: $\text{<cost of replacing device>/<cost of installation>}$. Value < 1 means the replacement is less costly than a new installation; value $=1$ cost of replacement is the same as a new installation, value >1 cost of hardware replacement is larger than cost of a new installation. Costs must include everything (human resources, update in software, physical changes, issue solving...). <p>One option, to save cost is to have a subset of users use device of one vendor and other disjoint subsets of equal size have devices of other vendors; when approaching step 2 switch of devices can be performed between 2 sub sets (devices will be reused).</p>	
Success cases	
<p>When the replacement is performed no substantial issues are reported, the switch should be as smooth as possible in functionality terms (of course usability issues are expected). Mean cost of replacement should be strictly less than 1, the lower the better.</p> <p>In cases where the value is around 1 (± 0.1) further analysis should be considered, to research if the reason for the costs is due to external factors to the switch (for example not applying the proper procedures or deficient training).</p>	

Procedure type	☐ Analytical ☒ Operational ☐ Technical ☐ Programatical
Procedure general description Demonstration of the capability of plug and play in real life environments by deploying different compatible devices using the same software.	

Procedure script
<p>There must be at least 2 different compatible device sets. Devices that perform the same operation (for example taking blood pressure measurement), but are provided by different vendors, or are different models of the same vendor.</p> <ol style="list-style-type: none"> 1. When deploying, define sets of users according to the device model used. 2. Make sure that the user issue reporting procedure takes the different device models into account, and monitor issues.
Success cases
<p>All deployment subsets should have a comparable number of issues reported by users. Issues are expected, but the difference of issues should not be very high.</p>

Procedure type	☐ Analytical ☐ Operational ☐ Technical ☒ Programatical
Procedure general description <p>If the tester fails to find a compatible device/component to switch with, then a dummy device can be programmed to test that in fact the system is capable of using any surrogate device.</p>	
Procedure script	
<p>Given a component X (must include the source code for its adaptation):</p> <ol style="list-style-type: none"> 1. Analyse the connections with the platform and the connections with the device/component it self 2. Make a copied piece of software where the connections with the device are simulated through a simple user interface or random number generator. <ol style="list-style-type: none"> a. Alternatively, create a compatible device/component, for example using the appropriate ontologies (generalization of used concepts) that simulate the connection with the platform. 3. Test the system <ol style="list-style-type: none"> a. open the application, analyse and annotate the functionality/services that the specific application offers b. annotate the functionality/services properly functioning with the surrogate/simulated device/component 	
Success cases	
<p>For a successful implementation of the showcase by the system is achieved if the system is capable of performing all the functionality and services with the simulated component (3.b. is equal to 3.a.)</p>	

Indicator	Tested/supported sensors
Value centred in	☒ Technological Provider ☒ Service Provider ☒ Assisted Person ☒ Caregiver ☒ Government
Description	
<p>The purpose of this Indicator is to have a list of tested devices, to be able to compare the potential of cost reduction. Each data point consists on:</p> <ul style="list-style-type: none"> ■ Device Type (compatible sensors should have the same type, if a sensor has 2 or more types, then consider each type as a separate data point) ■ Device Brand and Model. ■ Purchase cost per unit. Cost of a single unit for an end user ■ Bulk Costs. Price that service providers would get if bought in bulk. When different bulk offers write them all, e.g. 50 €/1u, 45 €/5u, 40 €/10+u; means if 1 to 4 unit is ordered the budget price is 50 € per unit; if ordered 5 to 9 units the price per unit is 45 €; if 10 or more units are ordered then the price per unit is 40 €. Bulk costs should not include shipping, or packing fares. ■ Installation cost per unit. Any possible installation cost derived from having to send the unit to the user, or dispatching a technician. It may indicate the effort of installation. ■ Software/Drivers required. List of required artefacts needed for the device to work. ■ Subjective quality of the sensor (between 0 and 10). Preferably have a set of experts evaluate the quality and report the mean score. ■ Number of users that have tested this device. <p>For each sensor type we can deduce important information:</p> <ul style="list-style-type: none"> ■ The number of compatible devices; as an indicative of the potential for substitution. Difference between the most and least expensive devices, gives a measure of the potential cost reduction in substitution. ■ Cost vs quality ponderation, gives an overview of the current market for the device type. 	

Indicator	Questionnaire for TP & SP
Value centred in	☒ Technological Provider ☒ Service Provider ☐ Assisted Person ☐ Caregiver ☐ Government

Description
<ul style="list-style-type: none"> ■ In your opinion, can the procedure be performed with little technical knowledge? ■ Does the PnP showcase have benefits from the economic point of view? ■ Can the procedure be performed without help by an assisted person or a caregiver? ■ Do you have any suggestions for the improvement of the procedure? <p>Target Description: Technical provider and service provider focus group</p>

Indicator	Questionnaire for End-user
Value centred in	☐ Technological Provider ☐ Service Provider ☒ Assisted Person ☒ Caregiver ☐ Government
Description	
	<ul style="list-style-type: none"> ■ Did you feel there was enough information provided about the procedure? (assisted person) ■ Did you think the training/introduction was useful? (assisted person) ■ Did you feel there was enough support from the care organization? (care-giver) <p>Target Description: End-user focus group</p>

4.4.3. ADVANCED DISTRIBUTION

The platform allows applications developed on top of it to communicate with others and share resources regardless of the node on which they are deployed. Of course in order to have a minimum degree of distribution there must be more than one single node in the platform. And this distribution is truly "advanced" when the communication and resource sharing is effortless and seamlessly transparent (applications are oblivious to the actual node where they or the resources are located), and when the nodes are separated beyond the usual reaches of common platforms.

Nodes in the platform can be distributed not only within the same network of an end user environment, but can also be located in server backends (owned by service providers or third parties) or could even be mobile, being worn by the user while outside his/her house. This represents advanced distribution not only within the user home but also mobile and cloud-based. And applications and resources can be communicated across all these nodes no matter where they are.

This adds another level of distribution, where nodes from different users and spaces can virtually co-exist in other shared nodes. This is most visible in cloud-based scenarios, where the server of a provider can host virtual nodes for each of

their users, while being a node for them itself.

4.4.3.1. TECHNOLOGY PROVIDER'S VIEW

Developers can choose from all the different possibilities where to deploy their applications, and can be sure that they can access all of the features of the platform regardless of the node they choose. They do not need to worry about communication protocols or endpoints addresses since this is hidden by the platform distribution features. For what they know, the resources or applications they require are virtually in the same place where their application reside. If the applications they develop are designed to be cloud-based, the server side of the application can discriminate between which user node should be addressed in every interaction. Distribution capabilities can also be used to set up load-balancing solutions.

4.4.3.2. SERVICE PROVIDER'S VIEW

All the different possibilities for deploying applications release any constraints on service providers to decide how and where they want to deploy the platform and their applications into. Whether in a computing device at the users' home, a smartphone, a tablet, a web application or a combination of any or all the previous, new and existing applications can be compatible with the platform.

4.4.3.3. ASSISTED PERSON'S AND CAREGIVER'S VIEW

Advanced distribution is transparent for the end users. They interact with their applications without having to worry about where exactly is their data stored or which device should they use to access it. A kind of "computational load-balancing" can be set up as well, moving resource-consuming applications to the more powerful nodes, while, for instance, the mobile phone node of the assisted person can carry only light-weight apps.

4.4.3.4. GOVERNMENT'S VIEW

The possibility of having different types, sizes and distributions of nodes manufactured by different companies boosts competition and opens more choices for purchases.

4.4.3.5. RELATIONSHIP WITH OTHER SHOWCASES

4.4.3.5.1. COLLABORATION

- Plug and Play. Distributed nodes connected through "remote" channels (e.g. cloud servers) enhance pluggability of more systems.
- Scalability. By relying on distributed nodes, it is easier to increase the capabilities and/or amount of nodes.
- Administration. It is possible to administrate distributed nodes from, for instance, a "server-side" node.

4.4.3.5.2. NOT TO BE CONFUSED WITH

- Resource Sharing. The fact that resources are shared across nodes is not the same, but a benefit from advanced distribution.

4.4.3.6. EXAMPLE USE CASES

4.4.3.6.1. HOME NODE A common scenario for node distribution is having a computing device at home and also a smartphone or tablet, both running the platform. An application running on the smartphone can be aware of the status of all devices connected to the home node so, for instance, it could turn on the lights when the smartphone node detects it arrives home, and then turn them off when it's away. For the application it is transparent, it does not need to access the home network or a cloud service to do so. And it can do it with or without a cloud backend service to act as intermediary (there are options for both).

4.4.3.6.2. CLOUD Cloud-based communication is a key aspect of advanced distribution. Applications can be deployed in a backend server and handle end user information as if it was right there. For example a cloud-based application that detects user movements in his/her house can get the sensor information, associated with the right user, from the platform, without having to set an explicit communication channel with the user home nor the exact location sensors inside it.

4.4.3.7. EVALUATION OF ADVANCED DISTRIBUTION SHOWCASE

4.4.3.8. TESTING IF AN APPLICATION/DEPLOYMENT COMPLIES WITH THE SHOWCASE

Procedure type	☒ Analytical ☐ Operational ☐ Technical ☐ Programatical
Procedure general description	
Determine if the deployment being evaluated is indeed distributed.	

Procedure script
<ol style="list-style-type: none"> 1. Analyse the architecture of the whole scenario or application, identifying where the software components are physically installed. 2. If software components are distributed across 2 or more physical nodes, then there is “Distribution”. Next step is to determine if this distribution makes use of the platform advanced features for it. 3. Determine if the software components in the distributed nodes communicate with each other through one of the Advanced Distribution features of the platform: <ol style="list-style-type: none"> a. Local discovery and peering b. Remote discovery and peering through Gateway c. Remote node virtualization through Remote-API d. Multi-tenancy (only combined with the above)
Success cases
<p>An application scenario is considered to be designed to use Advanced Distribution if the 3 steps above are verified, with the last one having validated at least one of the options available. If any of these fails, the application was not designed for an Advanced Distribution scenario. This does not mean that it is wrong or poorly adapted, only that this showcase does not apply. Do not continue with further procedures for this SC.</p>

Procedure type	☐ Analytical ☒ Operational ☐ Technical ☐ Programatical
Procedure general description	
Confirm the usage of distributed processing in real life environments	
Procedure script	
<p>To demonstrate the usage of distribution in real life environments, the deployment must comply with at least one of these scenarios</p> <ol style="list-style-type: none"> 1. Local distribution: there are several processor nodes in the deployment site that communicate with one another through local networking. This usually implies that 2 or more devices (e.g. smart phone, tablet, pc, set top box, etc.) are to be installed in the deployment site. Each device will be preconfigured with a set of software modules that interact with one another independently of their physical location (the device on which they are running). 2. Cloud: where services require remote operations to be performed on cloud servers. In this case Internet connection is typically required. 	

Success cases
Applications designed to benefit from Advanced Distribution do indeed work seamlessly through any nodes no matter where they are. This means the service can be offered seamlessly as long as the networking requirements apply (e.g. for Cloud scenarios the internet connection is stable).

Procedure type	☐ Analytical ☐ Operational ☒ Technical ☐ Programatical
Procedure general description	
Confirm the availability of distributed nodes	
Procedure script	
<ol style="list-style-type: none"> 1. Deploy the software modules of the application scenario under test in arbitrary different nodes than those of its habitual operation (This is the default situation during the centralized lab tests). Nodes must be configured exactly analogous to the original ones. 2. Perform all or a set of the typical application operations and determine that the fact of being deployed in different physical nodes is seamless and unappreciable to the end user: the application behaves in the same way. 3. For the cases where a multi-tenant scenario is deployed, one or both of these must apply, provided that data sets and configuration are carefully replicated in the “new” nodes: <ol style="list-style-type: none"> a. It should be possible to have the “local” node(s) changed while maintaining the “server” nodes the same, and still accomplish step 2. b. Alternatively to a), the “local” node(s) could be maintained while the “server” node is changed and step 2 should still be accomplished. 4. (Optional) For double-checking the actual distribution, a “packet sniffer” could be used to certify that the communication is really taking place. 	
Success cases	
Successful results in the above steps indicate that an application that was designed to benefit from Advanced Distribution does indeed work seamlessly through any nodes no matter where they are, or which ones they are. Even changing the physical nodes for new ones keeps the application operative. Even in “cloud” scenarios where a node is hosted in a remote server.	

4.4.3.9. MEASUREMENT OF THE VALUE

Indicator	Service Provider Focus Group
Value centred in	☒ Technological Provider ☐ Service Provider ☐ Assisted Person ☐ Caregiver ☐ Government
Description	
Questions: <ul style="list-style-type: none"> ▪ How “seamless” would you qualify this distribution? ▪ If you currently have “single-node” applications, do you think they could benefit from a distributed scenario like this? How? ▪ If you currently have “cloud-based” applications, how would you compare, in terms of complexity, your current connection to the server to these advanced distribution features? 	

4.4.4. SCALABILITY

It applies when increasing the number of deployments or users. Typically deployments are homes where the system is installed; increasing the number of deployments means that the system is installed in more homes. For this the system has to be easily installable and configurable; also the system must be resource optimized, many deployments will increase the need for, for example, more power in cloud services.

4.4.4.1. TECHNOLOGY PROVIDER’S VIEW

Developers usually do not take in consideration mass deployment in their implementation phase, so a platform that helps personalize the system for each deployment, as well as optimize and in some sense high the complexity of massive deployment will be an advantage when the system is actually mass deployed.

Massive deployment implies that the real-life usage of the system is high, thus the probability of failure is increased.

4.4.4.2. SERVICE PROVIDER’S VIEW

Scaling the number of users is the cornerstone of any Service Provider’s business plan, understandably the more users the system is capable of handling the more income those users will produce. Optimizing the costs of incrementing the number of deployments is essential, this costs must not overwhelm the overall profit margin of the operation.

4.4.4.3. ASSISTED PERSON’S AND CAREGIVER’S VIEW

The most value End Users will get from scalable services is divided in essentially in cost and warranty. The more users are using the system the cheaper it gets to run

such system, thus the price per user could be reduced. At the same time the more users get to use the system will help such system guarantee the essential features of it.

Massification of service use, will have to be accompanied by the possibility of personalization, so end users are not left feeling they are just one more. Also quality of the service must not be compromised, end users expect same quality at less price when the system is scaled.

4.4.4.4. GOVERNMENT'S VIEW

Optimizing the costs, and complexity of increasing the number of users is essential for making AAL services cheap for everyone. Lowering the effort barrier to mass deployments will also increase the competition, another pressing factor in the cheapening of the operation costs per user.

The competitive prices per user not only directly helps to introduce AAL services for the population which, depending on the nature of the AAL services, will greatly reduce other indirect costs from the state budget. For example general introduction of health services will reduce the need of the population to turn up in hospitals, thus the number of resources for hospitals will be reduced.

4.4.4.5. RELATIONSHIP WITH OTHER SHOWCASES

4.4.4.5.1. COLLABORATION

- Administration, easiness of configurability (whether it is remote, or local) will influence the capability to install the system in more deployments. Administration tools such as a Digital Store concept are also central to easily scale deployments.
- Advanced Distribution, especially in cloud based systems, distribution is an essential part for scaling the number of deployments.
- Personalized Content, is a mechanism to scale the number of users for a server, proposing the users that need a service the option to use it.

4.4.4.5.2. NOT TO BE CONFUSED WITH

- Evolution, increasing the number of users within a single deployment is not the same as scaling the number of users.

4.4.4.6. EXAMPLE USE CASES

4.4.4.6.1. HOME NODE A home based application, for example an energy optimization service, is first installed on 10 homes for testing. Then it is released and 1000 homes install it. The installment of the new systems must not be more costly than that of the initial deployments, in fact the costs should be reduced. Management of the new systems should also not be more expensive, in fact the opposite.

4.4.4.6.2. CLOUD In cloud-based solutions the scaling is measured purely by number of users, increasing users and managing the new users must not be more expensive. The cloud concept has this feature at its core, so the platform must take advantage of it, and improve if possible.

4.4.4.7. EVALUATION OF SCALABILITY SHOWCASE

4.4.4.8. TESTING IF AN APPLICATION/DEPLOYMENT COMPLIES WITH THE SHOWCASE

Procedure type	☒ Analytical ☐ Operational ☐ Technical ☐ Programatical
Procedure general description	
Find indicators about the application's capability of scaling	

Procedure script	
<ol style="list-style-type: none"> 1. List all available documentation. This may include, but not limited to: user manual, installer manual, design documentation, adaptation schema, issue report system, help fora, developer's documentation, Code comments, Commercial brochures, etc. All of these may be in different formats: web page, paper. 2. Go through all the available documentation looking for: <ol style="list-style-type: none"> a. Easiness of installation, Installation process should be as simple as possible. Installation manual should be straight forward, without forgetting about possible issues. If the process is simple enough for the user to perform, it may be indicated in the user manual. Nonexistence of installation manual may indicate also that there isn't enough documentation, take this into consideration. b. Procedures for self-configuration. Configuration is important to many applications; the code should consider that there may not be configuration. The application should still be able to operate with default configuration values, or by creating a default configuration file with it does not exist. c. Initialization procedures. The application is prepared for initialization issues like for example when there are dependencies to other modules and such modules are not available the application must be prepared to deal with the situation. Options may be: wait until the module is available, warn about the unavailability of the module, or disable the functionality of the application that involves the module (being the rest of it operational). 3. (optional) Estimate the installation costs/time per deployment pondered with potential number users per deployment. 	
Success cases	
<p>In step 2 there are clear indicators found that prove the application has been designed and prepared for easy installation and deployment contingencies.</p> <p>In step 3 the estimated costs per user is lower as the number of users (and maybe deployments) increases.</p>	

Procedure type	☐ Analytical ☐ Operational ☒ Technical ☐ Programatical
Procedure general description	
Attempt to scale the deployment in number of nodes/devices	

Procedure script
<ol style="list-style-type: none"> 1. Deploy the system 2. Increase the node/device number (preferably more than 2 times the last node/device count) 3. Repeat step 2 if possible
Success cases
System response is equal or better each iteration of step 2. Installation of new nodes/devices is easy and almost configuration-less.

Procedure type	☐ Analytical ☒ Operational ☐ Technical ☐ Programatical
Procedure general description	
Attempt to scale the deployment studying the evolution of cost/per deployment	
Procedure script	
<ol style="list-style-type: none"> 1. Deploy the system in at least 10 sites (or a server with at least 10 users/tenants) <ol style="list-style-type: none"> a. Measure the mean deployment time and cost per tenant, see indicator below. 2. Increase the tenant/user number (preferably more than 5 times the last user count) <ol style="list-style-type: none"> a. Measure the mean tenant deployment cost, see indicator below. b. Monitor user issue reports, perform questionnaires, or focus groups to find how the discomfort of users associated with initial installation evolves with the number of deployments 3. Repeat step 2 if possible 	
Success cases	
<p>Values measured at 2.a. must be equal or lower than previous values (earlier iterations of 2. and 1.a.). If costs are approximately equal (+- 10%) further research will be necessary to determine if the costs are derived from scaling actions or other uncontrolled variables.</p> <p>Issues reported by users in 2.b. should drop as with more users the deployment phase should increase in efficiency.</p> <p>The Platform must behave equally independently from the tenant numbers.</p>	

4.4.4.9. MEASUREMENT OF THE VALUE

Indicator	Scalability Deployment costs
Value centred in	☐ Technological Provider ☒ Service Provider ☐ Assisted Person ☐ Caregiver ☐ Government
Description	
Mean Deployment cost values, and its evolution through the attempt of scaling an application should show the real value of scaling a deployment Costs and time should also be compared to profit, in total terms; which will also show the reasons to scale the deployment in the first place. Each data point contains the following fields: <ul style="list-style-type: none"> ■ Pilot ■ Increment of deployments. The number by which the number of deployments has been increased, only in initial deployment this number is equal to the total number of deployments. ■ Mean cost per deployment ■ Estimated mean profit per deployment, in mean how much is the service provider profiting from each of the deployments (taking into account variable costs) 	

Indicator	Service provide focus group
Value centred in	☐ Technological Provider ☒ Service Provider ☐ Assisted Person ☐ Caregiver ☒ Government
Description	
Questions: <ul style="list-style-type: none"> ■ How would you evaluate the priority for applying scalability in your Service(s)? ■ Have you ever found scalability problems while deploying large scale services? 	

Indicator	End user focus group
Value centred in	☐ Technological Provider ☐ Service Provider ☒ Assisted Person ☒ Caregiver ☐ Government

Description
<p>Questions:</p> <ul style="list-style-type: none"> ■ What pattern do you expect prices to follow when a service provider scales their services? ■ Have you ever been excluded because a service that you needed/wanted wasn't available in your zone? ■ Do services with lots of users make you feel more comfortable? (reliability) ■ Do services with lots of users fail in personal assistance? (quality)

4.4.5. EVOLUTION

The showcase “Evolution” implies the change in time of the particularities of a single deployment, whether it is hardware, software, features (installed services) or its context. From the application/services point of view it is possible to develop new functionalities or services as “extension” of previous provided functionality in order to add new tools or new instruments useful, for example, to simplify the installation task or automatic update task; the new functionalities may also be useful for troubleshooting and fixing of some malfunctions. Moreover, from the architectural and HW point of view, “evolution” can be interpreted as the adding of sensors or actuators without changing the core functionality of the system, but with the goal of improving general performances. Last but not least, “evolution” of the context is also possible, for example going from a single-user deployment to a multi-user deployment.

4.4.5.1. TECHNOLOGY PROVIDER’S VIEW

Evolution gives technical providers the ability to integrate new functionalities or to increase the number of services with minimal effort. Moreover, the addition of new features and services brings competition because the technology providers can increase the potential of its products and services compared to a competitor who has already developed additional services over the same service/application.

4.4.5.2. SERVICE PROVIDER’S VIEW

Evolution enables Service Providers to offer fully customizable services for users. Each user will only use the services that she/he needs, and if this needs change new services can be added. Upgrading users from one service category to a superior category, evolving the single deployment; in this way, service providers can evaluate the opportunity of new possible business models.

With evolution capabilities Service providers are capable of quickly and economically adapt to new circumstances. Adapting to new circumstances within each single deployment, as users likely change the context in which the service is offered. Also

adapting to global circumstances such as new legislation, technology, science and medical advancements can be achieved by evolving all the deployment sites.

Evolution offers new instruments for testing the applications and checking if a service, not currently integrated, will work. In this way they could offer customizations of specific services that, for example, have been demonstrated to be effective in other sites. Moreover, the "evolution" of a deployment (in terms of "upgrading") can be a valuable asset for service providers, for example it may be possible to solve some recurring issues in deployments.

4.4.5.3. ASSISTED PERSON'S AND CAREGIVER'S VIEW

For the end user, the extension of the functionality of a platform component has considerable advantages, for example through "evolution" it can be possible to customize an application according to specific needs of the assisted person. From the caregiver point of view, on the other hand, the same application can be specialized based on monitoring needs; in this way different evolutions of the same application are available, increasing the level of knowledge of the tools and the level of satisfaction.

4.4.5.4. GOVERNMENT'S VIEW

Evolution showcase is important for government agencies as it allows a better and more effective management of financing instruments. In fact, for the institution, is fundamental to invest for the improvement of existing applications or services that are already popular among end users; the improvement can be obtained through extension of functionality of application and services which contribute to the provision of new solutions required by the market but not yet realized. Moreover, "evolution" helps keep the market active, and always looking for innovative solutions that can be used for evolving current deployments.

4.4.5.5. RELATIONSHIP WITH OTHER SHOWCASES

4.4.5.5.1. COLLABORATION

- Plug and Play. From HW point of view, the capability to choose from PnP different sensors can provide "evolution" of some functionality.
- While evolution focuses on the capability of "replacing" or "adding" components or services to extend the solutions, the "tool" being used is Services Integration.

4.4.5.5.2. NOT TO BE CONFUSED WITH

- Scalability, even though should increase the resources and performance in a system, not always allows evolution in the same system.

4.4.5.6. EXAMPLE USE CASES

4.4.5.6.1. HOME NODE A sensor that detects the gas leak is installed in the home of the user. In case of the detection of danger situation, an actuator promptly closes the valve system; as evolution, in the home scenario it is possible to incorporate an application able to send an alert during the leak out of the domestic environment; the application can be seen as an extension of an existing service contributing with its functionality to enrich the list of useful tools for prevention/alert solutions in home environment.

4.4.5.6.2. CLOUD An application able to monitor environments thanks to the anchors, or to collect environmental or health parameters of the end user, transmits this data to an application in the cloud. Another cloud-application can extend the functionality of the previous application; for example, the evolution can be a novel functionality for the automatic anomalies evaluation that, starting from data acquired from the anchors (and transmitted to the cloud), extract and processes this data separately with the objective of providing a new feature for the platform.

4.4.5.7. EVALUATION OF EVOLUTION SHOWCASE

4.4.5.8. TESTING IF AN APPLICATION/DEPLOYMENT COMPLIES WITH THE SHOWCASE

Procedure type	☒ Analytical ☐ Operational ☐ Technical ☐ Programatical
Procedure general description	
The purpose of this procedure is to verify if the deployment being evaluated can be extended (i.e. can “evolve”) from analytical point of view	

Procedure script

1. Using available documentation, especially the technical manuals, analyse the application evaluating if it is possible to add new functionalities or services as “extension” of previous provided functionality, by certifying one or more of the following:
 - a. The application is separated into different modules (bundles), each containing separate “pluggable” features
 - b. The application functionality is separated into different platform members (callees, publishers. . .), thus decoupling its own business logic into independent modules communicated through the platform
 - c. The application relies on libraries or platform managers that are susceptible of being upgraded/updated.
 - d. The application relies on libraries or managers that can incorporate logic through external resources (example: the Situation Reasoner can add new rules).
 - e. The Ontologies used by the application, and the application components, allow for third party extension. Other developers are capable of extending or mapping the concepts to fit other applications/purposes.
 - f. The service profiles called are generic enough so that when new services are introduced they will be matched.
 - g. Introducing new service profiles does not affect the application, when there is a service request the application is capable of managing multiple responses and responses from different components (the treatment is generic enough).
 - h. Hardware devices are abstracted, and the application is agnostic about the number of devices and their nature. This allows the application to be able to deal with the addition of different devices and sensors to the deployment.
 - i. The application is inherently multi-user. This allows deployments to evolve from single to multi-user.
2. Determine if the options above that are covered by the application also allow that these “separated modules” can be updated or upgraded to a new or different version without affecting the application as a whole (other than adding new functionality).
3. Report analysis; see Number of extension opportunities indicator.
4. Analyse, if possible, the process by which an application can be upgraded. Implementation of showcase Administration will ensure this.

Success cases
An application can “evolve” if it can be upgraded/updated either by replacing it with a new version, adding new modules to it, or new apps or managers that interact with it and enhance it. This is what the procedure above attempts to check, but given it can be a complex process, there are different degrees to which an app can comply with “evolution”. It is up to the analyst to determine how “hard” or “easy” it is to fulfil the upgrade, but success in absolute terms is achieved by “enabling” this evolution, in any of the mentioned procedure steps.

Procedure type	☐ Analytical ☒ Operational ☐ Technical ☐ Programatical
Procedure general description	
Demonstrate the evolution capability in real life environments	

Procedure script

The objective is to show evolution of deployed applications. There are several ways to approach the problem:

- **Forced Increments.** If the application is already modular, the deployment plan for individual users can be forced in an incremental way. This means that the number features that a user gets at the beginning of operation is less than what she/he gets at the end; where one or more evolution procedures were applied in the middle.
- **Natural evolution.** When a new feature is demanded by stakeholders and this feature can be easily introduced to the deployment by means of one or more evolution procedures.

An Evolution Procedure can be one of the following:

- **Module add.** New Modules are added, these can be software modules, adding new functionality, or hardware components. In the case of hardware components adding instances of the same device is also considered evolution, as the application will have to deal with an increasing number of data points.
- **Module Swap.** A part of the deployment is swapped by another, different component which may have more features or is more efficient. This may be swapping a hardware device with another device (different in model and/or brand), or a software module is swapped with another module, preferably not an updated module, but a module produced by another provider altogether.
- **Integrated Applications.** When a set of applications, which work both as standalone and integrated with one another (i.e. they are also compatible with the service integration showcase), is available; it is possible to evolve by increasing the subset of integrated applications installed in the deployment. Essentially evolution is achieved by installing a new integrated application that was previously not installed.
- **Context change.** The context if the deployment changes, for example increasing the number of users using the system.

In all cases the core parts of the application do not need to be changed/updated.

1. Make sure the deployment is working as specified
2. Perform one evolution procedure, either to selected developments or batch all
3. Make sure to record it, see Evolution procedures indicator
4. Repeat, if possible

Success cases
<p>Evolution procedures can be performed smoothly:</p> <ul style="list-style-type: none"> ■ The user may only notice the new features/capabilities of the system being now available. ■ The costs of an evolution procedure is significantly lower than a new deployment <p>Evolution is a tool that allows service providers to quickly adapt to user needs and other circumstances. While technology providers can offer these adaptations at reduced costs. Users benefit by being able to customize their system to their need by only adding the features and/or capabilities that fit them.</p>

Procedure type	☐ Analytical ☐ Operational ☒ Technical ☐ Programatical
Procedure general description	
The test consists in testing the functionality of an application/service after the introduction of some specific plug-ins, extensions or new features.	
Procedure script	
<ol style="list-style-type: none"> 1. (optional) Locate if there are plugins extensions already shipped with the application. In the following steps assume one of them is a new one, i.e. do not start it until 3.c. 2. (optional) Locate existing plugins or extensions compatible with the application. In the following steps assume one of them is a new one, i.e. do not start it until 3.c. 3. Run an application: <ol style="list-style-type: none"> a. verify the application/service is working properly b. discover and annotate all the functionalities of the application c. include a new functionality (service, event, UI. . .) d. verify the correct operation of existing features provided by the application e. verify if the new functionality/service is working properly 	
Success cases	
In step 3.e. the new functionality is proven to work properly. This shows that the application is capable of accommodating new functionality without changing core artefacts of the system.	

Procedure type	☐ Analytical ☐ Operational ☒ Technical ☐ Programatical
Procedure general description	
Test if the application can manage evolution through the addition of hardware	
Procedure script	
<ol style="list-style-type: none"> 1. (optional) Locate if there are multiple hardware devices already shipped with the application. In the following steps assume one of them is a new one, i.e. do not start it until 3.c. 2. (optional) Locate existing hardware devices compatible with the application. In the following steps assume one of them is a new one, i.e. do not start it until 3.c. 3. Run an application: <ol style="list-style-type: none"> a. verify the application/service is working properly b. discover and annotate all the functionalities of the application, making sure to use the available hardware. c. Include a new hardware device. It might be just a new instance of a device already connected to the system, or a completely new hardware type. d. verify the correct operation of existing features provided by the application e. verify if the new hardware information is being processed. This might be indicated by better measurements in case of increased number of devices, or new measurements being considered when a new hardware device type was added. 	
Success cases	
In step 3.e. the added hardware is proven to work properly, and it is being used. This shows that the application is capable of accommodating new hardware without changing core artefacts of the system.	

Procedure type	☐ Analytical ☐ Operational ☐ Technical ☒ Programatical
Procedure general description	
The test consists of developing a test plug-ins, extensions or new feature that can be integrated into the application.	

Procedure script
<ol style="list-style-type: none"> 1. Take the indicator: Number of extension opportunities; select one extension possibility (preferably the easiest) 2. Develop a simple extension module, make sure the use of this new extension by the core is reported. Allow also easy configuration of the behaviour of the module for the tester, for example displaying a simple interface to the person in charge of the test, so she/he can control the response of a particular data request that is offered by the simple extension to the core system. 3. Use this module in the technical test procedure.
Success cases
<p>The extension can be developed, without the need of changing the core of the application, and with relative easiness (overall effort should be approximately the same, or less, as the estimated effort).</p> <p>The extension can be added to the core system, and the system works as expected. The tester can certify the new module is being used. If possible the tester may also prove, by introducing extreme values, that the application is protected from non-standard extensions.</p>

4.4.5.9. MEASUREMENT OF THE VALUE

Indicator	Evolution procedures
Value centred in	☒ Technological Provider ☒ Service Provider ☐ Assisted Person ☐ Caregiver ☐ Government
Description	<p>For each Evolution procedure performed, while executing operational test procedure, we can collect data to analyse the quality and value of evolution. Each data-point consists of the following values:</p> <ul style="list-style-type: none"> ▪ pilot ▪ time since deployment (the initial installation of the deployment) ▪ approach {forced, natural} ▪ type of evolution procedure {module add, module swap, integrated application, context change} ▪ number of deployments involved in the evolution ▪ mean estimated relative cost of evolution per deployment: $\frac{\text{cost of evolution}}{\text{cost of installation}}$. Value < 1 means the evolution is less costly than installation; value $=1$ cost of evolution is the same as a new installation, value >1 cost of evolution is larger than cost of a new installation.

Indicator	Number of extension opportunities
Value centred in	☒ Technological Provider ☒ Service Provider ☐ Assisted Person ☐ Caregiver ☐ Government
Description	
<p>During the analytical test procedure, count the number of extension opportunities. Each data-point consists of:</p> <ul style="list-style-type: none"> ■ Pilot ■ Application ■ Easiness {trivial, requires small modifications in the core, requires may modifications to the core, impossible, N/A} ■ Brief explanation on how to provide extension 	

Indicator	Questionnaire for TP & SP
Value centred in	☒ Technological Provider ☒ Service Provider ☐ Assisted Person ☐ Caregiver ☐ Government
Description	
<ul style="list-style-type: none"> ■ In your opinion, can the procedure be performed with little technical knowledge? ■ Do you think that Evolution SC can bring added value in terms of number of services offered? ■ Can Evolution SC improve the quality of an application? ■ Do you think that thanks to Evolution SC you can evaluate new business models? ■ Do you have any suggestions for the improvement of the procedure? <p>Target Description: Technical provider and service provider focus group</p>	

Indicator	Questionnaire for End-user
Value centred in	☐ Technological Provider ☐ Service Provider ☒ Assisted Person ☒ Caregiver ☐ Government

Description
<ul style="list-style-type: none"> ■ Did you feel there was enough support from the care organization? (care-giver) ■ How often did you use the application/service before its “evolution”? (assisted person) ■ How often did you use the application/service after its “evolution”? (assisted person) ■ Did the “evolution” of the application/service have a positive effect on your workload? (assisted person) ■ Do you have any suggestions for the improvement of the procedure? (assisted person and caregiver) <p>Target Description: End-user focus group</p>

4.4.6. ADMINISTRATION

It is possible to remotely (and of course locally) connect to platform nodes (or sets of nodes) and manage the status of the platform itself. It would therefore allow technical providers, service providers or even caregivers to execute maintenance operations such as updates, reboots, log inspection and similar. It provides a way to check on the status of the system and if necessary provide help and support if problems arise. With remote administration facilities, providers can manage several users’ platforms in a unified and accountable manner.

4.4.6.1. TECHNOLOGY PROVIDER’S VIEW

Remote administration, and even on-site administration if necessary, gives technical providers the ability to make the system recover from certain errors and inspecting their cause, as well as many other technical maintenance tasks such as updates or statistics collection.

4.4.6.2. SERVICE PROVIDER’S VIEW

Costs are reduced by being able to manage nodes remotely without sending personnel in-place. Service provider administrators can check on the status of their applications, but in general it is the same that can be done by the technology provider administrators, or just delegating these tasks to them.

4.4.6.3. ASSISTED PERSON’S AND CAREGIVER’S VIEW

Given the technical nature of system administration is unlikely that any end user would directly benefit from access to this features, but in certain, less technical cases, it could be useful to allow some caregivers to remotely access administration basic features such as reboots. This could be helpful for deadlock situations or when service providers give limited support.

4.4.6.4. GOVERNMENT'S VIEW

Allows government stakeholders to participate in deployments as administrative peers, without technical intervention. In this case, like Service provider's advantages, costs are reduced thanks to remote and unified administrative capabilities.

Administrative subcontracting might create a new interesting market to nurture new business models.

4.4.6.5. RELATIONSHIP WITH OTHER SHOWCASES

4.4.6.5.1. COLLABORATION

- Resource Sharing. Administration facilities can manage configuration data of different applications since these are sharing their content across the platform.
- Advanced Distribution. In advanced distributed scenarios the administration can happen remotely from a node located at a different place.

4.4.6.6. EXAMPLE USE CASES

4.4.6.6.1. HOME NODE Understanding a "Cloud use case" one that happens with a provider or caregiver accessing remotely the administration features, a "Home node use case" would be one that sees the provider present at the Home node location, using the Administration features. This could happen in a scenario where an end user reports a malfunction to the provider, and this provider does not have the means to remotely access the Administration features, or could not solve the problem remotely. Direct access to the platform system administration features in this case would be unlimited and should help identify any problem and get the system up and running again.

4.4.6.6.2. CLOUD On the other hand, as mentioned above, remote administration would do the same but with the provider accessing a certain user system remotely (This, of course is preferable to being physically present).

4.4.6.7. EVALUATION OF ADMINISTRATION SHOWCASE

4.4.6.8. TESTING IF AN APPLICATION/DEPLOYMENT COMPLIES WITH THE SHOWCASE

Procedure type	☒ Analytical ☐ Operational ☐ Technical ☐ Programatical
Procedure general description	
Find among all the available documentation of the application, if processes, tasks, methods, practices or operations to perform local or remote administration are described	

Procedure script
<ol style="list-style-type: none"> 1. List all available documentation. This may include, but not limited to: user manual, installer manual, design documentation, issue report system, help fora, developer's documentation, Code comments, Commercial brochures. All of these may be in different formats: web page, paper. 2. Go through all the available documentation looking for: <ol style="list-style-type: none"> a. Update procedures. Look for phrases like "update the software", "upgrade version". b. Reboot procedure c. Log inspection d. Configuration procedures. Usually the configuration parameters are listed and the acceptable parameters described. e. Remote operations procedure. Maybe there is a way to do some or all of the above remotely, and this is explained in the documentation.
Success cases
<p>In step 2, there should be at least a mention to one of the procedures. Different degrees of success can be defined taking into account different factors:</p> <ul style="list-style-type: none"> ■ The number of procedures undertaken, better if all procedures are explored ■ The level of detail, better if detailed documentation exists (usually documentation is structured in hierarchy, going into further detail). ■ The complexity of the procedures, better if complexity is low (for example clicking a button is less complex than editing a file, which is less complex than editing source and recompiling) ■ How is intended to, Better if end users are capable to do it.

Procedure type	☒ Analytical ☐ Operational ☐ Technical ☐ Programatical
Procedure general description	
Analyse if the system is including frameworks or tools to perform administrative operations	
Procedure script	
<p>Look in the design documentation, and/or the source of the system for usage of frameworks. A good place to start is in the source, looking at the listed dependencies (for example in the dependencies section in pom.xml file).</p> <p>Frameworks that might help administration tasks include: OSGi services (compendium) especially config admin; or the platform's configuration framework.</p>	

Success cases
<p>The solution uses frameworks (careful with declared dependencies that are then not used) that facilitate administration tasks such as Software updates, rebooting, log inspection and configuration.</p> <p>These frameworks usually include the feature to perform the tasks both locally and remotely. Remote capability is always a plus.</p>

Procedure type	☐ Analytical ☒ Operational ☐ Technical ☐ Programatical
Procedure general description	
Demonstrate Administration capability in real life environments	
Procedure script	
<p>Use the Administration features during operation to solve issues, especially remotely. This should not be considered a test, as this might result in fatal interruption of the service operation.</p> <p>The objective is to demonstrate that software update, log recovery, system reboot, or configuration operations are manageable in real life environment.</p>	
Success cases	
<p>Administration tasks can be performed without interruption of normal operation.</p> <p>Bonus: for remote administration tasks, no technician had to be sent to the deploy site to solve issues.</p>	

Procedure type	☐ Analytical ☐ Operational ☒ Technical ☐ Programatical
Procedure general description	
Access a space and perform maintenance, analytics and configuration tasks on it as a provider	

Procedure script
<ol style="list-style-type: none"> 1. Access the administration interface of an already running environment. Access can be performed either by direct physical access to it (through hardware that is part of the environment) or through remote access. Notice that remote access may be deliberately disabled in certain scenarios due to security constraints. At least the local access must be verified, but the remote access is strongly encouraged. 2. Perform the following administration tasks, regardless of the order, in all available nodes (not all operations are relevant in all nodes): <ol style="list-style-type: none"> a. Install/uninstall an application in the environment b. Start/stop/update an application that is installed in the environment c. Change configuration of an application installed in the environment d. Collect logs from the execution of the environment e. Obtain information about the status of the running environment (installed application, available nodes, versions. . .) f. Shutdown/restart the environment (This may not be possible in certain environments).
Success cases
By being able to perform all basic operations needed to run and maintain the platform in any node of the environment, even remotely, providers are assured to be able to recover the system from stall situations. It also allows for support and problem resolution. In addition, when available, remote access allows to perform all the above without being physically present in the maintained environment.

Procedure type	☐ Analytical ☐ Operational ☒ Technical ☐ Programatical
Procedure general description	
Access a space and perform support and configuration tasks on it as an end user	

Procedure script	
<ol style="list-style-type: none"> 1. Access the administration interface of an already running environment. Access can be performed either by direct physical access to it (through hardware that is part of the environment) or through remote access. Notice that from the point of view of end users the configuration requires a specialized interface provided by the application being evaluated in the scenario. As such, remote access may not be present by design. At least the local access must be verified, but the remote access is strongly encouraged. 2. Perform the following administration tasks, regardless of the order, in the nodes designated as interface points for the end users. Notice that the availability of the tasks may be restricted depending on the type of end user, access location or by design, as it depends mostly on the application: <ol style="list-style-type: none"> a. Install/uninstall an application in the environment b. Start/stop/update an application that is installed in the environment c. Change configuration of an application installed in the environment d. Shutdown/restart the environment 	
Success cases	
<p>By being able to perform these basic operations, even remotely (where available), end users are assured to be able to recover the system from certain stall situations. It also allows for minor support and problem resolution, as well as personalization and configuration of applications. In addition, when available, remote access allows to perform all the above without being physically present in the environment. Because these tasks are oriented to end users, it need not be the Assisted Person who is in charge of performing these tasks, but it can also be a caregiver, thus reducing the perceived complexity and required technological experience.</p>	

4.4.6.9. MEASUREMENT OF THE VALUE

Indicator	Service provider focus group
Value centred in	☐ Technological Provider ☒ Service Provider ☐ Assisted Person ☐ Caregiver ☐ Government

Description
<p>Questions:</p> <ul style="list-style-type: none"> ■ Are the administration capabilities of the platform more or less convenient than the ones you may already have in place? ■ How do you think the administration capabilities of the platform will scale up when you want to manage a high number of users? Do you think it will be better or worse than what you currently do? ■ Do you currently provide remote assistance? <ul style="list-style-type: none"> 1. If so, do you think the platform features provide you any advantage? 2. If not, would you like to provide it thanks to the platform features? ■ How many personnel do you currently commit to these administration tasks and how many would you need if you used these features?

Indicator	End user focus group
Value centred in	☐ Technological Provider ☐ Service Provider ☐ Assisted Person ☒ Caregiver ☐ Government
Description	
<p>Questions:</p> <ul style="list-style-type: none"> ■ Have you ever had to act as “technical support” for a relative or friend? <ul style="list-style-type: none"> 1. and if you are a professional caregiver, for a person you were caretaking? ■ How capable would you consider yourself to provide this kind of technical support? ■ How often do you think you would be assisting in this kind of support? <ul style="list-style-type: none"> 1. Do you think using the platform would make it quicker or longer, or more or less often? ■ How would you value being able to solve these issues remotely? Do you think the person you are assisting would prefer to have you on site? ■ For professional caregivers that have more than one person under assistance: could you assist more people thanks to these features? 	

Indicator	Mean Administration task time
Value centred in	☐ Technological Provider ☒ Service Provider ☐ Assisted Person ☐ Caregiver ☐ Government

Description
The mean administration task time is the arithmetic mean of time taken to perform each administration task (in tests 1 and 2) for the system.

4.4.7. INTEGRATION WITH LEGACY SYSTEMS

The showcase “Integration with Legacy Systems” aims to underline the platform capability to manage HW, application and services that are earlier than current technology. Typically, the challenge is to keep the legacy application/services running while converting it to newer and more efficient code that makes use of new technology and programmer skills. A legacy system is not necessarily defined by age. Legacy may refer to lack of vendor support or a system’s incapacity to meet organizational requirements. Legacy conditions refer to a system’s difficulty (or inability) to be maintained, supported or improved. A legacy system (for open platforms) overcomes this problem because it is usually compatible with newly purchased systems. For example is fairly well known that more and more health sector uses ICT as a tool to improve services, but the introduction of new ICT-systems in public sector (e.g. clinics or hospitals) is often complex and requires a lot of time and resources (one reason is the complexity of health domain and the role of legacy systems in this domain). The introduction of innovation in ICT for healthcare always requires the integration and coexistence of new components with legacy systems, and the integration should be kept at a high level i.e. avoid complex integrations and keep simplicity for the end-user to exchange information between a new system and the legacy system. It is important to stress that the platform uses ICT instruments to make this integration easily.

4.4.7.1. TECHNOLOGY PROVIDER’S VIEW

With the “Integration with Legacy Systems” showcase, technology providers may be able to rapidly extend their legacy systems to new technologies (mobile application, web application and cloud applications). Open platform (and open-source protocols) enable technology providers to easily, quickly and freely extend the portfolio of capabilities and service offered, while maintaining robust and reliable legacy systems.

4.4.7.2. SERVICE PROVIDER’S VIEW

From the perspective of service provider, this showcase is the most beneficial in terms of resource optimization. In fact, thanks to the showcase described above, every adapted system already used by the service provider (e.g. billing system, health records, user management...) can be integrated with a little effort.

4.4.7.3. ASSISTED PERSON’S AND CAREGIVER’S VIEW

The showcase “Integration with legacy systems” is interesting for the assisted person since it allows the reuse of obsolete technologies and systems that may be present inside homes of elderly users.

4.4.7.4. GOVERNMENT’S VIEW

From the Government perspective, many benefits can be derived by the involvement of a greater number of users, since the reuse of legacy systems well integrated with the platform will have advantages both from the economic perspective both from the number of end users involved in the use of new “open” technologies. The financial resources that will be saved by integrating new application/services to legacy system will be spent, for example, for other purposes.

4.4.7.5. RELATIONSHIP WITH OTHER SHOWCASES

4.4.7.5.1. COLLABORATION

- Services Integration, permits to increase the service deployment flexibility, facilitating in some cases the use of legacy system.
- Evolution, extends the functionality of the platform and enhances the capability to manage obsolete devices or legacy system.

4.4.7.5.2. NOT TO BE CONFUSED WITH When properly balanced with Services Integration, there should not be any confusion with this showcase.

4.4.7.6. EXAMPLE USE CASES

4.4.7.6.1. HOME NODE In the user environment an obsolete computer system is present. The PC may still be in use because its data cannot be changed to newer or standard formats, or its application programs cannot be upgraded, but must necessarily be used by the end user. A new application for monitoring (and storing) of vital signs is available and the caregiver thinks it might be appropriate to use the new application (it could be certified with respect to new regulations). The application also works with the user’s computer, thanks to its capability of integration with legacy systems.

4.4.7.6.2. CLOUD A new safety service is provided by the home care services. A vendor is implementing this safety service as a stand-alone solution. However as this vendor has created an adapted component capable to share user identifier with a user Electronic Health Record (EHR) there is no need to duplicate the effort for a double registration. This component can also be used to exchange data between the home safety service and the remote EHR through the cloud using a web-service Gateway.

4.4.7.7. EVALUATION OF INTEGRATION WITH LEGACY SYSTEMS SHOWCASE

4.4.7.8. TESTING IF AN APPLICATION/DEPLOYMENT COMPLIES WITH THE SHOWCASE

Procedure type	☒ Analytical ☐ Operational ☐ Technical ☐ Programatical
Procedure general description	
Identify the Legacy systems integrated with the application.	
Procedure script	
<ol style="list-style-type: none"> 1. List all available documentation. This may include, but not limited to: user manual, installer manual, design documentation, issue report system, help fora, developer's documentation, Code comments, Commercial brochures. All of these may be in different formats: web page, paper. 2. Go through all the available documentation looking for: <ol style="list-style-type: none"> a. Centralised User management, databases, or authentication mechanisms. These will typically be core elements of legacy systems. b. Use of existing or third party services. For example web services offering land traffic distribution, weather, or nutritional recommendations. The use of the platform components such as the Internet Gateway, which transforms web services to and from universAAL IoT member model. c. Exporting services could be used when the interaction is initiated by a legacy system. In this case some platform components such as Service Gateway, or the Internet Gateway are strong evidence of this method being used. d. Standards, whether it is hardware standards, file standards, network protocol standards or framework standards, the use of such standards may indicate the attempt to interact with legacy systems. 3. Make sure the system which the application is interacting with is indeed a legacy system. Behind an action of the ones listed above there can be no system, or the same system as the system being tested. A legacy system is defined as any previous existing platform/infrastructure/device that cannot be directly plugged to the systems after adaptation. 	
Success cases	
In step 3 there is at least one legacy system verified. Legacy systems may be important to an application, because maybe there is no other way to replace the service; it might be an external service where there is close to no power to change.	

Procedure type	☐ Analytical ☐ Operational ☒ Technical ☐ Programatical
Procedure general description	
Verify if the specific application/service is integrated with legacy or external systems	
Procedure script	
<ol style="list-style-type: none"> 1. Identify the legacy system(s) which the application connects to. Use the analytical procedure results. 2. Analyse how the system(s) are connected. This connection can be through network, either IP (through a specific port), standard protocols (for example http), or through other networks; the connection can be done also using hardware connectors, or proprietary libraries. 3. Certify the connections are being used <ol style="list-style-type: none"> a. In case the connection can be monitored, for example using network sniffers, analyse the traffic for the data being transmitted between systems. b. Alternatively, severing the connection will make the functionality provided by the legacy system unavailable. 	
Success cases	
In step 3, it can be demonstrated that legacy systems are being used by the application, either as core part of the application (when the connection is cut, the application stops working altogether), or as part of the application (when disconnected, only part of the functionality is affected).	

Procedure type	☐ Analytical ☒ Operational ☐ Technical ☐ Programatical
Procedure general description	
Show the integration with legacy system by replacing part of it with new system that integrates with it.	

Procedure script	
<p>This operational procedure will span in a long period of time. It requires an original system that will be replaced by a new system, which will interface with the original system to perform already implemented operations.</p> <ol style="list-style-type: none"> 1. Make sure there is data about the usage of the original system. It may be necessary to deploy the original system to collect data about usability, effectiveness, Issues with the system, etc. 2. Deploy the new system for a subset of users. 3. Monitor the Issues reported from this sub set. Look for: <ol style="list-style-type: none"> a. Large discrepancies in the expected results from a user perspective. This will show the new system is radically different to the old one, and users are not capable of adapting to the new concepts. b. Discontinuities of the service. For example in a service that keeps favourite item lists, these lists are not maintained when introducing the new system. Another example is that a sub service is no longer available, in this case check this is not an administrative decision. c. General technical problems, this might indicate the integration with the original system is not working properly, or the hardware is incompatible with the new system. d. Deficiencies in service efficiency. When introducing a new system it is expected that the new system will be able to perform the same or more operations, and perform them in a more efficient way. If this is not the case, then something might be wrong in the interface between the original and the new system. 	
Success cases	
<p>In step 3 it can be shown the user acceptance level of the new system is comparable with the original legacy system. The new system is capable of replacing the original one, and the transition between the original and the new system can be performed smoothly. No large discrepancies arise in the interface between the two systems.</p>	

4.4.7.9. MEASUREMENT OF THE VALUE

Indicator	Questionnaire for TP & SP
Value centred in	☒ Technological Provider ☒ Service Provider ☐ Assisted Person ☐ Caregiver ☐ Government

Description
<ul style="list-style-type: none"> ■ Have you ever had to integrate new systems with old ones? ■ In your opinion, the Integration can be performed with little technical knowledge? ■ Do you have identified technical limitations? If yes, specify the type of limitation... ■ Does Integration with Legacy Systems SC have benefits from the economic point of view? ■ Do you think that integration with legacy systems can increase the number of end-users? ■ Do you have any suggestions for the improvement of the Integration? <p>Target Description: Technical provider and service provider focus group</p>

4.4.8. SERVICES INTEGRATION

Service Integration refers to the capability for different applications (even those developed by vendors who don't know each other) can cooperate and exchange data and logic effortlessly.

One platform model to achieve this kind of interaction is through **Semantic Interoperability**, the base of which is the modelling of the service logic and data through ontologies. Each application will view its own tasks (and data) as the domain restricted by the ontologies it uses, this allows for both very concrete and very general services. This paradigm also enables smooth integration between different services, even when such integration is not planned or even foreseen. By means of extension or mapping of the underlying ontology, each application will work on the same data resources with its own view, leaving to the semantic engine the mapping of the operations.

Logic can also be modelled through ontological concepts, one example of this is the OWL-S standard. A **Semantic Service Oriented Architecture**, is the extension of traditional Service Oriented Architectures, to use semantic definitions of the services instead of constricted APIs or interfaces. This way APIs are now ontological definitions that can be extended or mapped. The benefit is on one side easiness in integration and composition of services, as ontological models offer very low-coupling APIs with the same properties as the ontology itself; on the other it offers the capability of using virtual services, i.e. the logic may include calls to services that may or may not be even defined. Of course Semantic Service Oriented Architectures enable different vendors to implement the same service (see Plug and Play showcase).

The service integration showcase aims to stress the platform's ability to host new services, adding functionality to the current ones in a very smooth and simple way.

4.4.8.1. TECHNOLOGY PROVIDER'S VIEW

Services integration offers technology providers the capability to modularize to the extreme each application, making high quality products at a fraction of the price due to the re-usage of all the components already developed.

The capability of composing services enables Technology providers to easily create new components for an application, enhancing the reaction power to requisite changes.

Programming against virtual services, enables Technology providers to divide the development team, and lessen the inconveniences when one team fails to meet deadlines; in a sense it helps parallel development of dependent components.

Semantic Service Oriented Architectures enable technology providers to specialize on particular areas of knowledge. This fact will create specialized markets, for example for artificial intelligence algorithms, each with its own domain view (i.e. ontology); each niche can be easily re-used by other domains by means of ontological mapping without affecting either side view.

In general Service integration offers Technology Providers, especially small business, a very low entry barrier. Whether it is to a highly specialized market or a general one, the capability of offering products (such as components) is achieved easier than full solutions (developed in-house). In fact, for more general markets, full solutions can be offered easily by taking advantage of the open component market.

4.4.8.2. SERVICE PROVIDER'S VIEW

A service Provider will most certainly profit from the specialized competition in the Technology Provider market. The service provider may choose form different Technology Providers, or even choose several Technology Providers simultaneously, for cost optimization without necessarily affecting final quality.

For those Service Provider who act also as Technology providers, the specialized market will ease their own development as and open market for components, instead of applications, will make it easier to provide services by means of composing different components from this market.

Service integration offers flexibility, especially in the option to choose from different implementations. This flexibility may also be offered to the clients, as an asset of the service itself. Deployment flexibility and frequency can be an asset for clients, keeping the service highly personalized and updated.

The most important result from service integration is the interaction of different services, services may be exploited in new different ways, generally in more efficient ways. Examples of this include the Apple Health and Google Fit frameworks; both centred on the client's health, allow third party applications to cooperate in the final objective of improving the client's health. For a Service Provider in the Health domain providing different applications to its clients to track different aspects of their health will help gather more information and a more precise picture of the client's health, and therefore this will undoubtedly improve the service provided to the client. Of course Health is one domain where these new models fit very well,

but the same kind of interaction can be exploited for other domains, and even between domains.

4.4.8.3. ASSISTED PERSON'S AND CAREGIVER'S VIEW

From the end user's perspective the expectation is a radical change in the market, following the steps of the mobile app revolution; where a single, top-down, closed solution is opened to different vendors creating a new market where the users choose the platform and the services on top, without impositions or exclusions.

The most important impact of this flexibility for the end users will be the freedom to customize their own solution at a fraction of the price.

4.4.8.4. GOVERNMENT'S VIEW

The prospect of a revolution similar to the mobile revolution, which now is one of the most important sectors of the global economy, is indeed appealing.

Lowering the entry barrier to the AAL market, will for sure foster small business in the sector. These small business do not have to directly compete with big corporations already monopolizing the market, in fact these small business might overtake big companies if their solutions best fit the population interests. There have been other precedents for this in the internet sector, and the principal factors for this where the openness of the platform, the low entry barrier for small business and the free access to the consumers.

Fostering small businesses not only helps keep a healthy inflow of new, and innovative ideas to the sector (making it more robust); but is also helps with unemployment rates. Demand for highly qualified personnel will also impact the higher education indicators.

In general open, interoperable, flexible markets will benefit all.

4.4.8.5. RELATIONSHIP WITH OTHER SHOWCASES

4.4.8.5.1. COLLABORATION

- Plug and Play, can be played when different vendors attempt to provide the same service making them interchangeable.
- The fundamental basis for Evolution, is the service integration, even when the integration is not foreseen.
- Via Service Integration, Service Transferability can be easily achieve, especially for services that depend on localization.
- Ambient Intelligence, the openness and flexibility of the data and logic is a key factor to achieving Ambient Intelligence; especially where different vendors are involved.

4.4.8.5.2. NOT TO BE CONFUSED WITH

- Integration with legacy systems, is related to the integration with existing services, once integrated they may act as other services, and may in fact be integrated with other services.
- Service integration is the capability to integrate different services, Evolution, is this integration planned and executed in deployments over time.

4.4.8.6. EXAMPLE USE CASES

4.4.8.6.1. HOME NODE A user is enjoying a generic service, but he wishes the data generated for this service to be securely stored so only authorized people can access it. This could be achieved by installing a new service focused on the security domain. An automatic (or assisted) mapping of the initial service and the security service can be performed, so that now the security service can identify the assets (concept only available for security service, that may be mapped to the resources of the generic service) to secure and manage. During this integration neither service had to be reimplemented nor recompiled, just a new mapping ontology added.

4.4.8.6.2. CLOUD A health service is installed in the cloud, it is in charge of collecting data such as health sensor data, daily activities, and other context information. A Service provider may improve the service by adding a specialized Artificial Intelligence (AI) background service, for it to work the health ontology is mapped to the AI ontology, in other for the AI algorithms to properly identify the data. In this example the pattern recognition space is mapped by using the mapping ontology and the data gathered from the health application, equally the categories and the data sets (concepts unknown to the health application) can be mapped or deducted from the already mapped resources.

The background AI service can be then composed to enhance the service offered to the clients, so after the semantic integration and composition, clients will receive a new service that recommends how much exercise they should attempt today, or which menu they should eat; completely personalized and using already available data (without the need to input the same data again).

4.4.8.7. EVALUATION OF SERVICE INTEGRATION SHOWCASE

4.4.8.8. TESTING IF AN APPLICATION/DEPLOYMENT COMPLIES WITH THE SHOWCASE

Procedure type	☒ Analytical ☐ Operational ☐ Technical ☐ Programatical
Procedure general description	
Find indicators about the application's capability of Integrating new services, or already integrating with existing services.	

Procedure script
<ol style="list-style-type: none"> 1. List all available documentation. This may include, but not limited to: user manual, installer manual, design documentation, adaptation schema, issue report system, help fora, developer's documentation, Code comments, Commercial brochures, etc. All of these may be in different formats: web page, paper. 2. Go through all the available documentation looking for: <ol style="list-style-type: none"> a. The Ontologies used by the application, and the application components, allow for third party extension. Other developers are capable of extending or mapping the concepts to fit other applications/purposes. b. The service profiles called are generic enough so that when new services are introduced they will be matched. c. Introducing new service profiles does not affect the application, when there is a service request the application is capable of managing multiple responses and responses from different components (the treatment is generic enough).
Success cases
In step 2 there are clear indicators found that prove the application has been designed and prepared to be integrated in new services or future services can integrate with the application.

Procedure type	☐ Analytical ☐ Operational ☒ Technical ☐ Programatical
Procedure general description	
Include/integrate a new service with already running system demonstrate the capability for different applications/services developed by different vendors to co-operate/interoperate and exchange data and logic with each other without effort.	
Procedure script	
<ol style="list-style-type: none"> 1. Identify an application that is susceptible of integrating with the application being tested. 2. Run both applications. 3. Perform normal user workflows in both applications. <ol style="list-style-type: none"> a. Note information that was entered into each of them individually. 	

Success cases
<p>Information that is shared by both applications did not have to be inputted twice, the information was simply shared and both applications.</p> <p>In some cases when the information is already available the application does not request it, which might change the user workflow, this effect is a good indicator.</p> <p>Information from one application is affecting the other application. There are many manifestations, one instance is at business logic (for example Medication prescriptions affecting Nutritional program); another is they just share the data (for example a Nutritional program and Personal training application, that includes food management, show the same menus).</p>

Procedure type	☐ Analytical ☐ Operational ☐ Technical ☒ Programatical
Procedure general description	
Development of a simple module that integrates with examined service.	
Procedure script	
<ol style="list-style-type: none"> 1. Based on the included version functional manifest of the tested service, perform a functionalities test to ensure that all functionalities are still taking place with full capabilities based on the service related functionalities 2. Based on the related semantic model study the ability to easily add new functionality to the current ones. Some methods may include: <ol style="list-style-type: none"> a. Sharing information (context) with the whole environment b. Using exposed services. c. Overriding used services, enabling to slightly change the behaviour of different parts of the existing services, or to do parallel operations to dose performed by the original service. d. Mapping the semantic model. By including and mapping existing ontologies to the ones used by the original service, the latter will be able to process the new concepts. 	
Success cases	
On step 2, it is proven there are possibilities to create applications that integrate with the examined service.	

4.4.8.9. MEASUREMENT OF THE VALUE

Indicator	Service provider focus group
Value centred in	☐ Technological Provider ☒ Service Provider ☐ Assisted Person ☐ Caregiver ☐ Government
Description	
Questions: <ul style="list-style-type: none"> ▪ Was the process of including the new services more easy and flexible for you compared to your doing it with your original system? ▪ From a cost perspectives, evaluate the integration cost compared to the same integration based on your original system ▪ Add to the new services functionalities, have you profited from the benefit of having new very easy developed scenarios due to the integration process 	

Indicator	End user focus group
Value centred in	☐ Technological Provider ☐ Service Provider ☒ Assisted Person ☒ Caregiver ☐ Government
Description	
Questions: <ul style="list-style-type: none"> ▪ Have you found a multitude of open solution from different vendors in the market? ▪ Has the user the chance to choose between different applications/functionalities from different vendors/service providers? 	

Indicator	Government FG
Value centred in	☐ Technological Provider ☐ Service Provider ☐ Assisted Person ☐ Caregiver ☒ Government
Description	
Questions: <ul style="list-style-type: none"> ▪ Is the modular approach and the service integration (interoperable solution) flexibility a way to lower the entry barrier of the AAL market? 	

4.4.9. SECURITY & PRIVACY

The Security and Privacy showcase aims to demonstrate the platform capability of providing confidentiality, integrity and availability of data/resources to only authorized stakeholders of the system. Also, it aims to include the system ability to provide privacy mechanisms, allowing task, transaction or transfer of private data only if authorized by the owner of the data.

4.4.9.1. TECHNOLOGY PROVIDER’S VIEW

Adding security and privacy characteristics, traditionally have been a burden for technology providers. This is usually due to the difficulty while testing and debugging the application, and therefore is generally added at the end of the development, which makes it prone to errors on deployment phases. A good platform will help incorporate security & privacy on the whole development lifecycle, especially since the beginning of development; while not interfering with testing and debugging tasks.

4.4.9.2. SERVICE PROVIDER’S VIEW

For Service providers, it is important to have a robust security mechanism, so that clients who hired a service, receive the service while blocking clients (or malicious users) that did not pay the service from receiving such service.

Privacy is one of the most important indicators for quality in applications, providing security and privacy will rise the demand of clients and end users for the service provider’s products. In a platform that incorporates privacy and security by default, optimizing costs do not affect the performance of the security and privacy features.

4.4.9.3. ASSISTED PERSON’S AND CAREGIVER’S VIEW

As an end user, the assisted person will benefit from the Security and Privacy features, since his private data will be secured through several encryption protocols. Same, and in order to protect her/his privacy, the end user will be the only one able to access to certain data and application via the authorization features.

4.4.9.4. GOVERNMENT’S VIEW

From a Government ”legal” perspective, the showcase will be aligned with the ethical rules and enforcement of security and data privacy laws. Providing such a feature will definitely encourage government to recommend the platform for future use.

4.4.9.5. RELATIONSHIP WITH OTHER SHOWCASES

4.4.9.5.1. COLLABORATION

- Integration with legacy systems, legacy systems may have their own security mechanism, but these may be compatible with the platform mechanisms.
- Services Integration, the integration a generic security service into an application may be provided via service integration mechanisms.
- Resource Sharing, the security mechanisms are shared among all applications.
- Scalability, a smooth integration with security systems will help during smooth scaling of the deployment.
- Advanced User Interaction, personalized user interaction requires a good user authentication methods, as well as privacy considerations.
- Administration is a task that might involve at some point security (for example configuring users that are authorized to access an application, or accessing remote configuration feature) but the execution and maintenance of security is not part of the administration task.

4.4.9.6. EXAMPLE USE CASES

4.4.9.6.1. HOME NODE A user at home will enjoy privacy of hers/his data, on first instance when this data (for example the activity logs) are not leaked over the network, and if they are the protocol used is secure.

To access hers/his data the user will have to authenticate, by any means of authentication (for example user and password, or via 2 step authentication). The authentication and the context information can be combined to create the concept of session, by which the user is not required to authenticate again because they have been tracked and the system acknowledges that the user is still the same physical person.

4.4.9.6.2. CLOUD A user has to authenticate to the cloud service in order to receive the service. The sensible data has to be managed securely, making sure malicious users cannot access this data, directly from the server or through any type of automatic monitoring of the client-side or its communication with the cloud.

4.4.9.7. EVALUATION OF SECURITY & PRIVACY SHOWCASE

4.4.9.8. TESTING IF AN APPLICATION/DEPLOYMENT COMPLIES WITH THE SHOWCASE

Procedure type	☒ Analytical ☐ Operational ☐ Technical ☐ Programatical
Procedure general description	
Analyse the documentation for explicit mentions of security and privacy considerations.	

Procedure script
<ol style="list-style-type: none">1. List all available documentation. This may include, but not limited to: user manual, installer manual, design documentation, adaptation schema, issue report system, help fora, developer's documentation, Code comments, Commercial brochures, etc. All of these may be in different formats: web page, paper.2. Go through the documentation specially looking for:<ol style="list-style-type: none">a. Instances where a password is required. Check also the general quality of the password administration, for example if default passwords are provided or how easy is it to change the password.b. Identify the critical information, privacy wise, that the system is managing. Follow the process to certify that when this information is being stored or transmitted it is being done securely; preferably with standard and internationally acknowledge protocols/methods.c. Make sure the resources, software and hardware, that require access authentication are correctly managed, so the authentication mechanism may not be circumvented.d. Identify the critical components of the system, and verify the information they get, and how the get it, is protected against intentional and unintentional modification. Integrity will ensure that information is not changed, which otherwise may cause the system to take unwanted actions.e. Identify binding contracts in the system, for example buying systems, or critical services like fall warnings; and check all parts of the contract cannot deny a legitimate contract enactment. This ensures that when, in the previous example, an item is bought the buyer is ensured the seller will not take the money and leave.f. Verify all communications outside the system are secured, at least those that might transmit sensible information.g. Information security might also be critical at hardware level. Check hardware devices are certified.h. Technical information might include used and certification of security standards.
Success cases
Security has been taken into account throughout the design of the system and implementation of it. Security attributes such as confidentiality, integrity, availability and non-repudiation are ensured.

Procedure type	☐ Analytical ☐ Operational ☒ Technical ☐ Programatical
Procedure general description	
Check authentication procedures.	
Procedure script	
<ol style="list-style-type: none"> 1. Identify the user workflows that require authentication (by means of user password, or other means of credential provision). 2. Execute said workflows, when arriving at the authentication step: <ol style="list-style-type: none"> a. Provide valid credentials. b. Provide invalid credentials. 	
Success cases	
<p>Expected results at 2.a. are that the system recognises the user, and allows her/him to perform the operations she/he is certified to perform, and only those operations.</p> <p>In case of 2.b. the system does not allow the user to further the workflow. Warning is preferable, but this warning must not disclose any information about the authentication procedure (for example saying the user does not exist, when the introduced user does not exist, and</p>	

Procedure type	☐ Analytical ☐ Operational ☒ Technical ☐ Programatical
Procedure general description	
Use network sniffer to check the encryption of communications.	

Procedure script
<ol style="list-style-type: none"> 1. Set up the system in a controlled network. <ol style="list-style-type: none"> a. Make sure the network is monitored, IP datagrams can be read by the analyst using either hardware and/or software network sniffers 2. Run the system and perform all the user operations on it. When asked for input note the input values 3. Review the network log of the network sniffer. Check for communications from the devices involved in the system. Look for any sign of un-encrypted information, you may use the notes of the inputs taken in step 2 to search for failures in the communication encryption; clear text is also quite identifiable. Make sure information collected from devices and other delicate information (such as user location) is not being disclosed. The same process applies to incoming communications.
Success cases
In step 3 it can be certified external communications of the system are secured, and only the intended receivers can read such information.

4.4.9.9. MEASUREMENT OF THE VALUE

Indicator	Questionnaire for End-user
Value centred in	☐ Technological Provider ☐ Service Provider ☒ Assisted Person ☒ Caregiver ☒ Government
Description	<p>Discussion points:</p> <ul style="list-style-type: none"> ■ How can data security and privacy affect every day activities? ■ Is security something that gets in the way? You agree with the following statement: “inputting the password is very annoying”. ■ Do you take security precautions? For example changing default passwords, or using secure passwords? Or security is something being enforced (by the system or by people)? ■ Security and privacy only matters when it breaks. ■ There are a lot of laws regarding security and privacy. Do you know of any? What is your opinion of them?

Indicator	Questionnaire for TP & SP
Value centred in	☒ Technological Provider ☒ Service Provider ☐ Assisted Person ☐ Caregiver ☐ Government
Description	
Discussion points: <ul style="list-style-type: none"> ■ Is security central to the service? Or is it something that is added once the core business logic is working? ■ How would you evaluate the effort required to ensure security in your products? ■ Are your security and privacy standards very strict? Why (laws, business model, users do (not) request it,)? ■ Do you think it is feasible to “externalise” the security mechanisms? Using security frameworks will make the service easier to implement, or this usually does not work because the security frameworks does not fit my requirements. 	

4.4.10. SERVICE TRANSFERABILITY

Transferability is the ability to deploy a service on other countries and/or set-ups. Transferability has been widely achieved in the software industry, where the same product, is slightly adapted (namely the language used in the user interface) to be operational on other countries. In AAL domain Transferability is a bit more delicate as services may not be 100% software (as they may include human resources), general health/assistance workflows may differ, Culture plays an important role in may AAL service, and many regions have slightly different laws that affect the service.

Transferability in the AAL domain is a game changer, finally breaking the market free helping to reach the people in need of this technology.

4.4.10.1. TECHNOLOGY PROVIDER’S VIEW

A transferable service has to comply with certain technical, and otherwise logical, design rules:

1. Language internationalization is the first and most important rule in order to enable transference, especially to other countries, or where population speaks another language.
2. Application design must be modular, to enable the customization of the deployment-sensitive components. These include devices, services that involve human resources and/or legacy systems not provided with the application.
3. Data may also require adaptation to other cultures, have as an example a nutritional manager. Ingredients and dishes in one region, due to cultural

differences, may not be available on other regions.

4. Legal issues, especially with regard to data protection laws have to be taken into account.

Transferability opens the market, enables expansion; at the same time it brings competition that requires technology providers to be innovative and provide better quality/price than competitors. Competition is not the only result, collaboration is also possible, specialization on different aspects of the AAL problem enables different Technology providers to offer transferable sub-solutions, which other technology providers can use to integrate higher intelligence in their development.

This new market view opens the gate to new business models; handing the possibility to go from consultancy/custom-made-solutions for service providers, to white-label products, where requirements are driven by the technology providers, and then small customizations are developed for each client.

4.4.10.2. SERVICE PROVIDER'S VIEW

The ability to import transferable services from other technology providers, even from other countries, broadens the choice for service providers. This increase in supply will help to keep reduced costs, while increasing the general quality of the solution, and at the same time not giving up on customization of the services. Service providers will be able to choose solutions that have been demonstrated to be successful on other deployment sites, having greater grantee of the final product compared to contracting a technology provider to develop the solution from scratch or create an imitation product.

At the same time, Service providers may invest in new services to offer to their clients thanks to transferable AAL products.

4.4.10.3. ASSISTED PERSON'S AND CAREGIVER'S VIEW

True open markets always benefit the end consumers, they will get the services with better quality at competitive prices. In the case of open AAL market, will help people with special needs get services that otherwise would be far too expensive to service providers to provide, thus increasing the general well-being of assisted persons and satisfaction of caregivers.

4.4.10.4. GOVERNMENT'S VIEW

Open markets promote healthy competition that will help improve the overall quality of the AAL industry.

Successful technology providers will increase homeland's exports.

4.4.10.5. RELATIONSHIP WITH OTHER SHOWCASES

4.4.10.5.1. COLLABORATION

- Plug and Play, is useful when dealing with different devices in different deployment set-ups.
- Services Integration, enables seamless customization of services for different deployments, legacy system should be also customizable in such way.
- Extending the deployment of an application on other countries, involves some sense of Scalability.

4.4.10.5.2. NOT TO BE CONFUSED WITH

- Advanced User Interaction, even though the UI framework should include facilitation for internationalization, as explained, transferability may require a bit more than just language independence.

4.4.10.6. EXAMPLE USE CASES

4.4.10.6.1. HOME NODE In a home deployment a transferable application may be installed, just as any other application. External services, like home care, have to be adapted so that the local service provider receives the notification.

4.4.10.6.2. CLOUD In the cloud, the service may be installed additionally to the existing services. For example a new transferred web application is installed and linked in the home page; for users to use this new application it may require integration of user accounts and profiles.

4.4.10.7. EVALUATION OF SERVICE TRANSFERABILITY SHOWCASE

4.4.10.8. TESTING IF AN APPLICATION/DEPLOYMENT COMPLIES WITH THE SHOWCASE

Procedure type	☒ Analytical ☐ Operational ☐ Technical ☐ Programatical
Procedure general description	
Find indicators about the application's capability of Transferring to other countries or environments.	

Procedure script
<ol style="list-style-type: none">1. List all available documentation. This may include, but not limited to: user manual, installer manual, design documentation, adaptation schema, issue report system, help fora, developer's documentation, Code comments, Commercial brochures, etc. All of these may be in different formats: web page, paper.2. Go through all the available documentation looking for:<ol style="list-style-type: none">a. Translation/internationalization files. Clearly the best indicative the application is compatible with several languages is to find the translation files that hold the strings in different languages for all dialogues of the application.b. Access to databases, and other resources. Make sure data is also internationalized.c. Translations of the documentation, if the documentation, especially the user manual, is edited in different languages, this might indicate the application is intended for different countries.d. Any existing services involving human resources should be susceptible to be changed. For this optimally the services should be modularised. A possible mechanism is through the implementation of the Services Integration showcase.e. Any hardware devices are abstracted. There is no explicit reference to concrete brands/models or protocols used to interact with the devices are in a separated module outside the application's business logic. Basically if there is any hardware involved in the service, it is highly desirable that plug and play showcase also applies.f. Analyse any legacy system, it must be also be transferable. This means that either it offers different language use; or it is modularized and can be replaced on the target environment, if it is environment dependant (for example legacy systems for managing local human resources, have to be replaced on the target site)
Success cases
In step 2 there are clear indicators found that prove the application has been designed and prepared for running in different languages and environments.

Procedure type	☐ Analytical ☐ Operational ☒ Technical ☐ Programatical
Procedure general description	
Simulate the transferability of the service using platform's UI framework to another country or to other setting	
Procedure script	
<ol style="list-style-type: none"> 1. Configure the platform for UI usage, enabling access the UI Preferences editor 2. Start the system <ol style="list-style-type: none"> a. Edit the preferred language via the UI Preferences Editor now available in the UI main menu. b. Make sure the dialogues are all coherent and in the current configured locale 3. Repeat step 2, with new locale/settings. 	
Success cases	
All attempts in 2.b. should be successful.	

Procedure type	☐ Analytical ☒ Operational ☐ Technical ☐ Programatical
Procedure general description	
Demonstrate the transference of the service to set up	
Procedure script	
<p>Plan the deployment so at least one subset of deployments are using different set ups. These may vary in the devices running the software, or in the sensors used (whether it is number, type and/or brand).</p> <p>Make sure that the user issue reporting procedure takes the different setups into account, and monitor issues.</p>	
Success cases	
All deployment subsets should have a comparable number of issues reported by users. Issues are expected, but the difference of issues should not be very high.	

Procedure type	☐ Analytical ☒ Operational ☐ Technical ☐ Programatical
Procedure general description	
Demonstrate the transference of the service to different countries, where local services are being used or language/culture adaptations are performed.	
Procedure script	
<p>Deploy the same application on at least 2 different countries (the original plus at least an additional one), having a local provider on each of them.</p> <p>Each local provider should consider the translation and adaptation of the application (consider the opportunity to ask for the Transference costs indicator described below). The adaptation might include the use of local resources or services (for which Resource Sharing or Service Integration showcases might be useful). These tasks must not involve the modification of the core application.</p> <p>Each local provider should also take care of the customer service, and particularly of the user issue reporting.</p>	
Success cases	
Deployments on each country should have a comparable number of complains. If this is not the case, consider researching the root of the problem, as it might not be related to the transferability of the application. The application should work independently of language, and environment.	

4.4.10.9. MEASUREMENT OF THE VALUE

Indicator	Transference costs
Value centred in	☒ Technological Provider ☒ Service Provider ☐ Assisted Person ☐ Caregiver ☐ Government

Description
<p>The purpose is to measure the effort and costs of translating and adapting an application developed for another country.</p> <p>Break down the costs in the following categories:</p> <ul style="list-style-type: none"> ■ Translation, the costs and easiness of translating the application, and application data to the local language. ■ Human resources, costs and easiness of adapting or creating local human resource management similar to that of the original service. Take care also in any possible protocol changes that need to be applied. ■ Hardware needed, the costs of adapting the deployments so they include the required hardware resources. Note also if the available hardware resources in the local setup are compatible (i.e. 0 costs can be either because there is no hardware required, or because the hardware is already compatible) ■ Software adaptation costs, any other adaptation required. For example using different web services more appropriate for local use.

Indicator	Service provider focus group
Value centred in	☐ Technological Provider ☒ Service Provider ☐ Assisted Person ☐ Caregiver ☐ Government
Description	
<p>Questions:</p> <ul style="list-style-type: none"> ■ Is the ability to deploy your services in other countries interesting to you? ■ Is the extra effort in internationalization worth it? ■ Deploying in different deployment sites is interesting? ■ From a business perspective, allowing my services to be used on other platforms makes sense. 	

Indicator	End user focus group
Value centred in	☐ Technological Provider ☐ Service Provider ☒ Assisted Person ☒ Caregiver ☐ Government

Description
<p>Questions:</p> <ul style="list-style-type: none"> ▪ Have you ever been in the situation where a service you need is available elsewhere and you could not receive it because of language barriers? Or because of region conflicts? Or technical difficulties?

4.4.11. ADVANCED USER INTERACTION

When an application that to interact with the user, can do so in many ways. For example it can interact through different channels (i.e. multimodality) such as a screen or through spoken text (text to speech, TTS); it may also combine different modalities at once for any given interaction (i.e. output fusion), having to collect the input from the user through different channels too (i.e. input fission). Interaction may also be personalized to the user needs, for example Users that suffer from low vision disabilities may require bigger script, or no lettering at all in favour of TTS channel. The interaction may also take place in new paradigms such as Human-Environment interaction (HEI), in favour of the traditional Human-Computer Interaction (HCI); the difference being that in HCI all the possible channels of interaction are restricted to a single computer's peripherals, where as in HEI all available peripherals in the environment, (and sometimes the environment itself) may be used to interact with the users. HEI allow for features such as follow me interaction, where when the user moves, it is still possible for him to access the interaction; the system might have decided to change de display peripheral to the nearest screen, or change the whole modality; for course this requires extra processing to maintain privacy (context information of whether there are other users in the environment and if the information shown is privacy sensitive). Consistent look and feel is important in AAL because users learn to use the different applications faster and easier, specially avoiding distress to new systems.

4.4.11.1. TECHNOLOGY PROVIDER'S VIEW

There are many platforms and framework specific for UI tasks, most of them allow for a consistent "look and feel" within applications, and even between applications using the same framework. Some platforms allow for consistent look and feel system wise, across all applications as well as the system itself. These type of platforms not only allow for applications to be natively multimodal, and manage output fusion and input fission; they allow for system wise personalization, opening the gates to adaptability (allowing change in order to meet user requirements and preferences prior to the start of the interaction) and adaptivity (allowing the system to alter its interaction during run-time). These platforms generally work through the Independence between application and presentation layer. UI specific tasks, are generally one of the most tedious for software developers, using a platform that offers

all the features (multimodality, input fission, output fusion, adaptability, adaptativity, HEI, and so on may simplify developing the UI by enforcing best practice models such as the Independence between application and presentation layer, and declarative presentation.

Learning new systems is always time consuming, and may produce errors at first. Also the Independence between application and presentation layer, and the consistent look and feel constrain developer in certain cases for example when developing custom graphics such as in a videogame.

4.4.11.2. SERVICE PROVIDER’S VIEW

User interaction is one of the most important aspects of the system, after all the UI is the only thing the user will actually see of the software system.

Advance UI, offer the clients a sense of whole package, the system solves many of their problems, even though in reality there are many applications devoted to each sub task. It may be branded (especially on the consistent look and feel feature), so that this sense of problem solving is associated to the service provider’s brand; nourishing the bond of the clients with the service provider, helping them maintain the service.

On the other hand the consistent look and feel, in a scenario with multiple service provider (one offering a set of applications, and other offering another set), will make it difficult to make a difference.

4.4.11.3. ASSISTED PERSON’S AND CAREGIVER’S VIEW

User Interaction is where the end-users get most of the feel of the system, but also where they need the most help. Advanced User Interaction may reduce, and even eliminate the need for help, since the uniformity helps users learn the system quicker, HEI allows for more natural and efficient interaction with the system, and especially personalization helps the user feel the system is specifically designed for them, increasing their engagement. HEI, is an innovative and futuristic way of interaction that will make end users proud of ”owning”.

4.4.11.4. RELATIONSHIP WITH OTHER SHOWCASES

4.4.11.4.1. COLLABORATION

- Advanced Distribution, when interpreting the HEI, some sense of advanced distribution within the environment is required, but not performed by this showcase.

4.4.11.4.2. NOT TO BE CONFUSED WITH

- Personalized Content, even though the Advance User Interaction is personalized, this content is coming from applications themselves and not from the system proposing new content to add to the system; the content itself is not

edited, it is offered as is, so it is up to the application to personalize the content.

4.4.11.5. EXAMPLE USE CASES

4.4.11.5.1. HOME NODE App A and App B are installed in the home node of the user. The user does not notice there are 2 different applications, for him it is just 2 different tasks within the system. The interaction is personalized for both applications, so for the user the letters in display screens are big and in high contrast, whereas the volume on the TTS is set to low (because he has good hearing). Dialogues will be displayed in the nearest peripheral or in the smartphone if the user is out of his home, or the information shown is private.

4.4.11.5.2. CLOUD An application in the cloud may contact the home node or the smartphone node to interact with the user, then the workflow for the user is as if the application is installed in the home node.

Cloud based Advanced User Interaction is difficult to obtain, since the context information of the user may not be available, thus specific parts of the personalization cannot be performed, nor the HEI paradigm, for example a web app may have a desktop and a mobile app to contact the user, but there is no way to select one of them unless the user explicitly interacts with either of them. In Web-based communication CSS may be selected to fit the user's preferences, as well as complying with web accessibility standards; in such case the multimodality, personalization, input fission, and output fusion is entirely up to the client (that may require advice configuration to work as desired).

4.4.11.6. EVALUATION OF ADVANCED USER INTERACTION SHOWCASE

4.4.11.7. TESTING IF AN APPLICATION/DEPLOYMENT COMPLIES WITH THE SHOWCASE

Procedure type	☒ Analytical ☐ Operational ☐ Technical ☐ Programatical
Procedure general description	
Determine if the UI Framework is being properly used, by looking at all the available documentation of the application.	

Procedure script
<ol style="list-style-type: none"> 1. List all available documentation. This may include, but not limited to: user manual, installer manual, design documentation, issue report system, help fora, developer's documentation, Code comments, Commercial brochures. All of these may be in different formats: web page, paper. 2. Go through all the available documentation looking for indications of advanced UI: <ol style="list-style-type: none"> a. There might be screen shots b. Detailed user workflows on how to interact with the system to achieve different tasks, which might include a UI Framework c. Set up indications so that the application appears in the main menu. d. Message explanations, what happens or what is going on when the user gets a specific message e. Non-standard UI interfaces. Interfaces that are launched using other UI frameworks. When using different UI frameworks it will make the user confused, plus not all features of all frameworks might be achieved (in fact the opposite).
Success cases
<p>There is explicit mention to some form of User Interaction. This interaction should be done through the platforms UI Framework.</p> <p>There might be different addressed users, for example Assisted persons and care-givers; the documentation should define the different interactions for each type of user.</p>

Procedure type	☐ Analytical ☐ Operational ☒ Technical ☐ Programatical
Procedure general description	
Test the application is using the UI Framework to interact with the user	
Procedure script	
<ol style="list-style-type: none"> 1. Start the running environment, making sure at least the UI Framework and the application are started too. The preferred interaction should be the one indicated by the application, the user preferences, and/or the platform's default interaction. 2. Start the application interaction <ol style="list-style-type: none"> a. Start through the UI main menu b. Trigger application messages 	

Success cases
UI interaction is successfully done through the standard UI framework at either 2a or 2b. If Interaction can be achieved through the UI Framework it means that the application is using the platform's UI Framework.

Procedure type	☐ Analytical ☐ Operational ☒ Technical ☐ Programatical
Procedure general description	
Test the application is using the UI Framework and it is capable of multimodal communication, using different modalities	
Procedure script	
<ol style="list-style-type: none"> 1. Start the running environment, making sure at least 2 UI methods with different modalities and the application are started too 2. Start the application interaction on one UI method <ol style="list-style-type: none"> a. Start through the UI main menu b. Trigger application messages 3. Repeat step 2 for each additional UI method on different modalities 	
Success cases	
UI interaction is successfully done through the standard UI methods at either 2a or 2b	

4.4.11.8. MEASUREMENT OF THE VALUE

Indicator	Service Provider Focus Group
Value centred in	☐ Technological Provider ☒ Service Provider ☐ Assisted Person ☐ Caregiver ☐ Government
Description	
Questions: <ul style="list-style-type: none"> ■ Do you think adaptability and personalization of UIs is useful? ■ Will you be willing to loose "brand image" in UIs in favour of personalization, multimodality and adaptation? ■ Is the effort of adapting UIs to the platform's UI Framework worth the UI features (personalization, adaptability, follow me, multimodality)? 	

Indicator	End User Focus Group
Value centred in	☐ Technological Provider ☐ Service Provider ☒ Assisted Person ☒ Caregiver ☐ Government
Description	
Questions: <ul style="list-style-type: none"> ▪ How would you evaluate a system wide (i.e. not just the application) interaction? ▪ Do you think dialogs in applications need to be adapted to your personal needs? ▪ Would you feel comfortable using the UI system in follow me scenario (unfinished dialogs are displayed in the nearest capable device wherever the user goes)? ▪ How would you value a multimodal (capability of using different channels such as screens and text to speech) system? 	

4.4.12. PERSONALIZED CONTENT

Content is information and experiences that provides value for an end-user. Typically content can be seen as products like hardware, software, culture (books, music, films). Furthermore content can be interpreted in a general way, particular features and services to be considered content too.

Personalized content add even more value for each individual as it is tailored to her/his preferences. Content is personalized based on information contained in the user's profile, or knowledge on the user's context. Personalization might be done in several ways:

- by prioritizing the available content to each user's needs and preferences.
- by filtering the available content to each user's needs and preferences.
- by adapting the available content to each user's needs and preferences. The adaptation is not performed on the way to present the content, but in the content itself.
- by offering the user a selection of new content he/she might be interested in.

The personalization of content is always performed within the user workflow (i.e. no personal information is processed by third-parties).

4.4.12.1. TECHNOLOGY PROVIDER'S VIEW

Personalized applications are always more difficult to design and implement.

But proactivity of the system enhances the user experience and simplifies the workflow in some cases.

4.4.12.2. SERVICE PROVIDER’S VIEW

When content are services, and specially personalized services that the user is most likely to be interested in, there is a higher probability of having the user requesting such services. Service provider benefits from this just as from targeted advertising.

Other type of personalized content is interesting too. Personalized content makes users feel better cared for, increasing the satisfaction with their service provider.

4.4.12.3. ASSISTED PERSON’S AND CAREGIVER’S VIEW

Personalization makes assisted person feel more engaged with the service, which in most cases is what it is required for the service to provide better quality of life.

For example, a medication intake monitor service, offering personalized information and education regarding the medication the assisted person is taking (and specially when the user is not complying with the treatment) will make the end user understand the importance of complying with the treatment; instead of being annoyed by constant reminders resulting in the user ignoring the service all together.

4.4.12.4. GOVERNMENT’S VIEW

Personalized care systems are expensive. But when the system is able to personalize to each user’s needs the system not only has better acceptance, its chances of being a successful measure increase drastically.

4.4.12.5. RELATIONSHIP WITH OTHER SHOWCASES

4.4.12.5.1. COLLABORATION

- Ambient Intelligence, is used to detect the context that might trigger the proposal for new content.
- Security & Privacy, is an important part of the showcase, as the personalization of proposed content is not processed by third-parties, therefore maintaining privacy.

4.4.12.5.2. NOT TO BE CONFUSED WITH

- Advanced User Interaction, Advance User interaction personalizes the way to present the content to the user, where Personalized Content focuses on adapting the content itself to the user.
- Enhanced Market communication and distribution, is the method of distributing some content (services) but not necessary the way to propose all new content.

4.4.12.6. EXAMPLE USE CASES

4.4.12.6.1. HOME NODE A Health application is used by the user to monitor health parameters, and treatments prescribed by a physician. Under certain circum-

stances, when for example the user is not following the prescribed treatments, for example he is not exercising enough, the system will automatically push some information about his illness, and motivational messages in the hope of the user not giving up and to keep up with the prescribed treatment.

4.4.12.6.2. CLOUD A user that accesses a cloud service daily, is given the chance to access another service that he might be interested in, by processing the history of the user and the user's profile with in the cloud (no 3rd parties involved).

4.4.12.7. EVALUATION OF PERSONALIZED CONTENT SHOWCASE

4.4.12.8. TESTING IF AN APPLICATION/DEPLOYMENT COMPLIES WITH THE SHOWCASE

Procedure type	☒ Analytical ☐ Operational ☐ Technical ☐ Programatical
Procedure general description	
Find possible cases of Personalized Content being designed into the application by looking at all the available documentation of the application.	

Procedure script
<ol style="list-style-type: none"> 1. List all available documentation. This may include, but not limited to: user manual, installer manual, design documentation, issue report system, help fora, developer's documentation, Code comments, Commercial brochures. All of these may be in different formats: web page, paper. 2. Go through all the available documentation looking for: <ol style="list-style-type: none"> a. Dynamic workflows, especially in user manuals it may be suggested that some workflows change with use. b. Access to user profiles, It may be performed to personalize the available content. Pay attention especially when there is no apparent need, as this showcase might be precisely the reason for accessing the profiles. c. Explicit mention to content recommendation. User manuals may include screenshots where content is suggested, even if it is not addressed explicitly, suggestion of content is usually pervasive, for example at the main menu in one of the screen edges. d. Use of third party libraries/platforms for content suggestion, like advertisement. 3. Determine if there is some kind of content personalization. <ol style="list-style-type: none"> a. Check whether the available content is prioritized, filtered, adapted or new content is suggested. b. Check the results for each user have the potential to be different. c. Check that the privacy of the user is conserved, the sensible information is only handled locally (pay special attention when in case 2.d.).
Success cases
In step 3, the analyst determines there is at least one instance of content personalization.

Procedure type	☐ Analytical ☒ Operational ☐ Technical ☐ Programatical
Procedure general description	
Prove that in a real environment, new content is suggested and the suggestion is personalized to each user.	

Procedure script	
Important prerequisite is that the content is monitored, for example via visitor counter in a web page.	
<ol style="list-style-type: none"> 1. Make sure the suggested content is not part of the installation, the reason being that the demand for this content will unequivocally mean the user is requesting this content. 2. During operation gather the information from the content monitoring process 3. Analyse the demand for content <ol style="list-style-type: none"> a. Check not all, or the majority, of your users are requesting the same content. This might be a sign the content suggestion is not actually personalized. b. Check demographics on the content demand are coherent. For example suggestion for products available on one region is only being demanded from that region. Another example: suggesting a service targeted to women, men should not get the service suggested, even less demand the service. c. Consider that not all targeted people are going to demand the content. Low demand may mean either the content is not interesting at all, or that the content is being very personalized (the target group is further reduced due to personalization). Further analysis may be required, using tools like questionnaires or focus groups. 	
Success cases	
In step 3 the analysis shows that there is content that is being suggested, and that this content is suggested in a personalised way.	

Procedure type	☐ Analytical ☐ Operational ☒ Technical ☐ Programatical
Procedure general description	
Proof in laboratory conditions that content is personalized, and that this personalization is done locally, respecting user's privacy.	

Procedure script
<ol style="list-style-type: none"> 1. Create user workflows for the system. These workflows must include the screens where content personalization should be. 2. perform tests, at least twice <ol style="list-style-type: none"> a. Each test should represent a different persona (age group, gender, preferences, conditions, impairments...). Each test can be performed by a tester impersonating the persona, or by actual users b. Allow the user to use the system freely. This will enable the system to collect relevant information for the personalization process. Make sure the relevant profiles are filled in. c. Execute the workflows agreed on step 1, annotate the results (e.g. taking screenshots on each step of the workflow).
Success cases
Results of step 2.c. show that each user has had different experience with the workflow, clearly showing content personalization for each of them.

4.4.12.9. MEASUREMENT OF THE VALUE

Indicator	Service provider focus group
Value centred in	☐ Technological Provider ☒ Service Provider ☐ Assisted Person ☐ Caregiver ☐ Government
Description	<p>Service Providers should engage in discussion to find the value of the showcase. Discussion topics:</p> <ul style="list-style-type: none"> ■ whether proactivity can be important for your service or not, and why ■ Do clients/patients find Personalization important? <ul style="list-style-type: none"> ● Do they feel better cared for in personalized conditions? ● Does this personalization increase the engagement of the user with the service? ■ Compliance with the need of personalization from the users is of great value; it could be a competitive advantage; Personalization could be a key point in marketing efforts. ■ Importance of Advertising as part of marketing for my service ■ In system personalized service suggestion mechanisms is a new marketing tool, and it can be exploited for improving the sales of certain services

Indicator	End user focus group
Value centred in	☐ Technological Provider ☐ Service Provider ☒ Assisted Person ☒ Caregiver ☐ Government
Description	
<p>Users and caregivers engage in discussion to find value in personalized content. Possible discussion topics:</p> <ul style="list-style-type: none"> ■ Personalized content makes users feel more cared for, and it helps them increasing the engagement with the service. ■ Organization and selection of content is confusing for users. How will proactivity affect the efficiency of interactions with the system? ■ Importance of Privacy for users and caregivers ■ Would I like to receive relevant content <ul style="list-style-type: none"> ● Do users like to get suggestions about new content specially selected for them? If this content is paid services, is this considered advertising, and how does it make it feel for users. ● what kind of relevant content might be interesting 	

4.4.13. AMBIENT INTELLIGENCE

Ambient intelligence is the ability of a system based on distributed computing devices, sensors and actuators that enables it to act or react to the conditions of the environment and the status of the users. Thus it can provide solutions and services to those users specifically tailored to them, to cover their needs at that moment in that place. Ambient Intelligence can take full advantage of environments with pervasive technology, having access to context information, user inputs and computational power from "everywhere". It must be noted though that Ambient Intelligence is a capability provided by the system, but in order to leverage it, an application must make proper use of it.

4.4.13.1. TECHNOLOGY PROVIDER'S VIEW

Ambient intelligence is at the core of the platform in ReAAL, as it allows developers of applications to take advantage of all the contextual information gathered by the sensors, act on the environment, or get information about the users, all in a unified way. It is their responsibility though that this "ambient information" is properly used and presented to the user in a convenient way, to produce the effect that the desired feature of the application emerges from the system for the user when needed.

4.4.13.2. SERVICE PROVIDER’S VIEW

The customized and context-aware response of applications that benefit from ambient intelligence can be seen as an advantage over other competence applications that confine themselves within their own data and capabilities. In applications that focus on remote monitoring and similar, the information gathered from the environment can provide an overview of the status of both that environment and the user (provided that the right sensors are in place).

4.4.13.3. ASSISTED PERSON’S AND CAREGIVER’S VIEW

Interactions with an ambient intelligence application are more personal and familiar since they are already tailored to the user needs and environment. These applications can also reduce user interaction efforts since they can react to the conditions of the user environment on their own. In summary, applications that make proper use of ambient intelligence should be easier to use and require less user intervention.

4.4.13.4. GOVERNMENT’S VIEW

It is the goal of Ambient Assisted Living to reduce long-term costs of healthcare assistance thanks to continuous monitoring, of health conditions and prevention. Information and responsiveness is maximized in a full Ambient-Intelligence environment.

4.4.13.5. RELATIONSHIP WITH OTHER SHOWCASES

4.4.13.5.1. COLLABORATION

- Resource Sharing. Realizing Ambient Intelligence requires applications to share their information about context.
- Plug and Play. Environment information is available when possible, without modifying Ambient-intelligent applications.
- Advanced Distribution. Access to context is available no matter which node it comes from.
- Services Integration. Ambient-intelligent applications can compose complex services depending on context information and other applications available.
- Advanced User Interaction. The interfaces look and feel and their location can be personalized to the user depending on his/her situation.
- Personalized Content. The content provided to the user can be personalized depending on his/her situation.

4.4.13.5.2. NOT TO BE CONFUSED WITH

- Resource Sharing. Just sharing content of applications is not enough for realizing Ambient-intelligence, but only a basic, reduced subset of it.

4.4.13.6. EXAMPLE USE CASES

4.4.13.6.1. HOME NODE In most scenarios home nodes are deployed together with sensor and actuator device networks, enabling the full potential of ambient intelligence. One of the most direct uses of ambient intelligence is the definition of context-aware rules, executed by applications such as the Situation Reasoner (included in the platform). It would be possible to define a rule that under a certain temperature threshold it activates HVAC systems. However this is a very basic home-automation scenario. More advanced usage of ambient intelligence has to be integrated within applications, such as determining that because the lights in the bathroom are switched on and the rest are off, it is not probably a good moment to notify the user about his daily exercise results, but it may be a good moment for notifying nutritional information, but only if the user is carrying a mobile device.

4.4.13.6.2. CLOUD Because ambient intelligence is based on the environment of the user, a cloud-based scenario may not be the best match to showcase it: the information about the user environment comes from the sensors and actuators around the user, not from the cloud. However it is possible to locate the logic of an ambient-intelligence-enabled application within a remote cloud server. Advanced forms of context reasoning may be resource-consuming, and would not be a good idea to perform them in user devices, especially in scenarios where user devices are constrained, such as smartphone and tablets. An example can be daily activity recognition, where all the ambient information recorded during a long period of time can be analysed by complex algorithms to extract user behaviour patterns and detect anomalies.

4.4.13.7. EVALUATION OF AMBIENT INTELLIGENCE SHOWCASE

4.4.13.8. TESTING IF AN APPLICATION/DEPLOYMENT COMPLIES WITH THE SHOWCASE

Procedure type	☒ Analytical ☐ Operational ☐ Technical ☐ Programatical
Procedure general description	
Demonstrate Ambient Intelligence features of an Application	

Procedure script

1. Ambient Intelligence (AmI for short) is delivered by applications (the platform is a mere container that provides the ability to do AmI applications). As such, not all applications that benefit from the platform are designed to provide AmI scenarios. Therefore the first step in the evaluation is determining if the application scenario under analysis is in fact an AmI application.
2. If it is, it is still difficult to provide a detailed checklist of features to demonstrate, since AmI covers many different approaches and features, and it is up to developers of the application to come with AmI features that make their solution unique. Nevertheless, there are a set of features that indicate that an application satisfies an AmI scenario, but depending of the application, it may accomplish more or less. At least two, including a), can be considered sufficient:
 - a. The application consumes context information from sources other than its own software modules.
 - b. The application incorporates this information from external sources to its own business logic, providing different results under different contexts.
 - c. The application can generate new, previously non-existent, information derived from the analysis of gathered ambient information, which at the same time can be consumed by other applications.
 - d. The application can react to user needs immediately as she/he requests it, even if the request is a complex combination of different subtasks.
 - e. The application can predict the needs of the user and react accordingly. For example activating the heater system when the user is heading home on a cold winter day.

Success cases

Compliance with the above steps and some of the subpoints indicate that the application makes use of AmI principles to provide its features. However this is not an absolute measurement. While accomplishing the above can be considered providing AmI, there is a range of how much Ambient Intelligence is put into an application. This is not an indication that an application is better or worse depending on how much AmI goes into its business logic. Some AAL applications (the most basic ones) may not need to go this far to provide their features (for instance home automation-control ones).

4.4.13.9. MEASUREMENT OF THE VALUE

Indicator	Service Provider Focus Group
Value centred in	☒ Technological Provider ☒ Service Provider ☐ Assisted Person ☐ Caregiver ☐ Government
Description	
<p>Questions:</p> <ul style="list-style-type: none"> ■ Do you currently provide services that can qualify as Ambient Intelligent? <ul style="list-style-type: none"> ● If so, would the platform features make it easier to provide these? ● If not, would you think about providing them taking advantage of the platform features? ■ How familiar are you, and if so, how do you value, the following concepts, in the scope of AAL? <ul style="list-style-type: none"> ● Semantic services ● Context awareness ● Context reasoning ● Pattern recognition ● Service composition ● User interaction ■ Do you think your customers would acknowledge Ambient Intelligent applications enough to consider it an advantage or a valuable differentiation to conventional non-AmI solutions? <ul style="list-style-type: none"> ● Is there any other scenario where you find Ambient Intelligence an asset, something valuable for you? ● Do you consider the investment on AmI solutions worth it? 	

Indicator	End User Focus Group
Value centred in	☐ Technological Provider ☐ Service Provider ☒ Assisted Person ☒ Caregiver ☐ Government

Description
<p>Questions:</p> <ul style="list-style-type: none"> ■ Do you perceive any advantage or a valuable differentiation to conventional non-AmI solutions? ■ What do you think about applications and devices gathering and analysing your personal and environmental data, even if this data is not shared with anyone? ■ Do you think it is worth the investment on an amount of sensors and devices considering the benefits shown by AmI applications?

4.4.14. ENHANCED MARKET COMMUNICATION AND DISTRIBUTION

An important aspect of AAL solution production is the communication with the market, knowing what is needed, what users want; as well as good and direct distribution channels, AAL applications need to reach the appropriate people.

An open platform needs a powerful tool for market support, a tool that puts in direct contact all stakeholders involved in AAL. Such tool could be thought as part project management tool, where different teams are responsible of different aspects of the final product, as well as being able to communicate, and track all the ongoing tasks like feature requests, new milestones/releases, malfunctions, etc. Another view to this tool could be thought similar to digital stores, concretely application markets; where users are able to buy, download, and install AAL services as well as searching, browsing and rating the most interesting AAL services for them.

4.4.14.1. TECHNOLOGY PROVIDER'S VIEW

This kind of market tool enables Technology providers to promptly get the true needs of the market, both from users' end and from service providers. It promotes the capability of Technology providers to adapt and provide technologies able to fit the needs, this promotion turns into profit as early and/or quality producers will get the advantage over the market.

The differentiation of the roles of Technology provider and Service provider enables more independence and creativity on the technology provider, as well as distinct business models.

4.4.14.2. SERVICE PROVIDER'S VIEW

Knowing what clients need is crucial for any business model. A market tool that enhances communication and distribution on the market is the perfect tool to get ahead of the demands and always meet what it is expected from service providers.

The ability to communicate with the market, rather than with a technology providers in particular, promotes competition and yields the option to choose from

different providers the best fitted solution for the service provider.

Distribution of the service, for many Service Providers is more complex than just installing the application at the deployment. In many cases the service requires allocation of human resources (for example, doctors), that in some cases will need to go to the deployment (for example nurses). In traditional digital stores this fact is not accounted for; thus the market tool must accommodate this.

The solution is to bundle software, hardware, and human resources as a service offered to the final user. This feature enables the creation of personalized service bundles, for example the same service can be offered with or without hardware; clients that require hardware will opt for the first, users that already have the required hardware will opt for the second. Bundling of AAL services may go a level beyond, and offer different package options (for example a package that has services A and B versus another that contains A, B and C), helping Service Providers manage their offer by keeping it in line with their billing and organizational structures.

4.4.14.3. ASSISTED PERSON'S AND CAREGIVER'S VIEW

Access to all available services is an essential part of what an end user needs, without this access there is no chance for end users to receive or even know there are AAL services that can make their life easier. A digital store is a mechanism that fits this objective cleanly, but it must be easy to use, browse the needed services, as well as allow feedback to providers about their real needs.

The capability to get services in bundles simplifies things for end users, plus usually bundled services have better deals.

A digital store enables users to get services from different providers, not being restricted to a single provider gives more freedom and flexibility to choose not only between different providers, but several simultaneously. For example Provider A offers a service that Provider B does not, and vice-versa; the user has the freedom to hire both services.

Some services are restricted by location (for example housekeeping services) or external services (like communication with local authorities), which means not all service providers are able to expand where the demand physically is. The market tool promotes that successful services of this kind can be offered elsewhere, Service Providers will emerge where the demand rises; since the hardware and software resources can be shared and transferred.

4.4.14.4. GOVERNMENT'S VIEW

An open marketplace for hiring AAL services, is the best way to get the population assisted by AAL services.

Marketplaces also stimulate competition, and offer grounds for new businesses to flourish.

A "central" repository of AAL services, is an advantage; as for regulations and required specifications can be monitored on AAL services so they comply with health standards.

4.4.14.5. RELATIONSHIP WITH OTHER SHOWCASES

4.4.14.5.1. COLLABORATION

- Services Integration. The main way to make different bundled services to work is through service integration, services must be able to interoperate in different circumstances, such as different deployments.
- Plug and Play, is the main instrument by which device dependent services will be able to be deployed elsewhere (probably with different devices).
- Resource Sharing, is important to optimize resources in each resource, bundled services will rely mostly on this capability.
- Service Transferability, will be required for services that can be offered in several countries.
- Scalability, will benefit greatly from the store tool, since it simplifies deployment to the point that end users can do it by themselves.
- Evolution, is evident when new AAL services are installed in a deployment, thus making it evolve.
- Evolution, is an important aspect to take into account when dealing with store operations and products.
- Personalized Content, application proposals may be done either by the home system or by the digital store. Purchase history will be the main source of information to personalize the services and application suggestion list for each user.

4.4.14.5.2. NOT TO BE CONFUSED WITH

- Service Transferability, is the capability to deploy on other countries, whereas Enhanced Market communication and distribution is one method to do so.

4.4.14.6. EVALUATION OF ENHANCED MARKET COMMUNICATION AND DISTRIBUTION SHOWCASE

4.4.14.7. MEASUREMENT OF THE VALUE

Indicator	Developer Questionnaire T0
Value centred in	☒ Technological Provider ☐ Service Provider ☐ Assisted Person ☐ Caregiver ☐ Government
Description	There are questions related to possible digital store usage and perceived quality in T0 questionnaire, these indicate the interest of developers in the possibility of having a digital store.

Indicator	Service provider focus group
Value centred in	☐ Technological Provider ☒ Service Provider ☐ Assisted Person ☐ Caregiver ☐ Government
Description	
<p>Questions to be answered:</p> <ul style="list-style-type: none"> ▪ Structure of an AAL Digital Store <ul style="list-style-type: none"> • Is it beneficial to have the roles distinguished in AAL Digital Store? • Is it a good idea to think of AAL services as sets of software, hardware and human resources? Or does this categorization need to be changed? To what? • Will the bundling of services be used? Will different bundles be useful in your business model? ▪ AAL Digital Store communication <ul style="list-style-type: none"> • Would you, as a Service provider, react to feature requests or suggestions for new apps from users? • Do you think browsing the available resources provided by technology providers will inspire you to include interesting components in your AAL service? ... Or create new AAL services? • Do you think an AAL Digital Store should have enough mechanisms to judge the successfulness of AAL services in, for example other countries? • Will it create interest to deploy successful services in your region? • Does AAL Digital Store need to include the required mechanisms for interacting with Technology providers? ▪ AAL Digital Store distribution <ul style="list-style-type: none"> • Do you think the digital store is a good tool for your business? Why (not)? • Do you think the digital store will attract new users to hire your service? 	

Indicator	End user Focus group
Value centred in	☐ Technological Provider ☐ Service Provider ☒ Assisted Person ☒ Caregiver ☐ Government

Description
Questions to be answered: <ul style="list-style-type: none">■ Structure of AAL Digital Store<ul style="list-style-type: none">● Is it a good idea to think of AAL services as sets of software, hardware and human resources? Or does this categorization need to be changed? To what?● Do you think the bundles of services are beneficial as a potential client?■ AAL Digital Store communication<ul style="list-style-type: none">● Do you think service requests are something you will use?● Do you think service requests will be complied when issued?● While browsing the available services, which information will make you decide if you need it, and where you should install it or not?● How do you think problems should be reported to support specialists?■ AAL Digital Store distribution<ul style="list-style-type: none">● How do you envision the perfect distribution of an AAL service when you click “buy”?

CHAPTER 5

VALIDATION AND ANALYSIS

This chapter is devoted to provide proof of the validity of the methodologies designed in the previous chapter. This proof is aligned with the results. First validation of the SDS model is discussed by providing some improvements to existing SDS platform which enable key aspects provided by the model. Then the model is used in standard and extreme cases, exploring in each case different aspects of the model, and therefore showing the methodology is capable of modelling and providing solutions to complex SDS. Then the 14 showcases for AAL platforms are validated with different stakeholders, not only providing proof of the value of these showcases, but validating the methodology as well. Finally 2 full cases of the cocreation methodology execution in the context of large scale deployment European research projects are discussed.

5.1. MODEL VALIDATION

This section will present different SDSs designs, in one case a validated implementation, for which the proposed model has been used. These system models should be interpreted as example cases to the proposed design models and patterns, as well as a validation exercise to demonstrate the design models themselves are applicable to many system architectures.

5.1.1. FULL EDGE SDS IMPLEMENTATION: INDOOR LOCALIZATION BENCHMARKING TOOL

The evaluation and comparison of complex Ambient Assisted Living (AAL) systems is still far from being a reality [119], there is a lack on standards to evaluate these systems, maybe because of both the diversity and the complexity of AAL services and solutions. Evaluation and assessment of components, services, and platforms for AAL systems is not only extremely interesting to efficiently offer AAL services, but it also essential to ensure the effectiveness of said services; and ultimately promoting the progress and success of AAL technologies.

EvAAL is an international competition on AAL supported by the AALOA association [120] and organized by the universAAL project [90]. It aims at advancing the state of the art in the evaluation and comparison of AAL platforms and architectures. EvAAL aims at contributing to AAL disciplines in the same way as other competitions have contributed to their respective areas. The main technical objectives of the competitions organized by EvAAL are to:

1. enable the comparison of different AAL solutions,
2. experiment with benchmarking and evaluation methods,
3. identify relevant AAL problems, requirements and issues,
4. identify new and original solutions for AAL.

The method to enable the comparison of different AAL solutions is based on establishing suitable benchmarks and evaluation metrics that will be progressively refined and improved with time. In particular not only algorithms, methods and technologies will be compared, but also cost, deployment effort, user acceptance, among other metrics will be considered. This method will promote an environment in which researchers, students, practitioners and industries can compare their solutions thus demonstrating to the potential market the possibilities of the different solutions.

The first edition (2011) a single track on the topic of "Indoor Localization and Tracking" was selected. Being a key component of many AAL services, and of particular importance to the context model of the user, from which many other services may derive. Global Navigation Satellite Systems (GNSS) are a common solution for localization, however they are largely unavailable for indoor environments. The scope of this track is to award the best indoor localization system to be applied to, or used by, AAL services.

A tool for collecting participants results of indoor localization and tracking was developed. This tool was specified using these simple use cases:

Competitor sends location data The competitor must be able to send easily location data (2D coordinates + timestamp) to the system. The system upon arrival of the data must analyse, display and store it, in order to produce an evaluation of the system, verify the received data is correct, and to log the benchmark for further post processing and evaluations.

Competitor receives context events The competitor receives device activation and deactivation events through the same mechanism as for previous UC. During the execution of each benchmark the actor will manipulate some of the predefined devices, these events will reach the competitor (in the form of deviceID + 2D coordinates of device + status + timestamp). Devices include a set of switches in the test environment, the TV, exercise machines, bed sensor, or presence sensors.

Execution of benchmarks Before the first competitor arrives, the system is configured with the desired paths and areas of interest, this configuration remains for all the competitors identical. The system announces the imminent start of a path (previously requested by the evaluation committee), so the actor gets ready for the test and knows exactly when to start moving. Then it guides the actor through the path, by giving queues (either visual or auditory, preferred both), queues include normal stepping, imminent pause, pause time and imminent pause end. After each path the all objective evaluation criterion for the competitor's solution must be calculated, displayed and stored.

5.1.1.1. SEMANTIC MODEL

The tool was implemented over universAAL IoT Platform, it was then necessary to define the ontological concepts required for the application. Luckily universAAL IoT also offers an upper ontology with many generic concepts. One of these concepts is *Location*, which being a semantic class it is defined in many ways. A *Location* can either be a *Place*, a *Way* or a *Point* in space, and Locations can be contained in one another, or adjacent between them. Places can be indoor or outdoor, yielding the notion from a room, all the way to a country. Ways are paths that connect Locations together. But the most interesting upper Location concept is the *Point*, this concept defines a point in 2 or 3 dimension Cartesian space with regard to a set coordinate system. These points are also useful to define areas of interest, like rooms, or zones within a room, this is a particular concept in the competition for lower resolution tests.

Although the upper concept for *Point* might be enough, it was too generic when dealing with the specific problem dealt in the EvAAL system. For example there was the need to keep track of certain times, concretely the time declared by the competitor, the time the object was created, and the time it was received in the AAL Space. Coincidentally in the latest version of universAAL (version 4.0.0) these timestamp distinctions will be part of the official context event model.

The upper ontology also offers another useful concept, the *Device*, This is the upper concept for any hardware in the AAL Space. This concept is intended to be extended by a full device ontology, however the specific *CompetitionDevice* (in figure 5.1) provides additional information about the device in the context of the competition.

Nevertheless there are concepts unique to the EvAAL competition like the competition it self, competitors and the phases that the competition is divided into.

The *Competition* concept is used to define a subject for when *Phase* changes (to be announced through a context event), also useful to notify other nodes about the connection to use to get to common resources like the paths.

Phases are the concept corresponding to the phases in the benchmark testing, per competitor there are 3 phases: installation, testing, and benchmark. Each phase can have a status, which can be that the phase is started, or stopped; but it can be also paused, to later be started again (only for installation phase) and it

Figure 5.1: EvAAL ontological concepts.

can be cancelled (only for benchmark phases); these are known as the enumeration *PhaseStatus*. Installation phase measures the time it takes the competitor to install his/her solution in the evaluation environment, therefore a lapsed time is needed to synchronize correctly this time. Benchmarking phase needs to be instantiated by the concrete benchmark being done, for this it has a property defining the *benchmarkID*, which will be used to correctly load the path in slave nodes.

Lastly, but not less important the *Competitor* concept is used to have a subject for location events, the predicate for these location events is that the competitor publishes a certain *LocationPoint*. *Competitor* has also a property to identify each competitor, by their name, this is very useful to automatically categorize the results and logs of the evaluation process.

5.1.1.2. CONTEXT MODEL

The communication mechanism to connect components is the universAAL IoT context bus. The main three context events used in the systems are:

Subject	Predicate	Object
Competitor	Publishes	LocationPoint
Competition	StartsPhase	Phase
CompetitionDevice	isActivated	Boolean

Table 5.1: The most important context events for EvAAL system

Each context event has a specific purpose. For example *LocationPoint* events are used by the competitors' software package to publish the estimated location by the competitor's solution. On the other hand *CompetitionDevice* events are used to communicate the competitor's about the activity of devices. Phase events are used to synchronize the procedures through the actual competition event.

5.1.1.3. COMPONENT MODEL

Figure 5.2: EvAAL components in relation with the main context events.

Once the connectors are defined, the business logic of each component can be concreted.

eval.terminal This is intended to be a singleton component in the AAL space, since it is the user interface intended for the evaluator team. It enables the evaluators to monitor every aspect of the benchmark controlling the phases.

eval.log The log component is a very simple component that listens to every EvAAL event (*LocationPoint*, *Phase*, or *CompetitionDevice*) and takes note of it on a text file.

eval.resourceServer The functions of this component are just two, but accomplished through the same mechanism: an http servlet. The first function is to receive all the logs from all the nodes (see 5.1.1.4) and categorize them appropriately. The second function is to securely serve the ground truth, i.e. benchmark paths, and AOI definitions to the terminal.slave components.

eval.terminal.slave It displays the installation timer, the testing phase results even during the benchmark phase it displays the real path and the metrics. But it does this in a passive way, it does not control the phases, that is something only done by the terminal component.

eval.socket.server The main component for the competitor to interface with, it opens a server side TCP/IP socket connection which admits simple ascii text commands that are translated to *LocationPoint* context events. It also translate *Device* event into simple ascii text notifications (through the same port).

eval.knx.device The sole purpose of this component is to translate the events collected by universAAL's KNX exporter into the *CompetitionDevice* events required by the system, by changing the id and adding the location.

eval.bike In the same line as the `knx.device` component, this component's mission is to publish device events, only events from the static bicycle installed in the Smart House Living Lab.

dummy.device A third publisher of device events is actually a test component. This test component is just a simple interface that configures the device events to send, and sends them.

5.1.1.4. DISTRIBUTION AND DEPLOYMENT MODELS

Thanks to the distributed framework that is `universAAL IoT` there are some interesting features that can be developed for the evaluation system.

Most notoriously, the ability to connect directly given a default configuration. This enables competitors, once at the competition place, to run the competitor's package in their machines, as they have been doing for their tests at their own lab. `universAAL IoT` will recognise this node and include it in the `uSpace` automatically, connecting the buses and therefore exchanging the context events.

All nodes will share at least the `eval.log` component, so logs are produced at all nodes, and recovered at the `resourceServer` component. The `resourceServer` component, along with the log and terminal components is allocated in the Main node, thus centralising the control and collection of results.

There can be any reasonable number of Slave nodes, containing a `terminal.slave` (and `log`) component, used mainly to display the status of the phases and data collected to competitors, evaluation team, and spectators.

The Comp Node (the competitor's software package) contains `eval.socket.server` and the `eval.log` components, by default does not include the dummy device component, but with a simple command this component is enabled for competitors use. When the Comp node is not able to join the AAL space, then a remote server has to be set up, to do this just install the `socket.server` component in a Slave node (or the Main node, but the slaves are generally computationally less loaded), and then the competitor can connect to the AAL space.

One important step that has to be taken with multi node configuration, is a system time synchronisation. Time is important, it is used to interpolate the real position and errors in timing can lead to metric errors.

5.1.1.5. SYSTEM RESULTS

Along this three editions the `EvAAL` system has had important improvements that have had effect on the usability of the system. Especially on the competitors side, even though the interfaces have remained practically the same throughout three editions, competitor's perception of the system has improved, as seen on table 5.3.

	2011	2012	2013
Main Function- alities added	Location events, Guidance	Device events, Phases (installation time, testing pre- visualization), slave, metrics	Path distribution, log recovering, Peer display
Improvements		Guidance, display, storage	MW2.0, Interface, Phase synchronization, Path editor, post metrics, Path verification
Issues	UPnP D&P, data storage, post metrics	post metrics, knx proprietary library reliability	MW Join
Number of Par- ticipants	7	8	7
Mean perceived easiness	7	8,29	8,67
Mean comple- tion easiness	7,33	8,25	6,85
Mean debug as- sistance	6,17	6,5	8,33

Table 5.3: summary table per year

5.1.2. HYBRID CLOUD-EDGE: PASSAGE

PASSAGE is an European project proposal submitted to the H2020-SC1-FADTS-2018-2020 DT-TDS-01-2019¹ call devoted to develop Trusted digital solutions and Cybersecurity in Health and Care. This section quotes the relevant excerpts from the submitted proposal document.

PASSAGE relies in the experience and leadership of its partners in the AHA domain to address the identified open challenges, and cover technology gaps. PASSAGE would enable the new generation of smart AHA Digital Services addressing more open, connected, transparent and nimble technological frameworks that provides dynamic and self-adaptive configuration capabilities, friendly experiences through intuitive and ubiquitous interactions, ambient pro-activeness, learning ca-

¹<https://ec.europa.eu/info/funding-tenders/opportunities/portal/screen/opportunities/topic-details/dt-tds-01-2019>

pabilities, data processing and modern graph-based storage and analytic services under a protected data & privacy environment. However, current landscape of platforms in the domain have addressed these needs partially, and frequently without a multidimensional approach. PASSAGE would use a multi-dimensional approach considering every economic, social and technological aspects, what makes PASSAGE a unique approach for implementing the new generation of smart AHA services. PASSAGE would address these identified needs and propose the first representative of a New Generation Digital Platform in the AHA Domain named PASSAGE DP.

PASSAGE DP is created as a global frame for new digital services that will follow the IoT Reference Architecture (HLA) [121] functional model described by the AIOTI-WG3 which is compliant with ITU-T Y.2060 IoT Reference Model, OneM2M reference architecture and IIC's Industrial Internet Reference Architecture (IIRA). From a high level architecture view the PASSAGE DP is comprised of two main blocks: The Cloud Environment and the Smart Living Environment 4.0 (edge).

5.1.2.1. OPEN INTEROPERABLE DIGITAL PLATFORM FOR AHA: CLOUD ENVIRONMENT

PASSAGE DP cloud platform would include enough different specific features aligned with AHA needs enhanced by inclusion of means to improve: scalability, self-* mechanisms, semantic interoperability and add new services to provide connectivity, device discovery / lookup, cloud-based storage, security and semantic annotation. The Cloud Environment would include specific cloud-based features aligned with AHA needs enhancing slicing, virtualisation and virtual functions management. This includes a semantic framework, with aim to improve current state of the art on semantic interoperability, which operates using specific semantic models, components, techniques, reasoning-enabled repositories and data models for deep insight and smart processing capabilities of data, including aspects of security; including:

- actuation and semantic translation of data and control commands, exploration, and discovery
- repository of semantically annotated components and services
- semantic adaptation of a working system with minimal interruptions

Cloud Environment would also be able to analyse the scalability and practicality of semantic technologies, increasing interoperability between heterogeneous AHA services, by applying data quality improvement techniques on the basis of existing standards and recommendations. In order to ensure the interoperability, interconnection and access to data from different open and legacy platforms, digital platforms, devices and edge components, a Broker Component (based on FIWARE Context Broker/ORION and AIoTES²) would be implemented to provide

²ACTIVAGE IoT Ecosystem Suite (AIOTES) is the main Technological framework developed in the ACTIVAGE H2020 project: <http://activageproject.eu>

publish/subscribe functionalities and expose data in-line with standardized interfaces (i.e., NGSI9 and NGSI10). In this way, Cloud Environment will also deploy a back-end cloud infrastructure that enable PASSAGE DP with data storage, semantic annotation and analytics. Cloud environment would provide security and privacy by-design semantic protection models, including anomaly detection and blockchain components (COMSEC). Finally, the knowledge acquired from the data captured would be used to automate and intelligently execute different tasks based on Self-* mechanisms. These mechanisms would be implemented to respond to capture, process and applied self-* behaviour needs in a working environment with minimal interruption (e.g. no need for complete restart the system), giving support for the AHA environment and automatic decision-making included.

5.1.2.2. OPEN INTEROPERABLE DIGITAL PLATFORM FOR AHA: NEW GENERATION SMART LIVING ENVIRONMENT 4.0

The role of edge computing to date has mostly been used to ingest, filter and send data to cloud systems. In the AHA domain, homes are equipped with sensors, devices and gateways which transmit raw data to the cloud with minimal processing and actuation capabilities. PASSAGE would disrupt this state of play moving essential functionality, computing, storage, actuation and analytic power to the edge: the living space which is close to the senior user (a.k.a Edge Space. The new Smart Living Environment 4.0 (SLE 4.0) is featured by data protection and security at every node, with privacy full control by users, computing capacity to run AHA Smart Services, AI capability for adaptation of the ambient and actuation to short-time and long-time variations of user's life conditions, interoperability and orchestration capability of a high variety of sensors (ambient, health), smart devices (smart watches, service robots), UI devices, and data & services exchange capacity with other home automation platforms (Google Home, Alexa), energy efficiency and transport platforms.

Edge Platform Hardware reference design. PASSAGE will deliver the first reference design for providing a native edge hardware. This reference specification will offer a trusted execution environment for SLE 4.0 that:

- Collects environmental information, including both context (home status) and personal (health, cognitive, etc.) data
- Processes this data in a secure form, which includes the execution of embedded intelligence algorithms and edge AHA smart services;
- Transmits in secure way information, processing results or alerts to network using the available interfaces (NB-IoT, WiFi, 4G)
- Interacts with the user using different available modalities (e.g. through a TV screen, or touch panel)
- Manages the node's power for low power and mobile applications; and power failure recovery for sensitive applications.

PASSAGE DP will produce two implementations of this reference hardware design, one based on traditional STBs (Set Top Box, a device connected to the TV, acting as a home gateway), and a second wearable node which the user would carry with them.

5.1.2.2.1. PLATFORM MIDDLEWARE SLE 4.0 would be implemented through a digital space centred on the user through ubiquitous computing capabilities, which offer services directly to or collect information from the users. Since the edge is the segment nearest to the users, it will have to adequately relay information to and from the user, as well as store, infer and apply the unique preferences and personalization its main user has. From architectural point of view, SLE 4.0 middleware is an aggregator of a distributed set of heterogeneous devices, sensors or actuators (Nodes, compatible with the Edge Hardware reference design) with different properties and functionalities (Capabilities). The SLE 4.0 middleware provides the necessary functionalities, mechanisms and interfaces to enable secure interoperability between nodes to orchestrate all the available nodes' capabilities offering them in an integrated way to applications (Smart Services). Through this smart orchestration and dynamic discovery, all the Smart Services in an Edge Space can share and use these capabilities, while providing users with a highly personalised environment.

Edge Spaces may overlap: a node may join more than one space at the same time, sharing its capabilities with all the joined spaces, but keeping data segmented in each space. The Edge space must be able to connect with cloud environment, enabling cloud applications to exchange data with the edge applications. "proxied" capabilities act as normal edge capabilities but are effectively provided by cloud components. Additionally, Edge Spaces may extend remotely, in order to offer "roaming" experience to the user, enabling the enjoyment of both local and remote spaces (e.g. controlling home remotely through mobile phone). For these remote operations, SLE 4.0 will provide a standard interaction with PASSAGE DP cloud platform; optionally complementing the local capabilities, with artificial intelligence, big data and other features provided by the cloud.

The core of the SLE 4.0 middleware will be based in the results of the UniversalAAL IoT platform, and other related projects such as SENSINAT, Cloud4All and GPII and will offer specific semantic models, mechanisms and repositories for :

1. automatic discovery of Edge nodes (adding distribution model as context)
2. aggregation of capabilities and data sharing (member management), including aspects of security, privacy and accessibility
3. data storage and embedded intelligence (specialised platform components and semantic models)
4. advanced user interaction (adaptable, personalised and accessible) with context awareness and hyper-personalisation methods

5. Edge Space interconnectivity (advanced dynamic distribution model)

5.1.2.2.2. SERVICES RUNTIME CAPABILITY SLE 4.0 main goal is to provide a set of functionalities on building a new generation of smarter AHA services and solutions based on the use of AI technologies. These prebuilt functionalities are the Edge Smart Services, components presented as service members. These set of Edge Smart Services take advantage of the capabilities of the edge space to profit profusely from the edge computation on local data such as data analytics, machine learning, long-term tracking and abnormality detection, all thanks to the wealth of information gathered through time from relevant sensors and other services, without sending any data to outside the Edge Space. Some examples of Smart Services form the initial use case are:

- Final services built on these Edge Smart Services can easily conceptualize and implement smarter functionalities based on the share AI of Edge components (e.g. using pattern recognition to detect and anticipate emergency situations).
- The shared functionalities among Smart Services will enable the composition of new services from existing ones; operating in such high level abstraction that even users could define these services (e.g. create a service which learns when the user gets home and according to environmental conditions (outside and inside home) prepare lights, temperature, etc.).
- It is also possible to learn behaviours of users; detect behaviour changes and provide suggestions to improve the quality of the services offered by the different nodes in the Edge Space (e.g. The system recognizes the user progressively forgets things, and suggests brain training games from the marketplace).

5.1.2.3. DATA MANAGEMENT, DATA SHARING AND MARKET-PLACE

PASSAGE DP would focus on providing data management, data sharing and marketplace capabilities to reduce the current fragmentation in the AHA domain and enable the interoperability requirement about the data being used for activating single digital marketplace(s), there is also a lack of semantic interoperability across diverse datasets that refer to the same data entities with same type but with different format data (semantics). Nowadays, most of the data related to health-care and well-being collected reside in a wide variety of fragmented systems and databases, including operational systems and online databases and proprietary data warehouses. As part of the PASSAGE DP data management approach the work initiated on Ontologies and AHA Data modelling, already initiated in ACTIVAGE LSP, will continue and furthermore will focus on consolidating and the enlargement of SAREF to the Active and Healthy Living domain.

There would have been different marketplace enablers:

Interoperability and Data Management offering a unified way of accessing & querying vast amounts of AHA-related data. Building blocks for semantic interoperability and interoperable data exchange capabilities.

Data Model for Sharing using linked data principles as main standard from the W3C, provides interoperability of diverse AHA datasets using existing ontologies for AHA.

Trusted Data Sharing a blockchain-based Trust Management and Access Control platform enabling secure and trustworthy data sharing.

Application store will provide an open approach for data-driven applications based on interoperability common semantics and trusts frameworks.

5.1.2.4. SECURITY, DATA PROTECTION AND PRIVACY

The project takes advantage of edge technology to guarantee data protection and security, while providing full control to the users. Along with the IoT layer (edge), cybersecurity provisions and services are applied at the network and cloud layers. All such tools are intertwined under a secure-by-design architecture, while a common set of security protocols and standards (including standardised data encryption and cryptography techniques) will be used to ensure interoperability among the security features.

In this context, registered user entities will enjoy end-to-end security through the "universal" PASSAGE security framework and user interfaces. This way users can keep/exert full control over their privacy and security rules (enforcing their own standards of data security GDPR compliance), and only parts of the data they wish to share are exchanged on demand. So, PASSAGE can contribute through the development of blockchain-based trust-ensuring frameworks for sensitive data exchange, enhancing data protection and traceability aspects. The main component will be an Open Distributed Ledger (ODL) implemented by blockchain technology. As a result, data transactions and services between members of the wider AHA ecosystem will be executed without a central infrastructure (a federated approach is applied - on the basis of the IIS reference architecture - for the realisation of an open, secure data ecosystem made up of trusted parties).

5.1.3. FEDERATIVE BLOCKCHAIN OVER P2P: H-CHAIN

H-Chain is an H2020-SC1-DTH-2018-2020 SC1-DTH-08-2018³ proposal seeking a prototype for a European interoperable Electronic Health Record (EHR) exchange

³<https://ec.europa.eu/info/funding-tenders/opportunities/portal/screen/opportunities/topic-details/sc1-dth-08-2018;freeTextSearchKeyword=;typeCodes=0,1;statusCodes=31094501,31094502,31094503;programCode=null;programDivisionCode=null;focusAreaCode=null;crossCuttingPriorityCode=null;callCode=H2020-SC1-DTH-2018-2020;sortQuery=openingDate;orderBy=asc;onlyTenders=false;topicListKey=topicSearchTablePageState>

solution. This section contains relevant excerpts for the submitted proposal.

H-Chain would make use of and extend the results of a series of successful previous projects in security and privacy and EHRs data structure and storage, to focus on harmonization and integration of current available health data silos that boost social and economic potential benefits of health data exchange. In this sense, H-Chain aimed to advance the state of the art of eHealth technology by providing a holistic approach, GDPR and the security of network and information systems directive (NIS) compliance, and targeting all layers of citizens' health data and care provision management, such as creation, storage, access, services acquisition etc., to address:

1. Harmonization and semantic standardization of eHealth data.
2. Secure and high quality integration and interoperability of current health data silos.
3. Incentive programs to drive social and economic benefits resulting from adoption of communication strategies that boost use and exchange of data among all involved actors.

H-Chain's implementation approach is based upon three pillars:

GDPR and NIS directive compliance. Current European legal framework requires confidentiality and protection of every individual identifiable health data that could be transferred, received, handled, or shared by healthcare professionals and citizens.

Data harmonization and semantic standardization. H-Chain will have the capability of acting as semantic interoperability layer to allow integration and interoperability of heterogeneous data types from different distributed data sources. The proposed semantic statement, based on arbitrary ontologies, allows the extensibility to new subsequent types of data (e.g. -omics data) while interact transparently with citizens/professionals.

Enhancement of citizen's access and control of their EHRs and health care providers, including quantified self-data or m-health apps and devices, cross border health provision and engage professional assistance towards a truthful information access, reduction of data fragmentation and inaccuracy caused by communication delays and coordination errors

5.1.3.1. TECHNOLOGICAL APPROACH

The technological approach of H-Chain consists in deploying APIs that can be easily integrated with current health systems, without need of changing current structures or working processes. Citizens will control data that are kept in the original database. At any given moment, citizens can manage access permission to their

data. Accesses will be securely traceable with the combination of blockchain technologies and semantic smart contracts. This technology is based on a distributed approach, on top of which H-Chain will build a federated schema. As blockchain based system, each of the nodes participating in H-Chain based ecosystems (namely, each of the data silo connected to the H-Chain system) collaborate to protect shared data and lead to the creation of a trusted history of transactions between organization, so that every citizen's transaction is documented and nobody can go back and alter past data (unless using a versioning system). In addition, to this, H-Chain enables an authorised party (i.e. Local or national health authority) to certificate the transactions, safely and publicly since they are traceable and documented.

H-Chain Core architecture is defined by its principles:

Validation is fundamental to establish trust, which is essential to create a secure environment and one of the main features required to create a EHRs exchange ecosystem. Before any exchange of data, it is necessary to validate the identity of users and entities that take part in the communication. This process is known as Proof-of-identity, which certifies the e-ID associated to an entity or user, actually represents their identity. There will be a trust framework mechanism where the proof is built by providing a digital signature of the e-ID by trusted certification authority; revocations of this trust are also part of the system. Thus through their process any member of H-Chain will be assured the identity is equivalent to the associated e-ID. When users, or entities, interact with the system, the system needs to be assured the e-ID is valid, and being correctly used. Proof-of-Authentication assures the authenticity of the identity. It is implemented by launching a challenge only the true holder of the e-ID can answer it. The technology will be based on Semantic Smart Contracts (SSC), these needs to be validated too. When a user, or entity, issues an SSC they will need to digitally sign it, so that anyone can validate based on the proof-of-identity that the user, or entity, agrees with the statements and procedures in the SSC.

Access to private data In order to access private data the technology will require both a Proof-of-Authentication and Proof-of-Access. This means that the user, or entity, who wants to access data needs to prove their identity and that there exists a set of SSC issued but the legitimate data owner(s) which explicitly allow the access, as well as the type of access. Both need to be validated by the storage backend where the requested data is hosted, then this access will be logged in an audit log (containing time, e-ID reference, and SSCs references), for monitoring and evaluating security. This log will also allow the owner of the information to know who is accessing their information. After all validations, and audit are performed, the access is granted to the requester.

safety and non-repudiation Blockchain offers these two important properties. Blockchain stores information, by grouping it, so that it can be distributed amongst all nodes, through the replication of the data in distant nodes. At

the same time through this grouping, a relation between these is created using cryptographic algorithms; this means that any change on anyone group will affect related blocks recursively, which makes it increasingly difficult to enact any change on a particular group without any of the participant nodes in the network detecting it. This inalterableness of the data means that the property of non-repudiation (any user, or entity, cannot deny data) is assured.

Distribute Storage Because Blockchain is an invariant storage there needs to be a solution which allows deleting data, because of the need to apply the erasure right specified by GDPR. H-Chain needs to allow not only erasure right but also all other rights; therefore, storage will be protected through the mechanisms previously described (safety, non-repudiation, access control), by storing only un-erasable data like validation certificates, SSCs and audit log (among other) in the blockchain. While the actual infrastructure for storing private data can be either traditional databases, cloud provision or any other storage adapter allowing to integrate with existing systems, while protecting the data through Proof-of-Authentication and Proof-of-Access.

Because individuals may not have access to storage infrastructure H-Chain also offers a distributed disk (as a local storage adapter), enabling them to add data without support of any specific infrastructure. The distributed disk will only hold encrypted information (using SSCs), which is only accessible under the storage terms previously mentioned, additionally the data will be safe from accidental destruction as it will be replicated throughout different nodes.

Data Harmonization through a semantic based interoperability layer, i.e. the interpretation of the meaning of the data, oriented to protect data quality and safety, built on clinical domain knowledge and standards, i.e. FHIR⁴ among others. In H-Chain vision semantic interoperability will be supported by hierarchical structure, based on the semantic web principle and ontologies. On the upper level the semantic definition of data, defined on a structural interoperability, i.e. definition of data formats that ensures that data can be interpretable at the data field, while received data is preserved, at the same time, defined on a foundational interoperability, i.e. healthcare systems are able to exchange data without requiring the ability for the reviewer party to (re)interpret the data.

These principles allow the different actors within the H-Chain community to interact securely and smoothly. Citizens for example cannot only be assured their privacy rights are being enforced, they can collect all their data from different sources, know where it is being used, and benefit from enhanced trust on institutions and procedures. This trust is based on the enforcement Controller Authorities exercise by providing the trust networks required to validate contracts, individuals and entities. Hospitals and health networks can manage the data they already have through

⁴<https://www.org.hl7/fhir/overview.html>

Figure 5.3: H-Chain Architecture vision

existing systems while adding a layer of security for accessing said data granting patients the legitimacy over their data. Health professionals will have access to more data about their patients (as with the appropriate permissions of the patient they can access data sources outside their current silo), as well as enhanced tools to process and generate said data. Finally, research institutions will be able to close the loop between health care and health research, by requesting actual data from the whole population, which can be shared with them following the strict legislation for said process.

Blockchain and semantic technologies will not only enable unbeatable scalability and extensibility to other domains, it provides a secure consensus platform where semantic smart contracts (SSCs) are the basis for enforcing privacy rights. A first vision of the H-Chain architecture (see figure 5.3) would serve as initial work to develop the five essential principles stated. The core functionalities of H-Chain are not domain specific, allowing the core to be reused in other privacy and security sensitive domains.

Distributed Storage H-Chain will use peer to peer communications giving it its federative nature. Blockchain infrastructure already provides distributed storage, yet this storage is constrained to a ledger, where entries are immutable. Healthcare is one of the most suitable segments for Blockchain application, as privacy, security, rapidity, transparency and interoperability characterize this technology. Indeed, it can create a reliable and secure environment for data exchange, where the patient take full control of his/her information, sensitive information are secured, every modification and access is traced and registered immutably. Finally, it will be possible to build a layer that do not change ev-

everything that already exist, ensuring interoperability and smart contracts for a new paradigm in Healthcare.

Parallel to this storage, H-Chain will use a pluggable, distributed storage solutions to provide a mutable distributed storage. These may be the traditional Databases which the data is already hosted on, or other type of storage provision like cloud storage, or distributed disk technologies. Allowing for users CRUD (Create, Read, Update, and Delete) operations on their data; as well as sharing it with other entities. The operations will be governed by the semantic smart contracts (SSCs), concretely Data Transaction Descriptors in the Private Information Transfer ledger (see below). Operations need to be verified by presenting a Data transaction Descriptor and by presenting Proof-of-Authentication, generally a challenge response by the peer community. All operations on a specific entry will be logged, thus at any time the data owner can know who, accessed what data, when, and with which authorisation.

Data has two important properties: temporal locality and geographical locality. The first property will help less resourceful nodes to delete older data and keep the latest accessed data, which has higher probability of accessing. The second will allow federation members to keep their data locally, as the data has more probability to be accessed from neighbouring nodes. Yet older and remote data can still be recovered, even though it will require more time. Both properties apply to the ledger blocks (which contain geographically located entries as part of the federation of the blockchain), as well as to the mutable, distributed storage (a.k.a distributed disk).

Because data will be replicated amongst many nodes, the system will be highly fault tolerant, with varying degree of replication of data, different robustness can be achieved, this robustness can be adjusted depending of the frequency of access to said data. The distributed storage, along with the rest of the H-Chain core will be able to run configure-less, redundant nodes, which can automatically overtake one another when the need arises. For example a Hospital can run two or more H-Chain instances, each working independently, in normal configuration they will be load balanced (using COTS load balancers), when one server is turned down for maintenance (or fails), the rest can take automatically the load.

Semantic Smart Contracts (SSC) Smart Contracts are the building blocks of Blockchain technologies (see section 2.3.4.1). H-Chain goes a step further in the use of Smart Contract, adding semantic value to this technology . In this way, a Semantic Smart Contract will perform as an extension of Smart Contracts, where the language used to represent it has semantic capabilities. This enables it to perform semantic inference not only extending the capabilities, while simplifying the implementation, but also allowing complex operations such as matchmaking, search and reasoning. Most H-Chain core services will rely on a blockchain architecture to offer invariant secure storage of records.

These records may contain Public Key Infrastructure statements, Public information in semantic form, Token transactions, or SSC; the type of records may be extended to accommodate new features . The blockchain allows for a decentralised management of big sets of sensitive data . In the health domain this is very interesting as there is no central authority to rely on to offer data management, which might otherwise imply change of jurisdiction, different procedures, and of course the risk of exposing sensitive data during transit or due to a central breach . Decentralised management is possible thanks to the cooperation of all nodes in the H-Chain network, who work to concatenate blocks of records by solving complex functions . The valid blocks are those, which are linked, in the longest chain of blocks; as the longer, the chain the harder it is to change any of the contents within . Federative Blockchain scales local operations to the global in a progressive way, adding all local transactions to the same or linked blocks, which allows local operations to be fast; but transactions after some time are reflected globally.

Distributed Private Key Infrastructure (DPKI) In order to achieve end to end encryption and verification H-Chain will implement asymmetric encryption systems based on public and private key pairs. The e-ID is henceforth considered one of these key pairs given to each user. Even though there are many existing standards for e-IDs (i.e. Open eID), most would either already include key pairs, or these can be assigned to them. One of the biggest problems when implementing an asymmetric encryption system is key distribution, concretely solving the problem of given a destination, how to get its public key. This is where a Public Key Infrastructure (PKI) is needed. In H-Chain we propose using blockchain to provide said infrastructure in a distributed way, hence DPKI. The ledger will contain PKI statements, concretely Public Key Announcements, Certifications, and Public Key Revocations, all in the form of special SSCs. As an additional feature, Public Keys could be signed by other entities. This model enables for example the certification that a particular user is employed by the signing entity, or if a smart contract is introduced in the signature metadata, certificate that a user is qualified to perform specific actions (e.g. a digital form of diplomas).

H-Chain Token cryptocurrency H-Chain will have the capability to define a cryptocurrency used to transfer wealth for given services and information. This will be managed using the most advanced cryptocurrency technologies. Due to the ethical implications of using highly volatile currency for health management services, the idea of using cryptocurrency in H-Chain is to incentivise health habits in citizens, incentivise the donation of data to research organizations, settle international, health insurer or health services issues, and finally compensate entities for the maintenance of the blockchain infrastructure (i.e. mining).

Public Semantic Information A public information repository is supported by

H-Chain, this repository will be integrated in the blockchain's ledger. This facilitates distribution, as well as auditing and validating public data. This repository has the capability of publishing semantic statements, according to arbitrary ontologies. This capability will make semantic information available to the public, so that any H-Chain node can perform semantic queries locally, and hence privately (as no centralized entity would receive the queries). To make the information available for the public, it must not be encrypted, however to ensure the data can be trusted it must always be signed by the data provider, who in turn must be authorised by a trusted authority in the system.

Private Information Transfer When Private information needs to be transferred between entities (e.g., between patient and physician), there needs to be a secure methodology to ensure the communication is secure end to end. We propose a similar protocol to Pretty Good Privacy (PGP), where the information is encrypted using a symmetric session key, which is then encrypted with an asymmetrical public key (gathered from the DKPI); except the cyphered content and cyphered key are not bundled together. On one side the cyphered content is stored in some accessible storage accessible by all recipients (i.e. distributed mutable storage; see Distributed storage); on the other the cyphered session keys are used as part of the data transaction descriptor in the ledger, a special form of SSC. This is essentially tokenizing health information, where the token is the SSC which is used to access the actual data. Because the data transaction descriptor is stored in the immutable ledger, it means that there is a trace of the access rights given to the data, providing strong auditing capabilities to the system. Data transaction descriptors can also be revoked through another SSC which makes reference to it.

Private Space Builder The private space builder is the intermediate layer between the core H-Chain services and the application layer. It will deal with CRUD operations in the distributed storage; validate and authenticate using a SSC the DPKI; Manage the data transactions descriptor hierarchies associated to the user; and present public and private accessible information in a unified usable, unencrypted form to local authorised applications. This is a novel Member model, allowing components to interact with the SDS in a controlled and local way. When local queries are performed private data does not leave this private space where it resides. This way an additional layer of privacy is added, sensitive data is not used in service requests, thus not even trusted entities have the capability to break privacy using secondary indicators (e.g. time and address of the requests). This concept is very similar to the Web 3.0 vision[19], in fact H-chain core features fully implement Web 3.0.

On top of the H-Chain Core components, standard applications and tools can be provided. These would be more tailored solutions for the Health domain, and particularly the management and use of Electronic Health Record (EHR). Yet in

some instances these solutions can be generic enough within the target domain to be used as intermediary services to build larger applications.

Professional Assistance Tools (PATs) On top of the basic interfacing, specialised tools can be addressed for specific professional needs. Such as managing patients, schedules (or queueing EHR recovery process), viewing EHR, writing reports, producing test results, or using Decision Support Systems amongst other possibilities.

EHR Management Tools Health care entities will need tools to manage their EHR. The typical, day-to-day usage of the EHR will be covered by these tools, like searching, assigning professionals (i.e. resharing the patients EHR with a particular professional, thanks to the extended permissions provided by the data transaction descriptor SSC), and editing. Other technical operations can be securely managed by these tools such as archiving or bridging between the H-Chain model and the local entity's system of management.

SSC Template Tools are pre-signed SSC which users, through client tools, can easily instantiate into concrete SSCs. As everything else in the ledger, they may also be certified by a third party certification authority ensuring users the ending SSCs can be trusted. These templates are meant for when users need to easily sign-in in to something, from hiring health insurance, buying health packages, requesting specific services, acquiring apps (see pHAs below), or donating data to researchers (information requests).

Public Information Applications (PIA) Entities publishing Open Access Data to the public information repository can create, adapt or use tools to facilitate this process. The expected information will be initially information requests (a special kind of SSC), Health care services information (such as clinics, their capabilities, location, and timetables), Open Access Data (such as weather predictions, air quality measurements, etc.), pHAs code (see below). A set of tools will be provided for Certification, certification management and validation. This are tools meant for entities, particularly a Controller Authority, which validates third party information, requests, contracts, and even software; since all the information in the public space must be certified for ensuring trust.

Personal Health Agents (pHAs) The patient requires a minimal set of tools to manage their EHR as well as consent. These clients will be designed to run either on mobile environments (mobile apps) or web clients (web apps). Corporations will be able to personalize their clients for branding purposes.

On top of the clients, and their basic management, pHealth Agents (pHAs) will be able run on these clients as part of the client specification. Similarly to PATs, pHAs are programs which run locally, and access private and public information on demand (through the private space builder); the difference is that pHAs executables are held as another piece of public data in the public information repository, and as such they may be certified by a certification

authority adding trust layers ensuring the executable code does not have malicious intentions. These agents can be semantically matched against the private space, i.e. pHAs may require specific conditions from the user's data in order to be executed. This matchmaking is done locally, which enforces privacy.

pHAs are micro services that address small problems for example reminding the patient to take their medication. The pHAs services will provide the necessary elements to deploy user application such as Medication management, Medical event scheduler and reminder, General health notifications (UV radiation level predictions, Pollen count predictions, Air pollution measurements), Exercise monitoring & motivation, Nutritional management, etc. The concept of digital prescriptions, software which is imposed on patients to improve their health and habits, can be easily implemented through H-Chain.

Because pHAs may be active and matched all the time, users should have control over which pHAs to subscribe to (activate). This may be managed using traditional "installation model", where the user actively selects each agent to be installed individually from any potential pHA marketplace (following the federative nature of H-Chain). Additionally other discovery models may be implemented, for example a pHA semantic filter is enabled and configured by the patient allowing him/her to automatically subscribe to the newest developments. A default filter definition may be proposed by the client implementation, or by the health provider, or the physician.

5.1.3.2. DIRECTIVE COMPLIANCE

H-Chain ambition: GDPR compliance Mechanisms. Table 5.5 summarises the components that lead the necessary mechanisms to fill with above elemental mechanisms to provide GDPR compliance platform.

5.2. SHOWCASE EVALUATION

The showcases tool set is specifically designed to extract value from open platforms. And not just general value, through the different showcases, which offer a unique perspective on what the platform does, specific value can be examined. This process allows us to look deeper into the benefits and impacts of the platform, as a whole, as well as in its bisected form.

In this section we will have a look at how showcases have been implemented in ReAAL; the potential value that, through open platforms, different stakeholders can achieve; and finally we will analyse what the showcases mean for every stakeholder.

The list of 14 AAL platform showcases and their full definition can be found in section 4.4. However some showcases in the list were not evaluated, or at least not fully:

Advanced User Interaction None of the ReAAL pilots, with the exception of TEA, had real interest in universAAL's UI framework, integral to the show-

Basic Principle	H-Chain method of enforcement
Eraseure	The personal information is never stored, as only Public Key information and Semantic contracts are stored in blockchain; once a public key is revoked all associated SSCs are revoked too. The data is always encrypted, when no valid SSC points to them, the object is removed from the distributed disk (for information stored in cloud infrastructure, specific mechanisms will be implemented).
Restriction of processing	H-Chain data can only be accessed with a valid SSC or by the legitimate owner of the data. The SSC will specify the use for the data, and the system will validate the use with the certified application capabilities. Use for a specific use will only be allowed when 1) the solicitor (health provider) presents a valid SSC, where the owner (the user) allows the solicitor access and use of the data and 2) the application presenting the request on behalf of the solicitor can be validated to perform the described use, and there is a Proof-of-Authentication from the solicitor.
Data portability	One of H-Chain applications can export (and import) data to (and from) many standards.
Rectification	The data owner has full CRUD permissions over the data, so the user themselves can correct their records.
Right to be informed	Every SSC will be displayed to the user, in human readable form; this will establish a 1 to 1, un-alterable relation between the consent the user signs and what the platform allows the solicitor to do.
Access	As legitimate owner of the data, through the private space builder, the user will be able to access all the information, not just the data which providers have.
Data minimization	SSC templates (for signing into services, and Information requests), will be validated and certified by controller authorities; the clients will automatically check for this validation, and ignore invalid SSCs. Without a valid SSC template, providers cannot get users to sign a valid SSC, which forbids them to gather more information than stipulated by the controller authority
Security and Integrity	The SSC for data access will hold meta information to ensure the data integrity. When the data is stored in the distributed disk, the security of the data is managed by the automatic replication of the encrypted data in different nodes.

Table 5.5: Summary of how H-Chain will implement GDPR

case. None of the pilots actually integrated universAAL's UI framework, thus this showcase could only be, very poorly, evaluated for its potential value.

Enhanced Market communication and distribution This showcase is special in the sense not a single application implements it. It describes the value of digital stores that sell applications and services, as well as some desirable characteristics arising from this concept. The intention of the evaluation team was to evaluate this concept, extract the value for such a feature. Since universAAL IoT had a tool implementing the digital store (called uStore), but was never deployed, and digital stores are integral to ITC markets the results of this evaluation would determine if additional resources were to be spent on an AAL service digital store (concretely uStore).

Security and privacy This showcase is extremely important, yet not many pilots actually implement. Additionally the functionality it describes was not fully implemented by universAAL IoT, and the sub set that was implemented was not used by pilots. Thus this value from unviersAAL IoT could not be demonstrated, let alone evaluated.

Administration This another case of an important feature detected, that universAAL IoT does not yet fully implement. Platform developers released statements regarding how to achieve this showcase with universAAL IoT in the later releases of the platform.

5.2.1. DEMONSTRATION OF SHOWCASES

In order to profit from the value of the platform through the showcases, it is necessary to experience the showcases. Applications or systems in general are said to implement a showcase if one of their use cases fall in the category of the showcase. In other words, the application can demonstrate showcase features through its operation.

5.2.1.1. SCRIPTS

To make the definition of showcase demonstration easier there were some scripts defined (see section 4.4). Scripts were designed to pace the system through a series of steps, which will force the showcase indicators to arise.

The next step is to initially map the applications deployed to the list of showcases. The aim of this mapping is to assure that most showcases can indeed be demonstrated and to plan the resources needed for this demonstration effort. After being provided with the showcase definitions and scripts, system providers were asked to evaluate which of the scripts, they felt their application is capable of passing. The results are composed as a percentage of the potential passed tests over the total tests, grouped by showcase. The presentation of the results is done through a diagram with hexagonal-honeycomb configuration where each cell represents a

showcase, the results are encoded in colour scale: the darker a showcase's cell, the closest to 100%, which will become the standard graphical representation for this kind of results.

Figure 5.4a shows the results from the self-pre-evaluation, which are detailed per pilot in table 5.7. This procedure was used to plan the evaluation tasks, as it will reveal the distribution of potential implemented showcases. As we can see there are 7 showcases for which there is good interest (represented with white coloured text), four of which are especially interesting: plug and play, service integration, evolution and service transferability.

Figure 5.4b, on the other hand, shows the results of the evaluation of pilots performed by a technical expert, following the analytical scripts, also results shown per pilot in table 5.7. It shows 6 showcases that could potentially be present, with one strong weight on resource sharing. The 6 showcases are not a sub set of the 7 in the self-pre-evaluation, evolution and plug and play are dropped, and advanced distribution is promoted in the evaluation priorities.

The explanation for this is simple, both evolution and plug and play are very difficult showcases to deduce from documentation and attached material; it is best certified by actually performing the operations (either technical or operational scripts). As for Advanced distribution, it was already a potential candidate in self-pre-evaluation; in this case the expert might have detected more instances where the showcase arise; also because the time between the evaluations, applications changed, and some have added certain degree of distribution to their requisites.

Figure 5.4c, shows the percentage of actual script test success rate per showcase. These results show that the actual implemented showcases are Service Integration, Resource Sharing, Plug and Play, Service Transferability and evolution; Which align with both analysis. There are however to mayor discrepancies, Enhanced Market distribution could be demonstrated beyond what both pilots and experts predicted; And integration with legacy systems could not be demonstrated as strongly as predicted.

(a) Global Self Pre-evaluation results

(b) Global expert analysis results

(c) Global Demonstrated score results

(d) Global distribution of data collected

Figure 5.4: Different results of overall showcase analysis, all pilot sites aggregated. Darker cells correspond to higher compliance, white text indicates compliance is over 50% .

5.2.1.2. VIDEOS

The idea of creating video material emerged during the planning of the tasks for showcase evaluation. Initially this material was meant for the execution of focus groups around showcases (see section 4.2.4); but the intention was to also use the videos as demonstration material of the showcases being executed at each pilot. In fact pilots were asked to record the execution of either Technical or Operational

Showcase	BSA	EIC-IL	IBR	IMA	NCSR	ODE	PRC	PGL	RNT	SCUPS	SL	TEA	WQZ
Resource sharing	SX	S	X	S	S		SX	X	SX			S	SX
Plug and play	S	S	SX	S	S			X		S	S	S	SX
Evolution	SX	S	S	S	S	S	SX	S	S	S		S	S
Service integration	SX	S	SX	S	S		SX	X	S	S	S	S	SX
Advanced distribution	SX		X		S		SX		S				X
Advanced user interaction			S										
Personalized content	X					S	X						
Ambient intelligence			SX					X					SX
Scalability	SX	S		S		SX	SX	S	SX			S	S
Service transferability	SX	S	S			S	SX	S	S		S	SX	S
Integration with legacy systems	S	S	X			X	S	S	S	S	S	SX	SX
Security & privacy	SX	S					SX	X					S
Administration		S		S						S			S
Enhanced market communication and distribution value		S											

Table 5.7: Legend: S indicates the showcase is identified in the Self-Pre-Evaluation for the pilot, X indicates the expert sees evidence of the showcase in the pilot; both only if the score is above 50%) (Associated pilots are not evaluated by expert)

scripts, preferably including users.

In the end the videos interests shifted towards dissemination and marketing objectives. This conflicted with the demonstration intention, thus the confidence of the evaluation team on videos as demonstration material is low.

5.2.1.3. SPECIFIC DEMONSTRATION FOR SERVICE TRANSFERABILITY

In the Description of Work, ReAAL explicitly stated that pilots would import applications from other pilots. The purpose of this experiment would be to demonstrate the platform benefits as interoperability and adaptability enabler. This represents specific benefits of the platform, since showcases are exactly the categorization of specific value; this means that the intention behind this idea could be matched to particular showcases, initially those showcases being Service Integration and Evolution describe. Yet when defining the showcases the transference of services between

deployment sites was identified in itself as unique value from the platform.

Application	Original pilot site	Importing pilot site	Other SC Demonstrated			
			Evolution	Plug and Play	Resource Sharing	Service Integration
Netmedical weight	RNT	ODE			x	
SmartLiving System	(DE)	RNT	x			
Rehabilitation Portal	ODE	BSA	x			x
Health gateway module (Nomhad Chronic)	BSA (TSB)	TEA		x		
LynxHH	IBR	TEA				x
Netmedical	RNT	IBR		x		
HWO	BSA (TSB)	IBR				x
MiBida Video Communication	RNT	SL	x			
Nomhad Chronic	BSA (TSB)	WQZ	x			

Table 5.8: Table of imported applications. Extended with showcases demonstrated through the imported application deployment.

In most cases the access to a new application enables the demonstration of other showcases, as shown by Table 5.8.

Another interesting case of Service Transferability is the case of Perche (PRC) pilot. This pilot operates the same applications developed for BSA. In a sense the whole pilot emerged from the possibility of transferring the technology to the site. This demonstrates the feasibility of importing applications at different stages of development of a service provider. Perche Pilot has successfully accomplished over 170 users (between both applications).

5.2.1.4. REAAL PROJECT DEMONSTRATOR

ReAAL had a booth in the European Summit on Innovation for Active and Health Ageing⁵. For this booth a Special demonstrator was developed, this demonstrator used modules from different applications throughout the project. Thanks to

⁵ http://ec.europa.eu/research/innovation-union/index_en.cfm?section=active-healthy-ageing&pg=2015-summit-exhibition-village2&ReAAL

the universAAL IoT platform these modules were already interoperable, and with small adaptations new scenarios were implemented.

Figure 5.5: General participation of pilots and modules in the demonstration scenarios

The first scenario was defined as follows: When the user wakes up, the action is recognised by activity monitoring application. After some reasoning (for example: it is Sunday and the user's profile has the appropriate settings) by the intelligent agent, the system triggers the concierge service to get breakfast at home.

For this scenario modules used were Ibermática activity monitoring, WQZ Intelligent agent, and SL Concierge service. Together this demonstrator successfully, and explicitly demonstrated Service Integration, Content Personalization, Plug and Play, and Resource Sharing; Although other showcases might have participated (e.g. Service Transferability as the demo was running on country none of the modules was explicitly designed for).

The second scenario, more related to the health domain was defined as follows: Abnormal behaviour and abnormal vital signs are detected through the appropriate sensors (mainly presence sensors and health sensors), an intelligent agent reasons this information and, if necessary, alerts the informal caregiver. The informal carer receives information about this abnormality as well as the history of the user, in a way that the informal caregiver can make a decision of action.

Modules participating in this scenario were provided by Puglia, Ibermática, WQZ and TEA pilots. Explicitly demonstrated showcases are Service Integration, Plug and Play and Resource Sharing. Other showcases might have been present, for example Evolution, as this scenario can be seen as a new service added to the pre-existing deployment show by the previous scenario.

5.2.2. VALUE OF SHOWCASES

In this section we will show the results of our research about the value of open platforms in general, as well as in universAAL IoT open platform in particular. For this we have collected data from different sources, namely from:

- Focus groups organized at each pilot, with local stakeholders. A total of 59 attendants, from different backgrounds and roles, across 11 sessions (each hosted by a different pilot), evaluating 10 out of 14 showcases.
- Data gathered from specific templates for each showcase, which pilots were asked to complete. 11 pilots provided different data for all 14 showcases.

For focus group data, the questionnaires have been processed to show the understanding of, value attributed to each stakeholder, and impact both positive and negative as defined by the OPEA framework. The data presented here has been averaged, as not every showcase had the same number of attendants evaluating it. Focus group data also categorizes the attendees through 3 categories: Government, Service Provider, and Technology Provider, the results will also show these different perspectives.

The templates for capturing showcase data gathered from pilots, have been specially tailored for each showcase. They aim at collecting recurring tasks, background information, and costs (and sometimes income) related to each of the showcases. In some cases data input is not consistent, nor sufficient to perform accurate analysis of the available data.

5.2.2.1. OVERALL VALUE

(a) Radar of positive impact for all showcases (b) Radar of negative impact for all showcases

Figure 5.6: positive and negative impact for all showcases

Figure 5.6a shows the distribution of positive impact per showcase. The dimension most valued in general is User Experience, in some showcases it is exchanged for Economic impact.

The most valued showcase is Evolution, followed by Ambient Intelligence, Service Transferability, Scalability and Personalized Content.

Figure 5.6b shows the negative impact for each showcase and by dimension. The dimensions with most negative impact are Technical and Ethics.

The showcases with most negative impact are Personalized Content and Ambient Intelligence, which were also part of the top most positively impacted showcases; this means that participants noticed as many advantages as disadvantages (positive winning by little margin). This result may be due to various reasons:

- the moderator might have been specially successful in engaging participants
- The videos might have been more keen on showing both sides
- These showcases are particularly complex for participants

The later might be the most probable reason.

Figure 5.7: Radars for Net impact (all positive minus all negative impact values)

Figure 5.7a shows the net (positive minus negative) impact value for each showcase, categorized by dimension. It shows that when compared the most significant negative impact is Ethical, whereas technological compensates negative with positive impact.

It can also be appreciated how the most valued showcases are Evolution, Services Integration, Scalability and Plug & Play. The reasons for each change, for most the mayor contributor is User Experience, while for Scalability the most significant contributor is economic factors.

It is very interesting to analyse the results by different participant roles. For example Figure 5.7b shows which showcases are more interesting for each role, Government officials like Service Transferability and Ambient Intelligence, Technology Providers like Evolution, and Service providers are very interested in Plug & play and Services Integration. We can also see the polarization on each showcase: when there is very positive view from one of the roles, one, or both, of the other roles are not as excited. The exception is Scalability, where the three roles agree.

It is also noteworthy how the complex showcases (Ambient Intelligence and Personalized content) is seen by role, the most adamant role with these two showcases are Technological Providers.

Figure 5.8: Self-assessed understanding for each showcase by different participant roles

Another indicator use for this analysis is the self-assessed understating of the videos for each role. Figure 5.8 shows how the different roles felt they understood each of the showcases, all average out as a good understanding for all showcases. Although as shown the least consistent, and least comprehension where the policy makers. Service Integration is the best understood showcase, and Resource Sharing and personalized content are the least understood showcase. This might help to explain the lower impact values for these last two showcases.

5.2.2.2. ADMINISTRATION

The value data for this showcase is provided by the templates filled by pilots, as it has not been discussed in any focus group.

From this data the first striking fact is that it has been the most reported showcase of all (41 entries, where the second runner had only 23). This denotes the high impact and need for a streamlined administration process for deployments.

Figure 5.9: Different results of data collected for the Administration showcase.

Almost every entry (for each administration task) states the task could not be simplified.

This data shows how administration is undertaken by pilots, as well as how much improvement is needed. For example most of the administration tasks are applied on each and every one of the deployments one at a time, when ideally administration tasks should be performed once and automatically replicated on each deployment. Again ideally remote administration is preferable, but only 41% are done this way. Mean time for administration tasks is over 6 hours, which is too long. All this is translated in very high administration task costs.

Though, one explanation for this data is the recurrence, or frequency of the tasks. According to pilots administration tasks are one time actions, and not repeated. Probably because these tasks are related to installation and configuration of complex set ups, that need to be streamlined. The platform should provide means to enable faster optimization processes even for piloting applications. Pilots have not demonstrated this showcase, and universAAL has not provided a concrete solution for this showcase; it would have been very interesting to compare this data with at least one application that truly implements the showcase.

5.2.2.3. ADVANCED DISTRIBUTION

(a) Positive impact as- (b) Negative impact
sessed by the different assessed by the differ-
stakeholders ent stakeholders

(c) Value Matrix assessed by the
different participants for different
stakeholders

Figure 5.10: Results from the focus groups evaluating Advanced Distribution show-
case

According to Focus Group participants this showcase has:

- Better value for Assisted persons and Technology providers, especially seen by Government
- Better economic impact
- Technical difficulties may arise, specially reported by Technology providers

The data gathered shows that the pilots who reported most based their advanced distribution on mobile-cloud models.

5.2.2.4. ADVANCED USER INTERACTION

The value data for this showcase is provided by the templates filled by pilots, as it has not been discussed in any focus group.

Figure 5.11: Distribution of types of interaction mechanisms used in the pilots.

All responses (N=11) deem the interaction to be adaptable to end user needs, as well as the interaction being mostly through portable devices (smartphone and or tablet). Only 2 pilots considered PC or touch screen as well.

The User Interaction feature has not been prioritised by pilots despite the fact they consider this showcase important. That means that a better promotion of the UI bus of universAAL IoT shall be made in the future; or at least that the UI system should be facilitated and made more accessible to developers, without constraining displaying choices.

5.2.2.5. AMBIENT INTELLIGENCE

(a) Positive impact as- (b) Negative impact
sessed by the different assessed by the differ-
stakeholders ent stakeholders

(c) Value Matrix assessed by the
different participants for different
stakeholders

Figure 5.12: Results from the focus groups evaluating Ambient Intelligence showcase

According to Focus Group participants this showcase has:

- Total positive and negative impact are very balanced, showing the complexity of evaluating this showcase
- Better value for Assisted persons
- Better User Experience impact
- Ethical complications are to be expected
- Organizational and technical problems can be an issue too.

It is noteworthy in this case the special negativity expressed by Technology Providers. Form the additional comments, their main concern is the scalability of the solution.

According to the data collected there are around 2612 sensors (typically environmental sensors: CO_2 , smoke, humidity, temperature, and flood detectors; with some presence sensors), and 1576 actuators (typically light actuators) installed in deployments across ReAAL pilots.

Involvement of the user highly depends on the piloting concept.

All reported services include reasoner toolkit, typically universAAL's Drools reasoner.

Costs for only 2 services have been reported: 500 €/deployment and 5.750 €/deployment. The high variance may be due to the small dataset, or to differences in accounting for costs. Not much information is provided to further research.

5.2.2.6. ENHANCED MARKET COMMUNICATION AND DISTRIBUTION

The value data for this showcase is provided by the templates filled by pilots, as it has not been discussed in any focus group.

According to the gathered data, only 3 services, out of possible 14, are using standard digital markets (Google play) to distribute their software. The download ranges (provided by the digital store) per service are 1, 10 to 50 and 100 to 500.

All those applications that are distributed through standard digital markets, require exactly 2 personnel members to maintain the distribution method.

Its relation to Service Transferability, encourages participants, both in focus groups and through questionnaires to give some insights to the future market in the AAL domain.

During the Pilot Experience focus group, pilot representatives a common market place was discussed; not forced by the moderator. From this discussion we learn:

- The number of 30+ applications of ReAAL could be insufficient to build a substantial Market place.
- Some Pilots/SMEs are not ready for sharing their solution in terms of deployment or marketing.
- However a common market place will definitely create the critical mass needed. Through the critical mass obstacles will be identified and removed.
- User demography must be taken into account, a digital store should be tailored to the AAL users.
- Government must support, laws must evolve and adapt to the emerging market and new distribution methods.
- B2B communication must be stronger in the AAL market and the market place should reflect that too.

5.2.2.7. EVOLUTION

(a) Positive impact as- (b) Negative impact
 sensed by the different assessed by the differ-
 stakeholders ent stakeholders

(c) Value Matrix assessed by the
 different participants for different
 stakeholders

Figure 5.13: Results from the focus groups evaluating Evolution showcase

The different participant roles highly disagree in this showcase:

The **Technology Providers** are very excited with this showcase (exceptional behaviour from this role). They see best value for their own role, all agree of the technological positive impact and all agree there are no economic, organizational, technical, nor ethical negative impacts.

All **Government** officials agree this showcase has positive impact on user experience, but contrary to Tech Provider they see positive impact on Technical criteria. Most of them express possible economic issues, and some are concern on organisational and ethical issues. They value equally this showcase for all stakeholders.

Service providers agree (as do government) this showcase has positive impact on user experience, their main concern is organizational, and they value highest value for assisted person.

As for the data gathered from the pilots, show that the most popular evolution strategy is to add a new module. The cost ratio of evolution (i.e. the investment to introduce an evolution in deployments) is between 4% and 7% of the original investment.

5.2.2.8. INTEGRATION WITH LEGACY SYSTEMS

(a) Positive impact as- (b) Negative impact
sessed by the different assessed by the differ-
stakeholders ent stakeholders

(c) Value Matrix assessed by the
different participants for different
stakeholders

Figure 5.14: Results from the focus groups evaluating Integration with Legacy Systems showcase

According to Focus Group participants this showcase has:

- Overall High value for Technology Provider stakeholder
 - Although Government and Service Providers think the best value is for Service Providers.
- Overall better organizational results
 - For Technology Provider there are technical advantages
 - For Service Provider the best value is user experience
- Technological risks

“It is not just a technological issue, i.e. a user is familiar with existing system and there are big issue related to change of technology. E.g. changing a TV set for a new one with a different remote control is a big issue. In short term this is a big drawback in my opinion. In long term an open platform is a big benefit.” Technology Provider focus group participant.

Data gathered showed that reported legacy systems are on average 4.5 years old. In general either the carer or the assisted person is in one way or another in direct contact with the legacy system; although there are 3 instances (out of 9) that neither are in contact with the legacy system.

5.2.2.9. PERSONALIZED CONTENT

(a) Positive impact assessed by the different stakeholders (b) Negative impact assessed by the different stakeholders

(c) Value Matrix assessed by the different participants for different stakeholders

Figure 5.15: Results from the focus groups evaluating Personalized Content showcase

According to focus group participants this showcase has:

- Very balanced positive and negative impacts, showing high complexity for evaluating the value

- Extremely high value for Assisted Persons.
- Extremely positive impact on user experience

All **Technology Providers** think there are technical and ethical implications.

For **Service Providers** There are good and bad organizational aspects; this stems from the fact that half expressed organisational as good and the other half voting against it. They account for good technological impacts, and consider there might be other aspects to be concerned.

“What about the quality of service? It is not dangerous to lose the human contact?” Technology Provider.

“I have a lot of end users with very different conditions from the point of view of the service that has to be provided, to have a personalized information without being processed or adapted by my Company is very useful. Gives us efficiency and is cost saving.” Service Provider.

No significant data could be extracted from the other data delivered by the pilots.

5.2.2.10. PLUG AND PLAY

(a) Positive impact as- (b) Negative impact
sessed by the different assessed by the differ-
stakeholders ent stakeholders

(c) Value Matrix assessed by the
different participants for different
stakeholders

Figure 5.16: Results from the focus groups evaluating Plug and Play showcase

According to Focus Group participants this showcase has:

- Better impact on user experience and economics
- Extremely low negative impact

The value is high across the stakeholders, but it is interesting the distribution of it according to the different roles. **Policy Makers** and **Technology Providers** believe the highest value is for the Technical Provider. Whereas **Service Providers** think the highest value is for the other stakeholders. **Service Providers** net value this showcase higher than the rest of the showcases.

“Platform has same protocol to connect, same system. Ultimately lower costs and effort thinking about what I have and what are the requirements. That’s what is being offered.” Technology Provider.

“Use of p&p technology allows me to save money.” Service Provider.

The other data gathered from pilots shows the mean cost of devices represented in table 5.9.

Category of device	Mean cost
Health	152,80 €
Environmental	85,00 €
Panic Bottom	51,25 €
Domotic	33,75 €
Average	80,50 €

Table 5.9: Mean costs of devices by Category

Collected data also shows the discounts of buying at bulk can be as high as 80%; whereas, on mean, pilots expect a discount of 16% from bulk purchases.

Data shows typical installation time is 10 minutes (not accounting for the diversity of devices), 70% of devices are self-installable by the user.

5.2.2.11. RESOURCE SHARING

(a) Positive impact assessed by the different stakeholders (b) Negative impact assessed by the different stakeholders

(c) Value Matrix assessed by the different participants for different stakeholders

Figure 5.17: Results from the focus groups evaluating Resource Sharing showcase

According to focus group participants this showcase has:

- High value for Technology Provider, as well as for Service Provider.
- Better impact on user experience
- Technology Providers think the benefits lay on economic and organizational aspects
- Service Providers and Government give value to technical impact
- Limited economic implications
 - Technology providers completely disagree

It is also noteworthy this is the least understood showcase, especially for Government representatives; it may stem from the video produced; it is recommended for future versions to review this video.

“In my opinion, end-user is not really interested in how much equipment he has at home and how many/which software applications are used, he is more interested in direct consequences for him, e.g. how much he pays for the electricity. In that context it may be important to reduce the amount of hardware in end-user’s home and integrate(...).”
Technology Provider.

“from the ethical perspective, the share of data can cause conflicts.”
Service Provider.

The other data gathered from pilots shows that the most common resource shared is Hardware (52%).

5.2.2.12. SCALABILITY

(a) Positive impact assessed by the different stakeholders (b) Negative impact assessed by the different stakeholders

(c) Value Matrix assessed by the different participants for different stakeholders

Figure 5.18: Results from the focus groups evaluating Scalability showcase

According to focus group participants this showcase has:

- High value for Technology Provider

- High economic benefits
- Some ethical concerns

“crucial that there is analysis of scalability before piloting.” Government Representative.

“Scalability is a precondition to be able to realise ambient intelligence. So scalability is most important.” (Service Provider)

“Large scale service integration could save costs especially for heart disease treatments. Same costs and better quality of life.” Service Provider.

“There are hundreds of thousands of e-health apps. The question is: where is the data going and what are you doing with it.” Technology Provider.

From the gathered data from the NCSR Pilot the economic value is clear.

Figure 5.19: NCSR pilot data showing costs per deployment for different number of deployments, actual and prospected.

For ODE the personnel involved in each scalability action were 6.2 and 14.7 person per 1000 users. This difference might indicate that each service has unique requirements per user. Although the difference of users (500 the first and 300 the second) might also indicate that scalability starts to pay off early, by reducing by half the number of personnel required per user every 200 new users and that will translate in lower personnel costs per user.

5.2.2.13. SECURITY AND PRIVACY

The value data for this showcase is provided by the templates filled by pilots, as it has not been discussed in any focus group.

There are only 7 entries for this showcase, denoting poor regard towards this showcase. Form the data the most popular mechanism to protect data is through username password (account) mechanisms. To encrypt information across networks the most popular mechanism is SSL (57%).

5.2.2.14. SERVICE INTEGRATION

(a) Positive impact as- (b) Negative impact
sessed by the different assessed by the differ-
stakeholders ent stakeholders

(c) Value Matrix assessed by the
different participants for different
stakeholders

Figure 5.20: Results from the focus groups evaluating Service Integration showcase

According to Focus Group participants this showcase has:

- Benefits for user experience
- Economic benefits
- Extremely low negative impact

Value for stakeholders highly depends on the participant. **Technology Providers** see themselves as the most beneficiated by the showcase. **Service Providers** think that users and themselves can profit more from this showcase.

“As a technology provider we saw the biggest benefit in services integration, because it simplifies the integration of different services through an open platform, when different services are already connected to the platform. The reuse of service components is also interesting and beneficial, but remains to be seen how reuse among competitors look in practice.

The prerequisite for this is that open platform becomes a de facto or de lure standard, so that it can reach a critical mass of integrated applications/services in its ecosystem. This success of an open platform is also very critical for preventing vendor lock-in, another benefit of a successful, widely recognised open platform.” Technology Provider.

No significant data could be extracted from the other gathered data regarding the value for this showcase.

5.2.2.15. SERVICE TRANSFERABILITY

(a) Positive impact as- (b) Negative impact
sessed by the different assessed by the differ-
stakeholders ent stakeholders

(c) Value Matrix assessed by the
different participants for different
stakeholders

Figure 5.21: Results from the focus groups evaluating Service Transferability showcase

According to focus group participants this showcase has:

- Best value for Service Provider; Service Providers disagree.
- Economic advantages

Technology Providers tend to agree there are good impacts but organizational, technical and ethical issues must be carefully considered.

Government Representatives value this showcase as the most interesting, even though they agree there is no impact on user experience, economical and organizational benefits are clear.

Service Providers have a very neutral approach for this showcase. This may stem from their difficulty understanding this showcase.

“For me what is very important of universAAL is that I see that if I want to have another functionality for my application or even a new application I do not have to develop, I would be able to import it and that is it. And that has been for me a very good discovery and that’s what I have been completely conscious of, of the importance of working with an open platform. That you can go there and say “well this will complement my application” , of if other say they are interested in part of my application. So this interoperability of the platform, for me has been very important.” Service Provider.

“In associated pilots we did not have the opportunity due to time, resources, etcetera to pilot the imported applications. But this I believe would be something of an immediate benefit from the open platform, if we had done this. Because this actually means that you do not just have this potential benefit when you import some application from somebody else but you actually experience that is is easier. You do not have to worry about implementation, you do not have to develop. I can really see the benefit of that.” Technology Provider.

The data shows that the cost of importing applications is significantly lower than the initial development costs. Therefore from an importing perspective (destination of the transferred service) transferring the service is economically more viable than developing a similar solution. From an exporting perspective (source of the transferred service) it also makes sense, as this provides the means to cover development costs, as well as the possibility of scaling up.

The best indicator for value for this showcase is the reports from pilots about their experience importing and exporting applications see the Pilot Appendices for more information. These specific enquiries to the pilots about the transference of services show that Service Transferability is preferable than developing from scratch, however there are certain considerations to ponder:

- The price has to be competitive, at least in the local market
- Original restrictions (like vendor lock in) must not be inherited.

5.2.3. REFLECTION ON SHOWCASE EVALUATION

Throughout this section we will discuss the general statements that the showcase data has taught us.

Completeness of the showcase list

One concern, early on, was about the completeness of the described showcases. Precisely how we will determine that the list is closed. We have observed that when discussing the value, focus group attendants would discuss different topics, thinking they were contributing to the showcase; when in actuality they were discussing another defined showcase. The fact that from those stray discussions no noticeable showcase, which had not been defined, has been discovered means that the list of proposed showcases covers the most pressing issues regarding the AAL domain. Yet the list is not closed, smaller topics and new concepts may require the definition of new showcases.

The Ethical criteria of showcase is not prominent enough

One general comment we must acknowledge, specially reviewing the data from the focus groups, is that the definition of showcases must be clearer on ethical aspects. Granted the showcases have to cover many details and perspectives, and that the videos shown at the focus group failed to show every nuance of the showcases; yet the descriptions of showcases have little orientation towards ethics. These issues may only arise when dealing with specific stakeholder perspectives. Thus for future versions of the showcases definitions it is highly recommended to add explicit ethical discussion for each.

Status on demonstration of showcases

One specific objective for this task has been the demonstration of the showcases both technically and in terms of value. Our determination has been to provide solid proof of the implementation of the showcases both for ReAAL applications as well as the aid universAAL IoT platform provides for such implementations. Thus many plans and effort has been devoted to gather these results. Many of the original plans could not have been carried out, due to resource, scope, or other constraints; thus we feel that we could have provided more evidence for the workings of the showcases. All in all we managed to provide very concrete evidence for the presence of most showcases across the project. The core showcases: Resource Sharing, Plug and Play, Evolution and Services Integration along with Service Transferability have been thoroughly demonstrated, if not through individual pilot actions through project-wise initiatives such as demonstrators and forcing importing applications on pilots.

5.2.4. CONCLUSION: KEY VALUE PROPOSITIONS

The top five most valued showcases, according to our data, are Evolution, Plug and Play, Service Transferability, Services Integration; and Scalability. In the next sections we draw value propositions centred on these showcases.

5.2.4.1. VALUE PROPOSITION EVOLUTION

This showcase is one of the few that proposes a novel business model. The business model for evolution is based on the improvement of the service. One interpretation is through a similar idea to service level (typical classifications being bronze, silver, gold and platinum); where each of the service levels has a different number of applications, users can then apply for upgrades in their service level. Another interpretation is diversification; by offering different services the customer can then select the combination of services that better fit his needs at that moment, with the option of changing the set any time.

These ideas for business models are extremely interesting for Technology Providers. As they know that both Service Providers and End Users are uncertain about the exact needs; this showcase offers the solution to cover the known problems while leaving the door open to more solutions.

Technology Providers are empowered by what the showcase enables, as they have more flexibility in their internal processes. Evolving solutions allow Technology Providers to deliver a working solution, while working on the next and solving the problems of the last.

Yet another innovative practice that this showcase can introduce is the capability of introducing Assisted Persons, who are typically reluctant to technology and/or aid in general (as this psychologically in many cases is perceived as defeat), in a gradual manner. By introducing less intrusive, easier to use services first the user can feel the advantages of the concept and makes them more comfortable with and susceptible of acquiring more services in the near future.

This concept can be translated into innovative marketing strategies in the AAL domain that have been proven in other domains. For example Freemium services can be offered, Service Providers offer one of their catalogue of services for free; typically one that is not very resource consuming; as the user gets comfortable and explores the limits of the service, she/he will probably choose to sign up and pay for other more complex services.

End users, through evolution, can access services that otherwise would be personally challenging, or even impossible. This friendlier approach to technology leads to better acceptance of it, eventually better acceptance correlates with compliance, which means that evolution is also a strategy for Assisted Persons to better their lives. For Service Providers such as national health care systems, compliance means that there will be less hospital admissions.

Evolution also enables the personalization of the care, by periodically reviewing the consumed services for a particular person, new useful services can be offered that better address new conditions in the person's life; as well as removing services that are no longer needed. For assisted person costs are reduced, as only the really needed services are consumed, and also this optimized services selection helps improve quality of life more efficiently.

Personalisation of care has advantages not only for assisted person, but for the Service Providers too. Service providers, can optimize resources on delivered ser-

vices, knowing those services are really used by people. Evolution also gives the opportunity for Service Providers to address the problems of their Assisted Persons. They are able to adapt to new circumstances, not just in the living environment, but social, economic, geological or any other situation that might arise in the region or globally. Evolution allows this circumstances to be addresses even if it means incorporating new service domains, which can be resolved in innovative ways, for the AAL domain. One way is to subcontract the new service to a specialised company, thus new ways of cooperation and markets can be created.

universAAL allows for the implementation of evolution showcase, and through other implemented showcases like Services Integration, and Resource Sharing; it also allows for very advanced evolution models. In fact universAAL IoT proposes an evolution-based deployment where it is always opened for new services and ready for new circumstances. All platforms, in general allow for evolution, but open platforms have extra advantages. Open Platforms allow for more advanced concepts of evolution, that is because of the openness nature of the platform application developers can look deeper into upgrading models and interoperability. Open platforms also do not have an “expiration date” , whereas closed platforms’ life cycle is dictated by the owner; and if the owner company goes bankrupt, the only way for the platform to continue surviving is to go open.

5.2.4.2. VALUE PROPOSITION PLUG AND PLAY

This showcase offers the possibility to choose the components during the deployment of applications. The first direct advantage is that components of the same class (that solve the same problem) are now in direct, and fair, competition. This competition forces Technology Providers to produce high quality products at lower prices if they are to strive in this market, thus translating in fewer costs and greater quality for Service Providers and End Users. While.

Plug and play certification ensures Service Providers and End Users that they have not fallen for a commercial trap, which we call “Vendor Lock-in” . This practice consists on Technology Providers forcefully developing technology which is only compatible according to the vendor interests. For example, producing an application that is exclusively compatible with devices fabricated by the producer of the application, will in theory, force customers to only use both the application and the devices from this vendor.

Vendor lock-in is a practice that promotes monopolies, and can only survive in monopolised sectors. Any sector which is monopolised will eventually be challenged by the market, policy makers and customers. Thus this showcase ensures the long term stability of the sector.

This is because vendor lock-in is not only a commercial trap for customers, the vendor is trapped too. This practice forces them to “verticalise” their business, e.g. producing both software and hardware. This means that the company that uses this practice must then compete at two (or more) markets, which is a risky investment since the only viable solution is for the company to monopolise both markets (take

all, or go home). This kind of boldness can only be undertaken by big companies, and even then big companies typically do not choose this path (see for example Google, probably the biggest company in the world, actively avoids vendor lock-in making their products highly interoperable is their main goal).

Most customers will be happy with vertical solutions, initially it means less hassle for them, but more demanding users will not feel comfortable with either the hardware or the software, or both. Competitors will enter the market with non-locked solutions targeted for demanding users; in turn the bulk of the user base will turn to these more free solutions. Take for example Nokia, their model was essentially a vendor lock-in where the users had to acquire both terminal and software; this model ended as soon as the openness of the Android System appeared in the market, no longer vendor lock-in is accepted by the bulk of users, and therefore Nokia was forced to change the model, once the biggest telephone company now it only produces hardware, and its net value is only a fraction of what it used to be. Thus Vendor lock-in is not only long way from being a long term strategy, it is very expensive to maintain.

Initially “avoiding vendor lock-in” may seem as a disadvantage for Technology Providers, however as shown, embracing plug and play is the best practice to follow ensuring long term stability.

Plug and play is the embodiment of freedom in the technological sector. It allows Service Providers to choose the best fitting solution (even if it not the cheapest) when they have to make technological decisions, and for those customers that do not want, or do not know how to choose there is always the option to get standard solutions. Typically the chosen solution will be one of the contributing factors to offering better services at lower costs.

The same can be applied to Assisted Persons: they can choose, after all these solutions are going to be part of their home, of their live; it is only natural for End Users to have the last word on these aspects. And if they do not care, then there is always a standard package. Forcing specific technology on users predisposes them to naturally oppose these solutions; whereas if customers feel the control over the technology, they will be better acceptance ratio. This freedom paves the way for happier, more engaged users, enabling the service to intervene more efficiently, improving dramatically the quality of life of the user.

Plug and Play and open platforms go hand in hand, they are essentially two views of the same philosophy. Closed platforms are not foreign to the plug and play concept, yet open nature greatly improve the usage by developers. This is why even closed platforms have their APIs related to Plug and Play open, or more open-inspired.

universAAL IoT implements Plug and Play, it has to since the community does not include any hardware developers. What really makes universAAL IoT the champion of plug and play is the innovation of the application of semantic interoperability as the cornerstone for plug and play. This allows even greater levels of flexibility when developing modules and hardware for the system.

5.2.4.3. VALUE PROPOSITION SERVICE TRANSFERABILITY

This showcase has attracted the attention of policy makers especially. This is because it resonates with the European spirit, solutions that work on some region should work on all. Not only the showcase puts forward an ideology, it establishes the foundations for a true open market for the AAL domain across Europe, even the world.

An important part, especially for the AAL sector, is that this open market is friendly to SMEs. By ensuring the solutions implement this showcase, the doors for internationalization are opened, meaning companies are no longer restricted to local markets.

Internationalization is not a new concept in economics, however in the AAL domain internationalization is a challenge, since it is not only the technology that needs transferring, in most cases the service as well. Services typically involve staff, processes and other resources; these need to be easily transferable to or supplied by the local market where the service is being transferred to.

Thus the full extent of Service Transferability is not just that the application can speak different languages, it needs to function under different management, with different processes. This can be achieved via another important showcase: Service Integration. Although it may seem trivial, language, cultural and administrative barriers can impede people from accessing services on their region. Service transferability ensures that local issues are modularised, and can be replaced for other modules compliant in other regions. This way people can access, without restrictions new concepts of services, proven to work efficiently, which will no doubt increase the quality of life.

The fact that a service has been proven to work on one deployment site, is a guarantee for potential customers of the product capabilities. High quality products will be showcased more, and thus attract more customers around the world. Implementing Service Transferability for Technology Providers is yet another form of marketing.

For Service Providers accessing existing technology is not only reassuring the service will work, it is also cheaper. On one side the gross development of the technology is already finished, thus acquiring it can be cheaper; although it has been shown that the added value of a proven solution can potentially be an investment. In general it is expected for Service Providers for this option to be cheaper than requesting for a development from scratch; importing a service means that the development costs can be diluted amongst several service providers who exploit the technology.

Another reason for transferred services to be more cost effective is the fact that constructs a market, thus technology providers compete not only locally. Seen from the customer side, there are more options to choose from, thus the final selection is not only restricted by the local market, other factors, such as price, contribute.

Technology Providers who offer transferable services access more Service Providers, and more End Users, both contribute for greater revenues. Scalability showcase

(which we will examine later) plays also a role here. Since more deployments (across different deployment sites and Service Providers) can influence, through economy of scales, the price tag for customers.

Since the market for Technology Providers is no longer limited by the local market, specialised solutions for minorities are now economically viable. For example particular solutions for people suffering from a rare disease can be successfully commercialised, and thus these people can now access solutions that otherwise would be impossible. This concept not only applies to minorities, exclusive solutions that target broader customer spectrum will emerge as the access to customers is easier; therefore innovative solutions are now available for everyone everywhere.

Both open and close platforms usually have features to help internationalization, in fact in technology slang internationalization is the process of making an application speak different languages. Granted this process is an integral part of the showcase, yet most platforms fail to recognise that internationalization is not just dictionary files, there are cultural, legislation, and administrative differences between regions. This concept is very innovative, and especially important in AAL, so we make the assumption that open platforms are better fitted to address these issues than close platforms do. This is because the philosophy of open platforms, allow third parties to take the core of the system and specialised, in this particular case, customizing the platform to fully comply with specific regions.

universAAL IoT, as any other open platform does, has internationalization methods (as an example: its user interaction framework is readily translated to many European languages, and has an internal model of over 200 languages). But thanks to the semantic interoperability proposed by universAAL IoT, it is not necessary for developers to customize the core to fit regional requirements, these can be fitted as normal application modules; again with the aid of the Services Integration showcase, Service Transferability is heavily improved.

5.2.4.4. VALUE PROPOSITION SERVICES INTEGRATION

The capability of integrating several services might at first glance seem a trivial task, however in technological terms is it a huge challenge, even when the applications are designed to interoperate forehand. With semantic interoperability different applications, whose developer teams have not even the knowledge of each other's existing, can automatically integrate thanks to common knowledge, which we call ontologies. This concept is a fable or a myth for most Technology Providers. In this section we will not focus on the how this is achieved, but on the implications of it.

When services easily integrate Technology Providers can take full advantage of other technology. Whether it is pieced or whole, recycling existing technology is always a good idea as there is no need to re-develop it, this makes it easier to develop new solutions as well as cheaper.

Aside from the evident economic benefits, if there are broadly available generic solutions for common problems, and these solutions can be immediately applied to specific problems, then producers of solutions can focus on "higher level" problems

better, and solve those. Thus more complex problems can be addressed, and then those become available to solve even more complex problems. Service Integration is the means to advance forward at increasing speed.

If we look at the market of solutions, or better yet partial solutions, common understanding, and the capability of making everything interoperable, means that companies can specialise on specific problems. This focus can be driven by the demand on specific problems or by the fact of the access to readily made solutions. Specialization means that better, faster, more efficient solutions can be developed; and access to these solutions is increasingly important for customers.

Service integration fosters competition. Solution providers will compete to offer the best solution at the best price. Yet it also promotes cooperation, where solution providers combine their technology to achieve new goals. Both concepts are beneficial for Service Providers, as with competition they can ensure they are offering the cheapest solution, whereas with cooperation the solution, and especially for complex problems, is the best.

In AAL the model of deployment, where the technology is deployed in order to assist people, has been very particular. In other domains the platform and the offered service are completely disassociated, but in AAL there is no clear separation between both. For example in smartphone domain, the platform, the device, and the service, the app, can be interchangeable, each can be provided by different vendors; and in fact different service providers can use the same platform to offer a user their different services. This has been dubbed “the App Revolution” . In contrast in AAL both the platform and the service are an indivisible unit; we see in many of our pilots and in the industry in general, that the service is bundled with the technology. This leads to situations where users may be forced to replicate resources if they want to enjoy different services, in the smartphone analogy it is like having to buy a new phone for each app to be installed.

Another important concept that the App Revolution for AAL will bring is that with Service Integration there is no longer need for big apps, i.e. complex services that server multiple objectives. Instead simple services can be developed to offer specialised solutions, and cooperate to provide the complex system that big apps propose. Needless to say, simpler applications are cheaper than complete systems and they are more recyclable, and adaptable which means that even if the integration costs are higher, they are diluted when they are reused for future systems.

Once the platform and the services are disassociated, then different Service Providers can offer their different services in the same deployment for the user, without hassle for the user. And thanks to Services Integration, the services can cooperate offering an integrated care feeling to the user, even if behind each service there is a different provider. New models arise; Technology Providers will set up and offer services focused on the deployment/platform; leaving specialised services to the experts. The most immediate impact for end users is that they are getting the appropriate service from the appropriate provider, therefore it is safe to assume that the quality of care, and also the quality of life, improve compared to fully integrated models.

Open platforms are better providers of Service Integration, since the internals of this integration are known, on the other hand closed platforms that do offer integration mechanisms, typically restrict developers to specific types of integration. universAAL IoT is the champion of Service Integration thanks, once again, to the concept of semantic interoperability. This is truly the core in universAAL's philosophy, the ability to make the integration mechanisms dynamic, enabling developers to define their own or conform to standard interfaces, and going beyond allowing the interfaces to be mapped even after the development is over. This feature makes it extremely easy for developers to integrate their applications even when the applications they want to integrate with do not follow the norm.

5.2.4.5. VALUE PROPOSITION SCALABILITY

Achieving “critical mass” , or the number of user beyond which your system is economically ensured, is necessary for every service endeavour. Scalability ensures the technology does not get in the way. The key point of Scalability is this that users get the same service, independently of how many users there are. As our focus group participants have taught us, this issue has to be treated with a strict ethical mind-set. It is not about having more customers, for more money; it is about offering more people better service.

As a consequence that economy of scale can be applied. This means that all stakeholders will be able to access (and sell) technology and services at more competitive prices. For any serious exploitation endeavour where the service is self-sustained (beyond critical mass), and not backed by subventions and public funds, scalability is a requirement. Costs of service provision need to be distributed amongst many customers, the more the better. In doing so not only each customer gets the same quality service they get it at an advantageous price.

If more services were to achieve critical mass, by applying different techniques (including previously mentioned Service Transferability), then there would be more and better, more competition and there would be more services available in many more domains. AAL market will really take on, and as with the App Revolution, more people will have access to services, and not just because they will be cheaper, because this trend will help people benefit from the plethora of services. Especially the stigma of getting aid will be mitigated as everyone will be getting more AAL services, even at younger ages. Accessing some AAL services earlier, in many cases, helps improve the quality of life in the long run. Certainly for Service Providers such as national health care systems, early prevention is a crucial factor in improving general health, hospital admissions, delaying institutionalization and ultimately optimising the whole system.

Scalability has to be addressed technologically, if the system is not easily installable configurable and manageable, then the time consumed will accumulate through all the deployments. This has some relation to another showcase: Administration.

Closed platforms are targeted towards scalability, in deed the open philosophy in this case does not provide a noticeable advantage. Closed platforms can more

efficiently provide optimizations of the platform for specific cases. Open platform on the other hand have to wait for optimizations to be contributed by the community. universAAL IoT has several means to provide scalability, however platform developers have targeted this showcase as the most immediate challenge to address. For example currently the platform has no remote deploying and configuration mechanism, it has however identified other open platform compatible with universAAL IoT that *will* address precisely these issues.

5.3. METHODOLOGY VALIDATION

Similarly to previous section, in this section we will explore some examples of research projects, but instead of validating the proposed models we will attempt to validate the proposed methodology.

The proposed methodology has been derived from the experience in many projects, in fact the following 2 projects have been part of this iterative process. This is not to say that the methodology used in each of them is distinctly different from the final proposition, therefore the examination of this experience should be considered part of the validation of the proposed methodology.

5.3.1. THE ADAPTATION PROCESS IN REAAL

The term ***universAALization*** was used in the ReAAL research project to refer to the integration of all functional components to be deployed with the universAAL IoT software platform prior to their deployment. The goal of the universAALization process is to resolve the dependencies between these components for the exchange of data and functionality based on the universAAL IoT open platform in order to achieve future-proof interoperability, adaptability, and extensibility to a wider extent.

The following is an abstracted workflow representing the patterns followed by pilot sites when adapting their solutions to universAAL IoT, i.e. the universAALization process. It represents a generic step-by-step approach for any Technology Provider to convert an existing solution to a compliant universAALized one.

5.3.1.1. ANALYSIS AND DESIGN

The first stage does not require any change yet. It is only an analysis of the existing situation which leads to a final decision-making step that will lay the path to the actual implementation stage.

5.3.1.1.1. ANALYSIS OF INITIAL SCENARIO In this stage the basic architectural distribution of the existing solution is analysed. The universAAL platform supports both local and server-based (cloud) scenarios, or a combination of these. The first analytical step will be to identify which one of these options corresponds to the existing initial architecture.

Client-based All the logic is run in devices deployed at the client side, whether that is in fixed or mobile devices at the Assisted Person’s home or in a mobile device the Assisted Person carries outside home.

Figure 5.22: Client based application

Server-based All significant logic is run in a server hosted by the deployer. It is possible that some amount of processing is performed by devices at the client-side, but only for the purpose of sending the information to the server, not performing any processing on it – such as a gateway gathering sensor data to send them over the Internet on their behalf.

Figure 5.23: Server based application

Client-Server Some data processing or business logic is performed by devices at the client side, while other takes place in a server. This may all be part of the same application or be combined by independent solutions – such as deploying a client-based application *and* a server-based application at the same time.

Figure 5.24: Client and Server based application

5.3.1.1.2. COMPATIBILITY WITH UNIVERSAAL IOT In this stage the constraints of the existing solution and where universAAL IoT is able to be deployed are identified. Identifying which one of the above scenarios corresponds to the existing architecture will help answer the following questions. These are aimed at identifying where the universAAL-aware nodes can be deployed (These are devices that can run universAAL IoT).

Where are applications currently running? In the initial architecture, identify in which devices and execution environments the business logic of the

applications is being executed – is it a JVM in Windows? In Linux? Is it an Android app...? Which versions?

Where could universAAL IoT run? Right now, the universAAL IoT platform can be executed in devices running Android 2.3 or above, and in Java Virtual Machines (JDK 5 as a bare minimum for the middleware) through an OSGi Container. Identify which of the existing devices in the initial architecture satisfies the requirement. For instance, in the case of OSGi, the hardware requirements are as low as mini-computers like the Raspberry Pi Model-A (700 MHz processor with 256MB of RAM), while for servers the requirements will depend on the number of clients to service and the complexity of the applications.

Can applications be universAALized “natively”? When universAAL IoT can be run in the same device as the initially deployed applications, consider if it would be possible to universAALize these applications directly. This “native” universAALization consists on having the applications communicate with a universAAL IoT instance through the APIs corresponding to their execution environment. Android apps and OSGi bundles can do this, using the universAAL IoT API in Android or in OSGi respectively (details are described in later steps). If this is not possible, alternative methods would have to be used (such as universAAL’s Remote API).

How will universAAL IoT instances connect to each other? If universAAL IoT instances can be deployed in more than one node, determine if and how this communication will happen. Instances in the same IP network can communicate directly to each other. In different networks (or through the internet) they can communicate with the AAL Space Gateway or the Remote API. Or not be connected at all, if that is not necessary for the deployment.

5.3.1.1.3. SELECTION OF UNIVERSAALIZATION STRATEGIES Answering these previous questions does not yet provide the right choice for universAALizing, but will help taking the final decision on which of the 4 options should be deployed, and how to implement it. These universAALization options provide value to the deployer in different ways; it is how the deployer benefits from having universAAL IoT. It is possible to combine more than one of these options. What options to take will depend on the aim of the deployer, the client/server scenario identified in the first step, and the restrictions identified by the previous questions.

Decouple devices from applications Instead of hardwiring devices to applications, the communication is abstracted thanks to universAAL IoT, which allows to connect newer devices from other technologies without modifying the applications, and allow other applications to benefit from existing devices.

Figure 5.25: Using universAAL IoT to decouple devices from applications

Decouple applications from each other Instead of hardwiring the communication between applications, it is performed through universAAL IoT, which allows to connect new or different applications providing or consuming the same type of information, without having to change them.

Figure 5.26: Using universAAL IoT to decouple applications, as producers and consumers of data and services

Decouple applications from servers Instead of hardwiring the communication of client-side applications towards the server side, the communication is performed by universAAL IoT, so for the client application it is like the server application is being run locally.

Figure 5.27: Using universAAL IoT to decouple applications from their servers

Virtualize applications in the server If universAAL IoT cannot run in the client side, an instance of universAAL IoT run in the server can virtualize this client, allowing client applications to benefit from universAAL IoT even if it is not run locally.

Figure 5.28: using universAAL IoT as an application virtualization solution

To consider that a deployed solution (application, service or a combination of any number of these) has been universAALized, at least one of these 4 options must be implemented. The result of this exercise determines in great measure the distribution model for the different solutions.

5.3.1.2. ADAPTATION

Once it is known which universAALization options to implement, where to run universAAL IoT and how and what to connect its instances to, it will be clear which application modules have to be universAALized (understanding application modules as pieces of executable business logic, not the overall application, which can be comprised of many).

The process of universAALizing an application module involves making it run in a universAAL-compatible container (Android, OSGi) and perform/allow some interaction through universAAL IoT buses. When an application module cannot run in those containers but still has to be universAALized, the Remote API can be used. In any case, it is imperative that the interaction with universAAL IoT is made in terms of ontologies.

5.3.1.2.1. SEMANTIC MODEL Ontologies are the data model in universAAL IoT. In order to communicate with universAAL IoT the data exchanged must be represented by ontologies. Developers must analyse the original data model of the application, and extract the basic information needed for the business logic, to a bare minimum, need-to-know basis. This helps identify the meaningful parts of the data that need to be shared (the “semantics”).

Ontologies are application- and hardware- agnostic, which means they should not model things like raw values of sensors, application-specific constants and the like, nor used to stream raw data (even at low frequencies). It must always be taken into account that other developers should be able to understand the data model to work with it without knowing anything about how the application or hardware work (even if nobody else will use it in the end). Developers should aim at having an ontology that is as future-proof as possible to reduce later needs of updates. This follows the criteria for ontologies, namely that ontologies should be: reusable, meaningful, extensible and future-proof.

Once it has been identified what kind of data the application is going to handle (sometimes referred to as the “domain”), it is necessary to inspect existing universAAL IoT ontologies to see if there is already one that can be used. It could be that:

- There is already an ontology that covers all the needs of the application: In that case, use that one.
- There is one that covers the needs of the application but would need more concepts, or more detailed knowledge: Use that one and create an extension for it that includes the missing concepts. Ontologies allow extensions: additional

modules that can define new concepts that “inherit” or “extend” existing concepts in other ontologies.

- There is none that covers the needs of an application: Create a new ontology specifically for the application, but trying to also make it as generic as possible so that other applications in the future can benefit from reusing it.

These terms can apply to one or more Ontologies, as sometimes domains overlap, a set of ontologies can provide alternative models for the same domain. In this case it is also useful to align the overlapping knowledge with a *mapping ontology*.

MEMBER MODEL The member model is provided by the 3 buses of the universAAL IoT platform.

In the Context Bus, information is shared in the form of Context Events. A Context Event represents a change in the physical context. These changes must be meaningful and relevant to the applications, the bus was not designed to “stream data” or share “raw” sensor data.

In the Service Bus, it is Service Requests that are shared, matched against Service Profiles. To build these it is necessary to start from a Service Ontology. Base Service Ontology is usually included in the related ontology module, but more can be created in the application modules. The module containing the Service Ontology must be available to both requesters and providers of services.

In the User Interaction Bus the information is shared as Forms. Forms are described in a default ontology included with the Bus. In general its concepts are enough for most applications, but just like any other ontology, they can be extended. However, UI Handlers that render the UI should be prepared to handle any new concept so this is not recommended. When used in declarative mode it can map forms to ontology resources.

When all needed ontologies are identified and ready, the application module must be adapted to make use of them and the universAAL IoT’s APIs for connecting to the buses. But before starting the actual implementation or modification of the code, it is best practice to analyse the software pieces of the modules to be clear about overall structure of the application, what needs to be modified, what needs to be created from scratch, and what “goes where”. The result of this process determines the members which will be connected to the buses, effectively bridging between the existing applications and universAAL IoT buses to connect to other applications and components.

COMPONENT MODEL Determine how the application logic is divided into different application modules, and how these interact with each other. This involves assigning different pieces of business logic, data representation (and/or ontologies), and, in universAAL IoT, also Bus members, to each module. Interaction between modules can happen in any way, not necessarily through universAAL IoT(although at least one will have to, for the overall application to be universAALized), but doing so will maximize re-usability and interoperability with other

applications. A module should be the smallest application part for which any configuration applies. Modules should have self-contained configuration elements (when needed). This means that the configuration of an application is the collection of configuration elements of all its modules, and each configuration element applies only to its module. Some global configurations may be replicated throughout the modules, but this will help increase the re-usability of each module.

As part of the component model it is also important to note that there is a selection of universAAL IoT managers (platform modules offering basic features) each application module will make use of. In particular, for applications that have a connection to hardware devices, it is necessary to determine which LDDI Exporter Manager to use, or if a new one has to be developed.

5.3.1.2.2. CODE IMPLEMENTATION Once it is clear how the software will shape up, the coding process begins. As mentioned before, universAAL IoT has two main native ways of accessing its API, based on the runtime container: in OSGi and in Android. That means that at least one of the software modules of the application (the one/those interacting directly with universAAL IoT) will have to run in these containers. Therefore they will be an OSGi bundle or an Android app, respectively. The creation and best practices for coding these modules will differ based on this.

OSGi It is logical that each application module being converted becomes an OSGi bundle. Bundles are the basic pieces of software in OSGi. Bundles contain an Activator that starts up the logic. The universAAL IoT platform is comprised of several bundles in OSGi, and these can be used from other bundles to access its core API.

- The universAAL IoT tools and documentation are oriented to managing bundles software as Maven projects. It is therefore recommended that the same is done when adapting application modules, but it is not mandatory.
- If possible, deployers should make sure that the app still works as expected when packed as a bundle and run in OSGi, before trying to universAALize the code. This may not be possible in all cases, depending on the adaptation and the original application, but is a helpful middle step to solve potential problems early.
- Once the bundle is created, create the needed code to access universAAL IoT API and create the needed bus members, following the decisions taken when designing the various model in the previous step.

Android In Android, application modules are usually stand-alone apps (although they can also be libraries). Because of the restrictions imposed by the Android platform, the way the universAAL IoT API can be accessed differs from the typical code writing that is performed for instance in OSGi. But first, there is a small set of additional design decisions to make.

- Choose whether to run universAAL IoT separately, or embedded in your app. The universAAL IoT middleware for Android is packed into a project that can be either installed as a stand-alone app in Android, or embedded into another app directly. Each option has its advantages and drawbacks.
- The universAAL IoT platform in Android, whether in a separate app or embedded, needs to be properly configured. While in OSGi this can be done with several configuration files, in Android it must be modified either at build time or at runtime through the adapted application. Something similar happens with the ontologies to be installed.

Remote API In those cases where an application module that needs to interact with universAAL IoT cannot be adapted to any of the runtime environments where the native API can be accessed, the Remote API can be used. This is a HTTP-based API that can be accessed by any software with Internet access – provided that the deployer has a universAAL instance running that publishes this API.

5.3.1.3. ASSESMENT, TEST AND EVALUATION

5.3.1.3.1. TESTING It is recommended that tests are performed through the development cycle of adapting the application to universAAL IoT (unit testing, integration testing, even test-driven development), but the adaptation process does not enforce this in any way. However, there are some features that can help with this task and are encouraged to be adopted by developers when adapting to universAAL IoT: The universAAL IoT platform has some plugins⁶ that enable unit testing bundles as if universAAL IoT was running during a normal execution in OSGi.

As to what to test, here is a list of suggestions that may help identify the targets:

- As usual, check the application and its modules work as intended.
- For cases when an application is being simply adapted to universAAL IoT, make sure the operation is the same as before adapting.
- Test the integration: that all the modules in conjunction make the overall application work as expected despite any possible re-arrangements or divisions due to the adaptation.
- Check that universAAL-based interactions are correct: that the Context Events are correctly sent and received, UI Forms are correctly rendered, and, especially, that Service Calls find their match. The latter is of special importance since the adaptation to a semantic-oriented service framework usually leads to some mistakes (more on this in the section about experience).

⁶<https://github.com/universAAL/itests/wiki/Integration-Tests>

- In general, remember to make use of all the tools available in universAAL IoT, such as the bus testers⁷, the log monitor⁸, or the command-line options.

As a final recommendation, use the logging system (log4j⁹) thoroughly and with a good strategy. This not only helps during testing but also provides a good way to audit the operation of the application and helps solving future unexpected bugs after deployment. It is also helpful when evaluating the proper universAAL IoT adaptation, as explained below.

5.3.1.3.2. VALIDATION The whole ReAAL project was oriented towards validation. This included not only the validation of the adaptation process it self, for which continuous coaching, as well as technical solution analysis where performed; but also validation of the universAAL IoT platform as a whole, as well as the feasibility, effectiveness and economics of the pilots' solutions.

For the purposes of validation of the adaptation process it is particularly interesting the analysis of showcases. This analysis may follow a similar methodology as presented in section 5.2. Analysing whether the showcases where achieved, and then evaluating the actual value for each given the final application, using the scripts presented in section 4.4. showcase compliance can be used as validation result for the end product.

5.3.1.4. GENERIC LESSONS LEARNED

Just like the adaptation process can be abstracted to a generic set of common steps applicable to everyone, there are some issues that have been encountered across several pilots. These identify aspects that should be improved in general by the universAAL IoT platform to facilitate adaptation in the future, regardless of particular specific issues that will always arise in any adaptation process.

- Developers approaching to a new platform such as universAAL IoT need first an off-line phase during which they can learn about the platform and the related component before starting any (pre)development steps. The quality, availability and diversity of the documentation is key at this stage to keep the motivation of the developer.
- As the platform is very rich in number of features and help methods, the technical team must well learn about the platform and look for the available features before deciding developing their own.
- The learning curve is steep specially at the beginning, requiring a lot of effort to learn the basics. However, the ability of developing applications on the top of the platform quickly evolve with the practice.

⁷<https://github.com/universAAL/samples/tree/master/integration.tests>

⁸<https://github.com/universAAL/tools/wiki/Log-Monitor>

⁹<http://logging.apache.org/log4j/2.x/>

- Training and coaching are key activities to ensure the success in adapting/developing universAAL IoT applications, but they need to be done by a universAAL IoT expert, otherwise common mistakes and pitfalls permeate into the final product.
- The most efficient way to coach a team that requires platform knowledge is using continuous communication. The forum of the platform is vital for knowledge sharing but on-line seminars and Skype calls are most effective tools for answering to questions from developers.
- There is the need to ease/customize the universAAL IoT applications development environment to maximize its acceptance from the developer side. Developer support tools for ontology design, test and debugging, as well as for configuration, deployment and maintenance are crucial for wide adoption of the platform.
- Main topics for support has been requested to setup universAAL IoT server and environment, and showing how to use the platform without using Eclipse, as well as modelling ontologies for the pilot applications.
- It is very important to study which exact ontologies to use, and how to extend them, to achieve an expressive and generic ontology that fit the pilot/application needs, before starting any implementation activity.
- For the coaching team, it is important to keep a look on the updates during development, especially during the ontology implementation, their optimization is key to the success of the universAALization and for the capability of implementing the desired level of interoperability.
- Typical mistakes have included the disregard of existing ontologies that could help develop a more complete modelling of the solution as well as the definition of very technical ontologies, for example the direct translation of interfaces to ontological concepts; falling in the category of data models rather than full semantic framework for the system.

5.3.2. ACTIVAGE MADRID DEPLOYMENT SITE CASE

The Deployment Site in Madrid of the ACTIVAGE research project is the focus of this section. The solution is tested with about 1000 participants, above 65 years old, scale frailty between 1 and 6 (in the Fried Frailty Index), with no cognitive problems and that can receive assistance from formal caregivers at home or at a day care centre. Participants will be given a smartphone (i.e. Samsung Galaxy S5) and some IoT devices depending on the concrete use case to test (e.g. smart wristband FitBit, NFC tags, IoT lamp, eBeacons, door sensor, etc.) Testing is performed in incremental iterative phases. Each new phase will include new services, new IoT

elements as well as more stakeholders who will be involved to the Madrid deployment site while it evolves, with the objective to achieve an AHA-IoT sustainable ecosystem.

5.3.2.1. ANALYSIS

The pilot draws from previous experiences in ReAAL(see 5.3.1), and City4Age (a project oriented to the application of smart city paradigm for healthier ageing) projects, proving some base infrastructure both in the AAL domain as well as smart city domain, or AAL for outdoor, urban areas. Analysis of User Needs was performed through the organization of focus groups. These showed, among other, the following key points:

- Localization, services provided in those places in which they move more often, like libraries or gymnasiums; instead of exclusively in traditional day care centres.
- Unobtrusiveness, the solution should not take over their current lifestyle nor disturb their activities in any way.
- Transparency, the solution should state clearly what it is doing, improving trust in it.
- Proactiveness, the services should only activated when they are really necessary.
- Personalization, for example activity recommendations should consider each user's preferences to assure best engagement.
- Reliability, single failure could cause users to distrust and abandon the solution.

These results might be unexpected by traditional AAL service providers, but this is because the target group are healthy and active adults, the objective being to delay as much as possible their decline.

These results helped re-define the use cases, identifying key elements of the architecture to address consumer needs. Any addition of confident, transparent and effortless components, should preserve, if not increase, their value for privacy and security and trust. Another important element to be included is associated with re-usability of technology and services across use cases.

There are 4 main use cases defined in the Madrid deployment site, the details of which can be seen in the tables 5.10, 5.11, 5.12, and 5.13.

Table 5.10: Deployment site Madrid Outside monitoring use case specification.

NAME	Monitoring assisted persons outside home
------	---

CONTEXT OF USE	This use case is provided outdoors. It can be requested by a family member or the elderly end user. It monitors the user activities outside the home by establishing patterns of behaviour, to detect changes that may be related to early symptoms of ageing, frailty and/or dementia. Data on activities outside the home are collected automatically and do not require the user interaction. When behaviour changes are detected, the user is notified so that the change in behaviour can be confirmed and proceed to analyse its risks.
STORY LINE	Luis is a retired university professor 71 years old. Luis lives in Madrid since he was a boy. He is single and lives on his own house. Nowadays, he likes to spend time with other friends at the day care center playing cards, domino and talking about football. He is an Atlético de Madrid supporter, he renews his season ticket every year, Luis likes to go to the Calderón Stadium on warm Sunday afternoon. He uses to move in an autonomous way around the city, using the public transport. Luis uses a wearable to control his daily steps, active time and sleep quality. Outdoor monitoring provides a monitoring outdoors service that tracks Luis' activities outside the home and in case of an abnormal behaviour (e.g. Luis is not going to the care center some days of the week), an intervention is delivered to Luis by notification on his phone asking him if everything is fine, and, if Luis does not answer his relatives are contacted in time.
INVOLVED USERS	Elderly person: +65 years old, with CFS level between 1 and 5, without severe health conditions, users coming from both private and public sector, non-dependent elderly people, active person living in Madrid near to the ACTIVAGE centers that are used to use public transport (EMT). Excluded persons with mental disease or cognitive impairment that are not able to sign an informed consent, Service provider / Care-giver: Assistance service provider that supervise daily activity trends of the elderly and intervene if needed through notifications, reminders and alerts.
GOALS TO BE ACHIEVED SYSTEM CAPABILITIES	Elderly : increased safety and security outdoors Service provider / Caregiver : Minimum false alert, reduction of undetected behaviour change that may lead to a risky situation Wearable sensors to detect activities and user behaviour. Gateway to collect sensor information and send it to IoT platform. Processing module for activity recognition and behaviour detection. Application for the intervention delivery, to the elderly or the relatives.

Table 5.11: Deployment site Madrid Exercise promotion use case specification.

NAME	Exercise promotion for fall prevention and physical activeness
------	---

CONTEXT OF USE	This use case is provided in an ACTIVAGE center, which can be located in a gym, library, a senior center or any other often visited communal place. It can be requested by the elderly person, a relative or a professional caregiver. The service consists of a device called Equimetrix which allows the measurement the postural stability of user using of a sensorised floor platform combined with a wearable camera positioned on the body trunk. Equimetrix offers balance assessment and training services via a graphical user interface. Users, in particular Frail users, will always be supervised by a professional caregiver. The measure of the postural stability could be correlated to the risk of falling. In addition, since the application provides balance training exercises, it contributes to prevent falls and promote exercise.
STORY LINE	Maria is an elderly woman 68 years old. She has been widow for the last 10 years and she lives alone. Often, her daughter Eleni, visiting her, because she is anxious about her mother's health, especially the last 2 weeks in which Maria fell twice. After discussing with her mother, they decide to attend the training in the care centres 3 times per week. After the first month of using the system, with the help of a supervisor, Maria feels more confident and her stability has demonstrably improved while at the same time Eleni feels happy that Maria is still independent.
INVOLVED USERS	Elderly person: +65 years old, with CFS level between 1 and 4, without severe health conditions, but that has at least suffered once a fall in the last year, users coming from both private and public sector, non-dependent elderly people, and active person living in Madrid near to the ACTIVAGE centers; excluded persons with mental disease or cognitive impairment that are not able to sign an informed consent . Service provider / Caregiver: Person that supervises the balance assessment of the elderly and guides him on the use of the Equimetrix.
GOALS TO BE ACHIEVED	Elderly : Increase postural stability and confidence (reduce fear of falling) Service provider / Caregiver: demonstrate the improvement (or decline) with empirical evidence.
SYSTEM CAPABILITIES	Complete Equimetrix system, able to estimate and train user's balance, including sensors (floor platform and webcam), Gateway to collect sensor information and send it to IoT platform, Real time processing module for balance recognition and Application for balance assessment and training. All running in each local ACTIVAGE Center.

Table 5.12: Deployment site Madrid Cognitive stimulation use case specification.

NAME	Cognitive stimulation for mental decline prevention
------	--

CONTEXT OF USE	This use case can be prescribed by relatives or the public (or private) health service, to prevent minor cognitive or memory problems. The service consists of a series of cognitive games, which serve to train the brain. The user interacts with cognitive games through a tablet or mobile phone and receives a score with the result of his training. The data with the results are sent to the IoT platform for processing and in specific situations notify the user of behaviours that can improve his cognitive health.
STORY LINE	Alejandra is 75 years old and lives with her husband, Fernando who is 80 years old. They have two sons who live far away, and they only communicate by the phone or over the internet. Alejandra is worried about Fernando because she has noticed that the last 3 months he has developed mild cognitive problems, which include memory loss but not severe cognitive decline. The exercises are installed in the tablets of their neighbourhood library and they offer also a tool for assessment and selection of the suitable cognitive training for him.
INVOLVED USERS	Elderly person: +65 years old, with CFS level between 1 and 4, without severe health conditions, users coming from both private and public sector, non-dependent elderly people, and active person living in Madrid; excluded persons with mental disease or cognitive impairment that are not able to sign an informed consent. Service provider / Caregiver: Person who supervises the trends of the results of the elderly and intervene if needed through notifications, reminders and alerts.
GOALS TO BE ACHIEVED BY THE SYSTEM CAPABILITIES	Elderly : Increase cognitive abilities and confidence. Service provider / Caregiver: demonstrate the improvement (or decline) with empirical evidence. Application to train and measure cognitive skills. Gateway to collect the results and send it to IoT platform. Processing module for trend, behaviour and cognitive decline recognition. Application for the intervention delivery, to notify the users.

Table 5.13: Deployment site Madrid Prevention of social isolation use case specification.

NAME **Prevention of social isolation**

CONTEXT OF USE	This use case can be prescribed by relatives, to prevent social isolation of the elderly living alone. The service is provided at the users home and aims at detect long stays of the user at home, and communication activities of the usage of the smartphone and a chat applications. The detection of these behaviours is transparent to the user, and is done through the use of home sensors and applications on the smartphone that send the anonymized data to the IoT platform. Once processed and analysed, interventions can be delivered back to the user. This service detects the interaction of the user with the ACTIVAGE chat application and the use of the smartphone with NFC Tags, collecting data of its use, as well as the position of the user, whether alone at home or outdoors.
STORY LINE	Maribel is 77 years old, divorced for the last 20 years and lives alone. She has two daughters and one son who live far away, and they only communicate by the phone. In the past, she used to go out with her friends in the neighbourhood around her age but nowadays she has only 1 of them left and she spends most of her time at home watching TV. One of her daughters that is aware of this situation has advised her to go to the closest senior centre and use the social solution. Over there, she has met new friends and she is going to social events but sometimes she feels depressed and stays at home alone. The solution detects that she remains home for long periods of time. Therefore suggests activities of her interest around the city such as temporary exhibitions in museums, theatre plays, dressmaking workshops, and promotes speaking with her friends through the chat feature of the solution.
INVOLVED USERS	Elderly person: +65 years old, with CFS level between 1 and 5, without severe health conditions, users coming from both private and public sector, non-dependent elderly people, living alone or spending a considerable part of the day alone at home; excluded persons with mental disease or cognitive impairment that are not able to sign an informed consent. Service provider / Caregiver: Assistance service provider that supervise daily activity trends of the elderly and intervene if needed through notifications, reminders and alerts.
GOALS TO BE ACHIEVED SYSTEM CAPABILITIES	Elderly : Reduction of loneliness and isolation Service provider / Caregiver : Remote monitoring and trend analysis Home sensors and smartphone applications to detect activities and user behaviour. Gateway to collect the information and send it to IoT platform. Processing module for activity recognition and behaviour detection. Application for the intervention delivery to the elderly.

5.3.2.2. ADAPTATION

For the Adaptation phase the key activity was to define a series of services, according to the identified user needs, and use cases. The following is a summary of all services identified:

Social Tracking , analyses some social interactions and activities the user has.

This information is used for evaluating isolation factors, or behavioural analysis. The service promotes social activities by inferring and learning those activities for which the user is more engagement.

Physical Activity Tracking , analyses the level of physical activity for each user, tracks its evolution over time, and provides context to the activities (for example the user might not done too much exercise because of weather or vacations). The service will encourage users to engage in their preferred activities, which are calibrated to the user's physical level.

Behavioural Analysis , behavioural patterns are recorded, so that changes in these patterns can be analysed. These changes might indicate minor or mayor health, or social issues like cognitive degradation or depression. Behavioural patterns can also be used to determine the best way and time to perform interventions.

Fall Risk Assessment , analyses the data from different sources, including the balance factors, level of physical activity as well as other factors, like depression, which may influence the probability of falling.

Isolation Risk Assessment , analyses the engagement of the user with their close social environment as well as their ability to increase their social circle. Early detection of isolation risks helps perform early intervention avoiding long term impact of isolation, as well as recurrent harmful patterns. Other than promoting social events, the service may include scheduling interviews with social workers or persuade the family to drop a visit.

Outdoor Tracker , through outdoor positioning systems, the service will analyse mobility patters, particularly walking patterns, any missed stop using public transportation, and erratic navigation (both walking and driving). This information might be useful to profile the user's abilities outdoors. Assistance will be provided, when detected and/or solicited, by sending a social worker or an informal carer to look for the user in distress.

Intervention , interaction with the user is unified under one service, helping the user stablsh familiarity and trust in the system. There are many intervention channels (such as telephone calls, SMS, LED light notification) and strategies.

For all these services to be provided and taking into account UCs, the system is being designed accordingly. The following are summaries of the SDS model used for the system development.

5.3.2.2.1. SEMANTIC MODEL The ontologies used are based on the ACTIVAGE Data model package, which includes many ontologies for AHA domain. The semantic model also relies heavily on the universAAL IoT Semantic model, some ontologies provided by this platform help define data models for user, locations, and devices.

Finally the semantic model is combined through a component called Inter Platform Semantic Mediator (IPSM), offered by the ACTIVAGE IoT Ecosystem Suite (AIOTES). Which through defined alignments, it can transform semantic streams of data from one data model to another. Alignments are transformation rules, where matched instances will be analysed, data extracted, and new instances will be created using the new structures defined.

5.3.2.2.2. CONTEXT MODEL Managed data can be categorized into 3 classes:

1. Health data comprises the information about heart rate, blood pressure, balance factor among other.
2. Activity data is composed of application usage information, sleep cycles, and number of steps.
3. Location data, which does not need to be very granular, or precise, is gathered through the smartphone's integrated GPS, proximity to BLE beacons, or connectivity to a set of Wi-Fi signals.

5.3.2.2.3. SERVICE MODEL The main service model used is that of a microservice architecture. Both specific services developed to implement those described in the previous section, as well as services provided by the AIOTES are web services.

In the microservice architecture, the context model is included as a service through the universAAL IoT REST API manager component, giving members the capability of publishing events.

5.3.2.2.4. USER INTERACTION MODEL The user interaction flows have been analysed for both end users and formal carers, for the Android and Dashboard applications respectively. Standard accessibility and user experience philosophies were applied. The implementation of UIs focused on providing users with a common interface for all applications. Users can access the applications through the MAHA app an application launcher, which could also be configured in kiosk mode for provided tables. As for the Dashboard Angular JS was used for a responsive, web driven client.

5.3.2.2.5. MEMBERS AND COMPONENTS The member model is based on both universAAL IoT, for the usage of the context bus as the main way to transmit information and the microservice architecture for most of the cloud operations.

One notable member is that of the historical service, which on one side accepts traditional web service calls, and on the other redirects and transforms these calls to the CHe component running the univiersAAL IoT cloud instance, which is the component holding all historical device information.

Components for each of the services were developed, some components were shared, such as the fore mentioned historical service component. Further analysis on the implementation determined other components were needed for the facilitation of the implementation of components and services, thus the developer was included as participant in the system, see figure 5.29.

Figure 5.29: Conceptual Architecture of the Madrid Deployment site with components in in each distribution scenario

5.3.2.2.6. DISTRIBUTION MODEL The key feature of the designed Architecture is the distribution, partly because of the need to deal with the mobility of the user. Thus the system is designed to handle these three user spaces:

- Home, through a set top box based on the Raspberry Pi.
- Mobile, through an android device, carried by the user. In this scenario, the mobile phone can connect to either the home system or the ACTIVAGE center system enacting fully fledged roaming capabilities, and being itself a service (i.e. the ACTIVAGE Center can use the presence of the Mobile as an authentication device for the user).
- ACTIVAGE Center, information kiosks using a single PC, users will be able to use it to receive particular services.

These three scenarios will be supported by a cloud node, to which all other nodes will connect to, sending the data for further analysis.

5.3.2.2.7. DEPLOYMENT MODEL The deployment model is based on the traditional IoT deployment model.

For gateways Andorid (for Mobile scenario), and universAAL IoT (for home and ACTIVAGE Center scenarios) where used as main component containers. An Android apk is manually installed in the target gateways, this is costly in human resources and in the future components will be provided to be able to automatically install and update the services on an Android device. Preconfigured packages including the universAAL IoT and all of the needed applications was used for the Home and ACTIVAGE center scenarios. Although universAAL IoT recommends the Apache ACE component for OSGi deployments and updates, this component is now deprecated; on the other hand deployments on these scenarios have not been as many to invest in an update infrastructure.

The cloud distribution scenario is based on docker as main container management. Each service is hosted in its own docker container, which means the DevOps deployment strategy is used for most of the services. Through Continuous Integration and Continuous Deployment, the services where compiled, tested, and if all was correct deployed. The blue/green deployment strategy was the most prominent used, with 3 phases: development (automatically deployed, and used for testing), pre-production (manually deployed and used as final verification), and production (used for providing services to the users).

5.3.2.2.8. SECURITY MODEL Using the unviuersAAL IoT platform for much of the system, some of the security features are used. In particular those security features provided by the REST API manager.

For the other microservices an auto-proxy solution is used, which creates valid SSL certificates using Letsencrypt service, for each endpoint.

An OpenLDAP service deployed in the cloud is used as authentication server for all nodes and services, for both end users and formal carers.

5.3.2.3. ASSESSMENT

ACTIVAGE as a whole is very oriented to asses the potential of IoT for AHA domain. This evaluation is main part of the work plan, shown by the fact that there are User Requirement assessment; technical validation of the AIOTES; specific evaluation collection and analysis tools; an evaluation framework for local, global, and a combination of both factors; Pilot and service assessment; Quality of Life assessment; among many other actions with in the project.

With all this support, the assessment in Madrid Deployment Site can be considered continuous. As part of this assessment, measurement of the engagement of the users with the different services through the IoT collected data will be the key aspect. In fact most of the data collected can derive this engagement, as well

as there being explicit data collected regarding the usage of each application. Finally usability, usefulness and ease of use will be assessed with a selection of users completing specific questionnaires.

Overall a set of Key Performance Indicators have been defined. These include Quality of Life, Economic Acceptability and usability indicators in order to provide policy makers and other stakeholders evidence to base any decision making process. These indicators will also demonstrate the accomplishment of the key objectives of the Madrid Deployment Site:

- Empowerment of users to live independently.
- Improvement of quality of life of the users of the IoT solutions and their informal cares.
- Support formal carers in their care of an assisted person.
- Improve efficiency and quality of social services.
- Reduce costs of providing social care services.
- Postpone or prevent institutionalization.
- Reduce opportunity costs (when e.g. a caregiver can work instead of caring for a relative).
- Increase the number of IoT services.

5.4. BRINGING TECHNOLOGY BEYOND THE STATE OF THE ART

This section corresponds to the accomplishment of the objective 5 (see 1.5). We will examine developments which had to occur in order to fulfil all the requirements for development of any SDS under the universAAL IoT Platform. The universAAL IoT Platform was selected as the SDS platform because not only complies with most of the model defined in 4.1, but also because it is focused on the AAL domain. However the universAAL IoT Platform was missing some key components in order to enable part, if not all of the features expected in the SDS models. Among the features it needed to incorporate are: a multi-cloud and multi-tenant connection system, a multi-platform REST enabler, automatic representation of ontologies; it also needed other improvements in its UI framework as well as in its security.

5.4.1. UNIVERSAAL IOT ENHANCEMENT WITH MULTI-CLOUD AND MULTI-TENANCY

In universAAL IoT Service calls, UI Requests and context events are the messages that are exchanged among components. Furthermore, the platform performs a

brokerage between members, called in universAAL bus members i.e. service provider and consumer; UI caller and handler; and between context publisher and subscriber. universAAL IoT defines the concept of uSpace as the technological support where the context (as defined in 4.1.2) of user (or users) can be expressed. Logically each bus member is within the same uSpace, however uSpaces may be connected to create greater uSpaces, effectively extending the context across networks. Therefore the communication in universAAL IoT can be described by three different layers of connectivity:

connector layer establishes a connection between different AAL-aware nodes forming a network of nodes, the uSpace. Possible protocols are, for example, TCP, reliable multicast (e.g. JGroups), Advanced Message Queuing Protocol, or Message Queue Telemetry Transport (MQTT).

brokerage layer works on top of the connector layer and realizes the mentioned communication patterns of call-based or event-based communication. Effectively coordinating the Edge network that the connector layer creates.

gateway layer works on top of the brokerage layer and connects different uSpaces, or external (space-less) bus members such as existing web services.

The first two layers are provided by the middleware, and have been thoroughly validated for edge network uSpaces. However when connecting universAAL as a service (see section 5.4.1.1), we are referring to the third layer. This section will center around the design, implementation, usage and validation of the multi-tenant gateway[110]. This enhancement is a direct implementation of the showcase for Advanced Distribution (see 4.4.3).

5.4.1.1. CLOUD SAAS MATURITY MODEL

As discussed in the distribution model (see section 4.1.7), the key difference for services running in the cloud is the capability to be able to handle multi-tenancy, in this case each uSpace represents a tenant. When deploying uSpaces in the traditional universAAL IoT sense, like at home, the system may need to handle any possible multi-user context within the uSpace, however it does not need to be aware of different uSpaces. In contrast, when a service is running in the cloud then multiple uSpaces can connect to that service and it must be assured that information of one space is not leaking into another space, thus maintaining their privacy.

One challenge is to determine whether universAAL IoT can be considered a PaaS with the applications running as SaaS or the platform and applications are bundled into a SaaS. This depends on how the brokerage and communication between different services in the cloud is realized. If the AAL platform is running bundled as a SaaS, the system needs to be instantiated for every tenant which results in high requirements on the infrastructure in the cloud, especially more memory consumption and processing power. The profiling and configuration of the services is an

important aspect to consider; common configuration parameters need to be adapted for every instance which requires appropriate configuration tools, which need to be also developed. This could be considered effectively a level 2 in Microsoft’s maturity model for SaaS (see 4.1.8.1).

Given all the problems a level 2 implementation yields, and the little benefits it produces; a level 3 implementation is targeted, where universAAL IoT is PaaS and any application is SaaS. As Level 3 is required by level 4, this is actually a necessary step to evolve in the maturity scale. Level 3 has an important implication, all applications running on top of the AAL platform need to be tenant-aware. To do this we needed to extend the message format to include information about the scope of the message, this is done trivially as the messages are semantic in nature, adding a couple of properties to the semantic model of the brokerage subsystem.

5.4.1.2. CONSIDERATIONS FOR MULTI-NETWORKS: DEFINING THE SCOPE OF MESSAGES

An uSpace can in theory be connected to an arbitrary number of distinct uSpaces, this can happen recursively. In figure 5.30 the red network is an uSpace in a home that is connected to the network *n1* which is deployed in the cloud. Network *n1* has connections to the networks *n2* and *n3*. Therefore, it is a necessity to add additional information to each member operation as to identify the *target* uSpace. However, this complicates the definition of member operations, also adding dependency to the setup. Each member would need to be configured with the web of networks; making the deployment more complicated and error-prone.

Figure 5.30: Considerations for multi-networks: AAL aware nodes are marked with “u”; the red network (top left) is an AAL Space for a tenant; gray blocks n1, n2, and n3 denote different networks that may also have tenants connected. Credit [110].

In some cases the network web is known, especially when using a private cloud. However in most scenarios the network web is not known and may even be dynamic. If the proposed static configuration is implemented then when a provider in a remote

network ($n1$) decides to add additional networks ($n2, n3$) to provide additional functionality to its clients each member in the whole web of networks needs to be re-configured.

One strategy to simplify this configuration is by omitting any information about the target in absolute terms. In the example above the client uSpaces would only have one connection and is only aware of this one cloud network. This boundary is indicated by a black line in figure 5.30. On the cloud side ($n1$), the information about the *source* is still needed in order to associate responses to the correct tenant. Equally if the communication flow is initiated in the cloud it is needed to have information about the concrete tenant(s) to address. The *scope* of a message is defined as the set of remote uSpaces that are involved in the communication flow, whether they are target or source. When no scope is selected only local members will be addressed (legacy applications do not need to be changed), however when remote members are involved the scope can be set up, either automatically or by the application (which means this application is tenant-aware). Because scopes only make sense to the local uSpace, they can be used in a relative manner, being managed at the local uSpace without risk of collision or need for update when the network web changes, the changes are only local.

5.4.1.3. COMMUNICATION BETWEEN CLOUD SERVICES

Tenant-aware applications will need to know which is the source of the communication flow, as to answer correctly to that request. They will also need to address specific scopes when the need to initiate communication flows with explicit uSpaces.

Figure 5.31 shows an example of a communication flow: a service in the red uSpace (App1) calls a cloud service (App2, e.g. an Agenda service). App2 needs access to a database that is provided by a second cloud service (App3, e.g. CHE). Neither the calling of App3 nor any responses App3 would make must be disclosed to the blue uSpace which is not involved in the desired communication flow.

Figure 5.31: Communication flow between services running in the cloud. Credit [110].

There are different strategies to avoid leaks. Multiple instances of a component could be deployed with each instance configured to a specific scope (read level 2

maturity level in section 4.1.8.1). Another possibility would be to handle messages in sequences: a service could only handle a message if the previous message is fully processed. The broker can ensure that a members can only use the scope that it was invoked with. However, a consecutive processing could have a serious impact on the overall performance compared to a parallel processing.

The services can have a wide range of security restrictions (see 4.1.9). Similarly to allowing a service to perform operations like access to databases, sockets, or files; the platform cannot fully guarantee that services are isolated and that data is not disclosed between uSpaces. In these cases Security is, to a certain degree, delegated to the application level. The handling of multiple tenants should be supported by the platform and the interfaces should make this handling as easy as possible for services.

5.4.1.4. VERIFICATION FOR THE MULTI-TENANT GATEWAY

The implementation of the multi tenant gateway was a proof of concept developed as part of the ReAAL efforts to make universAAL IoT more cloud compliant. It underwent a battery of tests to ensure the specifications were implemented correctly, the most important the privacy protection of scopes. For a detailed case of the verification of the multi tenant gateway refer to [110], where a scenario with different uSpaces (2 at home and one in the cloud) are set up and tested specifically for message handling and privacy enforcement.

The multi tenant gateway was developed as part of universAAL IoT release 3.2.0 (October 2014), and it has ever since been an integral part of the universAAL IoT core component set at the disposal of system developers, thus it has been proven to be production ready.

5.4.2. UNIVERSAAL IOT ENHANCEMENT WITH MULTI-PLATFORM CAPABILITIES, THE REST API

Alternatively to the local brokerage performed by the multi-tenant gateway, where local proxies are created for remote functionality and only if the local brokerage finds a match, the message is then sent to the remote site; there is the option to consider performing all the brokerage remotely. In remote brokerage all messages (service requests, context events and UI requests) are sent to the remote universAAL IoT network and are evaluated on the remote uSpace.

In any case when a message is received by the remote gateway, a scope is added to the message, with the scope being the ID of the uSpace of the sender. This scope is forwarded to all messages along the communication flow. When a message is received by the cloud gateway from within the cloud, the scope is removed and the message is forwarded to the respective tenant uSpace, removing the need of the tenant to handle any scope.

Remote brokerage does not need any local component to perform matchmaking, and the connectivity may be managed by a simple client (i.e. the AAL Platform

does not even have to be running locally), the main drawback is that it will demand higher bandwidth as every message needs to be sent to the remote gateway. This gateway is the component called REST API manager, which is in charge of managing client connections, receiving bus messages and delivering them to the universAAL IoT buses in the server node (see figure 5.32). The REST API manager is also capable of establishing reverse communication (for context subscription and service callee), for this it uses callback registration mechanisms. This requires clients to offer those web services, and the networking to allow the server to connect to them, requiring firewalls to be configured to allow it.

Figure 5.32: The concept of the REST manager is to offer a REST based member model access to the universAAL IoT Buses.

The REST API manager, as its name implies uses the Representational State Transfer (REST) architecture, popular with web services and increasingly popular for micro services. REST architecture relies on the http protocol methods/verbs, such as the standard GET, POST, DELETE, and PUT; together with a simple state representation, typically this is the Java Script Object Notation (JSON) or XML representations, which have many libraries in many programming languages. Effectively REST methodology makes syntactical interoperability based on simple verbs and simple notation of the information; so that any program with an http client and the pertinent serializer (transforming internal representation to the standard text representation) can create producer members of context, services or UI through the REST API. For consumer members, an http server listening to the appropriate port and http path, together with the appropriate deserializer (inverse of serializer) are required. Essentially this means that components can simply create their members regardless of the platform, programming language or operating system they are using. It is important to note that the REST philosophy applies to the members and not to the exchanged messages, these are delivered to the members (REST or universAAL IoT modules).

This manager is developed in response to the need to better implement Integration with legacy systems (See section 4.4.7), as the most popular inter process communication method currently is through web services, in concrete REST services as mentioned. The fact that universAAL IoT Native components can now interact with existing and future REST enabled components means that on one side the Service Interoperability vision is enlarged (see section 4.4.8), as well as a new potential routes for Evolution (see section 4.4.5). Non-universAAL IoT applications can now connect to a universAAL IoT hub, and access all the semantic interoperability features of universAAL IoT middleware, platform components, effectively offering the whole platform as a service, perfectly fitting for microservice architectures; in this context actually native universAAL IoT applications are completely optional, all components could connect exclusively through the REST member model(see figure 5.33).

Figure 5.33: The REST Manager is the main component which converts universAAL IoT into a full fledge PaaS for microservice architecture.

The REST web services the REST API manager exposes are divided in two categories space operations and member operations. With space operations clients may create custom spaces, or read the existing spaces. They can also declare members associated to any existing space. Member operations then allow members to operate (however, less RESTfull), for context publishers these operations are publishing events, through an http POST of the serialized context event; for the service caller, a serialized service request is posted. For consumer members, the operations are defining the callbacks, where the members can be contacted by the server, and the semantics of the member (subscription patterns, or service profiles).

The default serializer of universAAL IoT platform is for RDF/turtle or Terse RDF Triple Language. This notation is minimalistic, reducing the number of characters required to represent any RDF object; human readable; and standard, as it is used in the semantic domain extensively in documentation and in SPARQL for example. However this means that the clients of the REST API manager component need this (de)serializer, along with the JSON or XML used for the state representation of the space and member operations. JSON-LD (JSON for Linked Data) is another RDF serialization. It was designed to require as little effort as possible to transform their existing JSON to JSON-LD. In fact any JSON object can be transformed into a RDF/OWL representation just by adding a special property called context (defined as the semantic context of the communication the JSON document

is being used). Which means that JSON-LD is JSON compliant, and therefore any JSON (de)serializer will work.

Thus the JSON-LD serializer component was developed, this component not only added JSON-LD as a native syntax to universAAL IoT, for the REST API manager it means that it can now operate 100% in JSON. Because normal JSON can be almost trivially transformed into JSON-LD, and thus into RDF, now all clients can use this syntax to access semantic interoperability through the REST API manager and the universAAL IoT brokers. The component also opens new distribution models for SDSs, the same ones that the multi-tenant gateway has, but over standard protocols.

From a security standpoint, the use of standard protocols means that the communications of the REST API manager can be encrypted using Transport Layer Security (TLS). This not only ensures communication is confidential between client and server endpoints, it also opens the possibility for complex authentication and authorisation mechanisms through third party libraries and frameworks. Although something that the REST API manager needs to secure, which can not be secured using this framework, are the space operations. When a user logs in, the spaces they can access needs to be filtered (currently only created spaces through REST API would be available), the problem is that universAAL IoT currently does not have any supporting component managing this level of access from the different tenants and users. Arguably this is the next step to improve the tenant manager, the internal component shared by the multi-tenant gateway, the REST API manager and the brokers; to include access control of users to the different spaces. Member operations would also need to be protected, ensuring that a client, even with the right credentials, may not access information it is not authorised; however this will require an overhaul of the security mechanisms of universAAL IoT which manage components and the allowed operations by their members (currently only available through the Java-OSGi interface through the functional manifest tool).

5.4.3. UNIVERSAAL IOT ENHANCEMENT WITH SEMANTIC AND DISTRIBUTED CONFIGURATION FRAMEWORK

One of the key aspects when deploying universAAL IoT, and any of its components is configuration. As part of the deployment model, a good configuration tool is required. universAAL IoT is mostly containerized using the OSGi framework, and as such some of its components use the configuration service of the OSGi framework. This works fine for local instance management, but when the whole uSpace needs to be configured it may be more complex. Even more complex is the management of remote uSpaces in the context of many tenants, where there should be capability to easily batch configure all of them regardless of the numbers.

The configuration framework is an enhancement addressing precisely the deployment model issue. This framework allows for semantic definition of configuration,

in a similar way in which the UI Model extends the existing semantic model for the application, the configuration framework enables the usage of existing models to provide parametric configurations. The framework is also distributed, targeted for uSpace management each node is equipped with an instance of the configuration manager, as part of the core components of the middleware. The configuration manager connects to the control bus, a special kind of broker for deploy model management (other managers coordinate actions such as distributed installation of modules and connectivity), to handle specific requests for configuration.

Configuration can either be parametric, where a parameter is set to the particularly desired value, file for modules who need their own standard configuration, or dereferenced configuration, where the configuration itself (either parameter or file) is actually referenced through a URL. Configuration has a coordination identification system, where it both allows for configuration to be shared across the whole uSpace, or to individualise the configuration for each deployment of the components in each node. In fact this reference has 2 distinct levels, one meant for a component as a whole (distributed, or local), and the second for modules within each component, this would help manage components in the same way the OSGi configuration factory service does, for cases where the configurations consist on instantiating subcomponents, and configuring each instance independently. This coordinate system is also compatible with remote spaces, admitting uSpace Scope for all of its coordinates. This allows configuration to be retrieved and set not only locally, but remotely, also distributed shared configuration is managed, as well as affecting remote spaces too.

Modules registered within the configuration framework would receive their latest configuration, which might be the default one if there is no previous value. The mechanism used for this is an asynchronous communication channel, which forces components to react dynamically to configuration changes. A special kind of components, called configuration editors can read not only their own configuration, but all the existing configuration, as well as edit this configuration at any moment. This enables the development of tools aiding both manual deployment configuration as well as automatic configuration. Only a command line interface tool, for proving the configuration framework concept is currently developed.

The configuration framework is a direct reaction to the showcase of Administration (see section 4.4.6), as this was one of the interesting values for AAL platforms. The configuration framework can also be considered key for enabling other showcases such as scalability (see section 4.4.4), advanced distribution (see section 4.4.3), content personalization (see section 4.4.12), and service transferability (see section 4.4.10). In fact in the latter showcase, the ability to change the configuration of the components is crucial, e.g. changing the endpoint where the local service is provided.

The configuration framework, although inspired by the OSGi configuration service, it is not compatible with it. This stems from the fact that components are requested to be self configurable, providing at least a default configuration for each registered configuration entity. Some bridge enabling existing OSGi configuration service modules to be configured from the configuration framework would be ideal,

however there will be many pieces required, which would just be better to migrate the feature completely.

5.4.4. UNIVERSAAL IOT ENHANCEMENT WITH AUTOMATIC SEMANTIC REPRESENTATION

universAAL IoT, as many semantic systems, has an internal data representation framework. This framework is unique to universAAL IoT, namely because the platform needs to perform matchmaking over instances and classes. This feature is not typically present on standard semantic data representation libraries, as matchmaking is something that could be performed one level above: the inference level. The other reason universAAL IoT has its own data representation framework is for application developers to directly, and easily access the data and the classes.

This means that for universAAL IoT to represent an ontology one of two things need to happen:

- An OWL ontology, serialized in one of the native syntaxes of universAAL IoT, is loaded into the data representation package.
- A specific Java implementation of the ontology, using the universAAL IoT data representation framework (which is OWL compatible), declares the ontology.

If an OWL ontology is loaded directly, then the knowledge this ontology describes is handled by the matchmaking (and other semantic) operations the universAAL IoT engine performs. However the direct method is only designed for run time, therefore it is not available at programming time, something the other method allows. Having access to classes, instances, properties and so on in programming time is useful for developers, as they can now more easily interact and use the structures the Java defined ontology states.

The problem is that when a standard OWL ontology needs to be incorporated into the platform, then much effort is devoted into developing a Java defined ontology which maps one to one the statements the standard OWL ontology makes, in order to have the benefits of having symbols available at programming time. This severely affect the capabilities of the platform offering Administrative operations value (see section 4.4.6). For this reason a transformation tool is necessary. The Code Generator tool is a tool based on the code generator plug-in which is packed by default with the Protégé¹⁰, the free and open-source OWL ontology editor. The original code generator plug-in created a Java representation of the live ontology in Protégé, however this representation is focused on OWL API¹¹, a common data representation library in Java which is not usable in universAAL IoT.

The Code Generator tool creates code from a given ontology, using a pluggable template system reference, depending on which output may be a project in any programming code language or any text based file structure like SQL statements, or

¹⁰<https://protege.stanford.edu/>

¹¹<http://owlcs.github.io/owlapi/>

framework-specific interfaces. The user just needs to provide an XML file reference, optionally a basic configuration this XML defines, and set the output directory for the generated code.

Figure 5.34: The basic concept behind the Code Generator tool, using a pluggable defined template system it can output code which is tailored specifically to the input ontologies, and some variable values defined in the template system.

The new tool uses OWL API data representation library, as it is also the internal data representation Protégé uses, together with the Velocity template engine ¹². This tool works with a template system composed of an XML file (named template coordinator) and a set of Velocity Template Language (VTL) macro files (code macros). This template coordinator describes how to process the ontology, and any required additional information for generating the code accordingly. The purpose of template system is allow to decide which VTL macro to apply for each aspect of ontology (the whole ontology, each class, each datatype, each object property, each datatype property, or each individual defined), defining each output file for the output project, as seen in figure 5.34.

The Velocity template engine interprets files and executes VTL code where ever available in the file which is replaced by the output, thus VTL macros look very similar to the output other than the VTL controls some aspects. One particularity of Velocity is that it can access Java objects, thus it is possible to access all the OWL API representation within a VTL macro file. The template coordinator defines which ontology definition to feed to the VTL macro context (as a variable) thus the VTL macro, depending on how it was referred in the coordinator file, could access the OWL API representation of the ontology, class, property, datatype, or individual the OWL ontology defines.

Not only OWL API is compatible with most RDF serializations, each representation can be used to access all of the related axioms, annotations and assertions, as well as the related OWL API objects. Thus it is possible for example to create

¹²<http://velocity.apache.org/>

macros representing exclusively classes, and have all classes be structurally similar but each mapped to the OWL definition.

OWL API also offers the possibility to include some inference, which can also be accessed from the VTL macros. This opens an entirely new use case for the tool which is quite interesting in the context of semantic interoperability. VTL Macros, and template coordinators, could be designed with a particular ontology in mind, thus they could react to the processed ontology according to its relation to the referenced ontology. As an example they could create files for individuals which are subclass of a particular class, or make interfaces based on equivalency axioms. Essentially a combination of ontology plus a template system could create entire systems given an ontology extending the first, essentially making coding at semantic level.

The tool is also extended to handle multiple ontologies. Whether explicitly, or through recursive imports ontologies can be loaded and handled one by one using the same template. For the template to be able to handle this there is another level of iteration added: project. This means that macros associated to project level will only be executed once, but they have access to all the common variables of the transformation project as well as the OWL API representation of all ontologies to be processed (in dependency safe order).

Some example templates have been developed for the tool, amongst them a replacement for the original code generator Protégé Plug-in output, making the new tool an abstraction it. The code generator tool can be used as a Protégé Plug-in (used directly within Protégé), or as a Maven plugin (configured as part of a build lifecycle), or as a REST service (used as a set of web services). The new Code Generator tool has a lot of potential to be used by developers not so comfortable with semantic interoperability, transforming abstract concepts to syntactically familiar structures.

Figure 5.35: The detail of the full maven project template system, and how it is used to generate universAAL ontological representation for a single OWL ontology.

Particularly the universAAL IoT Java ontology template is also a template developed for this system. The system is based on the universAAL IoT Java representa-

tion of ontologies which is OWL compatible, thus this template is just transforming from one representation to another. The universAAL IoT Template system is composed by:

- 2 XML coordinators, which essentially makes 2 templates. One is designed to create a full Maven project with the purpose of later editing the output code (shown in figure 5.35). The second is designed to be used in conjunction with the Maven plugin, simplifying output for it and allowing direct translation and compilation as a module in a single command. Both templates will share most of their macros, listed in continuation.
- a Class Macro and an Enumeration Macro. Both class and enumerations are classes, however universAAL IoT does take a different implementation for both, therefore the XML has to detect when to use which.
- Ontology class Macro, corresponds to the universAAL IoT representation of the ontology which includes all of the axioms to be used in the inference processes.
- Factory Macro, is another Java required class to manage the ontology. It is used by the serializers to generate appropriate instances given a class.
- Activator Macro, is the project's main entry point.
- Because some universAAL descriptions are common across different macros, and they are complex to maintain, there is a common macro template that is imported when needed allowing to call specific macros as functions to generate specific structures.
- Other auxiliary project files such as POM Macro and license Macro, which are only used for the full project template and if configured appropriately.

Because the tool automatically generates the Java ontology modules for universAAL IoT given any ontology remote or local OWL file there are many advantages. Standard ontologies can be directly “compiled” for universAAL IoT from their URL reference, and used as any other Java ontology i.e. with programming time symbol access; this also means updates to the ontology are also managed automatically. 90% Ontologies in universAAL can be maintained as OWL file only; currently all ontologies in universAAL IoT are maintained as Java projects, and as part of their lifecycle they are transformed to OWL files to be published; with the tool the OWL file is the primary source and the code is generated from it. Not all Java ontologies can be processed this way as the use of Java for the data representation has been used to embed some ontologies with extra logic, this logic cannot be represented in OWL, and thus this process is not applicable. In any case the capability to directly create ontology components for run environment, or for later addition of extra logic, is enhancing the capability of universAAL to deal with new ontologies and new domains, effectively smoothing the evolution path (see section 4.4.5).

5.4.5. USER EXPERIENCE IMPROVEMENTS UNDER UNIVERSAAL IOT

universAAL IoT platform already had a reliable model for UI, and in fact most of the Advanced Distribution value for AAL platforms (see section 4.4.11) is inspired by universAAL IoT features. However it required some improvements, analysing it against the UI model proposed (see 4.1.4) such as the capability to better handle pull interaction methods, or added semantics for customization of modality rendering.

One key aspect of the redesign required was to assume that the communicating components are in different nodes for all operations. This way the needed network procedures were not hidden and, therefore, possible communication problems could be detected earlier in the design, at the same time optimizing usage of communication resources. Once the distributed procedures were designed, development of local procedures was even more manageable.

Figure 5.36: universAAL User Interaction Framework

The universAAL Framework for User Interaction in Multimedia, Ambient Assisted Living (AAL) Spaces is specified as Publicly Available Specification by the International Electrotechnical Commission (IEC) under the reference IEC/PAS 62883 Ed. 1.0. in the end of 2013. universAAL UI Framework separates the presentation and application layers and thus enables usage of pluggable user interfaces that are completely separated from the underlying applications that are using them. It consists of the following components:

- *UI Bus* provides methods for accessing distributed universAAL instances involved in the process of explicit interaction between the system and its user.

- *UI Model* contains ontological representations of the most common user interaction components i.e. the building blocks for sending messages (UI Requests) to the user and retrieving feedbacks (UI Responses) from the user that are shared among the bus peers. It enables modality-neutral description of user interfaces, inspired by the W3C XForms [92] specification.
- *Recommendations* are used for modality specific information that may be useful when presenting the forms. As the name implies Recommendations may or may not be applied and it is up to the UI Handler to use these recommendations, that if applied will enhance the appearance of the form.
- *UI Caller* is a part of an application that is responsible for explicit interaction with the user. It sends UI Requests to the UI Bus, and manages UI Responses coming from the user via UI Bus.
- *Dialog Manager* is a single instance in the AAL Space, that manages system-wide coordination, prioritization, personalization and adaptation of each user's dialogs.
- *UI Handler* represents forms (coming within UI Requests) to the user and packages responses (within UI Responses) to be delivered back to applications (more specifically to UI Callers). Depending on the functionality it provides, it may or may not use some supporting UI services.
- *Resource server* [122] stores and delivers resources, such as images, videos, internationalization files, etc., which cannot be exchanged via UI Bus, to the UI Handlers. .
- *Supporting UI Services* provide some additional optional services. An example is the UI Internationalization library, which simplifies the internationalization strings management for applications (and UI components sensible to locale, like the Dialog Manager).

Out of which UI Bus, Dialog Manager, Recommendations, and supporting UI services needed to be redesigned from scratch, maintaining, when possible, compatibility with existing UI Callers and UI handlers. Some additional UI Handlers, as well as new concept of UI Handler design, resulting in a library for UI handler building, are also part of the UI bus overhaul.

The usual user interaction flow is as follows (follow figure 5.36): Applications define what they want to present to the user, and what they need in return, in terms of dialogues as defined in the UI model. Once the dialogue is built, it is encapsulated in a UI Request (1) that is sent through the UI Bus by a UI Caller. The Dialog Manager then adds additional (adaptation and personalization) information to the UI Request (2-4) after which the UI Bus routes the message to the best UI Handler (5,6). The best UI Handler is chosen in the UI Strategy within the UI Bus based on several key parameters (e.g. if the UI Handler is physically nearest to the addressed

user, if it is the most suitable for the user's needs, etc.). Receiving UI Handler unwraps the dialogues coming from the applications and presents it to the user using its controlled interaction channel. Once the user provides the required input, this information is injected into the dialogue and sent back to the application via a UI Response (7,8).

5.4.5.1. BUS REDESIGN

The fact that universAAL buses are meant for distributed environment, means that there must be network protocols that support the interaction between peers. As for any universAAL bus the subcomponent in charge of using and responding to the protocol is called the *bus strategy*. The UI Bus strategy is a coordinated, call-based bus strategy, that has, in addition, also been layered. This layering has demonstrated to be not only a simplification of the understanding of the bus strategy as a whole, but indeed an optimal way to implement and maintain it. Furthermore, layered implementation also proved to be very beneficial with independent testing in different phases and bug fixing, as well as potential reutilization of different layers for other buses.

One key aspect of the design in this layered system is the continuous realization of the distributed model (see section 4.1.7). By assuming that the different defined members in the bus are deployed in different nodes, a clear message protocol is needed, as well as the information which needs to be exchanged between each member. Even though they are conceptualized in different nodes, the messages may be distributed within the same node, for example when the Dialog Manager is contacting a UI Caller which happens to be deployed in the same instance as the Coordinator no network message is generated, instead the message is directly delivered to the local member.

5.4.5.1.1. COORDINATION MANAGEMENT

The first level of the layering takes care of the coordination management. It is in charge of announcing, in case of the coordinator; and annotating, in case of the rest of the peers, who is the coordinator. The coordinator information is offered to the upper layers, since they in general have slightly different behaviour, depending on whether they are acting as coordinator or not. In the latter case reaching the coordinator with a message is also a valuable procedure, that is offered too by this layer.

It also implements the consistency strategies regarding the information of who is coordination. In earlier versions of the UI

Figure 5.37: UI Bus Strategy stack, each one has the network message exchange represented between the different members defined by the bus.

Bus strategy the coordination management was done by peers issuing a synchronous request for coordinator, until the coordinator responded. The latest implementation version anticipates the coordinator request, and as soon as the peer joins the AAL Space, the coordinator sends a message notifying the UI uSpace coordinator; thus optimizing the network load.

Maintaining the coordinated bus paradigm is part of the bus consistency strategies: when a peer requests coordination of the bus, and there is already a known coordinator, the second request will be aborted. If multiple coordinator requests happen to be issued at the same time, when there is no known coordinator, a random waiting mechanism is implemented so only one is granted coordinator status.

It also includes states and state transitions for when the coordinator is, for whatever reason, no longer capable of acting as coordinator. Normal peers can become coordinators, when the proper conditions hold true. The coordinator can turn into a normal peer, when those conditions no longer comply. These state transitions are announced via broadcasted events (see figure 5.37) so the bus consistency is maintained in all the peers. This makes automatic deployment model management automatic.

When a peer needs to send a message to the coordinator and the coordinator is not there, the call is suspended until the coordinator is announced. This kind of synchronization is part of the bus consistency forbidding the execution of procedures that require coordinator when no coordinator is available.

This first layer is generic enough to be considered as part of any coordinated bus.

5.4.5.1.2. REGISTRATION MANAGEMENT The second layer manages the information about the registered UI Handlers and their semantic profiles containing appropriateness for certain impairments and modality, location, supported language, context (like location) and other UI Handler-specific descriptions. Since UI Handlers can register and unregister their specific profiles that are used to select the best UI Handler for a given UI Request, this management is critical.

This layer enables the sharing of this information between peers and coordinator. Even though this information is only useful in the coordinator, in order to ensure bus consistency it is stored locally on each node. When the coordinator leaves, each peer automatically aborts the current UI Requests being processed by all the UI Handlers in the peer. When a coordinator is announced again, since it may or may not be the same peer who was previously the coordinator (even if it was the previous coordinator, the registration state will not be maintained), all registered UI Handler profiles are re-sent to the coordinator, without UI Handlers noticing. From the coordinator side, UI Handler Profiles that belong to a leaving peer are removed from the global registry.

This second layer can be considered generic enough to be used by a coordinated bus that additionally requires registration of matchable members and their profiles. Because members and profiles are handled in the most abstract way and seman-

tically, thus the fact that this layer is designed for UI handler registration is seen as a specification of the data model used which extends the abstract concept of member and member profile. The third and fourth layers contain the most abstract procedures of the UI Bus Strategy, and work very tightly together.

5.4.5.1.3. UI HANDLER BROKER Third layer manages procedures of the UI Handlers; devoted exclusively to brokering communication between UI Handlers and the Dialog Manager. Announcing user motivated actions (like when the user is logging in, or when the Form is completed), or pull actions, is a part of the abstraction procedures in this layer that will be used by UI Handlers.

The coordinator logic of this layer includes the capability of analysing the global registry of UI Handlers' Profiles and determining which is the best profile for a given UI Request. Once the best UI Handler is matched, the UI Handler is notified to process the UI Request.

Coordinator has also the power to retrieve a UI Request, in two phases, via the synchronous call called "Cut". First the UI Handler is notified to decommission the current dialogue, then the stored data up to that point is immediately sent back to the coordinator as the call response.

Bus consistency at this layer has several cases to look after. When a UI Handler profile is registered, the UI Bus Strategy will re-calculate the best UI Handler for each running UI Request, in case the new UI Handler profile is better for any of the current UI Request context. When a peer is leaving and there are pending cut responses expected from that peer, remember that cut call is synchronous, the suspended procedure is resumed with empty data. In this case the latest data stored in the UI Handler is lost, then again the chance this case appears is slim, there is a backup in the Dialog Manager logic, and there is very little that can be done to avoid it.

The brokering mechanism uses a weighted scale of, currently 9, parameters to insure the best possible adaptation and decision on the best available UI Handler at any point in time. Adapted UI Request is matched against all UI Handler Profiles on several parameters, given in table 5.14, and the UI Handler with the biggest score is selected as the best. Since different parameters have different meanings and different importance their weights form a geometric sequence with number 2 as a common ratio value (except for Last used UI Handler, which is a special case as it is determined by context awareness).

Weights also form a sum-free sequence meaning that no weight can be represented as a sum of preceding weights. Values were determined empirically having in mind that the importance of the each weight as well as behaviour in different test set-ups.

When there is no matching between UI Request and the UI Handler Profile the default match level failed weight of 0 is returned.

In case of two or more UI Handler profiles scoring the same value, a complementary score is given to the latest used UI Handler profile. This inhibits the UI Requests being directed to different UI Handler profiles randomly, enhancing the overall usability.

Parameter	Parameter name	Weight (hex)	Weight (dec)
1	Addressed user	0x80	128
2	Main modality	0x40	64
3	Alternative modality	0x20	32
4	User location	0x10	16
5	User impairments	0x08	8
6	Dialog privacy	0x04	4
7	Last used UI Handler	0x04	4
8	Dialog language	0x02	2
9	Dialog form	0x01	1

Table 5.14: Matching parameters

5.4.5.1.4. UI CALLER BROKER The upper-most, fourth, layer in the stack is in charge of managing the brokering communications between the UI Callers and the Dialog Manager. The palette of operations required by the UI Caller includes issuing a UI Request, suspending a previously issued UI Request, resuming a previously suspended UI Request, or aborting any issued UI Request (see figure 5.38).

In case of issuing a UI Request, which is an asynchronous call, the expected UI Response will be issued by the Coordinator; therefore the coordinator has to maintain a global list of pending requests and the issuer. In earlier versions the UI Response was send directly from the UI Handler, which not only broke the layered stack of the UI Bus Strategy, but also forced this network message to be broadcasted (with all its performance and security implications); since the UI Handler member's node does not manage the global (nor partial) pending request mapping, and both the UI Caller and the Dialog Manager have to be notified. Aborting is a very similar case, it is also an asynchronous call, there must be a pending abort management, however the response is directly the data.

Bus consistency is also maintained at this layer, when a node leaves the coordinator cancels all the issued UI Requests from any UI Caller in that node. On the node side when the communication with the coordinator is lost the UI Callers are notified that the connection has been lost and therefore their pending UI Requests must be re issued if the interaction with the user is to continue.

5.4.5.2. DIALOG MANAGER REDESIGN

Since the bus is only devoted to distributing the messages and performing the specific actions, personalization and context-awareness has to be performed by another component. This component is the Dialog Manager, it acts as the UI Bus manager, orchestrating the UI Requests. There should only be one instance of the Dialog Manager deployed in one of the nodes in the uSpace, concretely on the UI coordinator node. When the Dialog Manager registers to the UI Bus, this is the indicator that the peer in which the Dialog Manager is running will become the Coordinator. The existence of the Dialog Manager is why the UI Bus has to be a

coordinated bus, the features of this component have to be centralized in a single place in the uSpace.

The Dialog Manager is responsible of generating the general system interaction UI Requests. Enabling the notion of system by providing entry points for different interaction options, and presenting system-specific dialogues. System dialogues include, but are not limited to: the main menu, presenting stored messages, or the necessary interactions to enable switching between dialogues. None of these UI Requests are allowed to be issued by applications, only the Dialog Manager is allowed to present them.

Personalization in HCI domain refers to the optimization of a user interface according to the end user's needs and preferences. In general it can be user-specified or learned by the system. Personalization is a special kind of adaptation of the UI [123]. In scope of the universAAL UI framework it is done in the Dialog Manager where UI Requests coming from the applications are adapted so that they better fit the targeted user's preferences and/or special conditions. User interaction related parameters are stored as a UI Preferences profile so the user (or an automatic reasoner in place of the user) can update them at any time. UI Preferences subprofile, inspired by the standards ETSI ES 202 746 [124] and ISO IEC 24751-2 [125], extends the user profile and is also stored in a centralized Profiling Server where most current information is maintained. Communication between Dialog Manager and Profiling Server takes place via Service Bus. Initial initialization of the UI Preferences with stereotype data for each user is done automatically by the system based on the user's role (e.g. Assisted Person or caregiver) so that the update of this information is only voluntary and not obligatory.

Part of the personalization includes also the main menu contents. The main menu is the first conceptual dialogue where the user is able to trigger interactions with specific services. The contents of this menu usually represents the applications installed in the uSpace; although maybe the user, or any other stakeholder (service provider, deployer, system administrator, relative, etc.), does not want to have certain interaction trigger in the main menu of the user. For this reason the main menu is customizable per user.

The Dialog Manager is the main component in the UI Framework to interact with the other buses, especially with the Context Bus. Through the Context Bus the Dialog Manager can track the user's context and adapt the interaction to this context.

For example when the user physically moves, he or she changes their location, and if a tracking system is installed, this change in context is published as a Context Event to the Context Bus; the Dialog Manager gathers this information (as a Context Bus subscriber) and instructs the UI Bus to reallocate the current UI Request.

The Dialog Manager will also track UI specific context, i.e. the pending UI Requests for a given user. There are two types of management of these kind of context: persistent and non-persistent. The Dialog Manager will store persistently messages for the user, if the user decides to keep the messages. The reason to keep

pending UI Request is to enable the user to switch between them.

Another reason for it is to help prioritization of UI Requests, as not all requests have the same importance, and therefore trivial UI Requests must be interrupted and swapped with a UI Request that has more vital information. The Dialog Manager is queried before proceeding with any UI Request, this way Dialog Manager has control over which UI Request is to be shown at any given moment in time, but can also keep track of the context of each user.

The Dialog Manager is also sensitive to privacy of each UI Request, providing extra information for the UI Bus strategy to match the most appropriate UI Handler that is able to maintain the privacy required. This is especially useful when exchanging sensitive information with the user, like passwords or medical information. The privacy context not only relies on the capability of the UI Handlers but also on the people present in the uSpace, if the end user is alone (and the uSpace is capable of detecting this context) then the Dialog Manager can treat the privacy of the UI Request as any request with respect to privacy. This tracking is done thanks to the session manager component as part of the security enhancements, see section 5.4.6.

After analysis it was determined that most of the Dialog Manager attention is devoted to each user in an isolated manner. It made sense to design it in a way that there would be a separate instance per user, following a level 2 maturity model level (see 4.1.8.1). This way every process is user independent and there is no need to filter between users or user data. Also it makes personalization of general interaction processes possible, since different parts of the interface can be plugged in and selected by the user, or selected for their particular capability.

This possibility drove the modularity emphasis on the design of the Dialog Manager and enabled pluggable components not only for different interaction generation possibilities, but for internal procedures too. One of the most prominent is the possibility of performing different adaptations, each adaptation can be defined as a module making sure the Dialog Manager is ready for future information available in the user profile or in the uSpace context; as well as cherry picking the adaptations to be performed per user. The mechanism is so simple that usually plugging components can be combined to make more complex procedures.

One example is the previous main menu provision, there are basically two ways of storing the main menu per user; one is in simple text files another being stored as part of the user profile. Each method is implemented in separate modules, but both can be combined by an aggregator module that makes sure both (or any number of main menu providers) are executed when needed. In addition the aggregation module can be extended to also provide a search box that will filter main menu items from the result of its children providers.

The specialization of per user Dialog Manager, means that each user may have their own Dialog Manager running in any node in the uSpace. This reveals an interesting perspective for the possibility of de-centralising the UI bus and converting it in a non-coordinated bus. The UI Bus strategies can, instead of tracking a single SaaS maturity level 3 (see 4.1.8.1) Dialog Manager, which then manages a

level 2 User Dialog Manager; become a level 4 balancer issuing the messages to the specialized level 2 User Dialog Managers (which can effectively be level 1 as the implementation might be different as needed). These independent User Dialog Managers could have their own deployment model, having backups as needed, or being deployed on the cloud, edge or on mobile devices; however special emphasis must be set to the dialogue synchronization between them as to ensure a dialogue consistency. This has been partly managed by the current implementation of the Dialog Manager, as it can store the complete dialogues states as well as backup protocol between competing UI coordinators. It is precisely this feature that will enable the UI bus of universAAL IoT, and any application using it, to be natively fully compliant with GPII[51], GPII would feed the personal Dialog Manager with the preferences configured for the user in the GPII framework. In turn the personal Dialog Manager could learn common adaptations, which could be used to refine the personalization model in the infrastructure, using this infrastructure as one method to implement the ubiquitous deployment model.

5.4.5.3. APPLICATION PERSPECTIVE

Application developers have a very simplified view of the workings of the UI Bus. Through their UI Caller member, they are able to send UI Request, suspend, resume and abort (see figure 5.38). Because all dialogues (inside UI Request) are defined using declarative interaction, user data (the input

Figure 5.38: Lifecycle of UI Requests

to the proposed form) will be sent back asynchronously to the application's UICaller through a call back, in the semantic data structure the application used for the declaration. This process is independent on whether the dialog was ended by the user or aborted by the user, application, or, in extreme cases, the framework itself. This lifecycle resembles that of a call-based bus, in the sense that when a UI Request is sent is a petition to the UI framework to display it to the user (when, where, and how applicable, see figure 5.36 step 1) the result of this petition is a UI Response (in the most common use case, see figure 5.36 step 8) that is presented to the applicant in the form of an asynchronous callback.

5.4.5.3.1. RECOMMENDATIONS All of the concepts in the UI model are modality free, i.e. they are not restricted to any concrete modality. This enables the UI Framework to present the information in any modality, in a system-wise consistent way. But this forces application developers to think in a modality free manner, which is not frequently what they are used to.

There are some instances where the modality specific definitions have important information that add sense and semantic to the final result. Most of these cases come down to decisions that the UI Handlers have to make automatically without many cues, resulting in poor results. For this reason an ontological extension of the UI

model was added, Recommendations are modality specific cues that can be added to any element in the UI model. This extra information can be optionally interpreted by UI Handlers to enhance the results, or make the results more compliant with classical application UI design. As its name implies recommendations may not be interpreted, either because the recommendations available refer to a modality that the UI Handler is not managing, or because the handler itself does not take them into account, or ignores it for that specific case. Therefore the general idea of modality free UI design is maintained, but some degree of application side customization is allowed to enable better results.

The implemented Recommendations are mostly around graphical modality, allowing layout operations, size definition, and alignment.

5.4.5.4. DEVELOPMENT OF UI HANDLERS

UI Handlers are the components responsible for the transformation from modality free representation to an actual modal rendering as well as the actual interaction with the user. Depending on the communication I/O channel they cover specific modality and offer possibility to the end user to choose between alternatives for the same modality, thus realizing the paradigm of “pluggability”.

Different UI Handlers support different devices, modalities, modality-specific customizations, context-aware adaptations, etc. in order to better fit users with different needs and access impairments. It is not only possible but also welcomed that 3rd party UI Handlers are designed and developed by the experts in the field of usability, design and user experience. UI Handlers participate in the last chain of the presentation, thus some adaptations have to be handled by them, following the directives issued by the Dialog Manager, the declarative forms issued by the UI Caller as well as the potential recommendations. The currently available UI Handlers are offered as an example of such development.

Swing UI Handler [126, 47] is highly configurable, allowing different look and feels (L&Fs) to be loaded. These L&Fs are designed to work with the accessibility framework of Java, different screen resolutions (and relations), and be suitable both for touch screens and mouse operated terminals.

HTML5 UI Handler The particularity of a web based UI Handler is that its main use is to allow remote access and interaction with the uSpace. The architecture of this UI Handler is largely based on the architecture of the Swing based UI Handler; although the structure was adapted due to the web interaction paradigm following a pull approach as opposed to the push style the whole UI framework has.

Another important part of remote interaction is the special care that requires security, and authentication in particular. Not only the UI Handler must be able to recognise the user remotely (in our implementation we used simple username-password approach, but any credential exchange may be used); but it also has to recognise when the user has logged off.

Gesture UI Handler [127] uses hand gesture ontology defining 14 different hand gestures that are able to be used during the interaction using this UI Handler which manages a Microsoft Kinect. Modular design of the UI Handlers ensures easy update of any component allowing the usage of alternative devices for gesture recognition as well as arbitrary mapping of the gestures to desired actions.

Voice UI Handler Even though not currently provided, there were plans for the very begging of the universAAL project to do so. Actually the best way for developers to design their modality free user interfaces is to think *“How will the voice UI Handler read this?”*.

5.4.6. UNIVERSAAL IOT IMPROVEMENTS IN SECURITY AND PRIVACY

The universAAL IoT platform has always been a bit weak on security, mainly trusting the distribution being an edge topology and the deployments on this uSpace through functional manifests; a feature which ensured components only had members declared in their manifest, these being accepted by the user. But this falls short of the security and privacy value required from AAL platforms (see section 4.4.9) thus a clear improvement on security aspects was needed.

5.4.6.1. IMPROVEMENTS SECURITY

First all of the other components described in this section have had a security review during their design, in this subsection the results of this review is highlighted.

The Multi-Tenant gateway can perform a filtering of messages and these filters can be configured for the gateway component on both sides of the communication - the home and remote uSpaces. The filter can determine whether a proxy needs to be created in the remote site for service providers and context consumers, and whether a service call or context event needs to be sent or considered when received. These operations can be used to enhance privacy (as they check the content and semantics of the content of the message), security, and efficiency by restricting the set of services that are made available on the remote site. Currently the configuration of these filters is done as part of the connection configuration, and it may not be trivial for deployers, thus further tools and improvements are required, which could allow in the end the control of the incoming and outgoing data directly by the user. In fact one possibility is to extend the functionality of functional manifests to uSpaces, thus not only authorising components and their members, but also defining which kind of members and messages will need to transit inter-uSpaces.

The security of the REST API manager is minimal, in the sense that the only access protected is the web service endpoint itself. This only provides authorisation, using simple http authorisation which is managed internally by the component. One improvement definitely is to connect the component to the authentication framework

existing within universAAL IoT. As discussed above, another security improvement would be to manage the authorisation and access rights each user has to the different operations, this could be performed through the new authorisation mechanism (see 5.4.6.3). Yet another improvement, particularly for privacy, is to be able to define filters defining which kind of members and messages are allowed to enter and leave the uSpace, similarly to the multi-tenant gateway filters, and not only per uSpace but globally too. However all these improvements would require a managing tool to make it easier for components, administrators and even users manage these options.

As for the configuration framework, configuration is one of the most critical aspects in terms of security and privacy. Thus it is important that the configuration received, particularly remote configuration, can be trusted. For this the configuration framework will have to rely on the distribution security provided by other aspects of the universAAL IoT middleware. There is also the issue of the management of trusted configuration editors in the uSpace. One way to address this would be to make use of the functional manifest concept, extending it to the configuration framework. This concept could be very interesting as there could be editors only allowed to configure sub sets of the configuration entities.

Finally the UI framework overhaul caused not only changes in the bus in order to improve privacy and security, but also the addition of new components specifically devoted to security aspects of the whole platform, as well as new context models for managing privacy. The authentication message exchange is now not managed by the Dialog Manager, the authentication is responsibility of the UI Handler, after all it is the UI Handler the component which controls the direct channel to the user, and as such should better handle authentication. This fact opens the possibility for Multi-Factor Authentication(MFA), in fact the framework now allows for pluggable authenticators the combination of which enables MFA. Currently only authenticators for user password are provided (one using the user's security profile as back end, which salted password protection, another used only for testing purposes as it authenticates any credentials as valid).

The most important privacy concern the UI framework has to deal with is the case of multiple users. When private messages are sent to a particular user, the selection of the UI handler needs to consider the context where there may be more than one user in the room with the target user. The concept of spatial session has been created, similarly to device session, when a user authenticates to the uSpace it initially claims the whole uSpace, and this is always true for mono uSpace users. However if the uSpace has the capability of tracking users, then the session can morph to the most wide subspace in which the user is alone. For example, if the user is alone at home, the whole home is considered as the spatial session for that user, even if the user is only in the living room; when a new user is detected to enter the uSpace, but not the living room (maybe the new user goes to the kitchen), then the session of the original user is changed to occupy the living room. If the new user goes to the living room, the original user's session of the living room is temporarily disabled, relying only on the device level sessions the user may have opened (the user has logged in into devices such as smartphone). All this, and many more casuistic

is managed by the session manager which produces this privacy context from other available context for other components, such as the Dialog Manager, to consider when authorising or dealing with private information.

5.4.6.2. CRYPTOGRAPHIC SERVICES

The basis of digital security is based on encryption. Since universAAL IoT did not have an application-level access to encryption it is necessary to define the ontological description of encryption techniques. The produced ontology is based on other ontology standards in this domain, adapted to universAAL IoT.

The encryption module developed for universAAL IoT provides all the methods for encryption and decryption for some technique instances. Concretely DES, Blowfish for simple block symmetric encryption; MD5, SHA for 1-Way Hash, and RSA for Asymmetric Encryption have been implemented. These services are the implementation of the basic cryptographic services as described by the ontology, see figure 5.39.

Figure 5.39: Cryptographic services added to the universAAL security ontology

Encryption of any resource is performed via a service request of the *Encryption-Service*. This request has as input the *Resource* to be encrypted, the method to be used to encrypt and optionally one or more keys (several keys used for multi destination encryption). The output is an *EncryptedResource* which has the *Cypheredtext*, the original resource is serialised and then encrypted; the result is encoded as a *Base64Binary*. For decryption the request is the same except the *Resource* is an output and the *EncryptedResource* is an input.

Similarly to sign a *Resource*, a service request for the *SignVerifyService* must be issued. This call must at least include one keyring (multiple signatures can be processed at once), and both the method (for asymmetric encryption) and digest method (to generate the signature seed) must be specified. The output will be a *SignedResource*, which specifies the methods used to calculate the signatures and the signatures (in *Base64Binary*). For validating a *SignedResource*, this has to be the input (with all the valid references), as well as a keyring with a public key. The

verification process announces the result as an output in the service *verifyResult* property.

Finally a simple service for digesting resources is offered. The inputs are the digest method to be used, and the Resource itself. The output is calculated serialising the resource and digesting the resulting string to produce a *Base64Binary*.

Currently these basic services are being offered as application level services. Although the best solution would be to restrict the call of these services to the local node, avoiding some security risks (such as sending private keys through the network). One option is to add these services as core services of the universAAL IoT middleware. Another is to extend the Service Bus to restrict the callees members to be in the same node (or defined scopes).

5.4.6.3. AUTHORISATION SERVICES

The authorisation model is in charge of defining and verifying the access rights of users to information. The module allows subsystems to define access restrictions to resources (such as information or services). An access control ontology has been developed, which defines roles of users and access rights to any resource instance which is subclass of Asset. This enables the implementation of a semantic service which can be used to authorise access of users. Additionally this module also checks delegation access, making it an integral security service.

A delegation (virtual) form will be provided to the delegate for ensuring that the requested operations are allowed. These forms will detail exactly the operations being delegated and the restrictions of these operations. Delegation forms will be signed by the user issuing it, thus demonstrating the validity of the form itself. An additional measure should be placed while implementing delegation for delegates to be able to read encrypted documents (when applicable), these documents must be shared with them. Asymmetric encryption may produce multi destination encryption, which allows more than one keyring holder to see what other keyring holder has shared with him/her; a single document may be encrypted so more than one keyring holder has access.

Figure 5.40: update for universAAL's security ontology, regarding authentication and access control. In blue concepts imported from other ontologies.

The update of the universAAL IoT security ontology, which enables the authorisation service, has been developed (figure 5.40). The key concept of this update is the *AccessRight*. *AccessRight* defines whether a specific role may read, change, add and/or remove to a *TypeExpression*. In universAAL IoT a *TypeExpression* is a core concept of the data representation module, mapping the OWL concept of *Class Expression*, with this concept we can define references to any instance of a given class, a set of individual resources, or even resources which fulfil a particular set of rules (*Restrictions*). For the service to be able to be defined the protected access resources have to be subclasses of *Asset*, and the *Class Expressions* are restricted to asset subclasses.

Thus, to use the access control service, the classes, or instances which need to be restricted have to be defined as subclass of *Asset*. At this moment, a default access right can be defined describing the access rights for any user, if no default access right the system will assume the most restrictive access right, i.e. by default no one can access it. The main management of access rights is the concept *Role*, as in the role a user has. *Roles* can be hierarchical, as to ease the configuration of the security access. This allows for example to define a “subscriber role” which has read privileges over certain resources and then an “administrator role” to edit those resources; the “administrator role” will include the “subscriber role”, so users with the “administrator role” will also have read rights as well. In this example, if the “subscriber role” has to be changed then the “administrator role” will automatically inherit these changes.

Roles are associated to a user by means of the user's security sub-profile. A user may be assigned more than one role, so the access rights assigned to the user would be the union of the access rights of all of its roles and the included roles. A specific user may have extended Roles, through *DelegationForms*. *DelegationForm* is a subclass

of *SignedResource*, using the encryption services (described in 5.4.6.2), an authorised user signs the *DelegationForm*. Each form indicates for a user (*Delegate*) the *Roles* she or he can impersonate the *Authoriser*. When checking for authorisation, after the normal *Roles* are checked, and before the extended roles granted by the delegation forms are checked, each *DelegationForm* is validated. First it is checked that the delegate property coincides with the user requesting authorisation, and then the signature is validated against the authoriser user's keyring.

The access control feature is implemented as a service, so that other services and applications may use it to manage and verify access rights. This service is defined on alongside the basic cryptographic services in the security ontology. The authorisation service is simply defined as a universAAL IoT service that authorises an *Asset*. The access mode is indicated through the effect on the property (as defined by the service request). The user to be authorised is the involved user (also defined in the service request) and the call status of the universAAL IoT response indicates the result. *CallStatusSuccess* indicates the user has been authorised, all other status indicates the user was not authorised. Specifically a new *CallStatus* has been introduced in universAAL IoT core for this type of services. This *CallStatus* indicates a denial due to security restrictions; this adds the necessary semantics, meaning the service was matched and acknowledged, but not executed due to security reasons.

5.4.6.4. IDENTITY PROTECTION AND ANONYMIZATION SERVICES

The objective of anonymization is to disable a particular set of actors, for example the service provider, access to the knowledge of any user. Particularly the relationship between the user and user's data is critical. For this, an anonymization strategy is devised. The anonymization strategy is divided into 2 fundamental actions, one including the modelling of user identity and its data; the second would focus on the obfuscation of the relations of the identity and corresponding data.

Through the universAAL IoT profile ontology (one of the core ontologies of universAAL IoT), an ontological individual representing a user is related to its profile. The user's profile is nothing more than a container of subprofiles. Subprofiles are the final containers of the user's data for a particular domain. For example, personal information and health subprofiles that may be related to a user's profile and, thus, to the user is also included.

Additionally, since the 3.3.0 release of the universAAL IoT platform, a new feature was added: the definition of *PrivateResource*, a conceptual abstract class that enables developers to tag ontological concepts and individuals which are sensitive to privacy restrictions, by just making ontology instances or classes a subclass of it. By default all user profiles, and user subprofiles are tagged as private. This tag will help the module automatically identify those pieces of information that should be anonymized and secured.

An important restriction subprofiles must comply is that subprofiles do not contain identifying information. Identifying information is any data from which the

subject can be identified. For example name and surname are identifying information.

Once the data is structured in such a way that the user's identity can be separated from the rest of user data, the next process is to sever the correlation between both, thus keeping the data anonymous (figure 5.41). To achieve this, we designed a service that will cipher ontological properties of any ontological instance, and make them available only to a set of actors.

Figure 5.41: Different views from authorised and unauthorised actors of an anonymized property

The anonymization service is based on the encryption service defined in 5.4.6.2, which enables end-to-end encryption through keyrings (public-private key pairs) and multi destination encryption. The idea is to encrypt the reference of a property, so the system instead of holding the clear URI, it holds an encrypted version. *AnonymisedResources* store the encrypted URI, along with the symmetrical encryption key, but this key is also encrypted using the public keys of the actors that are allowed access to the relation (through the multi destination encryption feature); namely the user him/her self and any set of recipient users. Then, the encryption key may only be retrieved by the holders of the associated private keys, therefore only the holders of these keys may access the actual relation.

As long as the URI of the *Anonymizable* instance, the object referenced by the encrypted property, is random (i.e. it does not contain identifiable information) the object can be stored as any other object, and can be even used by the cloud provider. This usage will be only in an anonymized form, this gives more information than aggregated form, but maintains the anonymity of the data like aggregated form.

The anonymization module offers a service to retrieve an *AnonymousResource*, from an *Anonymizable* object (both concepts provided by the ontology defining the service). Typically the *Anonymizable* object is the user's profile, for which one or more of the references to the subprofiles will be ciphered. Once the device has retrieved this object, it can decipher the references and then query for those (to be used for reading as well as updating). Another service is offered to enable the anonymization of user profiles.

The “anonymize” and “deanonymize” service is defined in a similar manner as the basic encryption service (see 5.4.6.2), since instead of encrypting the whole resource it only encrypts the reference to the resource. The difference is that instead of encrypting/decrypting any resource it is restricted to *Anonymizable* type, also the only properties that can be (de)anonymised are those that are subproperties of the *AnonymousResource*. Thus to use this module, the ontologies have to define which objects and properties can be anonymised.

The service uses a special URI schema, which encodes the fact that it is an anonymized resource, the methods used to encrypt the URI, the encrypted URI and the encrypted session keys for each user who can access the relation. This is very complex; the reason for this is to enable different types of encryption for each property. If *Anonymizable* class would have the same properties as *MultiDestinationEncryptedResource*, as to hold all the information needed for decryption, this would force all the subproperties of *AnonymousResource* must be encrypted using the same process; i.e. all the subprofiles of a user profile would have the same encryption schema and even the same destinations, thus forcing the same actors for all data, and in some cases this might not be desirable (e.g. a doctor may have access to the health data but not the location data).

5.4.6.5. PRIVACY RIGHTS AS A SERVICE

The case of semantic services applicability to privacy is very interesting, abstract semantic services regarding privacy can be defined. In fact these abstract service definition could map a particular privacy regulation such as the GDPR. By doing so, not only the necessary parts of the regulation are modelled, but developers trying to implement compliance with this regulation can better understand in terms of implications, as well as transactions, what each service requires.

This concept gives rise to Right as a Service (RaaS), where each right is defined as a semantic service, which can be enacted by the affected entity. The implementation of these rights is delegated to the application developers. Validating if an application is regulation compliant would be as easy as ensuring each required service (in contrast with optional service) is implemented, potentially individual validation tests can be defined to ensure not only the service is implemented but the service is completely compliant with the specification.

Figure 5.42: Right as a Service architecture, where the implementation is independent for each Service Provider, and application, the service may be syntactically distinct however the semantics for each are univocally defined as a semantic service which can be accessed through a common interface.

The existence of these services also enables the creation of quality of life tools for service providers, data protection officers (DPOs) and system managers as well as end users. As an example, take a GDPR dashboard, this dashboard would display the available operations and rights to the user (see figure 5.42). For end users this means they can manage all their data and GDPR rights centralized, and individualized per application. For DPOs the dashboard would show operations such as sending breach notifications (which would also be enacted through RaaS, thus the concrete implementation will depend on the deployment). This control panel could be extended with other features, such as tools for verification of the running applications as well as security auditing tools.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

This final chapter will review and correlate the different methods laid out throughout the document, providing critical discussion on the overall results and how to improve the methods.

6.1. DISCUSSION ON MODEL FOR SEMANTIC DISTRIBUTED SYSTEMS

Through the participation in the different research projects, the Model for SDS has been refined. Not only have the requirements have grown both in number and complexity, but in abstraction as well. This forces the model to be updated with each iteration to be able to include this evolution.

Additionally the technology has been advancing, as well as the technology culture and consumption. Consumer electronics have been getting more advanced, faster, smarter, more connected, with more storing capabilities. As consequence the way users consume the services through the technology have changed; and expectations of the users have increased with this usage. The expectation is that users will continue to do so, but the argument in favour of SDS is that they can potentially adapt to these changes.

New terms and trends appear, in most cases these concepts are not revolutions, they are evolutions or specifications of existing concepts. For example when the Cloud Computing term appeared, many experts labelled it as "a branding name for Client-Server model", however beyond the branding of the term it self, the term actually emphasises the fact that the server is running on a third party provider; thus distinguishing between the service provider and the technology provider levels. This concept was not possible until the technology allow it, but it does not mean that the underlying model radically changes.

The proposed model has been used for over 5 years, over 5 projects, and it has demonstrated to accurately model the needed systems in different AAL related re-

search initiatives. These initiatives have been based on different distributed models such as Edge computing, cloud computing, IoT, and even blockchain. These initiatives have all been revolving around the AAL domain, whether they implemented pure health systems, such as electronic health records, ambient intelligence, or physical activity services. In each case the model mapped the systems in an abstract semantic and distributed system space of terms and concepts, which in turn makes it easier to communicate, understand, design, interact, maintain and extend the system.

The model helps all stakeholders understand and define SDS. Developers have benefited the most from this model, as they were able to establish a common language to use in the complex realm of SDS. The model helped them also provide easier to understand options to deployers, the people in charge of deploying the system. Other stakeholders, such as Service Providers, were able to synthesise the requirements and negotiate the provisioning of the infrastructure and devices, they were also able to better manage the technology and synchronize it with the business model.

The model also helps view the AAL platforms in a new perspective, identifying the features it needs to provide to enable certain aspects of the model. In this example the Multitenant Gateway is a required feature to enable complex distribution models, because it accepts exactly the same member model as any other universAAL IoT module, it also enhances the distribution model, by adding possibilities of deploying modules in different connected nodes. The REST API Manager, is a new member model enable, coincidentally this member model is the current trend in the industry for Cloud oriented systems, however because it is light weight and bridges with universAAL IoT traditional member model, it can be used in other distribution networks such as edge, to provide multi platform operation to the platform. Traditionally the Semantic Model of universAAL is restricted to the default ontologies provided by the platform, not that they are not AAL complete, but this improvement suddenly makes the Semantic Model of the platform extensible to the limits of OWL defined ontologies, which is not infinite but is vast nevertheless. The configuration framework addresses specifically the problems in the deployment model, particularly those with management of configuration of deployed components; the configuration framework facilitates the initial deployment by providing default working configurations as well as subsequent operations, adding flexibility and automation to configuration operations. In fact the configuration framework can produce new business models, by where the configuration is managed by a specialized third party provider.

Other model based improvements of universAAL IoT are not so grandiose, they look into optimizations of the existing methods, even from a conceptual perspective. Such is the case of the UI Framework, universAAL IoT already had a robust UI Model, however conceptually inherited many restrictions from previous designs and implementations; the UI framework refactoring is the first step to a fully compliant UI Model implementation, but also makes deployments involving the UI framework much more automatic (thus easing related Deployment Model operation). Security

Model is also another improvement over existing secure components, by extending the existing implementations adding the security functions to the semantic model enabling the homogeneous access to these functions from all the components.

The initial iteration of the model was developed for the development of the EvAAL suite. Even in this first iteration the model already provides perspective to the different complexities of a uSpace, pure edge tenant network. The formalization of the model actually enabled the development of the system in under 2 weeks of implementation and testing.

The subsequent iterations of the model made it more abstract, and more mature. Even for systems which apparently are already addressed such as the PASSAGE architecture, where the Semantic Model portrays the typical AAL models for health, location, user, etc. The Distribution Model is very interesting, on a first approach there are multi-tenant edge networks connected to a multi-cloud service provisioning system; however through their combination and the application of virtualization, not only can a tenant space stretch across the internet securely, the multi-tenancy problem can be inverted through the concept of proxied members, an innovation from the Member Model which is implemented through the Distribution Model and enhances the Deployment Model. Many of these functionalities are thanks to the incorporation of the different models as part of the Semantic Model. This allows the context to now incorporate the distribution and deployment model, enabling applications to react accordingly using semantic services added to the Service Model for that effect.

Specifically on the edge section, PASSAGE proposes stronger Context and Service Models through the taxonomy of capabilities. Capabilities not only provide context but also provide common Services in the edge. This conceptualization of functionalities of the components, as part of the semantic model homogenizes the very heterogeneous environment the edge can be, keeping it simple and secure. The Edge section also profits from the UI Model, making more emphasis on the portability concept, enabling users to interact with the system on the go, through smart phones, or wearable gateway, and through the specific platform oriented (open for other applications) home gateway.

The limits of the SDS Model are explored through the exotic architecture proposed by H-Chain. This blockchain federated Peer to Peer system, initially might just seem that the Distribution Model is the most heavy component. However the whole system proposes radical solutions throughout all the models. The system is completely semantic oriented, all the basic concepts of operation are formalized and in particular instances, such as the SSCs they are expressly designed to be extended. The Member model is essentially reduced to the concept of the private space, enabling homogeneous access to all kinds of context and services, both public and private, achieving web 3.0 information exchange ideals. The Component Model incorporates the components as part of the data exchange, particularly for the pHAs. But arguably the strongest is the Security and Privacy Model, as expected from a system devoted to provide secure EHR transfer where the patient/user is the owner of their information, this model guides the rest of the ideation process resulting in a

system which handles the complete security triad, as well as GDPR compliance by default.

When used in conjunction with the Cocreation methodology, the SDS model can be used in all of its three phases. During Analysis, it helps identification and classification of input systems, already providing basis for the adaptation phase; where these systems are formally interconnected and the features identified through the model are implemented; in the assessment phase the model is used as reference for the case evaluations and validations of the system, specifically the technical aspects.

The value oriented methodology, or more concretely the AAL showcases, have been clearly a wealth of inspiration for the SDS Model. Even though the SDS Model might have been used to ideate some AAL showcases, the value extraction of these showcases and the definition of further showcases truly impacts the SDS Model. So yet another use for the Showcase Methodology is that of a validation and improvement of the SDS Model. Evolution is enabled through semantic interoperability, precisely thanks to flexibility in both the semantic model and the member model; however to really take advantage of future developments one must make special emphasis on the Context and Service Models, particularly including possible context and services not yet available. Another showcase enabled specifically through semantic interoperability (once modelled in the Semantic Model) is Plug and Play, but in terms of the SDS model it specifically refers to deployment model, and how components such as devices can be replaced by components developed by another provider. The roadmap to market in AAL is determined by the capability of transferring services, another key showcase, SDS model specifically addresses alternative deployments in the deployment model, however the service and context models should also accommodate to fit different AAL level service provision. Service Integration key showcase is specifically addressed by the Service Model, in fact the service model has adopted the specific key value proposition by expressly exploring alternative service composition, and open marketplace concept of services. Finally the SDS model is a tool which would enable Scalability showcase in all of its levels, from providing redundancy subsystems, to multi-tenancy awareness.

6.2. DISCUSSION ON SHOWCASES

showcases was initially ideated as a value oriented analysis tool, however it has demonstrated to be useful for much more. This fact may on one side be a tribute on the need of such tool in SDS development, on the other hand the interference it produces when using all of the applications might indicate the tool itself misses some specialization, purpose restriction, to be able to focus the attention on the topics at hand.

The proposed Showcase Methodology is quite open, it has been used to provide a complete overview from a value perspective of AAL oriented platforms. SDS is not technically needed to provide this value, however it is best implemented through

the usage of SDS methodologies, such as the SDS model or the Cocreation Methodology. This is revealed by the intrinsic dependencies both the SDS Model and the Cocreation Methodology have with the showcases tools.

Cost effectiveness is a quite abstract concept which may be difficult to predict, specially at the beginning of any project, yet it is the key factor when taking decisions, not only on *how to develop the system*, but even more basic *why should the system (or specific functionalities) be developed at all*. Showcase Methodology, and particularly the existing tool set is a critical exercise to resolve the cost effectiveness issues, as once we access the actual value (by envisioning it or through empirical evidence) it is far easier to balance the effort to be invested.

This was the case in ReAAL, initially the experts ranked the depth of universAALization for 14 applications at level 1 or 2 and 9 applications at level 4 or 5 out of 5. Most of these latter applications were developed universAAL IoT Natively, which explains the high number. Once the cost effectiveness was contrasted through showcases, the first set of applications would invest more effort on improving universAALization depth, which would take more advantage of the interesting showcases universAAL IoT was already offering. As a matter of fact no relationship between universAALization quality (depth, complexity) and universAALization cost was found, which means that cost effectiveness in terms of achieving showcase implementation through SDS has even more potential. The showcases have helped ReAAL participants to articulate and recognize the value of universAAL IoT. Looking back, ReAAL might have benefited from making the showcases explicit from the beginning, and having the showcases guide the project more explicitly.

The Showcase Methodology it self is not influenced by the SDS model, after all the methodology describes a generic tool to extract value from any system. Nevertheless the application of the Showcase Methodology to the AAL domain, defining the AAL showcases, was highly influenced by the SDS model, explicit examples are Advanced Distribution, Advanced User Interaction, Security and Privacy, and even Administration can be directly mapped to their specific SDS Sub-models, which, no surprise, share denominations. These showcases are used as initial value propositions, and improved through the methodological refinement process. Other showcases bisect further specific SDS sub model, such is the case of plug and play and resource sharing; both look partially into the context model, but one focuses on the value of different context providers, and the other values the consumption of common context. The full analysis relating each showcase to all SDS sub models is depicted in table 6.1.

	Semantic Model	Context Model	Service Model	User Interaction Model	Member Model	Component Model	Distribution Model	Deployment Model	Security and Privacy Model
Resource Sharing	↑	←	←	←			←	←	x
Plug & Play	↑	↑	↑	↑	←	←		x	x
Adv. Distribution	x	x	x	x	x	x	←	←	x
Scalability	←	←	←		←	←	↑	↑	x
Evolution	↑	←	←	←	↑	x	←	x	x
Administration	←	←	←	←	←	←	↑	↑	↑
Int. w/ Legacy Sys.	←	←	←	←	↑	x	x	x	x
Services Integration	↑	←	↑	←	←	↑	x	x	x
Security & Privacy	←	←	←	←	↑		↑	←	↑
Service Transferability	↑	↑	↑	↑	↑	←	↑	↑	↑
Adv. User Interaction	←	←	←	↑		←	↑	↑	↑
Personalized Content	↑	↑	↑	↑		←		←	↑
Ambient Intelligence	↑	↑	↑	↑	←		↑	←	x
Enh. Market C & D	↑	↑	↑	←	↑	↑	↑	↑	↑

Table 6.1: Map between AAL platform showcases and SDS model. Up arrow symbolizes the needed involvement of the sub model to implement the given showcase, left arrow symbolizes the sub model may provide support to the showcase, but it is not required, x symbolizes the fact that both showcase and sub model need to take one another into account.

As for the impact of the Cocreation methodology on the Showcase Methodology, it is quite evident the high correlation, specifically on the need to involve all stakeholders in achieving a common objective. The process of definition of the showcases tool-set is very similar to the whole Analysis phase of the cocreation methodology, in particular requirement extraction and negotiation tasks. As for the Value assessment process of the showcase methodology, is comparable the Assessment phase of the cocreation methodology, but highlighting the different contexts and objectives.

6.3. DISCUSSION ON COCREATION METHODOLOGY

The cocreation methodology was ideated to apply cocreation framework on large scale research projects. The methodology allows adaptation of existing solutions to fit particular paradigms, even combining paradigms, optimizing value for all stakeholders. This optimization results in a better implementation of the adapted solution.

The fact that the methodology is divided into 3 phases, means that it can be pipelined, i.e. allowing the execution of various projects simultaneously each on different phases, with interference, regardless of participants or shared technology. The pipeline concept can also be used to apply results from one project to the next, or even plan iterations of the whole process.

The analysis phase lays out the ground work for the rest of the development of the system, especially assessing the viability throughout the earliest stages. Requirement extraction from the users ensures usability and acceptability of the system, but is valuable as a validation tool of the effectiveness of the use cases. In the case of use case definition, it is important to account for reusability of the use cases, which in turn will affect the number of reused components.

A balance between the services and the technology must exist during the adaptation phase. In most cases adapting services is more challenging, especially when these services involve human resources, however when the change is supported by technology, there tends to be less resistance. The technology it self can be adapted, or developed, through the MDS model. Identifying data models, data flows, internal services, user interaction flows, distribution patterns, component and their mechanisms all supervised under strong security and privacy management.

The only way to assure the system is working as intended, before upscaling to production, is through an assessment phase. Evaluation of the system in different axis, or dimensions, is very important; the results of this analysis will determine if the solution has any value, and exactly how much value for the different stakeholders.

The whole objective of the ReAAL project was to adapt existing solutions to create potential for interoperability and, in turn, a full fledge AAL ecosystem of services and technologies, a new economic market across Europe. The adaptation process followed the cocreation framework, learning along the way the best value for the different stakeholders, and particularly proving users access to new services. When an AAL SDS open platform, such as universAAL was proposed as the substrate for all this potential, the appreciation was not immediate. Developers indicated that the development of single applications can be done in less time and better results using alternative material. The return on investment for technology providers in SDS driven solutions was unclear, specially considering the required effort on universAALization, which particularly steep if the number of applications to be adapted was low. For technology providers with more interoperability oriented solutions, or more total number of solutions, the problem was more tangible and the benefits more interesting. All technology providers, thanks to the analysis, and particularly

the showcase value analysis, could appreciate the increment in value when applications were adapted to a SDS ecosystem. This experience highlights the need of the analysis phase, in particular the negotiation task.

Similarly the need for SDS model when adapting an existing solution was not clear. However results show that this adaptation is critical when the services, applications, and use cases need to be extended and made interoperable with other entities in the system as a whole. One particular example of this is internationalization; working together with vendors from other countries has its own challenges and has a high risk of delays. In addition, not all applications are fit to be used in other countries without thorough adaptation. For example, there are national differences in data protection requirements, and in legal classifications of application; and particularly locally provided AAL services which needed to be integrated in the tool. SDS model provides a common framework to be able to create interoperable solutions which can be easily adapted to different environments, legal frameworks, cultures, etc.

When assessing universalized applications, overall developers agree that the solutions performed much better. They see this quality improvement in standards conformance and in openness of architecture as a result of using SDS model, particularly semantics. Showcases helped better understand the value, and thus perceive potential quality of applications. Other developers argue that even if the quality of individual applications did not improve, it impacted the service integration positively. Not all pilots however exclusively report positive experiences, due to new issues arising after lab tests, which also negatively affected the quality of the services, which indicates that pre-production evaluation methods should be more aggressive. Improved application quality impacted the quality of the services positively, as experienced by the service providers and end users. For example, a service provider reported that it is easier to deploy and upscale the service, thanks to the semi-standardized deployment model.

The case of the Madrid Deployment Site of ACTIVAGE, with regards to the Cocreation Methodology is extreme. The local ecosystem is fully engaged, thanks to the involvement in all of the decision process of the services, technologies and use cases. The embrace is so tight the methodology is implemented continuously, in a sense the phases are being adapted to a dev-ops like production cycle; where each cycle the three phases are executed before deploying the next iteration of the system, as small as it may be. In fact because the small increments the pipeline in the whole process allows to examine each idea, feature, solution, individually and simultaneously with other ideas in the pipeline.

The tender is very multi disciplinary, including prominent assistance, medical, technical and business perspectives. This produces very complete results during the analytical phase. As new ideas are proposed, from either of the different disciplines, the team produces coherent analysis on user needs, as well as a selection of cutting edge technological solutions, which when approved move to use case definition and finally a negotiation, all flowing naturally from an appreciation of all participants towards the methodology.

The adaptation phase is directed by the technological partners, and as such it is typically more structured in terms of technology development. Full dev-ops is implemented, so that the validation and assessment phase can flow naturally from the increments. The usage of SDS model, both helps manage the internal adaptation providing interoperability between all the technological components, regardless of the underlying platform. Although the model is also useful in external interoperability, such as the fact that technological solutions from other deployment sites across Europe, and external providers, are integrated into the system.

The assessment phase is highly supported from the ACTIVAGE project, as it has within its objectives demonstrate the positive impact of IoT on AHA, locally and globally across the European deployment sites. In this sense the assessment methodologies, best practices, and results are shared amongst international partners which make the whole process more standardised, robust and effective. In general results of assessment show users engage in more active activities, promoting active and healthy ageing, this means that already active people prolong their active years, as well as activating people with lower activity level. Service providers report that the use of the technology is essential to provide quality service at scale, but also to provide new and innovative solutions to their customers. Finally policy makers react positively, particularly local small town administrations, where the resources to take care of ageing populations are not so extensive as in bigger cities, the kind of services and activities proposed by the deployment site, not only show the town as a modern age-friendly community, they help complement care people are receiving, improving the overall quality of life of the population.

The cocreation methodology is based on fundamentally the same dogma as the Showcase Methodology, that is examine and maximize the value for every stakeholder. In this sense, both having the same conceptual root, interesting concepts can be reused in each others methods. The case for Showcase Methodology influencing the Cocreation method, is strongest when looking in the analysis phase; the whole phase is oriented towards potential value, as seen from different stakeholders, and the analysis of the common value is performed through the negotiation task.

6.4. RESEARCH QUESTIONS

After all the research presented in the previous chapters, it is time to review the research questions asked in section 1.3. In this section the questions will be restated, but this time they will be answered given the results and validations presented.

6.4.1. SEMANTIC AND DISTRIBUTED SYSTEM MODEL QUESTIONS

After developing a Model for SDS, the challenge was then to show it could be used in a plethora of scenarios, specially in AAL and similar domains.

Can this model be used in purely Edge Computing environments?

Yes. The model has been applied to a system devoted to benchmarking indoor locations in a closed Edge network. Including replicability of nodes, and component membership alternatives. The model has been proven to work on Edge-type distribution model, however this exercise also examined the extend of the member model (where competitors had several alternatives to connect their component); the component model (where literally every competitor was presenting a different implementation of the same component); and deploy model (where the same component was deployed in different nodes providing not only redundancy but benchmarking functionality).

Can this model be used for IoT, hybrid Edge-Cloud environments?

Yes. The model has been tested against a distributed edge-cloud concept. PASSAGE has emphasized the need to have powerful home environments capable of running most day to day services without the need to send data to the cloud, how ever when this needs to happen, this still performed in a secure way. The model was stressed particularly on semantic, context, service, and distribution models. The resulting SDS was completely semantic framework, capable of adapting to new data formats through the inclusion of new ontologies. It also had a comprehensive context model, based on the semantics, but also including the whole system (and all of its models) allowing the system to react to different situations, services, components, distributions, and configurations. The SDS also emphasizes how services need to be managed not only independently in Edge and Cloud environment, but in the interoperability space between. The inclusion of virtualized edge spaces, as well as proxied capabilities, opens new possibilities in the distribution model.

Can this model be used in Blockchain based environments?

Yes. The model has been successfully tested against arguably the most exotic SDS concept, which is based on federated P2P distribution model and blockchain member model. Although the innovation, and thus the stress, might initially be though to be on the distribution model, the actual model being most impacted is the security; the whole system is devoted to security and privacy management, and the rest of the piece fall around the functions offered by this model. Built in conjunction with a strong semantic model, which is also extensible enabling the whole SDS to adapt to new applications and domains, the security model defines the concept of Semantic Smart Contract. Around this concept, other models can really emerge, particularly the context model is now a whole new concept based on the set of semantic statements the user can access, which extends to the member model affecting how users' components access the system. From this innovative view of service provision emerges, even considering the constrained environment for services P2P and particularly Blockchain technologies are, where delays may make it impossible to efficiently deliver services.

6.4.2. EVALUATION QUESTIONS

The challenge of examining the value for SDS was tackled through the application of a multi-dimensional value oriented methodology, which would also have to be developed.

What is the technical value and impact of AAL Semantic and Distributed System platforms?

After developing a value oriented multi-dimensional evaluation methodology, it was applied to AAL SDS platforms. This resulted in 14 showcases listing specific values SDS AAL platforms brought to all stakeholders involved in AAL services.

1. Evolving an SDS is something valuable, planning for the future and ensuring the SDS does not have to be replaced means that all stakeholders save costs.
2. Choice is highly valuable in any economy, for SDS stakeholders to be able to choose which components to add to the SDS is important, and impacts not only the costs, but encourages the growth and innovation of component markets.
3. Being able to expand markets is also interesting, successful AAL services can be transferred to new regions (through AAL SDS platforms), not only enabling technology providers to expand markets, but also enabling the demand side to access new markets, encouraging innovation and growth of the AAL market.
4. When services can be combined, this enables new services, new use cases to emerge. This allows for specialization in the AAL domain, breaking verticality of existing solutions.
5. Scaling the solution is crucial to reach more people, and more people in need of AAL services.
6. Privacy must be maintained, specially in the highly digital world that is today. Not only is necessary for legal liability management, it also builds trust in the system.
7. Recycling resources, optimizes the consumption and costs of the system.
8. Systems need to adapt to the users, this is another resource optimization strategy.
9. Having alternative and secure connectivity options not only allow a system to adapt to the environment (not needed a replacement system for this environment), it also provides choice for stakeholders on where and how to provide and consume the different resources.

10. AAL SDS Platforms enable the creation of a market of services and applications, priming the economy for an AAL service revolution akin to the mobile app revolution.
11. Access to simple day to day home automation services is important in AAL, not only these can integrate with the AAL services, they also help reduce energy consumption and avoid risks.
12. By standardising the provision, and general administration of AAL services and application service providers can focus on providing higher quality services at the same cost.
13. It is also important for SDS AAL platforms to interact with existing services, for the same reason the system needs to evolve, it needs to respect the established systems proven to work, and already optimized.

6.4.3. METHODOLOGY QUESTIONS

One of the crucial factors for success of any SDS in the AAL domain is the acceptance, not only by the users, but by all the stakeholders involved. Therefore a Cocreation methodology is proposed, ensuring all Stakeholders, and specially the users, are accounted throughout the SDS development process.

Can this methodology be applied to the development of any generic AAL SDS?

Yes. The cocreation methodology was applied to the development process of AAL SDS, using universAAL IoT as platform, to all of the solutions provided by the large scale project ReAAL. Not only is challenging using the methodology for the development of SDS, but in this case the application was done to all 21 solutions, resulting in a generic best practice and guideline set.

Can this methodology be applied to the development of IoT based AHA solutions?

Yes, The cocreation methodology has successfully applied to the development of a single SDS composed of many services and applications. The methodology allows to coordinate the efforts and commitments of the different stakeholders, while focusing on the benefits for the end users. The methodology is being used so extensively it has been adapted to dev-ops like phasing and pipelining, continuously provisioning new features, services, applications, and value.

6.4.4. TECHNOLOGY QUESTIONS

After the development of the Model for SDS, the technology needed to be adapted, the following questions ensure this adaptation was performed.

How can heterogeneous systems interconnect including all the associated semantics of their data collection, processing and displaying?

By using the REST API Manager any component running in any node, despite the underlying platform or operating system, can connect to an universAAL IoT Instance, interacting with those components as well as other components using the same member access. The semantics are preserved, which means that a minimal set of semantic model needs to be present in the component used connect, the transmission of said model, when the receiving node does not possess it, is handled through JSON-LD context definition. Context transference, and service provision can be now interconnect, however displaying is not currently a capacity the REST API Manager can offer.

How can semantic technologies be enhanced by advanced cloud distributed systems, maintaining privacy between the different user side sub-networks?

Using the Multi-tenant Gateway component, SDS can grow across networks using universAAL IoT native member model (and REST member model). The same component enables cloud spaces to interconnect in multi-cloud configurations, for provision of context and/or services in a secure way by providing encryption, and filtering.

How can AAL semantic and distributed systems include advanced user interaction mechanisms?

The universAAL IoT's UI Framework natively provides distributed operation as well as adapted, context aware, multi-modal, and multi-language user interaction. Significant improvements have been performed in the UI Framework to be optimized, as well as adding new features like modal recommendations as well as enabling simpler development of new UI Handlers.

How can legacy systems be more easily connected with semantic systems? how can non-semantic systems be enhanced with semantic capabilities?

The Codegenerator tool enables the creation of code, data or configuration files to be generated automatically from any ontology. Through the definition of template systems, legacy and non-semantic systems can at least interpret semantic data, and data structures. The tool can be used in different ways, as a pipeline to transform rdf graphs for the receiving system, or as configuration tool for describing data structures for the target system in the proper format; or it can even create specific code for the target system which can perform the interaction itself.

How can AAL semantic and distributed systems include strong cybersecurity and privacy protection mechanisms?

Although security never should be considered a closed topic, some advancements in the universAAL IoT platform have been presented. These advancements include the extension of authentication process, allowing for multi factor authentication; management and enforcement of a Role based authorisation system, including the capability of delegation; model and implementation of common cryptographic services, which can be extended if needed; and finally an anonymization process and protocol, which can be reverse only for authorised entities.

6.5. HYPOTHESIS VALIDATION

The hypothesis enunciated in section 1.4 will be validated and discussed through this section. Hypothesis 1, enunciates the challenge that is combining semantic and distributed technologies in a generic way, as well as stating that the benefits grant it. To corroborate this hypothesis is good to look into the formulation and validation of the SDS model (sections 4.1 and 5.1) to find how difficult it is to effectively combine semantics and distribution. Although semantics and distribution initially are compatible, in fact the use of concepts such as URI implies semantic technologies already has built in distribution characteristics, when combined it is always necessary to share common understanding of the semantics in all the nodes participating in the system. And even though this is true, for the union of the knowledge in all of the nodes, and each node may only intersect with a subset of the semantics required by the components deployed in said node. Therefore additional complexities arise when dealing with ontological distribution, this is handled by the deployment model comparing the needed representation with the one available throughout the system.

Another challenge is found on the need to build efficient syntactic interoperability which works for the different components in the different nodes in the distributed system. As shown by the need of the creation of the REST API Manager (section 5.4.2), different member models are required to accommodate for the heterogeneity probability increase inherit in distributed systems, particularly those in AAL domain. This syntactic interoperability issue is tackled as part of the rationale of the member model.

With regards to the benefits, not only the SDS model gives some clues, as it highlights the flexibility for developing SDS, but particularly showcases (see section 4.4) should be considered. AAL platform showcases like Advanced Distribution, Ambient Intelligence, and Scalability may give higher value to the distribution factor, where as other showcases like Plug and Play, Service Integration and Resource Sharing tend to highlight the semantic interoperability benefits. However when looking individually into each of these showcases it is evident that adding the complementing paradigm exponentiates the value. The most evident case for value on the combination of both paradigms is the most valuable showcase: Evolution, which looks into

the capability of a system to adapt, which is impossible with an effective understanding of future circumstances (achieved through semantics) and communicating with other elements (addressed through distribution).

One of the most compelling advantages of SDS is the cost effectiveness in increased complexity environment. When dealing with complex scenarios, it can be shown how this complexity tolls on the cost effectiveness of the end result (seen on figure ??), specially non-semantic, and/or non-distributed systems. When SDS are used, and specially where the SDS model is exercised, then showcases begin to show their values, and thus the cost effectiveness in these complex environments rises, even compensating with the initial overhead compared with non-SDS development. This overhead is actually the reason why SDS might not be so popular, as non-SDS paradigms provide cheap, entry-level, and fast developments. However when the complexity of the system rises, new paradigm are needed, this explains the rise of virtualization and micro-service concepts as a solution to part of the complexity, however to really deal with all aspects only SDS offer the most advantageous methods. This is precisely why the showcase about Integration with Legacy Systems is so relevant, as it offers the transformation of existing non-SDS solutions into fully fledged SDS environment, without losing the investment on these systems; precisely what the experience in ReAAL showed.

Figure 6.1: Cost vs Complexity, in low complexity scenarios SDS present an overhead which makes non-SDS systems more cost effective. However where as non-SDS cost rises exponentially with complexity, SDS handles better the complexity making the growth logarithmic, thus beyond a given complexity level SDS is more cost effective.

Hypothesis 2, refers to the need of a formal model to help decompose and address the different dimensions of complexity of SDS, as well as emphasizing the need for this formalization to be validated in the AAL domain.

The SDS model uses many existing methods to address up to 9 dimensions of complexity of SDS. The SDS model is an attempt to formalize this high level abstraction of decomposition of the complexities into different views or dimensions of the problem, some of the individual models are far from formal, in the common understanding of formalization; however the models provide enough flexibility to accommodate unforeseen circumstances, a particularity which may not have been possible with a complete formalization of the model.

The formality of the problem, although not as complete as desired, has been proven to be apt for at least 5 study cases presented through out this document. This fact also helps corroborate the hypothesis theorem about the completeness of the complex dimensionality, at least for the most common AAL scenarios. Yet the model should not be considered dimension-complete, analysing and specifying the complete space of possible views for addressing the different levels of abstraction required for SDS is in fact an nondeterministic polynomial complete (NP-Complete) problem, thus the provided model is only the best approximation for SDS in AAL given the research time.

The Hypothesis 3, states the need for a formal multi-dimensional approach to evaluate SDS, as well as a validation of said method against AAL SDS. The Showcases Methodology has been devised as a value oriented multi-dimensional evaluation approach, particularly for technical functionalities of any system. Showcases are similar to use cases, where use cases describe how the system should respond, showcases answer why is it valuable for the different stakeholders.

Formulation of the methodology is quite complete, including different alternatives to address the tool set creation as well as the actual evaluation of each showcase. The methodology encourages the definition of different values, even if the values overlap; as finding non-overlapping coverage of the value could not be possible, even less provide any algorithm to ensure this property is maintained. On the other hand the methodology it self involves all stakeholders of the AAL domain, which provide multiple perspectives to the analysis, and finally the method encourages participants of the evaluation to analyse each value considering different dimensions (such as user experience, economic, organizational, technical, ethical,...).

The methodology has been extensively validated through the definition of 14 showcases of AAL SDS platforms, and the analysis of the value 10 of those provide to up to 60 participants.

Hypothesis 4, refers to the need for an unified methodology to optimally develop SDS in AAL, as well as its validation in real large scale pilots. The cocreation methodology has been developed expressly to address this need, and specially addressing AAL system development. The most critical part of the development is the need to provide value to all stakeholders involved in the system, and particularly the end users. The methodology makes extensive use of the other methodologies presented in the document, providing solutions to specific problems arising throughout the process. A unifying vision to approach development of SDS for AAL domain is provided through the methodology, its 3 phases ensure a) all stakeholders are accounted throughout the development of the system b) the technical adaptation of

the system complies with the requirements and c) the final product actually accomplishes the objectives in a safe way before turning to production.

The methodology has been extensively tested through 2 projects involving over 10.000 end users and individual stakeholders. Flexibility has been key to enable the adaptation of the methodology to the different particularities of each project and system. One example is, even though the methodology is defined as a waterfall/sequential model, it can also be used in combination with other non-linear development methods.

The hypothesis 5 prognosticates that a generalized SDS model will in fact produce a set of feature requirements not covered by any state of the art SDS platform, and as such new developments are required to provide these features in a reusable way. The whole section 5.4 is devoted to detailing the implementation of at least 5 reusable, flexible components for the universAAL IoT platform. This shows the effectiveness of the SDS model, as well as validating the hypothesis prognostics.

6.6. OBJECTIVES REVIEW

In this section we will review the completion of all the objectives stated in section 1.5.

In order to analyse Objective 1, on the analysis of SDS challenges, the first chapter to read is Chapter 2, in this chapter the current state of the art is reviewed, and one can sense how most of the literature and solutions either stress the semantic or the distribution technologies. Another source for the answer of the challenges is the Model for SDS described in section 4.1. Throughout the section, challenges in the form of different complexities, are discussed, and a methodological approach to the development of technological solutions is provided. Challenges are also listed in the form of AAL SDS platform showcases, described in section 4.4. Each of the showcases is a desired value from AAL SDS platforms, which if it were not a challenge, it would not have enough value to be explicitly listed. The final challenge for AAL technologies is the need to involve many stakeholders, AAL technology is not a direct relation between the developer and the client, as in most engineering industries, the solution tends to involve many actors; and the relation with the end user is particularly delicate one. To address this challenge, the Cocreation Methodology, explained in section 4.3, is proposed.

Objective 4 is explicitly addressed in section 4.1; providing a model for analysing, adapting and implementing SDS decomposed in the different complexity dimensions of the system. This model is validated against 3 different AAL related complex concept systems, explained in section 5.1; and 2 more systems in the process of applying the cocreation methodology to large scale piloting projects described in section 5.3; all in compliance with sub-objective 2.1.

Objective 3, addressing the development of a multi-dimension, value oriented evaluation methodology, this methodology is described in section 4.2, and executed for SDS AAL platforms in section 4.4. Both are validated thoroughly throughout

section 5.2, demonstrating the completion of sub-objective 3.1.

Objective 4 requires a general methodology for analysing, adapting, and assessing SDS focused on AAL, this methodology is the Cocreation methodology presented in section 4.3. This methodology has been validated through 2 large scale piloting research European projects, and the results are depicted in sections 5.3.1 and 5.3.2. The other 2 methodologies, the SDS model, and the showcases evaluation methodology, can be integrated in this methodology, as it is of a more high level scope closer to project management than technology development or evaluation.

Objective 5 states the need to update the technology to comply with findings throughout the research, namely the SDS model and the AAL SDS Platform showcases. These improvements are detailed in section 5.1, through the implementation of 5 new components for the unviuersAAL IoT SDS platform.

6.7. FUTURE WORK

The research showcased through this document has been demonstrated and validated, however it is not complete, and future research and development should be invested in the evolution of this area of knowledge. The following are some suggestions on the future lines of research to be taken parting from the present state.

As stated the Model for SDS is been validated in the AAL domain, however it is presented in such a way that most of it could be applied to other domains. To do this the first step is to review the complexity dimensionality of the problem, as new models might arise from the application of the model to other domains. Additionally some of the models them selves should be tweaked to better fit the new domain, for example the user interaction model is particularly strict, because in the AAL domain this model is specially needed, however other domains might not place so much importance on this model, or even may spare it all together.

The Value oriented, multi-dimensional evaluation method should also be submitted to the abstraction process. Throughout the document, it has been stated this method produces tools which should be considered complementary to use cases. Because the use case methodology is quite wide spread through engineering, we could hypothesise that the showcase methodology could eventually become as wide spread as use cases. For this it is necessary to adapt, and simplify the method, yet keep the main message of keeping the value perspective of the technology.

Most of the AAL SDS platform showcases, could be directly used in other non-AAL related domains, as they are stated purposely in the most abstract way precisely for this reason. However other domains will produce a different list of showcases, thus even though the proposed list could be useful as a starting point, it is highly recommended to redo the exercise to better capture the individualised value of the technology for the target domain.

As stated the AAL SDS platform showcases list is not complete, thus it is extremely interesting to continue adding showcases to better specify AAL value as a whole. In this context new showcases could be explored, for example showcases

showing the value of the data it self, resolving the data ownership and usage propositions between the different stakeholders, and how can raw and processed data can make for better quality of life in general. Another potential showcase might be the capability of the system to self report, inspired by the self properties proposed in in PASSAGE (see section 5.1.2) a system which can self monitor it self, by reporting performance indicators in real time can in it self be useful, but if this self reporting can be extended through advance anomaly detection, it means that systems could self-diagnose, and even attempt self-repair making scalable services more cost effective and trustworthy; definitely a feature which would be highly appreciated in the AAL domain.

As for the Cocreation methodology, there is room for improvement; for example, it can be adapted to other domains, so that it may be used for future projects. Thus it will be extremely interesting to research how to generalize the methodology to be domain and technology independent. This research will be an iterative process in itself, presented here are only the results of initial iterations. The maturation of the methodology can be directed not only by he adaptation of the methodology it self, but through the evolution of the services it assumes, as well as the extension of existing and future technological paradigms, and by applying it to a different set of stakeholders and target populations.

The results of this research are targeted to better the understanding of SDS in the domain of AAL. One possible use of this understanding would be the creation of standards, helping evolve the concept of service transferability. This concept has been explicitly addressed in the document, however the full impact has not been considered. This is because the impact of fully interoperable AAL services, applications and technologies would inevitably result in a world wide market for AAL, a truly commendable effort which lies just beyond this research, and is still the main topic for many research proposals. The impact of such market would be ground breaking in an world with ageing population, where a health competition of AAL services and supporting technologies will not only foster innovation, optimizing the provision of current services, and developing new services; access to needed services will be extensive, particularly for services devoted to statistically low casuistic which would otherwise be deemed not economically profitable to engage with (e.g. strange diseases which affect low fractions of the population).

Finally share the hope that even though this research was devoted to AAL domain, and in particular ageing populations, it still can be useful to other AAL areas, and other domains. As demonstrated AAL is a very strict domain, particularly with regards to the user, if adjusting the target population removes these restrictions, SDS might be more manageable through the presented research. SDS could also be applied in new ways for addressing the ageing population, instead of focusing on the elderly, we could use the technology on the other side of the life cycle, aiding adults in improving conception and child upbringing. It is astonishing how both ends of life are so similar, younger people have equally demanding requirements from any technology as elder adults; thus it is possible to apply the knowledge acquired with elder adults to develop systems for children, aiding their parents in their upbringing

and hopefully facilitating them to have more offspring, thus addressing ageing populations. As with ageing citizens, the problem is not one which can be solved with technology alone, many factors contribute to the problem and the problem should be approach from different perspectives such as politically, socially, and ethically (addressing the sustainability of population growth avoiding also overpopulation).

BIBLIOGRAPHY

- [1] “Ageing and health.” <https://www.who.int/news-room/fact-sheets/detail/ageing-and-health>.
- [2] “The EU Silver Economy: Market size, challenges and opportunities — ActiveAdvice.” <https://www.activeadvice.eu/news/market-insights/the-eu-silver-economy-market-size-challenges-and-opportunities/>.
- [3] “2018 World Population Data Sheet With Focus on Changing Age Structures – Population Reference Bureau.” <https://www.prb.org/2018-world-population-data-sheet-with-focus-on-changing-age-structures/>.
- [4] “Numbers & Statistics : All figures about aging — Silver Economy.” <http://www.silvereco.org/en/statistics/>.
- [5] M. Khezrian, P. K. Myint, C. McNeil, and A. D. Murray, “A Review of Frailty Syndrome and Its Physical, Cognitive and Emotional Domains in the Elderly,” *Geriatrics*, vol. 2, p. 36, Dec. 2017.
- [6] “The 2018 Ageing Report: Economic and Budgetary Projections for the EU Member States (2016-2070).” https://ec.europa.eu/info/publications/economy-finance/2018-ageing-report-economic-and-budgetary-projections-eu-member-states-2016-2070_en.
- [7] C. O. Weiss, “Frailty and chronic diseases in older adults,” *Clinics in geriatric medicine*, vol. 27, no. 1, pp. 39–52, 2011.
- [8] “Factsheet: Horizon 2020 budget.” http://ec.europa.eu/research/horizon2020/pdf/press/fact_sheet_en.pdf.
- [9] “Budgets - FP7 - Research - Europa.” https://ec.europa.eu/research/fp7/index_en.cfm?pg=budget.
- [10] legroju, “AAL programme: Funding for Active and Assisted Living.” <https://ec.europa.eu/digital-single-market/en/active-and-assisted-living-joint-programme-aal-jp>, Jan. 2014.
- [11] W. H. Organization, “World health statistics 2014: A wealth of information on global public health,” 2014.

- [12] J. Bousquet, J. Malva, M. Nogues, L. R. Mañas, B. Vellas, J. Farrell, M. Bewick, M. L. Kowalski, T. Strandberg, J. Alonso, K. A. Ranberg, J. Ankri, M. Barbagallo, Y. Ben-Shlomo, C. Berr, G. Crooks, E. de Manuel Keenoy, M. Goldberg, N. Guldemon, M. Illario, M.-E. Joel, E. Kesse-Guyot, J.-P. Michel, R. Pengelly, K. Ritchie, J.-M. Robine, V. Romano, B. Samolinski, H. Schulz, C. Thijs, J. Touchon, A. Zaidi, J. Apostolo, F. Alonso, V. Andreeva, S. Arnavielhe, H. Arshad, A. Avignon, M. Barbolini, A. Bedbrook, P. Bertone, H. Blain, A. Blasimme, S. Bonini, R. Bourret, P.-J. Bousquet, C. Brayne, J. Bringer, G. J. Buijs, L. Calza, T. Camuzat, A. Cano, A. Carriazo, A. Cesario, N. Chavannes, B. Combe, J. Coletta, J. C. de Sousa, T. Cosco, M. Criton, F. Cuisinier, P. Demoly, S. Fernandez-Nocelo, B. Fougère, J. Garcia-Aymerich, Z. Gutter, D. Harman, A. Hendry, D. Hève, G. Iaccarino, C. Jeandel, T. Keil, O. Krysz, D. Kula, D. Laune, D. Maier, P. Matignon, E. Melen, G. Mercier, G. Moda, I. Momas, A. M. Pinto, A. Nizinska, C. Oliveira, M. O'Neill, J.-Y. Pelissier, J. A. Pereira da Silva, D. Poethig, D. Porta, D. Postma, F. Puisieux, S. Rebello, C. Robalo-Cordeiro, F. Roubille, R. Santiago, A. Scott, P. Senesse, S. Slagter, H. A. Smit, D. Somekh, J. Suanzes, C. Tischer, A. Todo-Bom, V. Traver-Salcedo, M. Van Beurden, R. Varraso, J. Venne, I. Vergara, E. Villalba-Mora, P. Viriot-Durandal, N. Wilson, E. Wouters, M. Zins, and J. Mercier, "Operational Definition of Active and Healthy Aging (AHA): The European Innovation Partnership (EIP) on AHA Reference Site Questionnaire: Montpellier October 20–21, 2014, Lisbon July 2, 2015," *Journal of the American Medical Directors Association*, vol. 16, pp. 1020–1026, Dec. 2015.
- [13] Q. Yang and Z. Shen, "Active Aging in the Workplace and the Role of Intelligent Technologies," in *2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, vol. 2, pp. 391–394, Dec. 2015.
- [14] C. Parra, P. Silveira, I. K. Far, F. Daniel, E. D. D. Bruin, L. Cernuzzi, V. D'Andrea, and F. Casati, "Information Technology for Active Ageing: A Review of Theory and Practice," *Foundations and Trends® in Human-Computer Interaction*, vol. 7, pp. 351–448, Dec. 2014.
- [15] E. Commission, "Innovation Partnership on Active and Healthy Ageing." https://ec.europa.eu/eip/ageing/home_en.
- [16] B. Rechel, E. Grundy, J.-M. Robine, J. Cylus, J. P. Mackenbach, C. Knai, and M. McKee, "Ageing in the European Union," *The Lancet*, vol. 381, pp. 1312–1322, Apr. 2013.
- [17] A. P. Sheth, "Changing Focus on Interoperability in Information Systems: From System, Syntax, Structure to Semantics," in *Interoperating Geographic Information Systems* (M. Goodchild, M. Egenhofer, R. Fegeas, and

- C. Kottman, eds.), The Springer International Series in Engineering and Computer Science, pp. 5–29, Boston, MA: Springer US, 1999.
- [18] C. Pautasso, O. Zimmermann, and F. Leymann, “Restful Web Services vs. ”Big” Web Services: Making the Right Architectural Decision,” in *Proceedings of the 17th International Conference on World Wide Web*, WWW ’08, (New York, NY, USA), pp. 805–814, ACM, 2008.
 - [19] T. Berners-Lee, J. Hendler, O. Lassila, *et al.*, “The semantic web,” *Scientific american*, vol. 284, no. 5, pp. 28–37, 2001.
 - [20] Dan Brickley, University of Bristol and R.V. Guha, Netscape, “Resource Description Framework (RDF) Schema Specification.” <https://www.w3.org/TR/PR-rdf-schema/>.
 - [21] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions,” *Future Generation Computer Systems*, vol. 29, pp. 1645–1660, Sept. 2013.
 - [22] M. Compton, P. Barnaghi, L. Bermudez, R. García-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog, V. Huang, K. Janowicz, W. D. Kelsey, D. Le Phuoc, L. Lefort, M. Leggieri, H. Neuhaus, A. Nikolov, K. Page, A. Passant, A. Sheth, and K. Taylor, “The SSN ontology of the W3C semantic sensor network incubator group,” *Journal of Web Semantics*, vol. 17, pp. 25–32, Dec. 2012.
 - [23] A. Haller, K. Janowicz, S. J. D. Cox, M. Lefrançois, K. Taylor, D. L. Phuoc, J. Lieberman, R. García-Castro, R. Atkinson, and C. Stadler, “The SOSA/SSN Ontology: A Joint W3C and OGC Standard Specifying the Semantics of Sensors, Observations, Actuation, and Sampling,” p. 19.
 - [24] R. Das, A. Tuna, S. Demirel, and M. K. Yurdakul, “A Survey on the Internet of Things Solutions for the Elderly and Disabled: Applications, Prospects, and Challenges,” *International Journal of Computer Networks And Applications*, vol. 4, p. 1, June 2017.
 - [25] G. Fico, J.-B. Montalvá, A. Medrano, N. Liappas, A. Mata-Díaz, G. Cea, and M. T. Arredondo, “Co-creating with consumers and stakeholders to understand the benefit of Internet of Things in Smart Living Environments for Ageing Well: The approach adopted in the Madrid Deployment Site of the ACTIVAGE Large Scale Pilot,” in *EMBECE & NBC 2017* (H. Eskola, O. Väisänen, J. Viik, and J. Hyttinen, eds.), IFMBE Proceedings, pp. 1089–1092, Springer Singapore, 2018.
 - [26] E. I. Konstantinidis, G. Bamparopoulos, A. Billis, and P. D. Bamidis, “Internet of Things For an Age-Friendly Healthcare,” *Studies in health technology and informatics*, vol. 210, pp. 587–591, 2015.

- [27] D. Wortley, J.-Y. An, and A. Heshmati, “Tackling the Challenge of the Aging Society: Detecting and Preventing Cognitive and Physical Decline through Games and Consumer Technologies,” *Healthcare Informatics Research*, vol. 23, pp. 87–93, Apr. 2017.
- [28] M. Zallio, D. Berry, and N. Casiddu, “Adaptive environments for enabling senior citizens: An holistic assessment tool for housing design and IoT-based technologies,” in *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, pp. 419–424, Dec. 2016.
- [29] A. Melis, M. Prandini, L. Sartori, and F. Callegati, “Public Transportation, IoT, Trust and Urban Habits,” in *Internet Science* (F. Bagnoli, A. Satsiou, I. Stavrakakis, P. Nesi, G. Pacini, Y. Welp, T. Tiropanis, and D. DiFranzo, eds.), Lecture Notes in Computer Science, pp. 318–325, Springer International Publishing, 2016.
- [30] Y. Shibata and G. Sato, “IoT Based Mobility Information Infrastructure in Challenged Network Environment toward Aging Society,” in *2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pp. 645–648, Mar. 2017.
- [31] C. Kulczycki, R. Genoe, H. Marston, S. Freeman, and C. Musselwhite, ““If I want to know anything I just Google it”: Older adults’ functional and social leisure activities and technology,” in *Canadian Congress on Leisure Research 15: Engaging Legacies (CCLR15)*, (University of Waterloo, ON, Canada), 2017.
- [32] D. Faraoni and S. T. Schaefer, “Randomized controlled trials vs. observational studies: Why not just live together?,” *BMC Anesthesiology*, vol. 16, Oct. 2016.
- [33] M. Fernández-López, A. Gómez-Pérez, and N. Juristo, “METHONTOLOGY: From Ontological Art Towards Ontological Engineering,” in *Proceedings of the Ontological Engineering AAAI-97 Spring Symposium Series*, (Stanford University, EEUU), Facultad de Informática (UPM), Mar. 1997.
- [34] S. Staab and R. Studer, *Handbook on Ontologies*. Springer Science & Business Media, 2010.
- [35] A. Gangemi and V. Presutti, “Ontology Design Patterns,” in *Handbook on Ontologies*, pp. 221–243, May 2009.
- [36] A. Gangemi, “Ontology Design Patterns for Semantic Web Content,” in *The Semantic Web – ISWC 2005* (Y. Gil, E. Motta, V. R. Benjamins, and M. A. Musen, eds.), Lecture Notes in Computer Science, pp. 262–276, Springer Berlin Heidelberg, 2005.

- [37] M. C. Suárez-Figueroa, A. Gómez-Pérez, and M. Fernández-López, “The NeOn Methodology framework: A scenario-based methodology for ontology development,” *Applied Ontology*, vol. 10, pp. 107–145, Jan. 2015.
- [38] P. Casanovas, V. Rodríguez-Doncel, C. Santos, and A. Gómez-Pérez, “A European Framework for Regulating Data and Metadata Markets,” in *PrivOn@ISWC*, 2016.
- [39] F. Radulovic, N. Mihindukulasooriya, R. García-Castro, and A. Gómez-Pérez, “A comprehensive quality model for Linked Data,” *Semantic Web*, vol. 9, pp. 3–24, Jan. 2018.
- [40] L. D. Miltiadis, A. Naif, D. Ernesto, and C. Tai, Kwok, *Innovations, Developments, and Applications of Semantic Web and Information Systems*. IGI Global, Jan. 2018.
- [41] M. Schmidt, T. Hornung, N. Küchlin, G. Lausen, and C. Pinkel, “An Experimental Comparison of RDF Data Management Approaches in a SPARQL Benchmark Scenario,” in *The Semantic Web - ISWC 2008* (A. Sheth, S. Staab, M. Dean, M. Paolucci, D. Maynard, T. Finin, and K. Thirunarayan, eds.), Lecture Notes in Computer Science, pp. 82–97, Springer Berlin Heidelberg, 2008.
- [42] J. Broekstra, A. Kampman, and F. van Harmelen, “Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema,” in *The Semantic Web — ISWC 2002* (I. Horrocks and J. Hendler, eds.), Lecture Notes in Computer Science, pp. 54–68, Springer Berlin Heidelberg, 2002.
- [43] N. Chappell and H. A. Cooke, “Age related disabilities-aging and quality of life,” *International encyclopedia of rehabilitation*, pp. 1–3, 2010.
- [44] S. Oviatt, “Advances in robust multimodal interface design,” *IEEE computer graphics and applications*, no. 5, pp. 62–68, 2003.
- [45] B. Dumas, D. Lalanne, and S. Oviatt, “Multimodal interfaces: A survey of principles, models and frameworks,” in *Human Machine Interaction*, pp. 3–26, Springer, 2009.
- [46] A. Grguric, A. M. M. Gil, D. Huljenic, Z. Car, and V. Podobnik, “A Survey on User Interaction Mechanisms for Enhanced Living Environments,” in *ICT Innovations 2015* (S. Loshkovska and S. Koceski, eds.), Advances in Intelligent Systems and Computing, pp. 131–141, Springer International Publishing, 2016.
- [47] A. M. Gil, D. Salvi, M. T. A. Waldmeyer, P. A. Jimenez, and A. Grguric, “Separating the Content from the Presentation in AAL: The universAAL UI Framework and the Swing UI Handler,” in *Ambient Intelligence - Software and*

Applications (A. Berlo, K. Hallenborg, J. M. C. Rodriguez, D. I. Tapia, and P. Novais, eds.), vol. 219 of *Advances in Intelligent Systems and Computing*, pp. 113–120, Springer International Publishing, 2013.

- [48] M. Arnold, P. Hopewell, C. Paddison, P. Parry, and N. Sustache, “User centred design: How to design accessible products,” in *Proceedings of the 1st European UPA Conference on European Usability Professionals Association Conference-Volume 3*, pp. 22–44, British Computer Society, 2002.
- [49] D. Hawthorn, “Possible implications of aging for interface designers,” *Interacting with Computers*, vol. 12, pp. 507–528, Apr. 2000.
- [50] J. M. Morris, “User interface design for older adults,” *Interacting with Computers*, vol. 6, no. 4, pp. 373–393, 1994.
- [51] G. C. Vanderheiden, J. Treviranus, and A. Chourasia, “The Global Public Inclusive Infrastructure (GPII),” in *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS ’13, (New York, NY, USA), pp. 70:1–70:3, ACM, 2013.
- [52] J. Dean and S. Ghemawat, “MapReduce: Simplified data processing on large clusters,” *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.
- [53] J. S. Rellermeyer, G. Alonso, and T. Roscoe, “R-OSGi: Distributed applications through software modularization,” in *ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing*, pp. 1–20, Springer, 2007.
- [54] P. J. Leach, T. Berners-Lee, J. C. Mogul, L. Masinter, R. T. Fielding, and J. Gettys, “Hypertext Transfer Protocol – HTTP/1.1.” <https://tools.ietf.org/html/rfc2616>.
- [55] 14:00-17:00, “ISO/IEC 21778:2017.” <http://www.iso.org/cms/render/live/en/sites/isoorg/conten>
- [56] J. J. Garrett, “Ajax: A new approach to web applications,” 2007.
- [57] R. T. Fielding and R. N. Taylor, “Principled Design of the Modern Web Architecture,” *ACM Trans. Internet Technol.*, vol. 2, pp. 115–150, May 2002.
- [58] E. Christensen, F. Curbera, G. Meredith, S. Weerawarana, *et al.*, “Web services description language (WSDL) 1.1,” 2001.
- [59] P. Jamshidi, C. Pahl, N. C. Mendonça, J. Lewis, and S. Tilkov, “Microservices: The journey so far and challenges ahead,” *IEEE Software*, vol. 35, no. 3, pp. 24–35, 2018.
- [60] N. M. Josuttis, *SOA in Practice: The Art of Distributed System Design*. “O’Reilly Media, Inc.”, Aug. 2007.

- [61] D. C. Luckham and B. Frasca, "Complex event processing in distributed systems," *Computer Systems Laboratory Technical Report CSL-TR-98-754. Stanford University, Stanford*, vol. 28, p. 16, 1998.
- [62] P. Mell, T. Grance, *et al.*, "The NIST definition of cloud computing," 2011.
- [63] W. Zheng, P. Xu, X. Huang, and N. Wu, "Design a cloud storage platform for pervasive computing environments," *Cluster Computing*, vol. 13, no. 2, pp. 141–151, 2010.
- [64] C. Doukas, T. Pliakas, and I. Maglogiannis, "Mobile healthcare information management utilizing Cloud Computing and Android OS," in *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, pp. 1037–1040, Aug. 2010.
- [65] V. Koufi, F. Malamateniou, G. Vassilacopoulos, and A. Prentza, "An Android-Enabled Mobile Framework for Ubiquitous Access to Cloud Emergency Medical Services," in *Network Cloud Computing and Applications (NCCA), 2012 Second Symposium On*, pp. 95–101, Dec. 2012.
- [66] L. Lamport, R. Shostak, and M. Pease, "The Byzantine generals problem," *ACM Transactions on Programming Languages and Systems (TOPLAS)*, vol. 4, no. 3, pp. 382–401, 1982.
- [67] M. Crosby, P. Pattanayak, S. Verma, V. Kalyanaraman, *et al.*, "Blockchain technology: Beyond bitcoin," *Applied Innovation*, vol. 2, no. 6-10, p. 71, 2016.
- [68] C. Yoon, M. Huh, S. Kang, J. Park, and C. Lee, "Implement smart farm with IoT technology," in *2018 20th International Conference on Advanced Communication Technology (ICACT)*, pp. 749–752, Feb. 2018.
- [69] M. N. Kamel Boulos and N. M. Al-Shorbaji, "On the Internet of Things, smart cities and the WHO Healthy Cities," *International Journal of Health Geographics*, vol. 13, p. 10, Mar. 2014.
- [70] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang, "What Will 5G Be?," *IEEE Journal on Selected Areas in Communications*, vol. 32, pp. 1065–1082, June 2014.
- [71] Z. Shelby, K. Hartke, and C. Bormann, "The Constrained Application Protocol (CoAP)." <https://tools.ietf.org/html/rfc7252>.
- [72] J. D. Day and H. Zimmermann, "The OSI reference model," *Proceedings of the IEEE*, vol. 71, pp. 1334–1340, Dec. 1983.
- [73] Y. Ai, M. Peng, and K. Zhang, "Edge computing technologies for Internet of Things: A primer," *Digital Communications and Networks*, vol. 4, pp. 77–86, Apr. 2018.

- [74] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge Computing: Vision and Challenges,” *IEEE Internet of Things Journal*, vol. 3, pp. 637–646, Oct. 2016.
- [75] N. Ruest and D. Ruest, *Virtualization, A Beginner’s Guide*. New York, NY, USA: McGraw-Hill, Inc., 1 ed., 2009.
- [76] V. Moreno and K. Reddy, *Network Virtualization (Networking Technology)*. Cisco Press, 2006.
- [77] J. Turnbull, *The Docker Book: Containerization Is the New Virtualization*. James Turnbull, July 2014.
- [78] *Standards for Security Categorization of Federal Information and Information Systems*. Gaithersburg, MD: National Institute of Standards and Technology, May 2009.
- [79] R. S. Ross, “Assessing Security and Privacy Controls in Federal Information Systems and Organizations:: Building Effective Assessment Plans,” Tech. Rep. NIST SP 800-53Ar4, National Institute of Standards and Technology, Dec. 2014.
- [80] S. Hamilton and N. L. Chervany, “Evaluating Information System Effectiveness - Part I: Comparing Evaluation Approaches,” *MIS Quarterly*, vol. 5, no. 3, pp. 55–69, 1981.
- [81] J. W. Tipping, E. Zeffren, and A. R. Fوسفeld, “Assessing the Value of Your Technology,” *Research-Technology Management*, vol. 38, pp. 22–39, Sept. 1995.
- [82] E. Almquist, J. Senior, and N. Bloch, “The elements of value,” *Harvard Business Review*, vol. 94, no. 9, pp. 47–53, 2016.
- [83] W. W. Royce, “Managing the development of large software systems: Concepts and techniques,” in *Proceedings of the 9th International Conference on Software Engineering*, pp. 328–338, IEEE Computer Society Press, 1987.
- [84] “A Spiral Model of Software Development and Enhancement.” <https://www.computer.org/csdl/magazine/co/1988/05/r5061/13rRUwwslzm>, May 1988.
- [85] A. Cockburn, *Agile Software Development: The Cooperative Game*. Pearson Education, Oct. 2006.
- [86] L. Bass, I. Weber, and L. Zhu, *DevOps: A Software Architect’s Perspective*. Addison-Wesley Professional, May 2015.
- [87] M. Meyer, “Continuous Integration and Its Tools,” *IEEE Software*, vol. 31, pp. 14–16, May 2014.

- [88] M. Fowler and M. Foemmel, “Continuous integration,” *Thought-Works*) [http://www.thoughtworks.com/Continuous Integration. pdf](http://www.thoughtworks.com/Continuous%20Integration.pdf), vol. 122, p. 14, 2006.
- [89] S. Hanke, C. Mayer, O. Hoefftberger, H. Boos, R. Wichert, M.-R. Tazari, P. Wolf, and F. Furfari, “universAAL – An Open and Consolidated AAL Platform,” in *Ambient Assisted Living* (R. Wichert and B. Eberhardt, eds.), pp. 127–140, Springer Berlin Heidelberg, 2011.
- [90] *universAAL Project, UNIVERsal Open Platform and Reference Specification for Ambient Assisted Living*. 7th Framework Programme of the European Union, Grant Agreement No. 247950, 2010.
- [91] M.-R. Tazari, F. Furfari, Á. F. Valero, S. Hanke, O. Höftberger, D. Kehagias, M. Mosmondor, R. Wichert, and P. Wolf, “The universal Reference Model for AAL,” in *Handbook of Ambient Assisted Living - Technology for Healthcare, Rehabilitation, and Well-Being*, pp. 610 – 625, ios press ed., 2012.
- [92] J. M. Boyer, *XForms 1.1., W3C Recommendation 20 October 2009*. 2009.
- [93] monhaber, “Security recommendations in Azure Security Center.” <https://docs.microsoft.com/en-us/azure/security-center/security-center-recommendations>.
- [94] “ENISA Good practices for IoT and Smart Infrastructures Tool.” <https://www.enisa.europa.eu/topics/iot-and-smart-infrastructures/iot/good-practices-for-iot-and-smart-infrastructures-tool>.
- [95] *IoT Security Compliance Framework*. IoT Security Foundation, release 1.0 ed., 2016.
- [96] S. Hernan, S. Lambert, T. Ostwald, and A. Shostack, “Threat modeling-uncover security design flaws using the stride approach,” *MSDN Magazine*, pp. 68–75, Jan. 2006.
- [97] K. Kidholm, A. G. Ekeland, L. K. Jensen, J. Rasmussen, C. D. Pedersen, A. Bowes, S. A. Flottorp, and M. Bech, “A MODEL FOR ASSESSMENT OF TELEMEDICINE APPLICATIONS: MAST,” *International Journal of Technology Assessment in Health Care*, vol. 28, pp. 44–51, Jan. 2012.
- [98] W. H. Delone and E. R. McLean, “The DeLone and McLean Model of Information Systems Success: A Ten-Year Update,” *Journal of Management Information Systems*, vol. 19, pp. 9–30, Apr. 2003.
- [99] R. Studer, V. R. Benjamins, and D. Fensel, “Knowledge engineering: Principles and methods,” *Data & Knowledge Engineering*, vol. 25, pp. 161–197, Mar. 1998.

- [100] S. Staab, R. Studer, H. P. Schnurr, and Y. Sure, “Knowledge processes and ontologies,” *IEEE Intelligent Systems*, vol. 16, pp. 26–34, Jan. 2001.
- [101] L. Bellatreche, Y. Ait-Ameur, and C. Chakroun, “A Design Methodology Of Ontology Based Database Applications,” *Logic Journal of the IGPL*, vol. 19, no. 5, pp. 648–665, 2011.
- [102] E. Serrano, P. del Pozo-Jiménez, M. C. Suárez-Figueroa, J. González-Pachón, J. Bajo, and A. Gómez-Pérez, “A Multi-agent Architecture for Labeling Data and Generating Prediction Models in the Field of Social Services,” in *Highlights of Practical Applications of Cyber-Physical Multi-Agent Systems*, Communications in Computer and Information Science, pp. 177–184, Springer, Cham, June 2017.
- [103] H. S. Pinto, S. Staab, and C. Tempich, “DILIGENT: Towards a fine-grained methodology for distributed, loosely-controlled and evolvInG,” in *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI 2004)*, vol. 110, p. 393, 2004.
- [104] A. Gómez-Pérez, M. Fernandez-Lopez, and O. Corcho, *Ontological Engineering: With Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web. First Edition.* Springer Science & Business Media, Apr. 2006.
- [105] M. Grüninger and M. S. Fox, “Methodology for the Design and Evaluation of Ontologies,” 1995.
- [106] D. L. McGuinness, “Conceptual Modeling for Distributed Ontology Environments,” in *Conceptual Structures: Logical, Linguistic, and Computational Issues*, Lecture Notes in Computer Science, pp. 100–112, Springer, Berlin, Heidelberg, Aug. 2000.
- [107] L. Daniele, F. den Hartog, and J. Roes, “Created in Close Interaction with the Industry: The Smart Appliances REference (SAREF) Ontology,” in *Formal Ontologies Meet Industry* (R. Cuel and R. Young, eds.), Lecture Notes in Business Information Processing, pp. 100–112, Springer International Publishing, 2015.
- [108] R. Battle and D. Kolas, “GeoSPARQL: Enabling a Geospatial Semantic Web,” p. 17.
- [109] H. Nacer and D. Aissani, “Semantic web services: Standards, applications, challenges and solutions,” *Journal of Network and Computer Applications*, vol. 44, pp. 134–151, Sept. 2014.
- [110] C. Stocklów, A. M. Medrano Gil, A. Fides Valero, M. Girolami, and S. Lenzi, “Multi-tenancy Aware Ambient Assisted Living Platforms in the Cloud,” in

Ambient Intelligence (E. Aarts, B. de Ruyter, P. Markopoulos, E. van Loenen, R. Wichert, B. Schouten, J. Terken, R. Van Kranenburg, E. Den Ouden, and G. O'Hare, eds.), Lecture Notes in Computer Science, pp. 80–95, Springer International Publishing, 2014.

- [111] G. Booch, *The Unified Modeling Language User Guide*. Pearson Education India, 2005.
- [112] K. Örtengren, *The Logical Framework Approach: A Summary of the Theory behind the LFA Method*. Sida, 2004.
- [113] H. Camacho, L. Cámara, R. Cascante, and H. Sainz, “El Enfoque del marco lógico: 10 casos prácticos,” *Cuaderno para la identificación y diseño de proyectos de desarrollo, Madrid, España, Fundación CIDEAL*, 2001.
- [114] B. Jackson, “Designing Projects and Project Evaluations Using The Logical Framework Approach,” *UCN Monitoring and Evaluation Initiative*, p. 26, 1997.
- [115] D. Salvi, J. B. Montalvá Colomer, M. T. Arredondo, B. Prazak-Aram, and C. Mayer, “A framework for evaluating Ambient Assisted Living technologies and the experience of the AAL project,” *Journal of Ambient Intelligence and Smart Environments*, vol. 7, pp. 329–352, Jan. 2015.
- [116] J. B. M. Colomer, D. Salvi, M. F. Cabrera-Umpierrez, M. T. Arredondo, P. Abril, V. Jimenez-Mixco, R. García-Betances, A. Fioravanti, M. Pastorino, J. Cancela, and A. Medrano, “Experience in Evaluating AAL Solutions in Living Labs,” *Sensors*, vol. 14, pp. 7277–7311, Apr. 2014.
- [117] A. Bolderston, “Conducting a Research Interview,” *Journal of Medical Imaging and Radiation Sciences*, vol. 43, pp. 66–76, Mar. 2012.
- [118] K. Bittner, *Use Case Modeling*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2002.
- [119] K. Connelly, K. A. Siek, I. Mulder, S. Neely, G. Stevenson, and C. Kray, “Evaluating Pervasive and Ubiquitous Systems,” *IEEE Pervasive Computing*, vol. 7, no. 3, pp. 85–88, 2008.
- [120] *AALOA Association*.
- [121] A. W. Standardisation, “High Level Architecture (HLA),” *Technical specification.—2017*, 2017.
- [122] Stocklów, C., Grgurić, A., Dutz, T., Vandommele, T., and Kuijper, A., “Resource Management for Multimodal and Multilingual Adaptation of User Interfaces in Ambient Assisted Living Environments,” *Universal Access in Human-Computer Interaction. Applications and Services for Quality of Life, HCI International*, pp. 97–106, 2013.

- [123] I. Cingil, A. Dogac, and A. Azgin, “A Broader Approach to Personalization,” *Commun. ACM*, vol. 43, pp. 136–141, Aug. 2000.
- [124] *ETSI ES 202 746. Human Factors (HF); Personalization and User Profile Management; User Profile Preferences and Information*. European Telecommunications Standards Institute (ETSI), Feb. 2010.
- [125] *ISO/IEC 24751-2:2008. Information Technology – Individualized Adaptability and Accessibility in E-Learning, Education and Training – Part 2: ‘Access for All’ Personal Needs and Preferences for Digital Delivery*. International Organization for Standardization (ISO), Sept. 2008.
- [126] A. Medrano Gil, D. Salvi, MT. Arredondo Waldmeyer, P. Abril Jimenez, and A. Fides Valero, “Use of hierarchical model-view-controller architecture for user interaction in AAL environments,” in *Proceedings of the 3rd Workshop on Technology for Healthcare and Healthy Lifestyle (WTHS’11)*, ITACA, Dec. 2011.
- [127] A. Grgurić, M. Mošmondor, M. Kušek, C. Stocklow, and D. Salvi, “Introducing Gesture Interaction in the Ambient Assisted Living Platform univ-sAAL,” in *12th International Conference on Telecommunications CONTEL 2013, Zagreb, Croatia*, pp. 215–222, 2013.