UNIVERSIDAD POLITÉCNICA DE MADRID
Escuela Técnica Superior de Ingenieros de Telecomunicación



Enabling smart drone-supported territories: architectures, unmanned traffic management, and simulation-based decision-making

D O C T O R A L T H E S I S

Submitted for the degree of Doctor by:

David Carramiñana Jiménez
M.Sc. Telecommunication Engineering

Madrid, 2025




UNIVERSIDAD POLITÉCNICA DE MADRID
Escuela Técnica Superior de Ingenieros de Telecomunicación

Doctoral Degree in Programa de Doctorado en Tecnologías y Sistemas de Comunicaciones

Enabling smart drone-supported territories: architectures, unmanned traffic management, and simulation-based decision-making

DOCTORAL THESIS

Submitted for the degree of Doctor by:

David Carramiñana Jiménez
M.Sc. Telecommunication Engineering

Under the supervision of:
Dr. Juan Alberto Besada Portas

Madrid, 2025

Title: Enabling smart drone-supported territories: architectures, unmanned traffic management, and simulation-based decision-making

Author: David Carramiñana Jiménez

Doctoral Programme: Programa de Doctorado en Tecnologías y Sistemas de Comunicaciones

Thesis Supervision

- Juan Alberto Besada Portas , Full Professor, Universidad Politécnica de Madrid (Supervisor)

External Reviewers

Thesis Defense Committee

Thesis Defense Date

A aquellos que marcharon de pueblos como Valdelavilla en busca de un futuro mejor,
y a sus descendientes que con su ayuda tuvieron que repetir el camino de salida.

«Colinas plateadas,
grises alcores, cárdenas roquedas
por donde traza el Duero
su curva de ballesta
en torno a Soria, obscuros encinares,
ariscos pedregales, calvas sierras,
caminos blancos y álamos del río,
tardes de Soria, mística y guerrera,
hoy siento por vosotros, en el fondo
del corazón, tristeza,
tristeza que es amor! ¡Campos de Soria
donde parece que las rocas sueñan,
conmigo vais! ¡Colinas plateadas,
grises alcores, cárdenas roquedas!... »

Antonio Machado

Acknowledgement

Esta tesis sigue la senda de mi trabajo dentro del Grupo de Procesado de Datos y Simulación de la ETSIT-UPM. Quien entrara como becario hace 8 años para realizar su TFG escribe esta tesis gracias al conocimiento transmitido desde entonces por los profesores del grupo de investigación. Así, debo agradecer a los profesores Casar, Besada, Bernardos, de Miguel y Bergesio por su confianza para dejarme formar parte del Grupo y sus enriquecedores intercambios de ideas. En particular, he de distinguir la supervisión de mi director de tesis, el profesor Besada. Sin su paciencia, consejo, guía y constructivas discusiones esta tesis no habría salido adelante. También debo reconocer la ayuda de mis compañeros dentro del Grupo por su ayuda y la colaboración en algunos de los artículos que sustentan esta tesis.

En clave personal, debo dar las gracias a mi familia y, en especial, a mis padres por no dudarlo y permitirme salir de allí donde no existen las oportunidades siendo conscientes de que compraban un billete sin retorno. El éxodo en busca de un futuro mejor que iniciaron mis abuelos desde Valdelavilla se repite con mi marcha de Soria a Madrid. Esta tesis supone otro paso más en ese camino lejos de casa. Camino que solo es posible gracias a su paciencia, comprensión, esfuerzo y apoyo constante y absoluto. Espero que este agradecimiento y la consecución de esta tesis pueda paliar si acaso mínimamente los innumerables e irrecuperables momentos alejado de casa. Por último, Celia, has sido la luz que me ha acompañado y animado en los momentos de duda. Has hecho que el inhóspito Madrid lo fuese menos y me has apoyado con comprensión en el pedregoso camino del doctorado que culmina con esta tesis.

A todos vosotros, **gracias**.

En Madrid,
esperando volver a ver los *álamos dorados*
del camino en la rivera del Duero,
junio de 2025.

Abstract

Technological advances have gradually integrated in the operation of modern human settlements, becoming one of the main driving forces of economic, societal and environmental prosperity. This trend is reflected in the concept of smart territories, which are those urban or rural settlements that use technological solutions to improve their infrastructures and services efficiency and resilience, and the quality of life of citizens. The main technological pillars of smart territories are: instrumentation, interconnection and intelligence through the Internet of Things (IoT); provision of advanced smart services to citizens, and informed decision-making based on data. In addition, aerial drones can be a technological enabler by supporting wide area monitoring, providing communications in remote areas or delivering new aerial mobility services. In this context, this thesis proposes novel systems, applications and models for each of the technological pillars of smart territories, focusing on drones' usage and integration.

For IoT solutions, the usage of drones and smartphones as pervasive monitoring systems is explored. Two architectures are proposed considering IoT systems as Data Fusion and Resource Management processes. For wide area ground surveillance, the thesis proposes an architecture that allows autonomous and explainable control of a fleet of drones. The proposal has been deployed in an indoor testbed environment demonstrating that it allows to identify the task performed by each drone and to provide useful surveillance information. For ubiquitous healthcare, a reference mobile health architecture has been presented to standardize and streamline the deployment of such smartphone-based systems. A use-case based on SARS-COV-2 patients monitoring serves as a proof of concept of the proposal.

Regarding smart services provision through drones, Unmanned Traffic Management systems are needed to safely coordinate their operation. On this topic, the thesis discusses a simulation framework to represent, test and validate the European U-Space ecosystem. U-space is understood here as a smart service that supervises and monitors drone operations through real time interconnection of drones, operators, and surveillance networks. As a novelty, the model considers realistic traffic generation, operators and pilots' interactions with drones and the ecosystem, and both collaborative and non-collaborative surveillance networks. Also, the thesis proposes a novel authorization framework to strategically deconflict operations considering fairness and congestion management criteria. The proposal goes beyond European regulations, but it is based on them; is distributable, flexible and allow operators to declare their preferences through a set of token-gated priority queues.

Finally, on decision-support systems, a hybrid (agent-based and network-based) simulation framework is presented to enable risk-informed decision-making on interconnected critical infrastructures. The model is instantiated in a healthcare and ICT decision use-case to demonstrate that it can provide multi-level quantitative metrics to guide both operational and strategic decisions. Finally, to inform regulatory decision-making on the European U-Space deployment, the thesis provides insights on the airspace usage efficiency of different aerial operation representation models or flight plans. Through simulation experiments, it is possible to conclude that considering the underlying aerial trajectory allows to increase the number of approved operations.

In sum, the thesis proposes a diverse set of novel technological solutions to advance the deployment of drone-supported smart territories, focusing on key aspects under a common conceptual framework, and filling some relevant gaps in the literature. Proposed architectures, models and applications have been demonstrated through relevant experiments or proofs of concept.

Resumen

La tecnología se ha integrado en los asentamientos humanos, convirtiéndose en impulsor de la prosperidad económica, social y ambiental. Esto se recoge en el concepto de territorios inteligentes, que son aquellos asentamientos urbanos o rurales que utilizan soluciones tecnológicas para mejorar su eficiencia y resiliencia. Los principales pilares tecnológicos de los territorios inteligentes son: la instrumentación, la interconexión y la inteligencia a través del Internet de las Cosas (IoT); la provisión de servicios inteligentes y la toma de decisiones basada en datos. Además, los drones pueden ser otro habilitador al facilitar la monitorización de grandes áreas, proporcionar comunicaciones en zonas remotas o prestar nuevos servicios de movilidad aérea. En este contexto, esta tesis propone soluciones novedosas para cada uno de los pilares tecnológicos, centrándose en el uso e integración de drones.

En cuanto a soluciones IoT, se explora el uso de drones y teléfonos inteligentes como sistemas de monitorización ubicua. Se proponen dos arquitecturas considerando los sistemas IoT como procesos de Fusión de Datos y Gestión de Recursos. Para la vigilancia terrestre de grandes áreas, la tesis propone una arquitectura que permite el control autónomo y explicable de una flota de drones. La propuesta ha sido desplegada en un entorno de pruebas interior, demostrando que permite identificar la tarea realizada por cada dron y proporcionar información de vigilancia. Para la atención sanitaria ubicua, se presenta una arquitectura de referencia para sistemas de salud móvil (mHealth) que busca estandarizar y agilizar el despliegue de estos sistemas basados en teléfonos inteligentes. Un caso de uso basado en la monitorización de pacientes con SARS-COV-2 sirve como prueba de concepto.

Respecto a los servicios inteligentes mediante de drones, se requieren sistemas de Gestión del Tráfico No Tripulado que coordinen de manera segura su operación. En este tema, la tesis propone un sistema de simulación para representar, probar y validar el ecosistema europeo U-Space. Este ecosistema es un servicio inteligente que supervisa y monitoriza las operaciones de drones mediante la interconexión de drones, operadores y redes de vigilancia. Como novedad, el modelo considera una generación de tráfico realista, las interacciones de operadores y pilotos con los drones y el ecosistema, y redes de vigilancia colaborativas y no colaborativas. Además, la tesis propone un novedoso flujo de autorización para la autorización estratégica de las operaciones considerando criterios de equidad y gestión de la congestión. La propuesta se inspira en la normativa europea, sobrepasándola; es distribuible, flexible y permite a los operadores declarar sus preferencias mediante un conjunto de colas de prioridad controladas por tokens.

Por último, en cuanto a los sistemas de apoyo a la toma de decisiones, se presenta un modelo de simulación híbrido (basado en agentes y con redes) para informar decisiones sobre riesgos en infraestructuras críticas interconectadas. El modelo se aplica a un caso de uso con sistemas hospitalarios y de comunicaciones, demostrando que proporciona métricas multinivel para guiar decisiones operativas y estratégicas. Finalmente, para informar decisiones regulatorias sobre el U-Space, la tesis aporta datos sobre la eficiencia en el uso del espacio aéreo de diferentes modelos de representación de operaciones aéreas o planes de vuelo. A través de experimentos de simulación, se concluye que considerar la trayectoria subyacente aumenta el número de operaciones aprobadas.

En resumen, la tesis propone un conjunto amplio de soluciones novedosas para avanzar en el despliegue de territorios inteligentes soportados por drones. Se centra en aspectos clave bajo un marco conceptual común y aborda algunos problemas no tratados en la literatura. Las tecnologías, modelos y aplicaciones propuestos han sido probados mediante experimentos relevantes o pruebas de concepto.

Table of Contents

Acknowledgement v
Abstract vi
Resumen viii
List of Figures xii
List of Tables xv

1 Introduction: a conceptualization of drone-supported smart territories 1
1.1 Technological features of smart-territories 2
1.2 Drones as technological enablers of smart-territories 7
1.3 An architecture for drone-supported smart territories 10
1.4 Thesis contributions and structure 11

2 Monitoring architectures for smart territories 13
2.1 An architecture for multi-drone wide area surveillance operations 16
2.1.1 Review on UAS systems for multi-UAV surveillance operations 18
2.1.2 Description of the proposed multi-UAV control and sensing system . 19
2.1.3 A testbed for development and validation of aerial algorithms 22
2.1.4 Proof-of-concept implementation and execution results 26
2.1.5 Conclusion and limitations 28
2.2 An architecture for ubiquitous health monitoring though mHealth applications 30
2.2.1 Review on computing architectures for health monitoring 32
2.2.2 Data fusion in mobile health applications 33
2.2.3 A multipurpose mHealth architecture 39
2.2.4 Validation: architecture instantiation for SARS-COV-2 monitoring . . 46
2.2.5 Conclusion and limitations 49

3 Fair Unmanned Traffic Management: smart services to enable drone operations 53
3.1 An introduction to Unmanned Traffic Management systems 54
3.1.1 An introduction to U-Space: the European UTM concept 55
3.1.2 U-space: operational view 55
3.1.3 U-space: services view 56
3.1.4 U-space: architecture view 59
3.1.5 U-space surveillance information: collaborative identification 61
3.1.6 U-space surveillance information: non-collaborative sensors 67

3.1.7 U-space: deployment status and challenges . . . 69
3.2 Modeling and simulating the U-space ecosystem . . . 72
3.2.1 Review on simulation systems for the U-space ecosystem . . . 73
3.2.2 USPASIM: U-SPAce ecosystem SIMulation framework . . . 77
3.2.3 Modeling human behavior in the strategic phase: operator agent . . . 80
3.2.4 Modeling human behavior in the tactical phase: pilot agent . . . 92
3.2.5 Modeling collaborative surveillance sensors . . . 101
3.2.6 Modeling non-collaborative surveillance sensors . . . 115
3.2.7 Modeling surveillance networks . . . 122
3.2.8 USPASIM execution example . . . 124
3.3 Managing fairness and congestion: a token-based framework for U-Space authorization . . . 128
3.3.1 Review on fairness and capacity management methods . . . 130
3.3.2 Token-based authorization framework . . . 133
3.3.3 Policies for token distribution and airspace cost setting . . . 144
3.3.4 Experimental setup . . . 150
3.3.5 Results and discussion . . . 155
3.3.6 Conclusion . . . 165

4 Simulation-based decision-making in smart territories 169
4.1 A hybrid simulation framework for operational and strategical resiliency enhancing decision making . . . 171
4.1.1 Review on decision-support systems for smart, resilient territories . . . 174
4.1.2 A conceptual model of smart territories critical infrastructures . . . 178
4.1.3 DSS based on hybrid simulation . . . 185
4.1.4 Instantiation of the DSS for healthcare resiliency enhancement . . . 194
4.1.5 Validation use-case . . . 204
4.1.6 Conclusion . . . 213
4.2 Decision making at regulatory level: leveraging USPASIM platform for decision making . . . 216
4.2.1 Description of the regulatory decision problem . . . 217
4.2.2 Decision alternatives: review on unmanned aerial operation representation models . . . 218
4.2.3 Simulation-based decision-making support: USPASIM setup . . . 222
4.2.4 Comparative results on airspace occupation and computational efficiency to guide the decision process . . . 224
4.2.5 Discussion and conclusion: simulation-based u-plan representation insights for decision-makers . . . 229

5 Conclusion 231
5.1 Thesis contributions . . . 234
5.2 Limitations and future work . . . 235

References 237

List of Figures

1.1 Examples of technological architectures enabling smart territories 4
1.2 Smart territories technological enablers. 7
1.3 Elements within a Unmanned Aircraft System. 8
1.4 Conceptual architecture of technological elements in smart cities. 11

2.1 Data fusion and Resource Management processes as viewed in the JDL model 14
2.2 Technological elements in the proposed drone-enabled smart-safety monitoring use-case 17
2.3 Logical architecture of the proposed multi-UAV control and sensing system. . 20
2.4 Correspondence between virtual and real worlds entities in the proposed digital twin. 23
2.5 Digital twin system enabling indoor testing and validation of UAS applications. 24
2.6 Digital twin deployment in a physical lab. 25
2.7 Web interface that displays the Swarm Operational Picture. 27
2.8 Hardware and deployment architecture of the proof-of-concept multi-UAV control system. 27
2.9 Multi-drone control architecture experiment results. 28
2.10 Technological elements in the proposed mHealth-based monitoring architecture 31
2.11 Stages of the clinical process as data fusion layers. 35
2.12 Proposed multipurpose mHealth architecture. 40
2.13 Workflow in the proposed mHealth architecture. 41
2.14 Proposed multipurpose mHealth architecture. 48
2.15 IMUP application as an example of the proposed mHealth fusion architecture 49

3.1 Technological elements required for deploying a UTM system in a smart-territory 54
3.2 U-space services and expected deployment phases. 57
3.3 U-space architecture alternatives. 61
3.4 Entities considered in U-Space simulation platforms 72
3.5 USPASIM architecture 79
3.6 Operator agent interactions 81
3.7 Considered flight patterns to model u-as operations 84
3.8 Sequence diagram representing modeled interactions in the strategic phase. . 87
3.9 Alternative deconfliction strategies. 89
3.10 Example flight scenario generated by superposing flight patterns. 91
3.11 Example of simulated deconfliction process within the Operator agent. . . . 91

3.12 Pilot agent interactions . . . 92
3.13 Pilot decision engine finite state machine for mission start-up . . . 95
3.14 Pilot decision engine finite state machine for mission execution . . . 97
3.15 Pilot decision engine finite state machine for mission termination . . . 98
3.16 Alternative flight paths in polygonal flight phases . . . 99
3.17 Example of simulated flights as directed by the models in the `Pilot` agent. . 101
3.18 Considered flight patterns to model uas operations . . . 102
3.19 Considered flight patterns to model uas operations . . . 103
3.20 Relative visibility of emitters and received in a shared radio medium. . . . 106
3.21 MAC model for an emitter and a receiver. . . . 107
3.22 Physical description model . . . 110
3.23 Transmission errors and error correction model. . . . 111
3.24 Coverage analysis of collaborative protocols under proposed model . . . 114
3.25 Collision analysis for each collaborative protocol under the proposed model . 115
3.26 Interactions of non collaborative sensor agents with other agent's in USPASIM 116
3.27 General model of a non-collaborative sensor . . . 117
3.28 Plot distribution analysis of a drone detected by radar sensor. . . . 121
3.29 Distance and error analysis of drone plots as detected by modeled radar sensor. 122
3.30 Interactions of surveillance node agents. . . . 123
3.31 Fusion node agent processes . . . 123
3.32 Simulation scenario description . . . 124
3.33 Authorization results as depicted in the connected USSP. . . . 125
3.34 Telemetry information forwarded to the USSP by collaborative drones. . . . 126
3.35 Tracks forwarded to the USSP by surveillance sensors. . . . 127
3.36 Comparison between the regulation mandated FC-FS approach and the adopted deferred authorization scheme. . . . 134
3.37 Comparison between the single priority mandated approach and the adopted multi-priority approach. . . . 135
3.38 Graphical representation of the token-controlled access to priority classes in the BTA workflow. . . . 139
3.39 Representation of the weighted token cost computation. . . . 141
3.40 Authorization Policy Service interactions. . . . 142
3.41 Sequence diagram of interactions between U-Space services when implementing the proposed deconfliction framework. . . . 143
3.42 Representation of different policy objectives and encoding within the token-based framework . . . 144
3.43 Scenario 1 results. Analysis of the dependence of the flight authorization rate and the submission time. . . . 156
3.44 Scenario 1 results. Comparison by token allocation policy . . . 157
3.45 Scenario 1 results. Analysis of the dependence of the flight authorization rate and the operator perceived criticality/preference for each operation. . . . 158
3.46 Scenario 1 results. Analysis on the impact on efficiency of each implemented policy. . . . 159
3.47 Scenario 1 results. Analysis on the impact of policy to different operator behaviors. . . . 160

3.48 Scenario 2 results. Comparison of airspace cost comparison per operator type 161
3.49 Scenario 2 results. Comparison per operator type. 162
3.50 Scenario 3 results. Geographical comparison of airspace congestion without and with the implemented policy 163
3.51 Scenario 3 results. Statistical comparison of airspace occupation without and with the implemented policy 164
3.52 Scenario 3 results. Occupation and authorization rate analysis. 164

4.1 Technological elements in the proposed decision-support system 173
4.2 Example of influence diagram representing interdependencies of the electricity grid system. 183
4.3 Comprehensive view of the interdependencies among all considered systems. 184
4.4 Agent and system dimensions within a smart territory 187
4.5 Agent as a set of sub-agents modeling inter-system dependencies 188
4.6 Proposed multi-layered agent-based model 190
4.7 Typical steps in a risk-informed decision making process 191
4.8 Modular class design for agents and sub-agents based on multiple inheritance 192
4.9 Systems, agents, and sub-agents considered in the proof-of-concept healthcare and ICT simulator 195
4.10 Disease evolution model in Patient subagents 198
4.11 Hospital subagent model with different healthcare services 200
4.12 Attention quality effect on recovery probability 202
4.13 Validation scenario 205
4.14 Validation scenario results at both agent and aggregate level 208
4.15 Hospital B health services occupancy for different threat levels 210
4.16 Utilization rate of Hospital B services for the different threat levels 211
4.17 Contingency table with the number of accumulated deaths as merit function 212
4.18 Technological elements in a digital-twin for smart healthcare and infrastructures 214
4.19 Technological elements in the proposed decision-support system 216
4.20 Different alternative u-plan representation models studied in this paper. . . . 219
4.21 UML class diagram indicating the parameters used in each of the U-Plan alternative representations under analysis. 219
4.22 Geographical distribution and shapes of flights concurrently flown in the test scenario in a given time instant. 223
4.23 Horizontal (i.e.: 2D view) of a same operation represented with the four considered u-plan formats. 225
4.24 Comparison of height reservations depending on u-plan representation 226
4.25 Comparison of time reservations depending on u-plan representation 226
4.26 Comparison of conflict detection results depending on the type of u-plan representation. 227
4.27 Analysis on u-plan occupation efficiency. 228
4.28 Analysis on u-plan computational costs. 228

5.1 Smart territory technological elements considered across the thesis 231

List of Tables

2.1 Common mHealth features identified in each of the considered health data fusion layers. . . . 39
2.2 Services description and implementation guidelines in the mHealth architecture (1/2). . . . 43
2.3 Services description and implementation guidelines in the mHealth architecture (2/2). . . . 44
2.4 mHealth data fusion features required for the SARS-CoV-2 use case 47

3.1 ADS-B protocol parameters 63
3.2 FLARM protocol parameters 65
3.3 Remote ID protocol parameters 66
3.4 Comparison of Non Collaborative drone detection technologies 69
3.5 Ongoing SESAR research topics. 70
3.6 Comparison of U-Space simulation literature proposals 76
3.7 Operator agent specification parameters 82
3.8 Parameters needed to model traffic patterns and associated mission types. . 85
3.9 Simulated messages considered in the GCS-drone C2 channel. 93
3.10 Particularisation of the collaborative surveillance model for each protocol. . . 112
3.11 Parameters of active radar sensor model and example representative values . 118
3.12 Considered scenarios description 155

4.1 Comparison of modeling techniques for smart-territories decision support systems. 177
4.2 Declared critical infraestructures by country and proposed city systems. . . . 180
4.3 Agents considered in the implemented DSS simulation model 195
4.4 Subagents considered in the implemented DSS simulation model 196
4.5 Aggregated parameters modeling disease evolution 201

5.1 Summary of thesis contributions 234

Abbreviations and acronyms

ABM Agent Based Model.

AI Artificial Intelligence.

AMC/G Acceptable Means of Compliance/Guidance.

ANSP Air Navigation Service Provider.

AR Agent Interaction Rules.

ATCO Air Traffic Controller.

ATM Air Traffic Management.

BVLOS Beyond Visual Line Of Sight.

C-UAS Counter UAS.

C2 Command and Control.

CI Critical Infraestructure.

CISP Common Information Service Provider.

COP Common Operational Picture.

DAA Dectect And Avoid.

DB Database.

DCB Demand and Capacity Balancing.

DF-RM Data Fusion and Resource Management.

DL Deep Learning.

DSS Decision Support System.

EASA European union Aviation Safety Agency.

EHR Electronic Health Records.

EO/IR Electro Optical/Infra-Red.

EVLOS Extended Visual Line Of Sight.

eVTOL electric Vertical Take-Off and Landing.

FFTx Fiber To The x (node, home...).

GCS Ground Control Station.

GIS Geographic Information System.

GNSS Global Navigation Satellite System.

HAPS High-Altitude Platform Station.

HIS Hospital Information Systems.

HS Healthcare System.

ICT Information and Communication Technologies.

IoT Internet of Things.

IR Internal Rules.

ISAC Integrated Sensing And Communications.

ISR Intelligence, Surveillance and Reconnaissance.

JDL Joint Directors or Laboratories.

KPI Key Performance Indicator.

MAC Medium Access Control.

MAS Multi-Agent System.

MIMO Multiple-Input Multiple-Output.

MTBF Mean Time Between Failures.

NOTAM Notice To Airmen. Aeronautical notices to alert aircraft pilots of potential hazards or restrictions.

PCA Principal Component Analysis.

PID Proportional Integral Derivative controller.

RIS Reconfigurable Intelligent Surface.

RL Reinforcement Learning.

RTH Return To Home.

SCD Strategic Conflict Detection.

SCR Strategic Conflict Resolution.

SDR Software Defined Radio.

SDSP Suplemental Data Service Provider.

SESAR Single European Sky ATM Research Joint Undertaking.

SR System Interaction Rules.

SVM Support Vector Machine.

SWAPC Size, Weight, Power and Cost.

UAM Urban Air Mobility.

UAS Unmanned Aircraft Systems.

UAV Unmanned Aerial Vehicle.

USPASIM U-SPAce ecosystem SIMulation framework.

USSP U-Space Service Provider.

UTM Unmanned Traffic Management.

V2V Vehicle to Vehicle.

VANET Vehicular Network.

VLOS Visual Line Of Sight.

WSN Wireless Sensor Network.

Chapter 1

Introduction: a conceptualization of drone-supported smart territories

In the last decades, Information and Communication Technologies (ICT) have become a transversal enabler of developed societies. Indeed, modern communities have been erected on technological advances such as low Size, Weight, Power and Cost (SWAPC) sensors, ubiquitous fast communication networks and information processing capabilities. In this sense, human settlements (both in cities and rural areas) have integrated sensor networks, intelligent services and data-driven decision making processes to improve their efficiency, resilience, sustainability and citizens' quality of life.

This technological integration process, aimed at solving urban challenges, is generally referred with the concept of **smart city**. However, the term is usually vague and multiple definitions can be found in literature. This limitation is discussed in [SSG23] which identifies two different approaches in defining smart-cities. On the one hand, definitions with a techno-centric approach focus on the implementation of technological tools to enhance the operation of cities' infrastructures:

- Authors of [Hal00] defined a smart city as a *"city that monitors and integrates conditions of all of its critical infrastructures including roads, bridges [...] can better optimize its resources, plan its preventive maintenance activities and monitor security aspects while maximizing services to its citizens"*.
- Also, in [Har+10], smart cities are described as *"urban areas that exploit operational data, such as that arising from traffic congestion, power consumption statistics and public safety events to optimize the operation of city services"*.

On the other hand, definitions with a multidimensional or holistic approach can also be found in literature, considering other aspects such as economy, participative governance, or knowledge generation. In this sense:

- Authors of [GPN15] identify seven components required in smart cities: natural environment, city infrastructure, knowledge economy, human capital and creativity, governance and collaboration, institutional arrangements, city administration and public services;

with ICT technologies and data being a transversal tool.

- Similarly, in [Dam13] a smart city is defined as *" a well defined geographical area, in which high technologies such as ICT, logistic, energy production, and so on, cooperate to create benefits for citizens in terms of well being, inclusion and participation, environmental quality, intelligent development; it is governed by a well defined pool of subjects, able to state the rules and policy for the city government and development".*
- For authors of [KS21], a smart city is *"a city that combines information and communication technologies with human and social capital and public institutions to dynamize it is economic, social, environmental, and cultural development".*

To overcome this duality, a comprehensive concept is proposed by [MCK16]. It defines the smart city as a urban settlement with the following components: smart infrastructure, smart transportation, smart energy, smart healthcare, smart governance and smart citizens. Where "smart" refers to the ambition to improve economic, social and environmental standards using ICT as the backbone of the process.

The lack of a common idea of what a smart city is also extends to its application scope. Cities' are complex systems composed of multiple inter-reliant critical infrastructure systems, social and economic interactions and governance networks [MNS16]. Thus, their geographical and influence boundaries can be intertwined with those of other cities and rural areas. Moreover, a similar "smart" concept can be applied to other smaller (or larger) geographic contexts prompting new terms such as smart-villages or smart-regions [GNW22]. In this regard, authors of [NRG20] coin the term **smart territory**. It extends the smart city notion to territories: broad, dynamic geographic spaces defined from socio-economic dependences rather than administrative boundaries. This way, ICT enablers should be considered to address both urban and rural issues. This thesis embraces this second term to put the focus on technologies and applications that are valid for both urban and sparse geographical areas.

1.1 Technological features of smart-territories

Without detriment to the sociological aspects that literature includes within the smart territories concept, and given the subject of the doctoral program, this dissertation must undoubtedly focus on the technological aspects. These technological features will be presented in this section, including the Internet of Things (IoT) concept, smart services or applications and data-based decision support systems.

[MCK16] and [Ugw+24] identify the **IoT** as the backbone of smart territories. Under this paradigm, physical objects are interconnected with systems and applications in order to provide smart, reliable and secure services. As discussed in [Har+10], the IoT concept is built on three technological features:

- **Instrumentation**, which refers to the utilization of data acquisition devices (i.e.: sensors) to enable real-time monitoring of smart territories. In this sense, affordable, energy efficiency wireless sensors for environmental variables (e.g.: temperature, humidity, light) or positioning and identification (e.g.: RFID tags) allow a massive deployment

in cities to track real-word physical systems (e.g: infrastructures) status over time [SG19]. Also, widespread usage of smartphones with embedded sensors makes them a pervasive monitoring platform that have already been used to monitor citizens health, noise pollution [Kri21], and even earthquakes [Kon+15]. Finally, instrumentation does not only refer to monitoring physical entities, but can also be extended to virtual probes that capture information from social networks, trading systems or, in general, business processes.

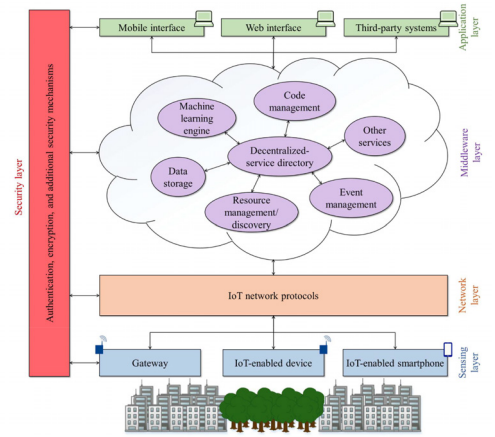
- **Interconnection** refers to an end-to-end integration of captured data, so that information can accessed and visualized anywhere, and meaningful fused insights may be generated. To this end, both physical connectivity (through communication networks) and logical data aggregation (through big data and cloud computing platforms) is required.

  As for **communications**, bandwidth, latency, range, mobility (i.e.: wired/wireless) and energy efficiency requirements are heterogeneous across diverse IoT applications [Yaq+17]. This means that there is a wide range of technologies ranging from LoRaWan for use cases with low data rate, low power consumption and high range; or ZigBee applications where shorter ranges are required. Similarly, different alternatives exist for applications requiring high bandwidths: fiber-based technologies (Fiber To The x (node, home...) (FFTx)) can be used if mobility is not required, with WiFi or public mobile networks (e.g.: LTE, 5G) being an alternative otherwise. Gradual deployment of 5G technology will help to homogenize the connectivity solution thanks to its different application profiles: Enhanced Mobile Broadband (eMBB), Ultra Reliable Low Latency Communications (URLLC), and Massive Machine Type Communications (mMTC) [Yan+22]. In the future, 6G will continue on these improvements.

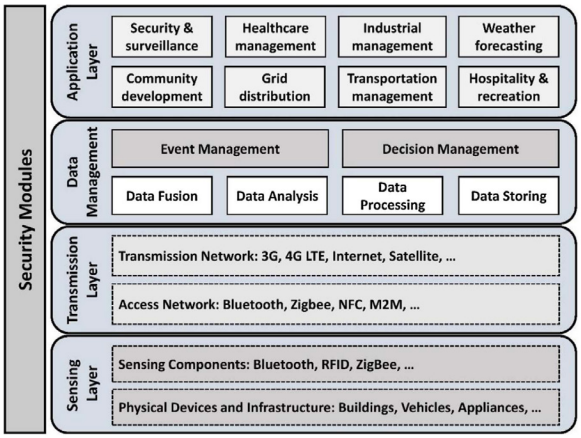
  Given the large amount of data collected and exchanged, **Big Data** techniques based on distributed data management and parallel processing are required, as discussed in [Has+16]. These platforms must enable data storage (in data lakes such as Amazon's S3), data interoperability (harmonising different data formats), visualization (interactive dashboards that can be integrated with Geographic Information Systems (GISs)) and real-time processing (usually through cloud-computing infrastructures and frameworks such as Spark or Hadoop). In this sense, different general-purpose technological stacks for smart cities exist in the literature. An example of those is the FIWARE framework [FIW24], which aims to be a comprehensive open source platform to interconnect and process smart territories data in a scalable way.

- **Intelligence** consists in including complex analytics, modeling and optimization techniques to improve the operation of smart territories and the provision of smart services. In this context, **Artificial Intelligence (AI)** algorithms are key to find patterns, detect anomalies and provide estimators from the large amounts of heterogeneous data available through the two previous IoT features. The range of learning algorithms is immense and depends on the final smart-application, going from traditional machine learning techniques (e.g.: Support Vector Machine (SVM), Principal Component Analysis (PCA)), to Deep Learning (DL) or Reinforcement Learning (RL)[Ull+20].

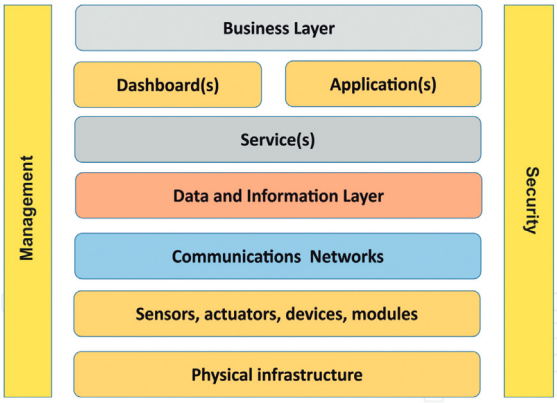
Figure 1.1 depicts some relevant literature examples of technological architectures for the deployment of smart territories. These architectures further support the claim that IoT is the main enabler of smart territories. Indeed, the architectures are articulated around different layers that implement the aforementioned features: a layer for sensors and instrumentation to



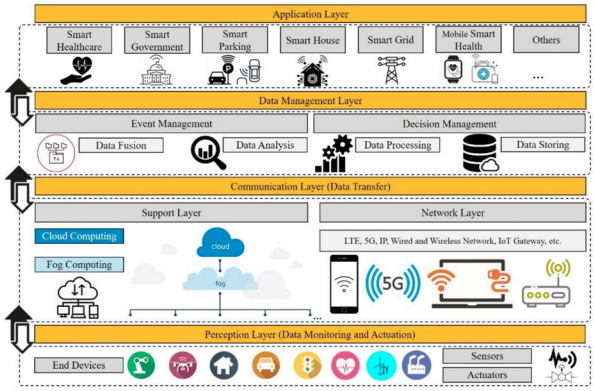
(a) Discussed in [Per+22] as a conceptual IoT model to deliver smart applications (e.g.: smart mobility.)



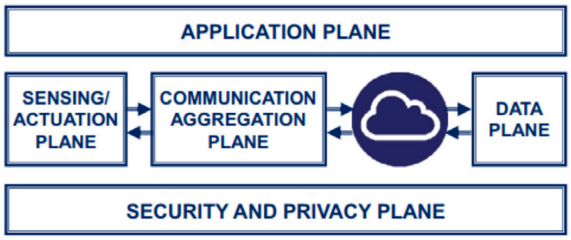
(b) From [SKH18], which proposes the architecture as an aggretated model based on a literature review.



(c) Developed in [Hai+16] as part of a smart grid use case in which the role of different communication technologies is explored.



(d) Proposed in [Tal+21] to enable smart applications deployment for parking, healthcare or environment and weather monitoring.



(e) Presented in [Hab+18] to discuss existing challenges for smart-territories deployment.

Figure 1.1: Examples of technological architectures enabling smart territories

perceive the state of physical entities, a layer for communication networks that interconnect sensors and computing resources, a data management layer that enables data integration and storage, and a layer for data analysis and intelligence. On top of this stack, examples show how it is possible to implement different application-level **smart services** across a wide variety of sectors.

Smart services are final applications (i.e.: oriented to consumers, citizens, infrastructure operators) that take advantage of the insights provided by real time data and connectivity (i.e.: the IoT paradigm) to improve the provision of a preexisting service or even provide new, innovative services. As discussed in [Tal+21], [Has+16], [SKH18] or [Per+22], smart services may include:

- **Smart transportation/mobility**. [Ola+23] reviews multiple cases where the use of IoT technologies can improve transportation systems in cities. For example, regarding road traffic, drivers positioning information from smartphones can be used to estimate congestion and plan routes intelligently to reduce delays. Similarly, traffic light cycles can also be adjusted in real time to favor the most overloaded traffic flows. In addition, the interaction of multiple drivers and travelers through intermediation platforms (such as Uber) can lead to ridership services that are a new mobility alternative in smart territories [MKA21]. Currently under development, Vehicular Networks (VANETs) will become an enabler of autonomous driving, that is expected to enhance safety and efficiency [LA21].There are also examples in other modes of transport such as railways where an IoT stack can be used to improve infrastructure maintenance [JKK18], or 5G communications can be used transversely for train signaling and automatic control [Ai+20].
- **Smart utilities and infrastructures**. A smart electrical grid makes it possible to optimize renewable energy generation and consumption processes (e.g.: integrating smart cars) in a way that improves energy efficiency and ensures grid stability [Fan+12]. To this end, smart meters make it possible to extract detailed demand patterns for each household, monitor the state of the network in real time and generate new forms of pricing with hourly discrimination [Kab16]. Similarly, smart water networks make it possible to quickly detect water leaks, pipe breaks and monitor water quality in real time, as discussed in [Gup+20] and [Lal+21]. Also, flood monitoring systems, as in [Gar+20] can provide early warning capabilities by monitoring water bodies, and include predictive models to automate water discharge in dams and improve territories' resiliency to flood-related disasters and droughts. Finally, as for urban structures and geotechnical systems, Wireless Sensor Networks (WSNs) can help to monitor slopes to alert of possible landslides. Tunnels and bridges can also be monitored in real time through connected gauges, accelerometers, or thermometers [Ber+20].
- **Smart buildings and factories**. Automated access control, room occupation tracking, efficient heating management, connected appliances and lighting are some of the common features of smart buildings [Jia+19]. This concept goes a step further in the case of factories, as it also refers to managing business operations with the help of IoT features. This includes predictive maintenance, inventory management, production process and supply chains optimization and quality control monitoring [SAD23].

- **Smart healthcare and safety**. Telemedicine services (in particular, mHealth which will be further discussed in Section 2.2.1) allow to increase the range of healthcare making ubiquitous and personalizable. Also, hospital management can benefit from IoT features to track patient occupation, drug inventory management or assist in resource allocation [Tia+19]. Meanwhile, from a law-enforcement point of view, networks of connected cameras and automatic processing algorithms allow for comprehensive surveillance systems [Joh19] that will help to fight crime at the expense of privacy concerns. In a broader context, the interconnection and aggregation of different sources of information helps emergency management (e.g., with mass alerts to cell phones [Gha+16] or the coordination of first responders [Bar+15]).

In addition to these smart-services oriented either to end users, or to improve services' operation; another key element enabled by IoT is **informed decision-making**. As discussed in in [Zha+25], [Ma+19], or [SE21]; this consists in the practice of taking regulatory and operational decisions based on quantitative information rather than experience or intuition. In this sense, **Decision Support Systems (DSSs)** are technological platforms that leverage IoT features to enable this process. As analyzed in [Chi14], different types of DSSs exist:

- **Data-based DSSs** are information systems that integrate multiple sources of relevant data and analyze it to generate insights that are useful for decision-making purposes. These tools rely on visualization and data analysis algorithms to identify trends and patterns.
- **Model-based DSSs** go a step further by providing prediction capabilities that allow simulating and comparing the outcomes of different scenarios (i.e.: what-if analysis). To do so, they may rely on statistical/mathematical models built on data (e.g.: a forecast model) or on full-fledged simulation models of complex systems (e.g.: digital twins). Such tools allow to implement an optimization loop (i.e.: simulation-based optimization) to tune a given decision alternative to increase its impact while also considering uncertainty.
- **Communication-based DSSs** provide collaboration and communication tools for a team of decision-makers so that information can be promptly shared. They allow to agree on decisions by consensus and to coordinate their implementation.

Figure 1.2 depicts a summary of the main technological enablers of smart territories discussed in this section. Nowadays, there exist partial deployments of the smart territory concept using existing technologies (mainly in cities, such as Porto in Portugal [San+18]). However, current technologies pose some **limitations to enable a broad adoption of the smart territory concept**. In this sense, [Aha+20] cites cybersecurity, interoperability across preexisting systems, and lack of appropriate standards as some of the current knowledge gaps. Also, high adoption cost of new technologies is preventing a successful roll out of smart city projects. Similarly, [Ali+23] identifies security and cost of operation issues as some of the current technological blockers. Additionally, authors identify ubiquitous connectivity as a pending task, as ensuring full network coverage can be challenging in remote locations. Overall, in extending the smart city reach to embrace the smart territory concept by covering wider areas, **economical costs and connectivity issues** arise as the main problems. In this sense,

alternative technologies to enable remote sensing and increase communications coverage can help to mitigate this gap.

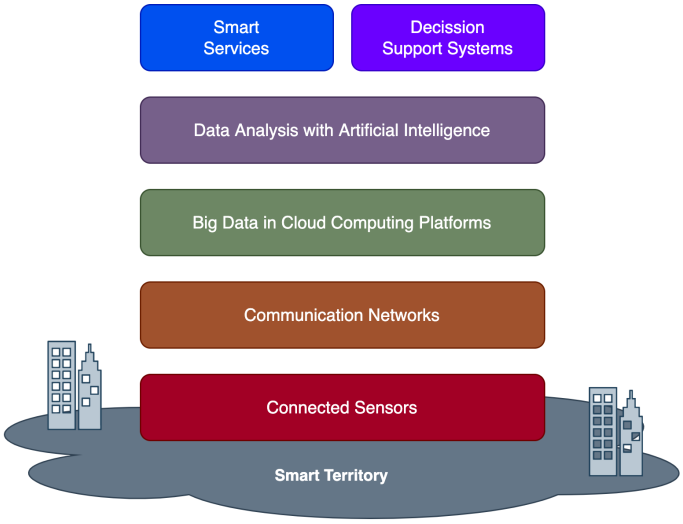


Figure 1.2: Smart territories technological enablers.

1.2 Drones as technological enablers of smart-territories

One of such enablers that might help scale smartness to greater geographical areas are drone systems. Usually referred as **Unmanned Aircraft Systems (UAS)** in technical terminology, drones allow to capture information from points of view otherwise inaccessible or economically unfeasible. In this sense, the development of economical and small-sized aircraft has opened up a new set of use cases and opportunities that are being exploited by companies and administrations. As a result, by 2050 it is expected that the commercial drone fleet at European level will reach 400.000 units, along with about 7 million aircraft for private use according to [SES16]. In fact, this trend is already happening as demonstrated by figures in [JUP25], which declares that 600.000 drones have been already registered in Germany or 100.000 in Spain (as of 2024). Similarly, authors of [Cla+25] have deployed drone detection equipment in 14 sites across Europe. With this reduced sampled, the project has been able to identify 35.000 individual flights in 2024.

As depicted in Figure 1.3 , UAS are composed of multiple subsystems. The aircraft, also known as the Unmanned Aerial Vehicle (UAV), is controlled remotely from a Ground Control Station (GCS) that exchanges messages with the UAV through a Command and Control (C2) channel. Such a channel may use an ad-hoc radio interface (mainly for Visual Line Of Sight (VLOS) missions, which are short range and allow visual supervision) or rely on public communication networks (mainly for Beyond Visual Line Of Sight (BVLOS) missions, which require remote supervision and navigation based on Global Navigation Satellite System (GNSS)). In any case, mission control and supervision may be performed by a human operator (i.e.: a pilot) or rely on autonomous control algorithms (albeit regulatory limitations). Thus, low latency and high reliability are key features for the C2 channel.

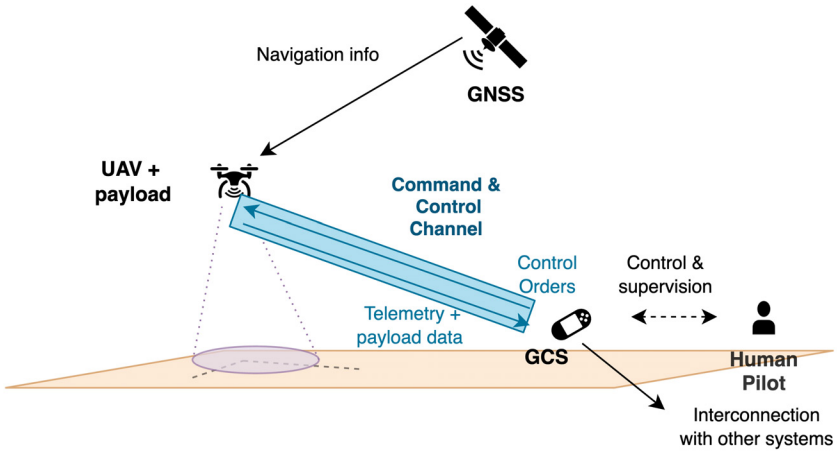


Figure 1.3: Elements within a Unmanned Aircraft System.

Regarding cellular-based C2 channels, UAV's altitude and high speed mobility present a challenge for existing cellular communication technologies (i.e.: 4G and 5G). As mentioned in [SBK21], base stations are usually oriented to serve users on the ground. Consequently, the aerial segment is covered by the antenna's sidelobes, which implies lower antenna gains and the existence of transmission zeros. This can result in significant fluctuations and interruptions of the received signal as the UAV moves. Furthermore, the lack of ubiquitous, high bandwidth coverage may be an impediment to some of the UAS use-cases reviewed below. In the future, the next generation of mobile communications (i.e.: 6G) is expected to address these issues, with aerial communications being one of the application scenarios cited in [Dan+20]. In this sense, and as discussed in [MLH21], massive Multiple-Input Multiple-Output (MIMO) is expected to perform adaptive 3D beamforming with narrow beams providing continuous coverage to drones 3D mobility. Also, Reconfigurable Intelligent Surfaces (RISs) may be used to reflect some of the ground-oriented power towards the sky. Drone location and high-precision localization capabilities are also foreseen thanks to Integrated Sensing And Communications (ISAC). As for coverage in remote areas, 6G technology is expected to integrate terrestrial and non-terrestrial (mainly satellite, but also High-Altitude Platform Stations (HAPSs)) providing seamless handover and even simultaneous usage. All this new enabling technologies together with the expected improvement in supported connection density and peak data rates [Liu+25] mean that 6G technology will help to accelerate the deployment of the drone ecosystem.

Coming back to the description of drone systems, UAVs can be equipped with a variety of onboard payloads (including RGB cameras, hyperspectral cameras, LIDAR or environmental sensors) that gather information and forward it to other systems directly or through the GCS. In this sense, 5G and 6G technologies enable the deployment of AI algorithms at the edge for low-latency data analysis. Overall, and as thoroughly discussed in [Moh+20], UASs can be used as **remote, mobile sensors** that can collect relevant information across different domains:

- **Industry and construction**. The use of drones to perform inspections of dangerous or inaccessible infrastructures such as telecommunication towers has been previously explored at [Bes+18]. In the construction field, the use of drones for the verification of requirements on construction sites has been reported in [Ash+16].
- **Mining and energy**. This includes the usage of drones for inspections of oil pipelines [Koc18] or high voltage lines. Remote sensing applications in this domain also include reconnaissance and mapping of mining areas in order to be able to estimate production [MD17].
- **Agriculture**. In this field, drones are being used to analyze the state of crops and to apply precision agriculture [Mur+16].
- **Mobility and transportation**. At present, UAS are already being used for road traffic estimation and highway pavement status monitoring, as discussed in [OMA20]. Also, railway track, tunnel and bridges inspections are explored in [Ael+24].
- **Security**. Drones are already being used for real time transfer of images (RGB or even heat maps) of emergencies such as fires, earthquakes or floods; which improve situational awareness [Swi16] . The use of automated drones is also being explored for the observation and detection of suspicious behavior over long periods of time, as in the case of the use of drones to detect non-compliance with coronavirus prevention measures in Singapore [US 20]

Regarding other drones usage, literature also shows how drones can be used to **interconnect other entities providing both communications and computing capabilities** [GZ23]. In this sense, UAS may be equipped with a mobile public communications (e.g.: LTE, 5G) base station in order to act as an aerial communication platform that complements terrestrial networks. As a result, it is possible to enhance communications coverage [WDZ19], temporary increase network capacity [LZZ18] or speed up service recovery after a disaster [MG15]. Also, UAVs may be used as relay nodes to gather data from a network of ground sensors, as in [Liu+21] where drones are optimally distributed to reduce power transmission required by those sensors. Continuing in the WSN context, drones may be used to wirelessly transmit power to ground devices in order to extend their lifetime in an affordable and robust way [XZZ18]. Finally, UAVs may be equipped with computing capabilities and serve as an edge server in which different users can offload latency-intensive tasks, as in [Du+18].

UASs can also be used to enable novel **smart aerial services** across different domains. In this sense, drones can be used for **delivery of goods** in time sensitive scenarios or to remote locations. From a commercial point of view, last-mile retail parcel delivery with an autonomous fleet of drones is being tested in operational environments by Amazon Prime Air [Ama23] and Google Wing [Win25]. However, there are still doubts about the feasibility of this concept from different perspectives: economic (e.g.: [AGC22] only detects a viable market in a 16-43 % of European population), social (e.g.: noise, acceptance of safety levels cited in [SV22]) and technological (e.g.: development of electric Vertical Take-Off and Landing (eVTOL) drones). A similar approach is also been explored for healthcare applications in which drones can delivery required drugs, blood or equipment (e.g.: defibrillators) in emergencies [Cam+24]. Another domain for drone-enabled smart services is **mobility**. In this sense, the

Innovative Air Mobility (IAM) concept aims to take advantage of drones integrated in a multi-modal transportation system to enable safe, secure and sustainable transportation of passengers [EAS25]. A particularization of this concept for the urban environment is known as Urban Air Mobility (UAM) which intends to deploy a network of vertiports across cities to enable passenger drone operations [CSF21].

To carry out these use-cases, a wide variety of aerial platforms are currently available in the market with different tradeoffs between flight endurance, maneuverability and payload weights. In this sense, multi-rotor aircraft provide great maneuverability and stability at the expense of lower flight times [Gar22]. These features make them appropriate for short inspection missions or precision agriculture. For instance, DJI Matrice 400 endurance ranges from 60 minutes with a 1 kg payload to half an hour with its maximum take off weight of 6 kg [DJI25]. Alternatively, fixed-wing drones usually offer greater endurance, larger payloads albeit evident maneuverability limitations. As a result, fixed-wing aircraft are appropriate for long-endurance surveillance mission. As an example, X8 Flying Wing aircraft in [Air25] provides an up-to 3 hour endurance. Other types of aircraft include the aforementioned eVTOL drones that aim to seek a balance between maneuverability and endurance by hybridizing the two previous designs. Also, cautive drones allow for almost infinite flight time duration with zero maneuverability. Overall, in view of the above, an appropriate aircraft selection is critical to a successful deployment of drone-based missions.

Finally, in order to leverage the capabilities of drones while maintaining safety, **Unmanned Traffic Management (UTM) systems** (which will be further discussed in Section 3.1 , particularly the European U-Space concept) are required. In itself, a UTM system is a smart service that mediates between drone operators, drones, and authorities in order to coordinate drone operations with safety and efficiency. Therefore, UTM arise as the enabler of drone operations in smart territories supporting the aforementioned sensing missions, drone-enabled communications and novel aerial services.

1.3 An architecture for drone-supported smart territories

Section 1.1 has discussed how cities are integrating technological solutions in order to improve their efficiency and sustainability. In this context, the IoT technological stack provides sensors, communication and data processing capabilities so that citizens and critical infrastructures can be monitored remotely and in real time. To this end, different technological solutions are possible including smartphones, 5G communication networks, cloud computing or AI. IoT-captured data may be analyzed and processed as part of a set of smart services that can range from smart healthcare to smart safety use-cases. The objective is to enhance the operation of smart territories' critical infrastructures systems and to deliver new service to citizens. Simultaneously, data can be used in data-informed decision making processes that can be complemented with simulation-based DSSs.

Also, in Section 1.2 it has been studied how drones are enablers across the whole IoT stack, being used for sensorization, communication and even computing purposes. Drones high

mobility capabilities is of particular interest in the case of smart territories that consider interdependency relationships that go beyond the administrative geographical borders. Moreover, novel services such as urban transportation or delivery can be provided to citizens. To do that, UTM systems are needed as another smart service that interconnects drones for their safe operation.

As a summary, Figure 1.4 depicts an architecture considering the previously identified technological elements of a smart territory. For each element, different technological solutions are shown. Particularly, it can be seen how drones appear as an enabler of different layers within the architecture. This architecture is not closed, and additional services, DSS methods, sensors or networking solutions may be incorporated.

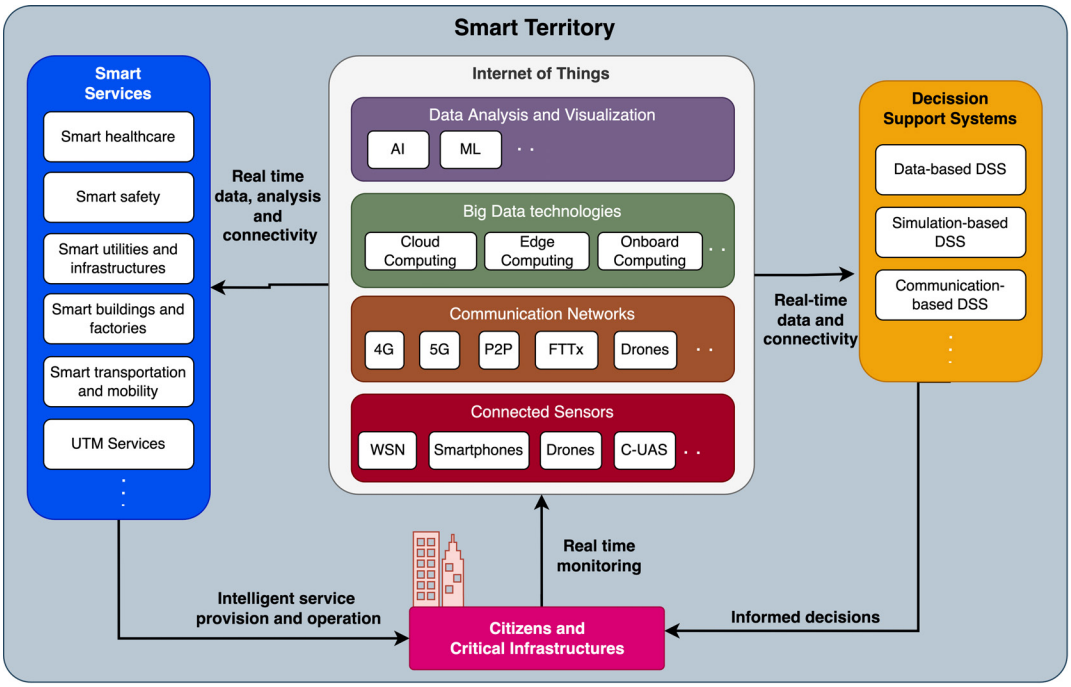


Figure 1.4: Conceptual architecture of technological elements in smart cities.

1.4 Thesis contributions and structure

In line with the architecture depicted in Figure 1.4 , this thesis dissertation presents monitoring architectures, fair drone management systems and simulation-based DSSs for different smart-territories use cases. Moreover, it does so with a special focus on UAS applications, using them as a building block of smart territories. With smart territories and drones as guiding threads, the thesis structure is as follows:

- Chapter 2 presents **architectures for ubiquitous monitoring** through data acquisition, interconnection and processing in smart territories. Particularly:
  - Section 2.1 explores how a swarm of UAVs can be used as a distributed sensor to monitor wide areas in law-enforcement setting. To this end, the section proposes a

control and sensing architecture that can leverage AI algorithms while retaining human supervision and explainability.

  - Section 2.2 addresses data gathering, integration and processing in a healthcare setting. In this application, a comprehensive mHealth architecture is proposed to leverage patient's smartphones for data acquisition. The architecture allows to extend healthcare reach to broad geographical areas while also improving attention quality.

- Chapter 3 is focused on discussing UTM systems (and particularly the European U-Space concept) as an **smart service** that supports a myriad of drone operations in smart territories.

  - Section 3.1 introduces U-Space as a technological ecosystem that relies on real-time information exchange between different parties to ensure safe and efficient aerial operations.

  - Section 3.2 presents U-SPAce ecosystem SIMulation framework (USPASIM): a holistic simulation platform that represents the whole U-Space ecosystem in order to streamline its development, validation and deployment process. Simulation models to consider human intervention and surveillance networks are proposed.

  - Section 3.3 discusses equity and congestion issues that may limit the provision of drone-based smart services. In this sense, the section proposes an alternative authorization framework that considers these issues and allows implementing different notions of airspace access fairness while contemplating operator preferences.

- Chapter 4 addresses how simulation models can be used as a **DSS** for decision-makers to take informed decisions at operational, strategical and even regulatory levels.

  - Section 4.1 presents a hybrid, formal model that combines Agent Based Model (ABM) and network-based models to represent smart territories critical infrastructures. A use-case that considers the interdependencies between healthcare and ICT infrastructures demonstrate how the model can be used at operational and strategical levels.

  - Section 4.2 describes a regulation-level decision-making use case in which the USPASIM platform is used to compare different aerial operations representations formats. The section shows how data obtained from the simulator can be used to guide regulator's decisions to improve airspace usage efficiency.

Within each section, Figure 1.4 will be particularized to present the corresponding contribution in the context of smart territories. Also, co-authored publications supporting the proposals will be cited together with relevant literature that describes the knowledge gap covered in each section. Finally, Chapter 5 summarizes the work and draws the main conclusions.

Chapter 2

Monitoring architectures for smart territories

As discussed in Section 1.1 , IoT is one of the technological pillars enabling real-time data acquisition and analysis to monitor smart territories. From the information processing point of view, sensors' data is usually analyzed jointly, integrating information from different sensors so that a holistic view of the monitored entity can be obtained. In this sense, the monitoring process can be considered as a **data-fusion chain**.

Conceptually, a well-known framework to articulate data fusion problems is the Joint Directors or Laboratories (JDL) data fusion model. Although it was originally envisioned in 1985 in the military domain for entity tracking purposes, its application can also be extended to other domains as it will be explored in this chapter. The model articulates information fusion through a set of hierarchical levels that gradually manage to extract more complex and abstracted information, allowing to finally understand the monitored situation. In this regard, and as depicted in the greenish boxes of Figure 2.1 , the JDL model levels are [SB09]:

- Level 0, or *signal-feature assessment*, carries out preprocessing of source information (i.e.: sensor measurements) to align data, standardize inputs and generate valuable features for higher level data-fusion.
- Level 1, or *entity assessment*, which consists in associating the previous features to estimate the attributes and state of an individual monitored entity.
- Level 2, or *situation assessment*, aggregates information from several entities to estimate the relationships between them and to recognize global situations and structures.
- Level 3, or *impact assessment*, which projects into the future the situation information to provide consequence analysis by assessing the impact of different courses of action.

For instance, let us imagine a video surveillance system consisting of a network of cameras to detect and track groups of persons of interest during a demonstration or any safety-sentive event. In that case, level 0 data fusion would correspond to image segmentation algorithms applied to each video feed to detect persons and estimate their position within each frame. Then, level 1 data fusion would associate and integrate detections from each video feed to

create a single estimate of each person position over time. At a higher level (i.e.: situation assessment), groups of related persons would be detected and tracked based on an analysis of each individual motion pattern. Finally, at level 3, previous information would be used to predict future situations and compare scenarios such as anticipating riots. Thereby, the scope and actionability of the produced information at each level is increasingly broaden.

So far, the data fusion process has been considered as a static process. However, in some cases, sensors can be configured to improve the results of the fusion process. In the previous example, cameras could be orientable, and their field of view adjusted iteratively to improve tracking quality. Thus, the concept of **resource management** appears, which consists of dynamically controlling and planning the sensor network operation in a feedback loop with the data fusion chain. Based on some literature, this can be modeled as an additional level of data-fusion: level 4 or *process refinement*. This level can be understood as a metaprocess that continually evaluates the fusion chain and manages it (i.e.: adaptive data gathering) to improve systems performance. Alternatively, and as visualized in the brownish boxes of Figure 2.1 , it can also be understood as a set of dual levels to those of fusion, that gradually divide the high-level monitoring objectives into tasks for each sensor, and finally into sensor-specific commands or configurations [BS09].

Finally, human intervention can also be modeled within a Data Fusion and Resource Management (DF-RM) process to consume the data fusion outputs and define the monitoring mission objectives. In this sense, Level 5 data fusion, or *user refinement*, connects the user with the fusion pipeline enabling fused information visualization and feedback and control of the process. Overall, the JDL data fusion model will be used through this chapter as a conceptual framework from which monitoring systems are build.

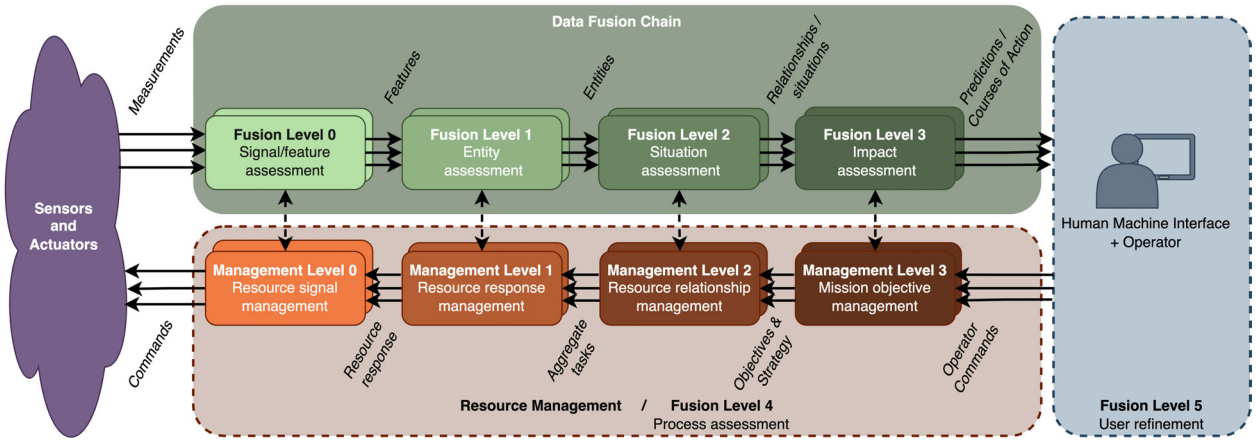


Figure 2.1: Data fusion and Resource Management processes as viewed in the JDL model
Graph generated by this thesis author inspired in similar graphs in [SB09]

From a technological point of view, a DF-RM process requires sensing and data gathering platforms that feed measurements to the chain. Particularly, in smart territories, cost, coverage and connectivity limitations must be overcome to capture information in an almost ubiquitous way, covering large geographical extents. To this end, mobile sensing technologies such as drones may be used for different remote sensing applications, as the ones discussed

in Section 1.2 . Also, WSNs can be used, with citizens' smartphones being an example of a massive network of connected sensing platforms. In this context, this chapter presents two novel end-to-end monitoring architectures that contemplate the full DF-RM process with different technological solutions:

- Section 2.1 explores how drones may be used to autonomously explore and surveil wide operation areas without requiring the deployment of a network of ground sensors and the corresponding communications nodes. In this sense, it presents a data fusion and control architecture for a swarm of camera-equipped drones. It does so taking advantage of AI technologies to both autonomously control the drones and to process gathered information. A key element of the proposal is that AI integration within the architecture allows to retain human control and explainable supervision.
- Section 2.2 analyzes how smartphones can be used in mHealth systems as a cost-effective platform to gather information from the population of a smart territory. In this sense, mHealth applications are discussed as a data fusion system to detect communalities between existing examples. Then, a health data fusion architecture is proposed as a reference to design mHealth applications. The key contribution of the section is that the proposed architecture is comprehensive, modular and considers specific health-related services. This way, it is expected to allow to implement assorted mHealth systems with software components reusing.

2.1 An architecture for multi-drone wide area surveillance operations

Due to their cost-effectiveness, ease of use, reduced operational risks, stealth and agility, the usage of UAS for Intelligence, Surveillance and Reconnaissance (ISR) missions has become widespread both in military and law-enforcement domains [Abd+21]. In addition, the operation of multiple drones may be coordinated (i.e.: a swarm of drones) to extend geographical coverage, temporal persistence and to provide an adaptation capability to operational changes. Therefore, the use of a **swarm of drones is suitable for surveillance / monitoring missions in smart territories**. Particularly, as drones usually rely on ad-hoc communication C2 channels, these systems are of interest in remote areas with limited mobile communications.

In this context, a usual ground surveillance mission for law enforcement purposes may consist in a set of drones that automatically fly to periodically explore and surveil a wide operations area. Drones are expected to incorporate Electro Optical/Infra-Red (EO/IR) sensors so that video streams from each drone are captured and can be analyzed in real time. The use of AI algorithms on the products of on-board sensors allow to monitor each aircraft surroundings and to identify and classify targets of interest in the ground (e.g.: suspicious vehicles, crowds...). Due to the sensors mobility, **a swarm of drones can be understood as a distributed autonomous sensor** that enables building an integrated operational picture through data fusion algorithms. In this sense, it is possible to build a Common Operational Picture (COP) which incorporates fused information from all relevant entities on the ground. The COP constitutes the main outcome of the swarm operation and is available for the decision-making, coordination and execution of higher-level law enforcement tasks.

Simultaneously, the swarm operation can be managed dynamically by adapting it to an evolving operational picture. Through intelligent algorithms, it is possible to ensure that the whole operational area is automatically monitored and that certain targets of interest are tracked over time. Therefore, the **swarm surveillance system can be considered as a DF-RM process that enables smart safety services**. In this sense, Figure 2.2 articulates the proposal of this section in the context of smart territories.

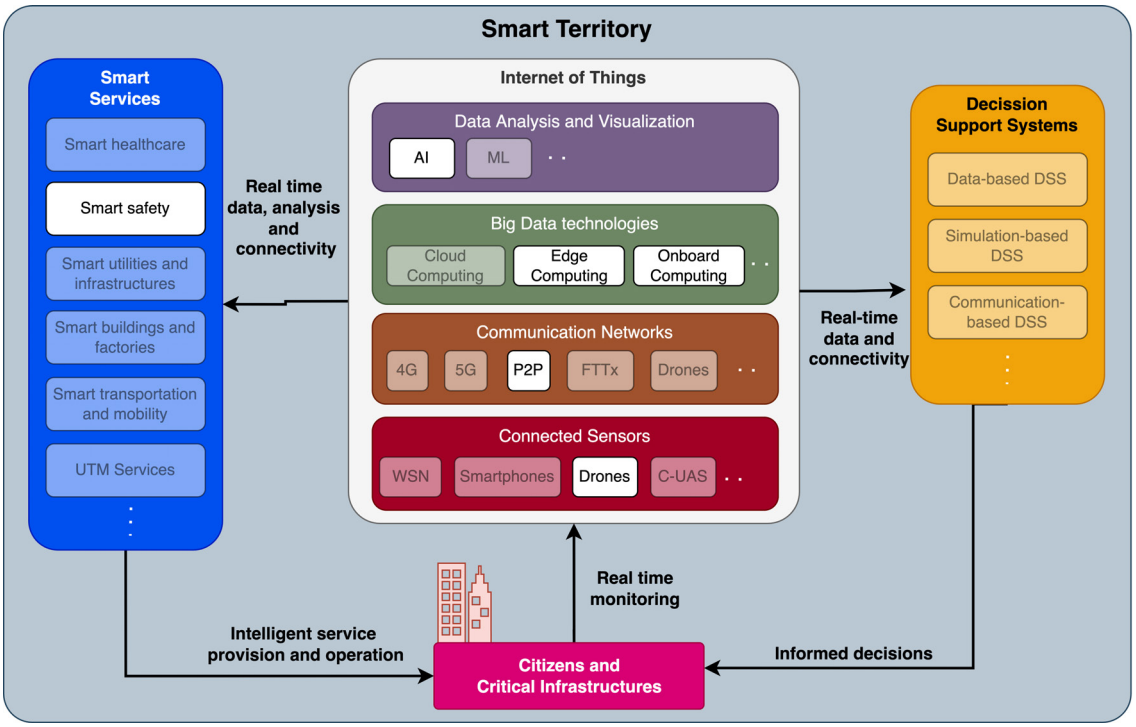


Figure 2.2: Technological elements in the proposed drone-enabled smart-safety monitoring use-case
Relevant technological elements are highlighted over the full architecture

Due to safety and accountability reasons, this autonomous operation shall be supervised by a human swarm operator. To this end, the operator requires having a detailed knowledge of the objective or task assigned to each aircraft, and the rationale/explanation behind each provided output within the COP. However, literature reviewed in Section 2.1.1 will show that the inclusion of AI techniques to automate drone operations usually hinders human intervention and supervision. Ideally, UASs operations should be automated with AI while maintaining behavior explainability and allowing the operator's intervention in the decision-making process. To overcome this gap, the **main contribution of this section is a DF-RM architecture that takes advantage of AI capabilities while supporting human supervision**. This architecture is described in Section 2.1.2 .

When testing and validating drone-based systems, it is common to encounter another problem: the complexity of performing real tests. Indeed, safety and regulatory requirements make it difficult to conduct real research flights. To solve this issue, Section 2.1.3 presents the second contribution of this section: **an indoor testbed with digital twin capabilities for developing and validating aerial algorithms**. Finally, Section 2.1.4 describes a proof-of-concept implementation of the architecture that has been used to prove its validity and usefulness. Also, qualitative and quantitative results exploring the performance of the system are provided thanks to the usage of the testbed environment.

The discussion and results of this section are supported by this thesis author's contributions to the following co-authored publications. Specific contributions in each paper are also identified below.

Supporting publications

- **[Car+25b]** - David Carramiñana, Lucas Rey, Raúl Arranz, Juan Besada, and Ana Bernardos. "An AI-assisted Multi-Drone Architecture for Surveillance Operations". In: *International Conference on on Drones and Unmanned Systems*. Granada, Spain, 2025, where this thesis's author has conceptualized, and designed the proposed multi-drone control architecture and collaborated with other coauthors in its implementation.
- **[Car+24c]** - David Carramiñana, Lucas Rey, Ana M. Bernardos, Andrzej D. Dobrzycki, Luca Bergesio, Juan Besada, and José R. Casar. "A Digital Twin Concept to Prototype Multi-Drone Based Applications". In: *14th International Conference on the Internet of Things*. Oulu, Finland, 2024, where this thesis's author has conceptualized and designed the proposed digital twin testbed, assisting in its implementation and validation.
- **[Arr+23]** - Raúl Arranz, David Carramiñana, Gonzalo de Miguel, Juan A. Besada, and Ana M. Bernardos. "Application of Deep Reinforcement Learning to UAV Swarming for Ground Surveillance". In: *Sensors* 23.21 (Jan. 2023), p. 8766. DOI: 10.3390/s23218766, where this author has collaborated in the architectural decomposition of the multi-uav control problem, the specification of the swarm task generation process, and in the definition of validation metrics.
- **[Rey+25]** - Lucas Rey, Ana M. Bernardos, Andrzej D. Dobrzycki, David Carramiñana, Luca Bergesio, Juan A. Besada, and José Ramón Casar. "A Performance Analysis of You Only Look Once Models for Deployment on Constrained Computational Edge Devices in Drone Applications". In: *Electronics* 14.3 (Jan. 2025), p. 638. DOI: 10.3390/electronics14030638, where the contributions of this author are related to the validation methodology, and assisting in its implementation and analysis.
- **[San+24]** - Rubén San-Segundo, Lucía Angulo, Manuel Gil-Martín, David Carramiñana, and Ana M. Bernardos. "Hybrid Artificial Intelligence Strategies for Drone Navigation". In: *AI* 5.4 (Dec. 2024), pp. 2104–2126. DOI: 10.3390/ai5040103, where the author has provided insights on the methodology and validation of the solution.

2.1.1 Review on UAS systems for multi-UAV surveillance operations

In line with the DF-RM process discussed previously, two problems must be addressed to implement multi-drone operations: control and coordination of the aircrafts, and the analysis of the captured information (the results of which are fed back into the control model). To this end, two alternative approaches can be found in literature with different levels of automation and human supervision:

- Implementations that require **almost continuous human control and supervision**. An example of this approach is presented in [Per+13] where the operator can manually assign surveillance tasks to each drone through a multi-UAV GCS. Also, video feeds are

provided so that the operator can analyze the situation on the ground.

- **Autonomous AI-based methods**. For the first problem, it is possible to use swarm intelligence (i.e.: biology-inspired, self-organized) algorithms such as genetic algorithms or particle swarm optimization, as discussed in [TDL23]. More broadly, [Cab+24] reviews other AI-based algorithms (not necessarily inspired in nature) that are able to autonomously coordinate the individual behavior of a large number of aircraft so that high level missions can be achieved. For instance, authors in [Liu+20] propose a RL based control framework to automatically manage a swarm of drones performing surveillance of an urban area. Regarding the second problem (i.e.: autonomous aerial image analysis), there is also a large set of examples that make use of AI techniques [CC19], such as in [Do+21] where a Convolutional Neural Network is used to detect moving ground targets.

From a comparative point of view, human intervention hinders scaling drone operations to a large number of vehicles, as it is discussed in [Dor+24] and [HS18]. Alternatively, the performance of autonomous AI-based systems is promising. Nevertheless, authors in [Ken+19] argue that the lack of supervision capabilities and decisions explainability discourages a direct adoption of AI techniques in systems where safety is critical such as aerial platforms.

To bridge this gap, the work coauthored by the author of this thesis in [San+24] explores the use of a hybrid and explainable AI for drone navigation in a simplified scenario. The approach is twofold. First, expert-rules are incorporated into a RL-based navigation algorithm to improve its robustness. Then, explainability algorithms such as LIME [RSG16] and SHAP [LL17] can be implemented to provide insights on the drone path. However, in that case, explainability insights lack operational meaning and is difficult to interpret by an operator. Instead, in [Arr+23], this author and others have proposed to achieve explainability as a result of dividing the navigation problem into two stages: task planning (swarm-level planning) and path planning (drone-level planning).

The contribution described in the next section explores the second approach from an architectural point of view. That is, the focus is not in the development of explainable control or data analysis algorithms based on AI, but in proposing a distributed computing system that allows the integration of such models with real drones, for validation, demonstration and operational purposes.

2.1.2 Description of the proposed multi-UAV control and sensing system

In order to allow drones' control automation while retaining drones' behavior explainability and facilitating swarm operator's supervision and control, the logical architecture presented in Figure 2.3 is proposed. It consists of a set of microservices implementing **an autonomous DF-RM process** that considers a drone control chain, and an information sensing and fusion chain. Services within the architecture may be distributed and asynchronously exchange information with each other to orchestrate the behavior of the drones.

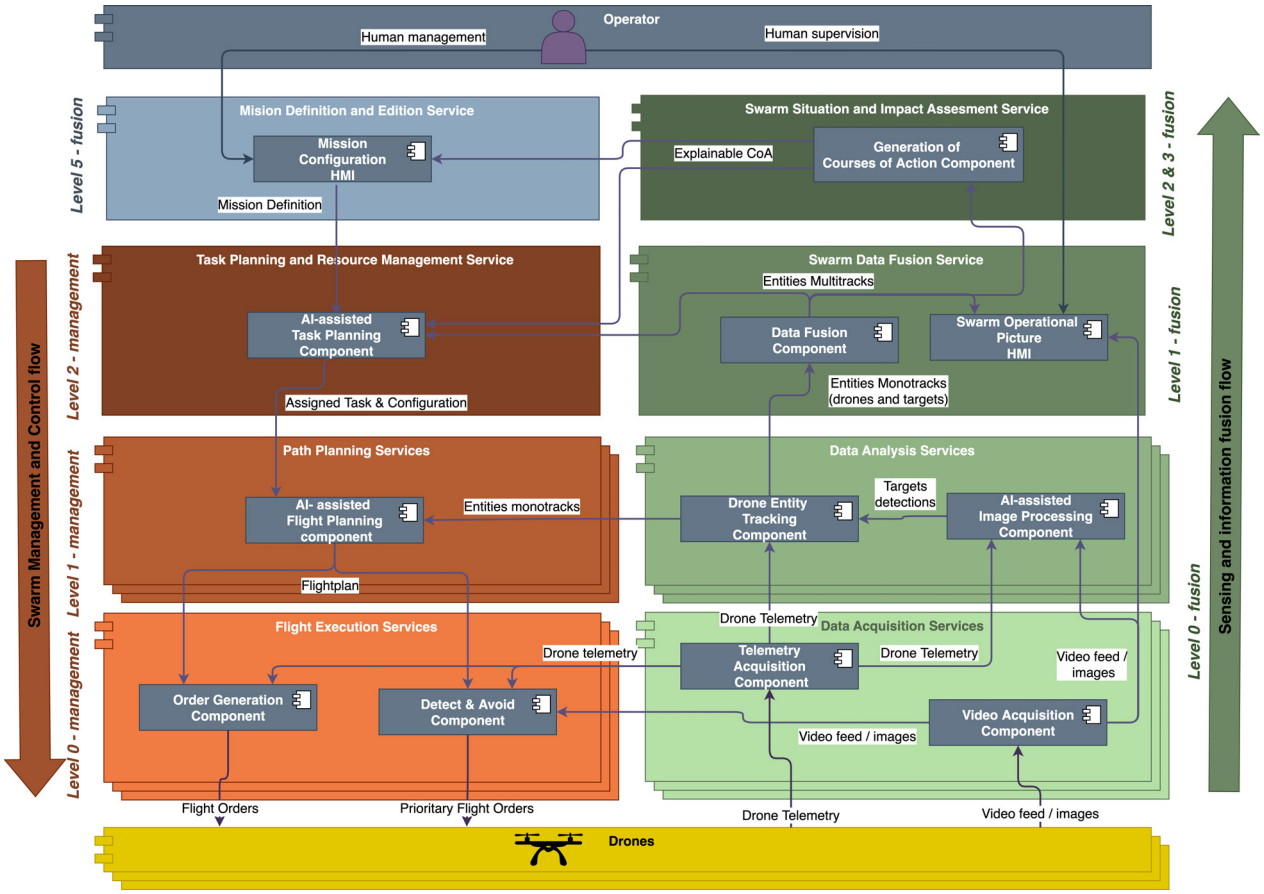


Figure 2.3: Logical architecture of the proposed multi-UAV control and sensing system.

2.1.2.1 Resource management: swarm management and control flow

Starting with the swarm management and control flow (on the left side of Figure 2.3 , from top to bottom), the swarm operator interacts with the **Mission Definition and Edition Service**. This service consists of a graphical interface that allows the operator to configure the surveillance mission to be executed. Configuration parameters include high-level information such as the operational area (i.e.: area to be surveilled), number of drones to deploy, target prioritization (e.g.: among different types of targets of interest), flight constraints (e.g.: no fly areas, battery requirements...). This service corresponds to the user refinement data fusion level.

With this configuration, the aforementioned two-stage swarm control scheme is adopted in the following services. First, within the **Task Planning and Resource Management Service**, the whole operational situation is analyzed jointly with the mission requirements. As a result, high-level operational tasks are assigned in real time to each drone within the swarm. Tasks are abstract objectives to be achieved autonomously by each drone such as:

- surveilling a given 2D area (a subset of the operational area) searching for new targets of interest,
- searching and tracking a given target of interest for which the last available position is

indicated,

- other meta-tasks (internal to the swarm health and operation) may be considered. For instance:
  - deployment/rollback from/to a safe "home" position.
  - ensuring swarm communications.

The set of tasks distributed among the drones must make it possible to accomplish the mission set by the operator, subject to changing constraints (e.g.: detected targets, drone positions, etc.). For this purpose, the allocation of tasks must be continuously re-evaluated. This process, which implements level 2 resource management (as it generates tasks considering the whole swarm), may be fully implemented using AI algorithms or using a hybrid approach to enhance explainability.

Afterwards, and in a decentralized way, each drone autonomously executes its assigned task, as managed by its associated **Path Planning Service**. This service implements level 1 resource management by considering each drone local context (e.g.: position history, detected targets and obstacles) and its assigned task to dynamically plan the most suitable flight trajectory (i.e.: the individual drone response). As a result, a trajectory is defined by a flight-plan (i.e.: a list of waypoints) for the near-future.

Therefore, a semi-distributed control flow is proposed: tasks are assigned centrally while decisions on their execution is decentralized to each drone. This way, it is possible to automate each drone operation (as the flight plan of each drone is automatically generated) while retaining insight into which task each drone is performing (thus providing explainability of their behavior). Moreover, by adopting flight plan-based control of the drones, greater stability and predictability of each drone's behavior is expected.

Finally, the **Flight Execution Service** serves as a gateway to the drone internal commands in order to guide and monitor the execution of the flight plan based on the drone's telemetry information. This way, it implements level 0 resource management. In addition, this layer can also provide obstacle avoidance capabilities (both with ground obstacles and with other swarm's drones).

2.1.2.2 Data fusion: sensing and information fusion flow

In parallel to the control flow, a sensing and information fusion flow is executed in order to provide operational ISR capabilities while also feeding back into the control flow (as shown on the right side of Figure 2.3 from the bottom to the top). In this sense, **Data Acquisition Services** capture video and telemetry information from drones and broadcast it to relevant services. One of these services is the **Data Analysis Service** in which images from each drone are analyzed to detect and classify relevant targets of interest, thus implementing level 0 data fusion. Additionally, depth inference models can be used together with knowledge of the drone position and attitude to infer target's position. In order to provide stability and continuity to the detections, a tracker is considered to provide monosensor (i.e.: only with information of one drone) tracks (i.e.: position, speed and optionally attitude and state information) of detected entities and of the drone itself.

At a superior stage of the fusion flow, the **Swarm Data Fusion Service** centralizes information from the whole swarm. Here, association and data fusion algorithms are used to fuse monotrack information from each drone into a single track per entity (i.e.: multitrack). Therefore, it implements an entity assessment process (level 1 data fusion). As a result, it is possible to build a merged Swarm Operational Picture (SOP) which constitutes the main operational outcome of the system. This view consists of the status information of all drones, positioning information of detected targets together with real-time video feeds and swarm statistics. This information is provided to the operator through an HMI so to inform its operational decision-making and allow swarm monitoring and reconfiguration. In addition, by providing the original video feeds, the operator can verify and understand the automatic detections provided by AI algorithms.

Finally, the swarm behavior may be assessed within the **Swarm Situation and Impact Assessment Service** to refine its operation. To do so, the system may provide smart suggestions on how to reconfigure the mission (e.g.: prioritization of targets, changes in the operation area) to improve the operational metrics. Also, situation and impact assessment modules (level 2 and 3 data fusion) could be considered to improve the provided operational ISR products.

2.1.2.3 Summary and mapping with JDL DF-RM model

In general, the proposed architecture follows the concept of a DF-RM process as described in the introduction of the chapter. Indeed, in Figure 2.3 , the fusion and management level or each service has been depicted. Also, the architecture allows the introduction of AI elements in several of its modules to improve system performance (e.g.: in the image processing and flight planning components). Likewise, as it is proposed as a set of interconnected services, its deployment and implementation are flexible and scalable to many drones. As a result, it possible to distribute each component in the cloud, at the edge or even on the drones themselves if on-board computing resources are available.

2.1.3 A testbed for development and validation of aerial algorithms

Algorithms and systems that control drones (such as the one proposed in the previous section) must be rigorously tested to ensure they can handle operational scenarios reliably. However, testing drone-based applications before deploying them in real-world environments presents a significant challenge. First, testing in real environments is often costly and logistically challenging, as it requires extensive resources and can be constrained by safety regulations and operational limitations. Second, due to the inherent complexity and variability of the operating conditions, tests should cover a wide range of environmental factors, including unpredictable weather, varying terrain, and dynamic obstacles; which may not be easily replicable. One can resort to test the algorithms in simulated environments. Nevertheless, relying only on simulations can oversimplify complex situations, ignore computational constraints imposed by the aerial platform and lead to algorithms that do not perform well in real environments.

To address these issues, literature (e.g.: [Lei+21], [LS22]) shows an increasing usage of advanced simulation tools and digital twins to create realistic testing environments, allowing

for comprehensive evaluation of drone systems under controlled yet representative conditions before they are deployed in the field. Particularly, digital twins extend traditional simulation techniques, providing a seamless interconnection between the physical and simulated elements so that both realities are coordinated in real time [Sin+21].

In this sense, **a digital twin system is proposed to validate the multi-UAV control and sensing system**. The indoor testbed system (conceptually described in Figure 2.4) consists of two types of controlable entities: small drones (DJI Tello, a small camera-equipped multirrotor [Ryz18]) and ground targets (Turtlebot robots); that are coordinated and synced in real time between a real environment (a lab room) and a virtual environment.

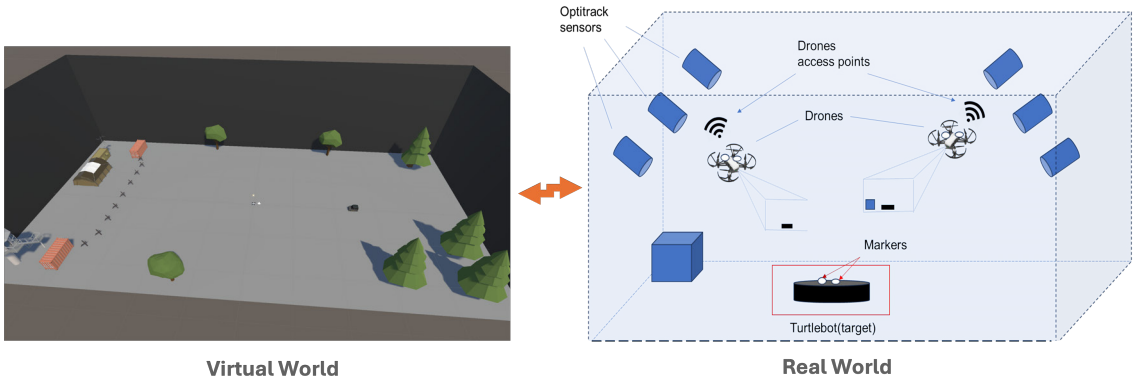


Figure 2.4: Correspondence between virtual and real worlds entities in the proposed digital twin.

The digital twin system serves as a tool for testing, enabling the repetition of scenarios, identification of bottlenecks and potential improvements in a controlled and risk-free environment. To this end, four types of services (depicted in Figure 2.5) are orchestrated:

- **Application level services**. This is the system or collection of services that are going to be validated through the digital twin architecture. These services are the ones governing the final drones' operation based on information gathered from the environment. To that end, these services easily interact in real time with both virtual and real worlds to retrieve information from relevant entities (i.e.: drones, targets...) or to modify their state (e.g.: move an entity). To enable experimentation with different deployments and communications systems, application-level services may be connected to the digital twin either directly, via low-latency networks (emulating edge computing) or via the Internet (emulating cloud computing).
- **Virtual world services**. The Simulation service, built using the Unity engine, generates a virtual representation of the entities found in the real world. It allows to replicate the movement of real drones and targets in an emulated environment that can be customized to replicate realistic operational scenarios. Moreover, it allows to simulate the field of view of each drone, so that a stream of simulated images can be extracted.
- **Real world services**. Regarding the services interfacing with the real world, the drone interface service acts as a relay to control and monitor real DJI Tello drones. To enable a precise correspondence between the location of real and virtual entities, the Positioning service provides accurate, high-frequency, low-latency location information simulating

a GPS system. This service is based on a comercial Optitrack system [Opt25], which provides accurate positioning through the detection of passive reflective markers using high speed infrared cameras. Finally, the Target control service interfaces with the movable ground targets which may be commanded to follow waypoint-based trajectories.

- **Digital twin enabling services**. The communications service serves as the central hub enabling data and video exchange between the previous services. Within it, a MQTT broker is used to exchange information messages in real time (e.g.: drone control orders, drone position updates...) through a series of standarized channels (named, topics). Also, to facilitate access to real-time video and image streams, a WebSocket relay server has been implemented. This server acts as middle point enabling multi-consumer reception of video streams. A proper configuration of this service allows interconnecting the services in a way that enables different uses of the digital twin. For example, it allows the application layer to be directly connected to the virtual world so that algorithms can be trained in a simulated environment. Alternatively, it allows connecting them to the real world for testing and validation, in a way that the virtual world becomes a reflection of the real one. Other hybrid approaches are also possible.

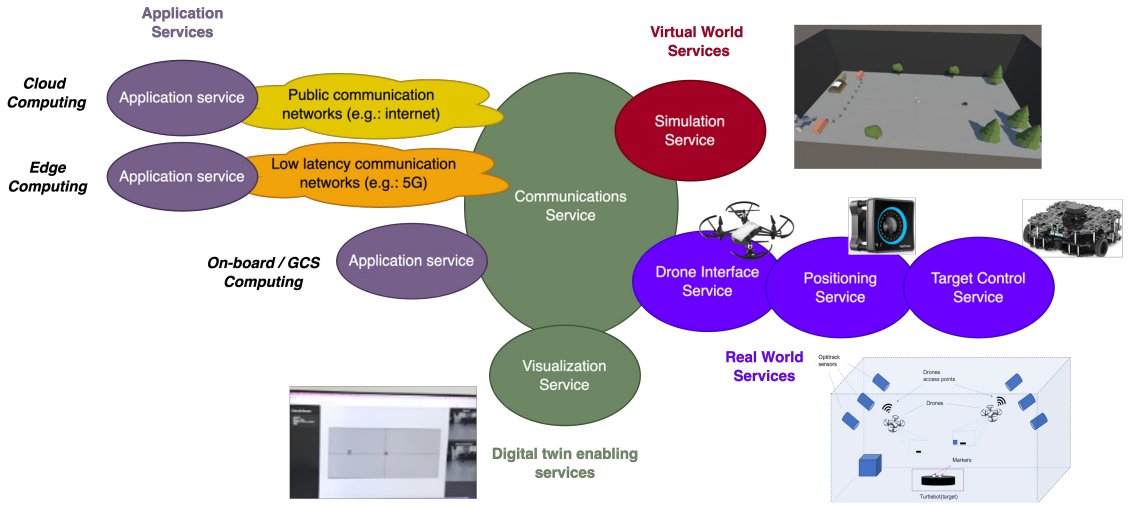


Figure 2.5: Digital twin system enabling indoor testing and validation of UAS applications.

The implementation of the hybrid virtual-real testbed environment is depicted in Figure 2.6 . It enables the following capabilities with regard to the validation and testing of the proposed multi-UAV surveillance architecture:



Figure 2.6: Digital twin deployment in a physical lab.
Note that both real (i.e.: DJI Tello, Turtlebot) elements and virtual elements (i.e.: synced virtual wolrld shown in the screen on the right side) are deployed. Also, the Optitrack positioning system enables a seamless integration of all elements.

- **Simulation of complex operational scenarios.** In security and defense contexts, the ability to simulate real-world threats and responses (e.g.: enemy attacks) in a controlled environment ensures that drone control algorithms and their operators are prepared for high-stakes missions, where precision and reliability are paramount. Similarly, for search-and-rescue missions, the digital twin allows for the testing of drones' performance in scenarios involving challenging terrains and dynamic obstacles. This ensures that, when deployed in the field, drones are equipped to handle the complexities of these critical operations. For instance, in Figure 2.6 , drones are flying in an indoor obstacle-free environment while simulated obstacle information is obtained (and considered for path planning) from the virtual representation shown in the visible monitor. Also, a video feed of the drone's field of view in the virtual world can be obtained for analysis.
- **Development, training and evaluation of artificial intelligence algorithms**. In this sense, its use is fundamental in training techniques such as RL, that require a test arena in which to iteratively evaluate different versions of the model as part of the optimization process. By leveraging the digital twin, it is possible to test and refine drone swarm control algorithms without compromising safety or other critical aspects of drone flight operations (i.e.: lack of replicability, regulatory restrictions...). In addition, AI-assisted vision algorithms can also benefit from the digital twin as it allows generating image datasets for training. This can be done both with fully synthetic datasets generated from a virtual world view, as well as with images taken with the drones where the ground truth is known thanks to precise positioning services.
- **Evaluation of communication technologies and deployment architectures**. It allows for the simulation of scenarios where the impact of advanced communication technologies, such as 5G, can be assessed, particularly in terms of latency. In addition to these communication technologies, a digital twin can also facilitate the evaluation of

real-time communication protocols, such as WebSockets, WebRTC or MQTT, which may be used for real-time video and data stream transmission respectively. These protocols play a crucial role in synchronizing drone behaviors. For instance, in [Rey+25], the proposed digital-twin platform has been leveraged to evaluate the performance of vision AI-based models under different connectivity situations.

2.1.4 Proof-of-concept implementation and execution results

In order to evaluate the multi-drone DF-RM architecture proposed in Section 2.1.2 , a proof-of-concept implementation has been carried out and integrated into the digital twin presented in Section 2.1.3 . At this point, it is important to remember that the objective of the section is not the development of AI algorithms but rather the design of an architecture that enables their integration, allowing for human supervision and the explainability of the swarm's behavior.

Therefore, regarding the **software implementation**, algorithms developed in the coauthored articles by other colleagues have been integrated. Therefore, the AI algorithms proposed by the authors of [Arr+23] have been used for the task planning and path planning services. Thus, the task assignment is solved by means of a set of predefined rules, and depending on the number of available drones. This task generation processes prioritizes detected targets monitoring (tracking task) over surveilling the rest of the operation area (search task, which are distributed among the remaining drones). Then, for task assignment, the Munkres algorithm [Mun57] is used to perform a globally-optimal task allocation in terms of drone-task distance. As for flight plan generation at drone level, it is carried out by a set of RL models trained with Unity's ML-Agents platform [Jul+20]. Models have been exported using the ONNX standard and loaded in runtime with the onnxruntime Python's library [dev21].

The architecture has been implemented and adapted to control a swarm of DJI Tello drones. Thus, the flight execution service is built on top of the DJITelloPy library [Esc24]. Drone guidance is implemented with a *carrot-chasing* path-follower assisted by a Proportional Integral Derivative controller (PID). The collision avoidance component relies on the modified voltage potential conflict resolution method proposed in [Eby95], which requires sharing each drone flight intent to predict conflicts. When a conflict is predicted, the model provides an alternative waypoint that tries to continue the current trajectory while avoiding the conflict.

Finally, regarding the information processing chain, the fine-tuned YOLO-based AI model implemented in [Rey+25] has been used to detect targets of interest and infer their position. Moreover, a tracker based on a Kalman filter has been used. As for the Swarm Operational Picture HMI, the web interface depicted in Figure 2.7 has been implemented to display the drone task assignment, entities location in a 2D map and real time annotated video feed.

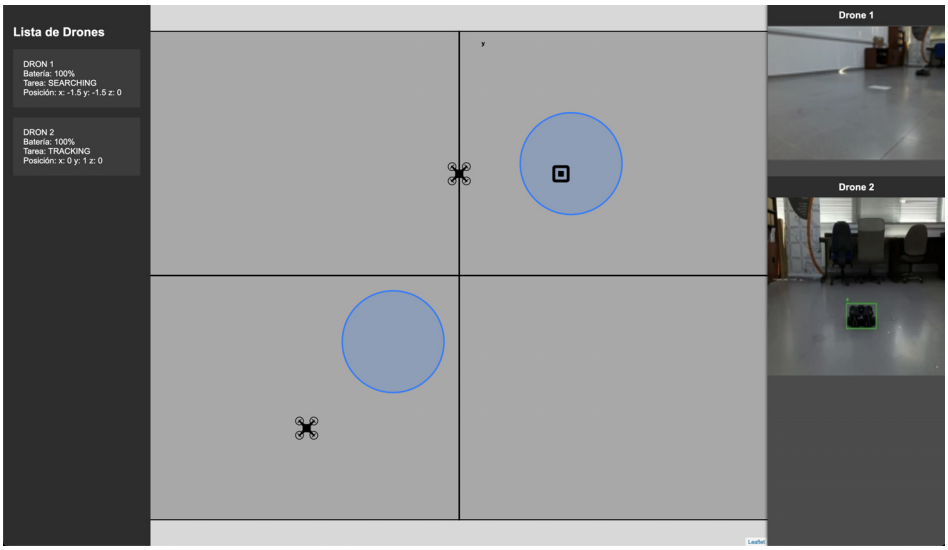


Figure 2.7: Web interface that displays the Swarm Operational Picture.

As for the **deployment and hardware**, Figure 2.8 depicts the deployment diagram of the proof-of-concept implementation. It shows how services have been distributed among different nodes. First, those services that centralize information and take decisions at swarm level have been centralized in a NUC mini-PC. On the other hand, each drone is directly controlled from a Jetson Orin computing module with an integrated GPU to be able to deploy AI models (i.e.: the aforementioned RL path planning module and YOLO-based object detector). Finally, communication between services is enabled thanks to a MQTT broker. Also, WebSocket protocol is used to transmit video frame by frame (i.e.: MJPEG format).

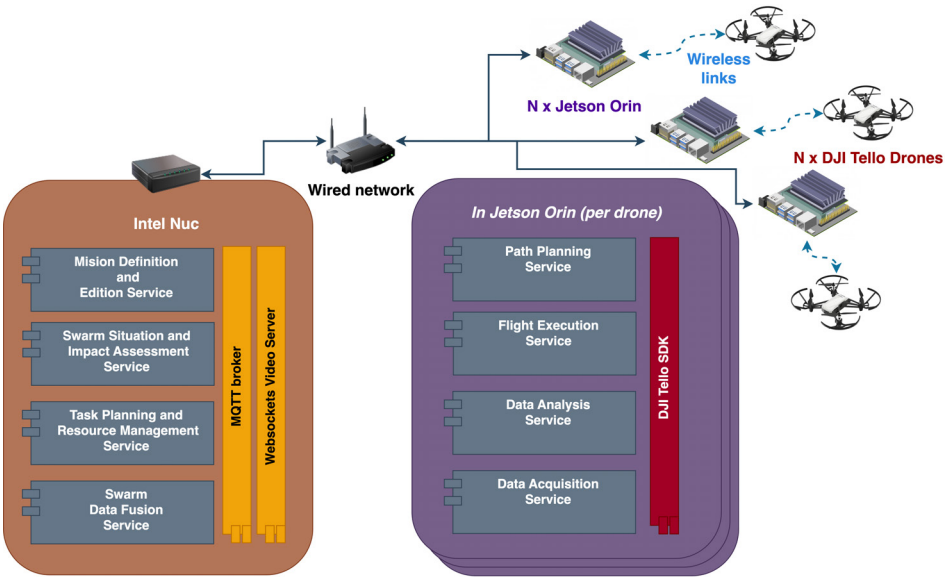


Figure 2.8: Hardware and deployment architecture of the proof-of-concept multi-UAV control system.

In that environment, **validation flights** have been carried out with a swarm of two drones

to assess the validity of the proposed architecture. In line with the contribution of the section (which is at the architectural or system level), the objective is not to validate the underlying algorithms but rather to ensure that the architecture allows to automate the control of the drones in a semi-distributed way (i.e.: tasks are assigned centrally with all available information whereas path planning is performed individually by each drone) and to analyze the collected information. To this end, Figure 2.9 depicts the trajectories followed by each drone and the trajectory described by the ground target (which is to be identified and tracked by the proposed platform). This figure demonstrates how the architecture allows the drones to be controlled to fulfill the surveillance mission. This can be derived from the fact that each drone follows different trajectories depending on the assigned tasks. Drone 3 has been tasked of tracking the target and thus its trajectory is concentrated around the moving target. On the contrary, drone 7 has been tasked of surveilling the whole operation area as demonstrated by its trajectory (and the field of view of its sensor over time) being spread around it. Moreover, both drones operate simultaneously over time in the same operational area without colliding.

As for human supervision, while the swarm automatically follows these trajectories in real time, the swarm operator can visualize the HMI previously depicted in Figure 2.7 . In that interface, it can find out which task has been assigned to each of the drones in order to understand their trajectory.

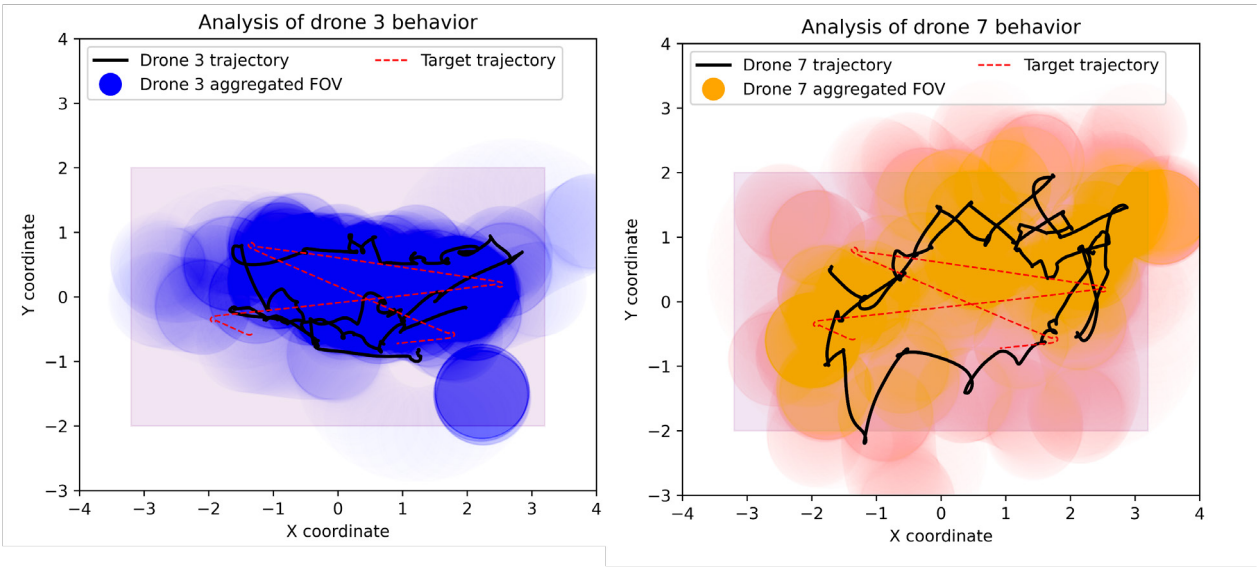


Figure 2.9: Multi-drone control architecture experiment results.
Superimposed blue and orange circles represent the field of view of each drone over time with darker circles meaning that those areas have been surveilled more. Purple rectangle depicts the operation area.

2.1.5 Conclusion and limitations

Remote areas in smart territories require purpose-built monitoring systems to overcome range, cost and connectivity limitations. In this sense, drones can be used as a low-cost, nobile sensing platform that uses ad-hoc communication networks. For example, the operation of a fleet of drones may be coordinated for ground surveillance missions for law enforcement

purposes. A swarm operation requires a DF-RM chain that fuses information from all drones into a single swarm operational picture for law-enforcement purposes, and dynamically adapts drones' flight path to fulfill mission objectives. In this regard, the potential of drones as a distributed and autonomous sensor can be leveraged through the inclusion of AI techniques to process captured information and automatically control drones. However, as discussed in Section 2.1.1 , this comes at the expense of a loss of drone behavior explainability (which is critical for safety) and reduced human supervision and control.

In this context, Section 2.1.2 has proposed a DF-RM architecture that integrate AI algorithms in the operation of a fleet of drones for ground surveillance missions. By hierarchically framing the drone control chain, it is possible to maintain explainability of the high-level task performed by each drone while operating them autonomously through AI algorithms (i.e.: semi-distributed control). Likewise, the architecture uses AI algorithms to process drones' videos, but provides an HMI (i.e.: the swarm operational picture interface) that allows visualizing the video stream of each drone to trace back AI's detections.

To test the architecture with real flights, an indoor testbed has been built as a digital twin for drone operations (as discussed in Section 2.1.4). In that environment, a proof of concept implementation has been tested proving that the architecture effectively facilitates an automatic operation of the swarm (both in terms of control and data analysis) while retaining explainability.

On limitations and further work, flights have only been conducted in the testbed environment and with a limited number of drones. Real flights with a greater number of drones would be useful to provide further insights on the architecture scalability and robustness to variable conditions. Also, tests under different network conditions (e.g.: 4G, 5G...) would be helpful to assess end-to-end latency and possible synchronization issues between the fusion and resource management chains. Finally, how to integrate swarms of drones within unmanned traffic management systems (as the ones described in next chapter) is still an open point.

2.2 An architecture for ubiquitous health monitoring though mHealth applications

In Section 2.1 , the use of drones for ubiquitous monitoring in surveillance missions has been discussed through the proposed swarm management and data fusion architecture. In this section, a mobile health (mHealth) platform will be discussed as a smartphone-based monitoring technology to improve smart territories healthcare provision.

As defined in [Wor16], **mHealth** consists of medical and public health practice supported by a technological framework that leverages mobile and IoT devices to enhance communication, collaboration and data collection. Also, mHealth services leverage the possibilities of AI to perform data analysis and health-related activities. Therefore, the objective of mHealth applications is enabling **continuous monitoring** of patients' condition, extending the reach of medical care from hospitals into the daily lives of patients. As a result, it is generally assumed that mHealth may contribute to:

- the sustainability of both public and private healthcare systems in terms of costs [Iri+17],
- to widening access to healthcare services in terms of location and entry into the clinical pathway [Meh+15]), which is critical in rural territories;
- and to improve the overall patient, clinician, and caregiver satisfaction [CZL22].

Due to these advantages, **mHealth systems are slowly becoming more prevalent in everyday life** as demonstrated in [del+20] [Fol+23]. Indeed, the scientific literature covers the use of **mHealth systems for a wide range of conditions** including breast cancer management [Shi+24], diabetes management [AOJ22], weight control [Sak+23], heart diseases monitoring [Lob+23], pregnancy care [Bje+21], overall health status [Chi+21c] or mental disorders [Hsu+24]. Generally speaking, mHealth services target two primary user groups: **patients** and **clinicians** [AOJ22], providing them tailored smart healthcare solutions at different stages of health and disease, such as [Anw+23]:

- pre-diagnosis (e.g., prevention, health assessment, health education),
- diagnosis (in particular, contributing to detect early symptoms),
- treatment (with the ability to analyze adherence and outcomes),
- or follow-up (e.g. automated risk analysis and comorbidities management).

In this sense, Figure 2.10 articulates how **mHealth systems are a way of delivering smart healthcare services in smart territories** based on smartphones, cloud computing capabilities and AI-based analysis. Moreover, health insights generated with these systems can be used by practicioners a decision-support tool (i.e.: data-based DSS).

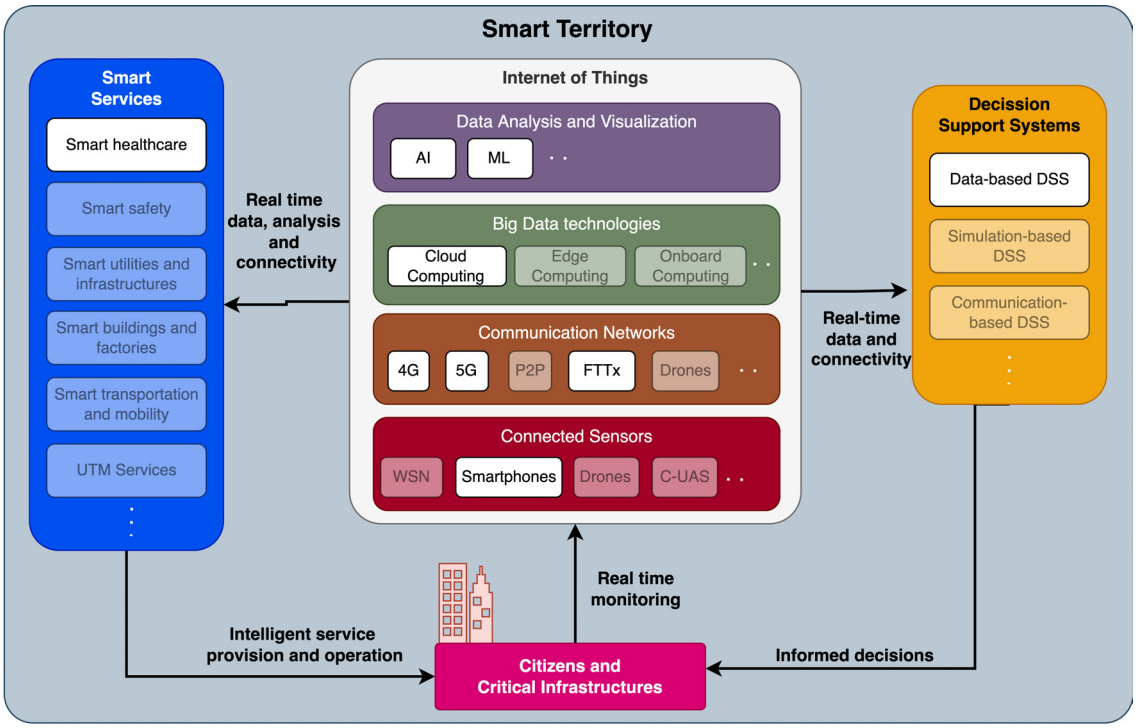


Figure 2.10: Technological elements in the proposed mHealth-based monitoring architecture
Relevant technological elements are highlighted over the full architecture

From a **technological perspective**, the literature review in Section 2.2.1 reveals that mHealth services are typically created through ad-hoc approaches, where researchers individually engineer and implement unique software architectures with distinct processing components. Thereby, the resulting designs are usually not interoperable, leading to fragmented solutions. This high customization for specific services poses challenges in reusing existing approaches to develop new services, and further complicates their extension to meet broader healthcare needs or address new diseases (e.g.: comorbidities) within the same mHealth framework.

However, there exists significant similarities in the data acquisition, processing, and user requirements across various mHealth systems. In this sense, Section 2.2.2 will argue that the **design of mHealth services can be approached as a data fusion problem**, at sensor, feature and decision-making levels. The fusion approach, which will be considered under the JDL data fusion model, facilitates leveraging the strengths of different data sources (smartphone and IoT sensors, patient reported data, clinician input, etc.) to build smart healthcare services.

Thus, the primary contribution of this section, as discussed in Section 2.2.3 , is to introduce a **mHealth architecture, grounded in the data fusion theory, that harnesses IoT and AI to steer and accelerate the delivery of mHealth services**. The proposal aims to streamline design and development efforts by minimizing the time spent on commonly reusable components, considering the information flows among stakeholders, while prioritizing the core functionality or "health logic," that offers significant value to the involved parties. This contribution can lead to reduced development costs, more extensible solutions (in terms

of functionality), easier to maintain systems and better knowledge management towards the delivery of personalized services.

Finally, in Section 2.2.4 , a case study is used to demonstrate the applicability of the data fusion architecture for remote monitoring of SARS-CoV-2 patients. The proposals and discussion on this section are based on the contributions of this author to the following publication:

Supporting publications

- **[Gon+24]** - Ana González Bermúdez, David Carramiñana, Ana M. Bernardos, Luca Bergesio, and Juan A. Besada. "A Fusion Architecture to Deliver Multi-purpose Mobile Health Services". In: *Computers in Biology and Medicine* 173 (May 2024), p. 108344. DOI: 10.1016/j.compbiomed.2024.108344, in which the contributions of this author have been focused in assisting in the literature review process, framing mHealth applications a data fusion problem, collaborating in the design of the reference architecture and participating in the development of one of the proposed validation use-cases (SARS-COV-2 monitoring).

2.2.1 Review on computing architectures for health monitoring

To facilitate the delivery of a mHealth service, a collection of computing and communication systems is essential. Typically, mHealth systems leverage users' smartphone applications and IoT sensors to gather the data needed for system operation. These systems are not usually monolithic, rather they often comprise various services constituting a **microservices architecture** [Kle+22]. Most mHealth services in the literature use a centralized or client-server architecture, i.e. all the collected data are gathered on a server with the logic to generate deeper insights [Ema+23], although strict regulatory and security requirements may discourage the use of centralized architectures. As an alternative, there are proposals using decentralized architectures (e.g., based on blockchain and IPFS, a distributed storage system [AOJ22]). However, such **distributed implementations** are not common. The additional complexity of such systems, together with other (not always technical) challenges, discourages developers from implementing distributed approaches.

Focusing on centralized architectures, the norm is to propose highly customized architectures for specific vertical use cases or for specific challenges (e.g., securitization of data management [Bar+22] or reduced computational resources in smartphones [EL17]). However, despite the differences in the architectures, a set of common functionalities can be identified (as detailed in Section 2.2.2). When designing a new mHealth service, it appears inefficient to overlook the previous knowledge regarding commonalities among existing services and opt for a design starting entirely from scratch.

In this direction, some authors before have already addressed the **development of generic mHealth architectures that can be adapted to different use cases**. For example, the client-server architecture in [Ber+12] consists of 4 layers: user layer (i.e., user interaction), communication layer, application layer (i.e., health functionality) and middle layer (i.e., data

storage, authentication). A layered approach is also proposed in [Bje+21], with a core layer managing data and relationships, an application layer managing APIs and user interaction, and an environment layer considering integration with external systems. The proposal in [BLG16] shows a flexible architecture, where part of the data pre-processing is delegated to mobile nodes. It also identifies generic nodes for communication, data aggregation, data processing and visualization. Authors in [Xu+17] recognize the value of cloud platforms for implementing mHealth systems. Their proposal includes a layer to manage multi-tenant cloud storage and access control, combined with data annotation and data analysis layers. In [Ban+15], a framework for developing a mHealth system is presented, based on the implementation of several functionalities: data acquisition, health knowledge extraction, persistent data storage, adaptive visualization, system management, and other optional services, such as intelligent alerts, recommendations and guidelines. This architecture focuses on data collection and monitoring functions. The most comprehensive flexible architecture found in the literature is the one presented in [Bo 22] and developed by IBM. It consists of an initial data ingestion layer, a storage layer, a data analysis layer, and a data access layer, all supported by common security and access control services. It also considers the existence of several user interfaces for data interaction and visualization.

Thus, most of the existing flexible proposals advocate structuring the architecture into a set of layers. Likewise, it also stands out that along with the need for a set of layers dedicated to data capture and analysis, proposals detect the need of layers dedicated to interaction, storage or user management. While these architectures may serve as initial guides for designers and developers, they remain being relatively **high-level ones, lacking detailed specifications required to implement the targeted mHealth system**. A complete identification of the blocks in each layer would help to develop common components that could be easily shared and reused.

Therefore, when compared with existing literature, the proposal on Section 2.2.3 also adopts the layered approach, yet it surpasses the current state-of-the-art by redefining it and elaborating modular services within each layer. These common services facilitate the design and development of heterogeneous services supporting various health logics. Also, a relevant novelty is that the proposed architecture is built on top of a conceptual data fusion chain as it will be discussed in next section.

2.2.2 Data fusion in mobile health applications

The first step towards defying a comprehensive architecture for mHealth systems is identifying features or requirements that are common across several existing systems or use-cases. To this end, a conceptual framework can help to guide this analysis task. In this sense, and as previously discussed the introductory paragraphs of Section 2.2 and Section 2.2.1 , mHealth applications are, in essence, systems that capture, processes and fuses data in order to provide useful outcomes/services for patients and practitioners. Therefore, **mHealth systems can be analyzed as a data fusion chain**.

At the beginning of Chapter 2 , the JDL model has been reviewed as a conceptual model to represent data fusion systems. Although the JDL model was originally envisioned for

decision-making in military environments, a domain transfer is feasible to apply it to the health domain. This is because the clinical method used by clinicians builds a holistic view (diagnosis) form individual attributes (symptoms):

1. The clinical process starts by gathering information from the patient about his condition (e.g.: symptoms, vital signs, life patterns). The clinician intrinsically translates patient input to standardized medical attributes and filters out non-relevant information. This corresponds to **Level 0 fusion** (signal/feature assessment).
2. In the next clinical process stage, health attributes are analyzed jointly to identify anomalous attributes and to provide a diagnosis on each of the patient individual conditions. This process correlates to **Level 1 fusion** (entity assessment).
3. As a next step, patient comorbidities are considered globally together with patient health records to obtain a holistic view of the patient health. In this case, the corresponding fusion level is **Level 2** (situation assessment).
4. Based on the comprehensive view, clinicians can provide a prognosis and to identify different treatments or courses of actions. This impact assessment stage is consistent with **level 3 fusion**.
5. Throughout the previous process, additional information might be required to refine the diagnosis in the form of complementary medical tests and to monitor patient evolution. Therefore, **level 4 fusion** or process refinement is performed here.
6. Finally, clinician-patient communication is also critical to feedback the process, which involves performing **level 5 fusion** (i.e.: user refinement).

From the above, it can be concluded that **as mHealth applications assist patients and clinicians through the clinical process, they implement a data fusion chain** covering the previously defined levels. In this sense, the JDL model can be adapted as depicted in Figure 2.11 to include the following health-related layers: health data acquisition , relevant health features extraction , intelligent data analysis and diagnosis, comprehensive patient health assessment , implementation of different health logic (ranging from prognosis estimation, recommendations generation, or notification of health events), health data visualization, and adherence support (both to treatment and to the mHealth system). Moreover, the model is extended to include additional levels for data acquisition and aggregation while also considering privacy and security issues in a separate layer.

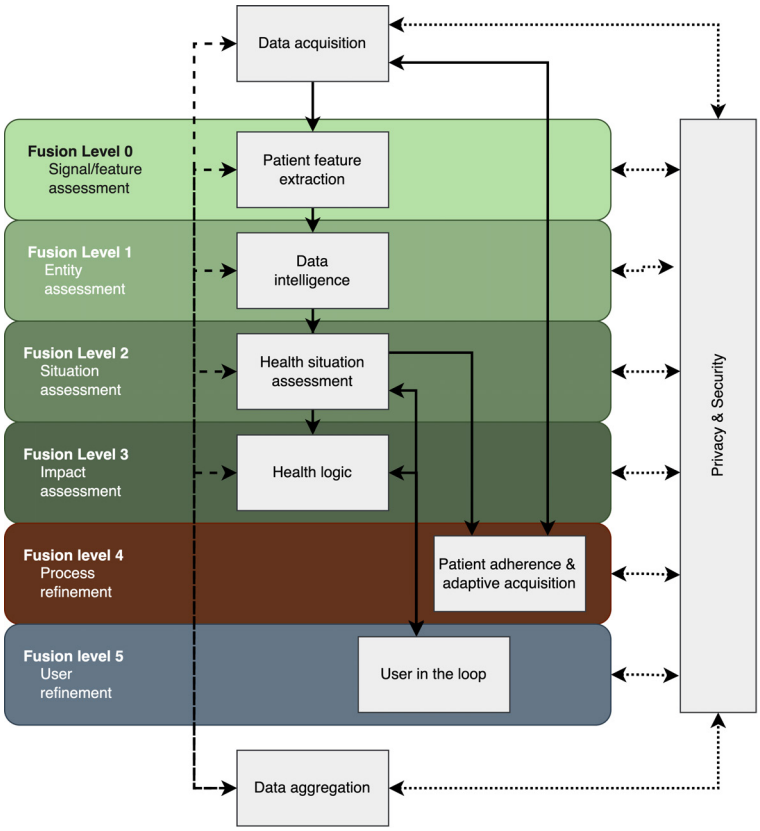


Figure 2.11: Stages of the clinical process as data fusion layers.

Overall, the previous health-centric data fusion model can serve as a conceptual framework to analyze and build mHealth applications. In this sense, a systematic review of more than 50 mHealth services in literature was been conducted in the coauthored paper [Gon+24]. The objective of the **review was to identify common mHealth applications features for each layer of the health data fusion model**. A summary of that analysis together with a description of each data fusion layer is included in the following subsections. Within those sections, common features are identified in bold and with a key between square brackets. Also, Table 2.1 at the end of this subsection provides an outline of these features, that will be then considered in Section 2.2.3 to build a comprehensive mHealth architecture.

2.2.2.1 Data acquisition

A precondition for data fusion is to gather relevant data from heterogeneous devices. For this purpose, patients' smartphones are commonly used. This data gathering may consist in **active sending of patient data [AC1]**, including information directly provided by users through forms or questionnaires (e.g., their self-perceived health status or mood [Hsu+24]), or measurements from non-connected devices (e.g., glucose levels [Vil+15]). Additionally, **smartphones embedded sensors [AC2]** can provide continuous and non-invasive contextual data such as location, social interactions (e.g., messaging, social networks usage or phone communications statistics [Bur+11]) or behavioral patterns (e.g., physical activity or sleep [AOJ22]). Finally, smartphones can also work as a **hub to external IoT / wireless devices**

[AC3] (e.g., wearable ones, such as activity bands [Ber+19], or non-wearable ones, such as weight scales or infusion pumps [Won+20]).

2.2.2.2 Data aggregation

Once data is acquired, **data aggregation and integration [AG1]** is a requirement to successfully perform the data fusion process. This includes supporting elements such as methods of data management, multi-source data merge and data storage, and strategies to integrate data into internal hospital systems. As discussed in Section 2.2.1 , a centralized platform (e.g., cloud computing platform) is usually required to interconnect the final devices and to provide both storage and analysis resources. It is expected that both structured and non-structured data stores might be used depending on the type of information to be recorded. Stores might be deployed on premise or on cloud service providers, as is the case of [Pie+21].

For the moment, the use of mHealth systems remains silo oriented, so they are often not fully integrated with existing Hospital Information Systems (HIS), which hold the registry of medical records, patient data and organizational information. However, full bidirectional **interoperability with Electronic Health Records (EHR) [AG2]** would enable the integration of mHealth-retrieved data with traditional hospital processes. Multiple standards can be used, such as the HL7 FHIR (Fast Healthcare Interoperability Resources) which enables the exchange of information between a wide range of different healthcare systems and health monitoring devices [El-+19].

2.2.2.3 Patient feature extraction

Collected data can be initially used to extract relevant features. Literature shows that a **data preprocessing [FE1]** can remove environment noise and motion artifacts that may have been introduced during data acquisition [Ban+15][Sar+21].

Then, data can also be processed to generate interpretable features (entities, patterns or events with a given meaning or possible medical relevance) which can be later used to make further analysis or take health-related decisions. Processing often takes advantage of AI algorithms. For example, the use Convolutional Neural Networks (CNN) for image processing is widespread using existing networks (e.g., Xception [Goh+24], YOLO [SRZ19]). DL can also extract complex information from large volumes of raw sensor data such as quality sleep features from vital signals [Ber+19] or movement patterns from IoT data [Sye+19].

2.2.2.4 Data intelligence

The previously collected features can be used to obtain information related to the patient's health. One way to obtain this is through **anomaly detection techniques [DI1]**, which can refer to detecting anomalous values or relevant changes in the evolution of features. This may include comparison with thresholds that indicate risk if exceeded (e.g., heart rate value to detect tachycardia or bradycardia [KA20]), or further data analysis to identify anomalous patterns [34] (e.g., detect variations in medication adherence or sleep quality based on IoT data [BLG16]). Also, an **inference engine [DI2]** can be used to obtain more sophisticated

health features, such as breathing state identification based on seismocardiogram (SCG) and electrocardiogram (ECG) signals [Cho+21].

As in the previous section, such approaches can be assisted by the application of **ML, data mining or AI solutions [DI3]** reducing significantly computing costs and time [KA20] and obtaining more accurate results [Jav+22]. AI can be used to learn from the data collected, automating analysis processes (by combining statistics and classification) and generating customized information that can feed decision support systems [GKK23].

2.2.2.5 Patient situation assessment

Deepening on the fusion chain, mHealth systems may perform a **health state evaluation [HA1]** by obtaining a comprehensive health status picture, evaluating patient's risk to a given disease [KA20] or developing predictive models for early detection of diseases and changes in health [Vir+22]. Relevant information (e.g., key health indicators, health risk) that may be useful for posterior decision making (e.g., recommendations, diagnosis) is extracted. Although the use cases in literature are generally focused on specific diseases, this layer could also perform an integrated comorbidity analysis to create an overall picture of the patient's health.

2.2.2.6 Health logic

Patient situation assessment can ultimately be used to support clinical decision-making, and to implement the health logic of each system. This might include patient recommendations (e.g., lifestyle suggestions and alerts), generation of alternative clinical or diagnosis courses of action (CoA, in fusion terminology) [Mor+21]. It can also enable different workflows depending on the health status, so proportional actions with human-in-the-loop (e.g., alerts) can be carried out [Un+21]. To handle part of these functions, it is usually needed to count on **rule-based functionalities [HL1]**, which executes a set of predefined rules based on specific conditions (which may be related to medical diagnosis and treatment protocols). Logical conditions and actions can be implemented to decide when to deliver a notification, or to decide which information is displayed [Cec+19]. Expert systems, case-based reasoning and other AI tools may also be used to implement the core health logic of the system. Algorithms for decision support can be used to take the above-mentioned actions based on patient information and provide real-time feedback [MHR20].

2.2.2.7 User-in-the-loop

User-system interactions can empower clinicians as decision-makers in semi-automated processes integrated into the clinical pathway [Faz+22]. This can be achieved through **communication [UL2] between clinicians and patients or caregivers**, through calls, chats or appointment scheduling. Patients can also **receive notifications [UL1]** related to reminders, alerts or communication with clinicians. **Data visualization [UL3]** (displayed as graphs or health indicators [Sob+18]) is also key in mHealth systems.

2.2.2.8 Patient adherence and adaptative acquisition

Within the data fusion process, accurate and continuous data collection is crucial to make the system knowledge base grow. Therefore, literature considers the usage of management processes to ensure sufficient data acquisition for the system's effectiveness. A key element element in those processes is ensuring strong user adherence (i.e.: the degree to which users or individuals follow and engage with the intended use, guidelines and recommendations of the digital system [McG+13]) to the system. **Adherence techniques [AD1]** encompasses factors like usage frequency, consistency, guidelines follow-up, task completion, engagement, and data collection. Examples within this layer include adapting the notifications delivered to the patient or the visualization of data (e.g., including gamification models to enhance user interaction [Cec+19]) to improve adherence. Meanwhile, adaptive acquisition refers to the intelligent adjustment of non-active data collection.

2.2.2.9 Non-fusion additional feature: privacy and security

Nowadays, **security and privacy [S1]** features are implemented by-design. Security refers to the protection of information systems from unauthorized access, disclosure, alteration, destruction, or disruption, while privacy refers to the right of individuals to control the collection, use, and disclosure of their personal information. Although they are different concepts, they are usually addressed through combined technology approaches. For example, in the EU, the General Data Protection Regulation (GDPR) is primarily concerned with privacy rather than security, although it does address security as part of its broader framework for protecting personal data.

Securing patient data may rely on encryption techniques [Ber+12] (e.g.: using public and private key encryption to protect user data during transmission and storage). Blockchain techniques can also be used for this purpose, as well as providing transparency and traceability in data management. Additionally, to ensure privacy and security, personal data may be stored independently of medical data [Lob+23]. Also, the regulations of each region may require an informed consent from the patient to exchange health information [Kio+21]. Therefore, in some cases, clinicians will only be able to access patient data with the patient's authorization during the clinical visit [Liu+23b]. Thus, it is necessary to provide authentication techniques so that only authorized healthcare professionals have access to patient information.

Data fusion layer	Identified mHealth features	
Data acquisition	AC1	Active sending of patient data
	AC2	Smartphone-embedded sensors
	AC3	External IoT/Wireless devices
Data aggregation	AG1	Data integration and aggregation
	AG2	Interoperability with EHR
Patient feature extraction	FE1	Data preprocessing
Data intelligence	DI1	Anomaly detection capabilities
	DI2	Inference capabilities
	DI3	AI-based solutions
Patient situation assessment	HA1	Health state evaluation
Health logic	HL1	Rule-based functionalities
Patient adherence and adaptative acquisition	AD1	Adherence techniques
User-in-the-loop	UL1	Notifications
	UL2	User to user communications
	UL3	Data visualization
Security and privacy	S1	Security and privacy

Table 2.1: Common mHealth features identified in each of the considered health data fusion layers.

2.2.3 A multipurpose mHealth architecture

Considering the common features identified throughout Section 2.2.2 and summarized in Table 2.1 , this section proposes a multipurpose mHealth architecture that integrates the previously established health-oriented data fusion model and harnesses the capabilities of AI and IoT within its workflow.

The proposed architecture (as depicted in Figure 2.12) is service-oriented: relevant services have been identified encapsulating related functionalities in each layer. This approach enables modularity as only those services that are relevant for each mHealth implementation are required. In this sense, Section 2.2.3.1 describes the different layers within the architecture (inspired on the previous data fusion model) and the required data flows.

Overall, the architecture aims to be a concept tool to guide the development of mHealth systems. It intends to help developers to identify and implement common reusable components that fulfills usual mHealth systems features. As a result, it enables developers to focus on the "health logic" and in the AI analysis algorithms. This is possible thanks to the architecture instantiation guidelines described in Section 2.2.3.2

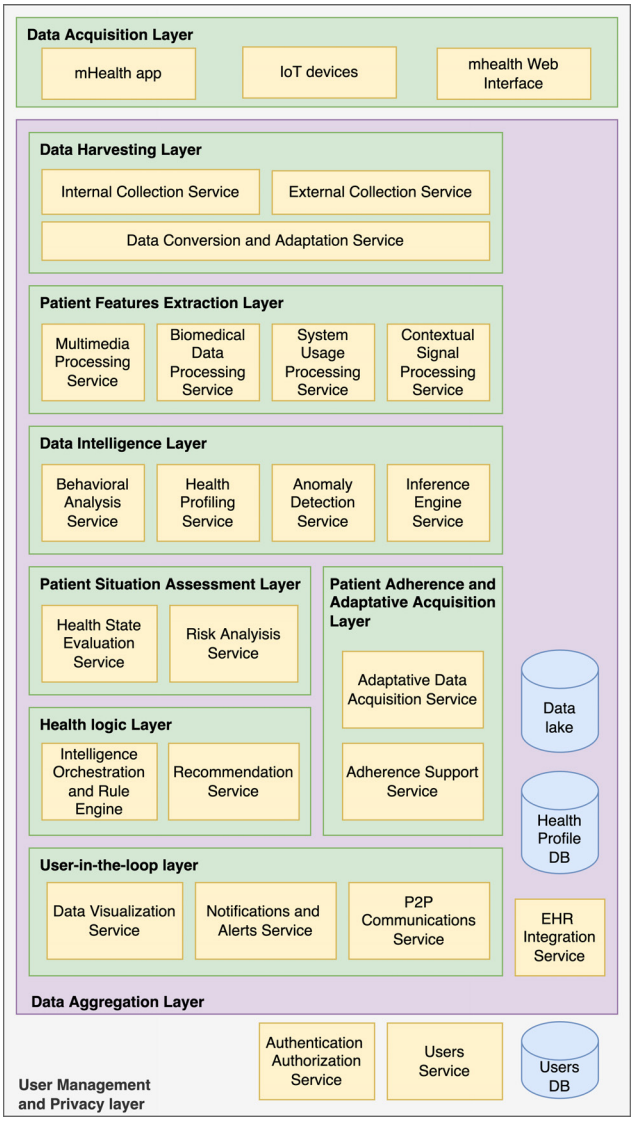


Figure 2.12: Proposed multipurpose mHealth architecture.

2.2.3.1 Architecture services and workflow

Within the architecture, data acquisition is integrated into two layers: the **Data Acquisition Layer** and **Data Harvesting Layer**. The first one includes apps and desktop interfaces, which allows requesting and retrieving data directly from users and connected IoT devices. The second one ingests this information (interaction 1a of Figure 2.13) together with data from APIs provided by IoT vendors (interaction 1b). As all raw data is received, it is passed through an adaptation and conversion service to maintain a uniform format and ensure integrity. Then, this data is stored in a data lake, as can be observed in the interaction 1c.

To enable the fusion and exploitation workflow, the **Data Aggregation Layer** includes the common features that allow storing and integrating the different sources of data. In this sense, the layer includes storage resources for raw (data lake) and structured data representing the patient's health status (health profile Database (DB)). These DBs are decoupled from the

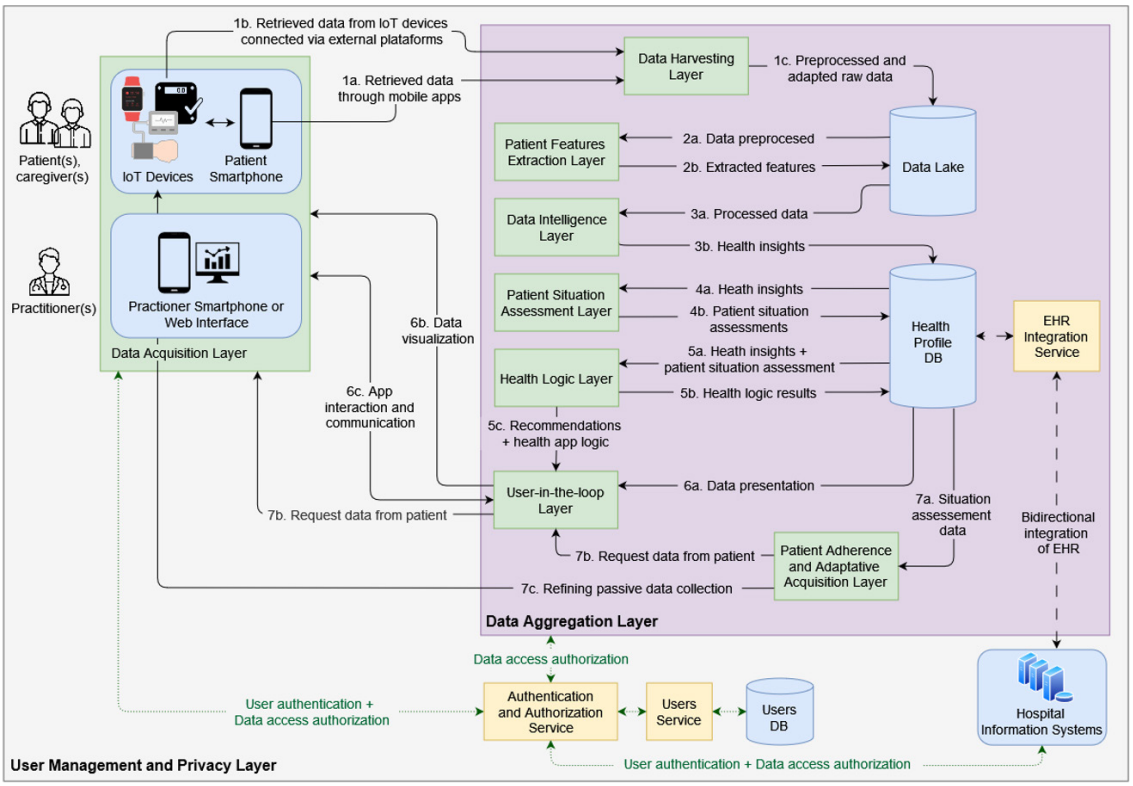


Figure 2.13: Workflow in the proposed mHealth architecture.

identity of the users for patient privacy and security reasons. Finally, this layer also allows the integration of the platform with the HIS by the EHR integration service.

Starting with the data analysis, the **Patient Features Extraction Layer** enables domain-specific processing thanks to independent services oriented to different types of data (as each type requires specific techniques appropriate for it, including AI): multimedia, biomedical, system usage and contextual. To carry out these analyses, the different services obtain the previously preprocessed data from the Data Lake (interaction 2a in Figure 2.13) and store the extracted features in the same database (interaction 2b).

Next, the **Data Intelligence Layer** encompass different services focusing on advanced analysis: complex information estimation (inference engine), detect anomalies, patient behavior analysis or health profiling. This analysis is fed by the information already preprocessed and stored in the Data Lake (interaction 3a in Figure 2.13) and stores the health insights obtained in the Health Profile Database (interaction 3b). Following on with the proposed data fusion chain, services in the **Patient Situation Assessment Layer** provide a comprehensive view of the patient's health and estimate risks (interaction 4a in Figure 2.13). The obtained health profile is persisted (interaction 4b) within the aggregation layer so that it can be visualized by users for decision-support.

Finally, within the **Health Logic Layer** (interaction 5a in Figure 2.13), the intelligence orchestration and rules engine service enables the integration of all AI-assisted services of the previous layers to execute the health logic of the system through a set of rule-based

or scheduled actions. Recommendations and course of action generation are also provided (interaction 5c) with a dedicated service based on the acquired information.

The **User-In-The-Loop Layer** considers services oriented to provide data visualization (interaction 6b in Figure 2.13 , which offers personalized data presentation, dynamic dashboards, advanced filtering, etc.), P2P communication, and notifications or alerts (interaction 6c). The behavior of these services is closely linked to the implemented health logic. Also tightly coupled to this layer is the **Patient Adherence and Adaptative Acquisition Layer** (interaction 7a and 7b Figure 2.13) for the management and adaptation of the data acquisition layer. The later also includes an adherence support service to encourage ongoing app use, remind them to actively provide information or to follow the recommendations and treatments.

Finally, as a supporting, transversal layer, within the **User Management and Privacy Layer**, the users service manages existing users and roles including sign up to the platform and log in the different applications, and the authorization and authentication service assigns users an authentication key associated with their specific permissions, which is used across the platform to guarantee that a user only access authorized information. Similarly, services within the platform are also registered in the service and use keys to access and decrypt DB information. Optionally, further enhancing privacy, the service also allows generating one-use time-limited access codes to ensure that patient's information is accessible only by assigned professionals, under their authorization.

A description of the services within each layer is performed in Table 2.2 and Table 2.3 . It is important to note that the architecture does not impose any specific implementation or protocol to use, this is finally decided in the light of the mHealth context. Also, AI can be applied across layers in the vast majority of services, from low-level feature extraction to high-level support to recommendations or courses of actions, going through risk analysis, behavioral analysis or profiling for different purposes.

Layer	Service	Fusion Features	Description	Notes on implementation
Data Acquisition	mHealth App	U1, D1, AC1-AC3	iOS or Android native app (e.g., final data visualization, communication, data retrieval) and serves as hub to gather info through the mHealth system.	OS-dependent APIs to acquire data from accelerometers, light sensors or positioning engines. App usage statistic or communication statistics may also be retrieved using virtual probes.
	IoT devices	D1, AC3	Wireless health devices that provide health metrics, such as smartwatches, smart-scales, oximeters, sensor clothing, sphygmomanometers...	Manufacturer specific APIs and SDKs can be used to retrieve data on NFC, Bluetooth or Wi-Fi. Standard protocols such as Matter may be used for health devices.
	mHealth web interface	U1, U2, D2, AC1	Clinician interface for information retrieval describing patient's self-perceived health and treatment details, data visualization and graphical insights generation, communication, treatment update, etc.	Web development frameworks such as React or Angular can be used to develop this interface.
Data Harvesting	Internal gathering service	AC1-AC3	Centralizes data obtained from the architecture internal interfaces (i.e., mHealth app and mHealth web).	Based on REST or streaming-based technologies (e.g. Kafka). To ensures data integrity, it may include outlier deletion, duplicate data deletion, time alignment, etc.
	External gathering service	AC3	Retrieval from external IoT, vendor-specific platforms.	Manufacturer specific APIs and SDKs when direct connection to device is not feasible.
	Data conversion & adaptation	AG1	Data standardization.	Ontologies (e.g. SNN), may be used to associate sensor data with interpretable concepts.
Data Aggregation	Data lake	AG1	Database/s, centralized repository for storing large amounts of multimodal raw data (patient's data, medical records, diagnostic images...).	Amazon S3, Azure Data Lake Storage and Google Cloud Storage are examples of data lakes provided as cloud-service.
	Health profileDB	AG1	Database/s containing user-linked electronic health profile and evolution: risk level, adherence to treatment, behavioral analysis, adherence to system, etc.	InfluxDB is a time series database, while MongoDB or PostgreSQL are example of OODBMS that can store health profiles efficiently. Cloud-service available.
	EHR integration service	AG2	Provides a bidirectional communication with the hospital services and databases, adapting format for compatibility and usability.	Standards (e.g., LONIC, HL7 CDA or HL7 FHIR) can be used to simplify clinical information transcription. Also, there are standards specialized in different areas of medicine, such as DICOM for medical images (e.g., radiology, cardiology imaging, and radiotherapy) or SNOMED CT for diabetes insipidus terminology.
Patient Features Extraction	Multimedia processing service	FE1	Processing of multimedia data sent by the patient (e.g., pictures, videos, audios) or the clinician (e.g., X-rays) to accelerate the data analysis.	Includes DL methods, e.g. CNN for object detection and classification from images or video, and NLP for disease detection from audios.
	Biomedical data processing service	FE1	Processing of bio-signals (collected from IoT devices) or relevant data for the clinical use case (self-reported by the patient), to obtain interpretable features, or to be used in further services.	Includes ML/DL methods for variations detection in vital signals, e.g. body temperature or ECG.
	System usage processing service	FE1	Processing of data related to system usage, including actively submitted data and metrics about patient interaction with the app.	Hybrid learning combining statistical metrics and Markov Models (MM) can be used to model standard behavior and analyze outliers. Also, ML can be used to classify activity (e.g., Naïve Bayes).
	Context processing service	FE1	Processing of embedded sensor data to extract information about the context of app usage (e.g., location) or life patterns (e.g., steps, sleep patterns).	ML methods can be used to analyze phone sensor value (e.g. Decision Trees).

Table 2.2: Services description and implementation guidelines in the mHealth architecture (1/2).
Fusion features refer to the common mHealth applications features identified in Section 2.2.2

Layer	Service	Fusion Features	Description	Notes on implementation
Data Intelligence	Inference engine service	DI2, DI3	It obtains new information based on the knowledge or combining different types of data.	AI algorithms (e.g., Regression algorithms, Decision Trees, Causal Graphical Models) or counterfactual analysis can be used to explain events, extracting causal relationships from data.
	Anomaly detection service	DI1, DI3	Detection of any data out of normal standards or with negative trend.	DL methods may be used, such as CNN, Isolation Forests, NLP, etc.
	Behavioral analysis service	DI1-DI3	Analyzes system usage or contextual data (e.g., location data) to obtain information about patient behavior.	Clustering techniques (e.g. DBSCAN, Naïve Bayes or Hidden MM) and predictive models (e.g., Logistic Regression) may be used.
	Health profiling service	DI3	Patient profiling in groups to obtain useful information about their state of health.	Clustering techniques to profile patients into severity groups or patterns of illness evolution.
Situation Assessment	Health state evaluation service	HA1	Health evaluation based on statistical analysis and classification algorithms.	ML algorithms can determine the health state using historical data (e.g., SVM).
	Risk analysis service	HA2	Detection of worsening or risk evolution.	Both statistical techniques and ML (e.g., SVM) CNN, NLP, etc.
Health Logic	Intelligence orchestration and rule engine	HL1	On the info from previous services (health state, risk level, adherence...), it decides which actions to take. It coordinates analysis modules. It dispatches instructions to customize info display, data collection, reminders...	Orchestration can be performed via REST or following event-based architectures. As for the rule-based execution, RulesEngine by Microsoft may be applicable. Alternatively, cloud-based or self-hosted lambda-like functions can be used to implement this service.
	Recommendation service	HL1	Delivery of recommendations to patients or clinicians on the health profile.	AI can be used for personalization, e.g. classification algorithms.
User-in-the-Loop Layer	Data visualization service	UL3	Dashboard components for information, monitoring and assessment.	AI tools for view customization. Use of multiplatform frameworks (e.g. Kibana or Grafana).
	P2P communication service	UL2	Communication between clinicians and patients (or caregivers).	WebRTC may be used to provide real time text, audio and video communications.
	Notifications & alerts service	UL1	Reminders and alerts delivery to the stakeholders (reminders of data input, alerts on risks, new message notifications, etc.).	Supabase or Firebase services can be used to generate PUSH alerts. Alternatively, SMS gateways services allow to distribute SMS messages.
Patient Adherence& Adaptative Acquisition	Adaptative data acquisition system	AD1, AC1-AC3	Adaptive sampling. Analysis of the quality and amount of data from sensors. Sensor collection status check-up.	Diagnostic tools for bad connection or insufficient permissions given to the app check-up.
	Adherence support service	AD1, AC1	Analysis of system usage to determine how to improve adherence and coordinate the needed actions (e.g. visualization changes, requests for information, etc.)	Clustering techniques can be used to identify adherence categories of users and customize actions to improve it.
User Management & Privacy	Authentication & authorization service	S1	Delivery of access to data when the user has permission, through the exchange of authentication keys and access credentials.	Custom-built, on standard protocol such as JWT or OAUTH or integrated with some external Identity and access management (IAM) services (Google Cloud IAM, Azure Active Directory, Okta).
	Users service	S1	User roles management and access to the user DB.	User Management Systems (UMS), often include the User DB, such as Active Directory and OpenLDAP. They are also provided by the several cloud-service providers.
	Users DB	S1	Database containing user personal data (e.g., name, age, telephone number...).	In the custom-built solutions, they are generally implemented using a generic database.

Table 2.3: Services description and implementation guidelines in the mHealth architecture (2/2).
Fusion features refer to the common mHealth applications features identified in Section 2.2.2

2.2.3.2 Architecture instantiation methodology

In the previous subsection, the proposed comprehensive mHealth architecture grounded on a data fusion process has been presented. It is intended to serve as a general blueprint to streamline the development of mHealth systems. In this sense, designers can instantiate the architecture by selecting the modules pertinent to their mHealth use case. To do so, the following four-step methodology is proposed:

1. First, designers shall **describe the goal of their mHealth system** (i.e., which service logic they want to implement) and identify the target users, data sources, devices (both for sensing and interface), and specific privacy/security constraints.
2. Subsequently, designers must **identify the functional requirements** for each stakeholder, mapping and analyzing them with **respect to the common mHealth Fusion Features** (as outlined in Table 2.1).
3. Then, designers shall **select which services are required for the architecture**. To this end, Table 2.2 and Table 2.3 also identify which mHealth features are fulfilled by each service to enable a quick identification of relevant services for each mHealth feature. Therefore, with the help of this information, designers can make informed decisions by selecting the necessary services so that their mHealth architecture is instantiated.
4. Finally, the fifth column in Table 2.2 and Table 2.3 suggest potential implementations and AI algorithms that can be used in each service, guiding the final development and implementation of the mHealth system. The tools mentioned in those tables include among others: predictive algorithms, anomaly detection algorithms, clustering algorithms. With those suggestions, the **designer can determine which of these technologies will be used to implement the mHealth system**.

2.2.4 Validation: architecture instantiation for SARS-COV-2 monitoring

In order to assess the value and feasibility of the architecture proposed in Section 2.2.3.1 , a use-case based validation can be carried out. In this sense, the architecture instantiation methodology presented in Section 2.2.3.2 is going to be followed to implement a real mHealth use case. As a result, it is possible to prove that the architecture is not a mere conceptual construct but a practical technical design that can be developed successfully with existing technologies.

In the supporting publication [Gon+24], several use-cases are discussed to demonstrate that the architecture fulfills mHealth systems needs of a wide range of use-cases with assorted requirements. From those use-cases, the author participated in the implementation of a **SARS-COV-2 patients monitoring app** which was developed through the proposed methodology as presented in the following subsections.

2.2.4.1 Step 1. High level definition of the mHealth system to be developed

The first step within the methodology is to describe the mHealth system to be developed by identifying potential uses, selecting which devices are to be integrated and outlining the expected health logic. The selected use-case intended to reduce the care load of healthcare systems through periods of hight SARS-COV-2 transmission. Also, high contagion risk advised against in-person monitoring of non-critical patients in primary healthcare centers. In this context, a mHealth system would enable patients to report their symptoms, clinicians to easily track their patients remotely (reducing healthcare workers workload) and have an alert system in case of health condition deterioration.

Therefore, the intended mHealth application is targeted to both patients and clinicians. Interaction with the mHealth system is expected through both smartphones (for patients) and a web app (for clinicians). Finally, the expected health logic consists in remote monitoring of large number of SARS-COV-2 patients including early warning risk notifications.

2.2.4.2 Steps 2 and 3. Identification of mHealth data fusion features and architecture instantiation

With the background information gathered in the previous step, it is possible to identify which mHealth features are required from those depicted in Table 2.1 . Therefore, relevant features for the considered use-case have been selected in Table 2.4 (step 2 from the methodology).

AC1	AC2	AC3	AG1	AG2	FE1	DI1	DI2	DI3	HA1	HL1	AD1	UL1	UL2	UL3	S1
Active sending of patient data	Smartphone-embedded sensors	External IoT/Wireless devices	Data integration and aggregation	Interoperability with EHR	Preliminary data preprocessing	Anomaly detection capabilities	Inference capabilities	AI-based solutions	Health state evaluation	Rule-based functionalities	Adherence techniques	Notifications	User to user communications	Data visualization	Security and privacy
✅			✅		✅	✅			✅	✅		✅	✅	✅	✅

Table 2.4: mHealth data fusion features required for the SARS-CoV-2 use case

From those features, it is possible to select relevant services layer by layer with the help of Table 2.2 and Table 2.3 . The result is the instantiated architecture that is depicted in Figure 2.14 . First, in order to consider patient reported information (AC1), it is necessary include the *mHealth app* and *mHealth web interface*. This information is forwarded to the system using the *internal collection service* and the *data conversion and adaptation service* from the Data Harvesting Layer. Data pre-processing (FE1) is also required to handle symptoms reported by the patient through the *biomedical data processing service* which allows removing non-consistent data.

Relevant features are then used in different services of the following layers. Particularly, the *health profiling service* is used to group patients depending on their current health status so that clinicians can be efficient in patient follow-up (e.g.: massive notifications to a group of patients). Also, the *anomaly detection service* (DI1) helps the implementation of the *risk detection service* to alert both clinicians and patients in case of sudden worsening. The obtained information serves the Health Logic Layer which is implemented through the *Rule engine service*. The objective of this service is to orchestrate the behavior of the overall mHealth system including event generation based on thresholds (e.g.: patient risk exceeds a value) to fire notifications and other actions.

Regarding the User-in-the-Loop Layer, the *data visualization service* (UL3) is needed to display the patient's status to the clinician in an adaptive manner (e.g., selecting charts / indicators to quickly interpret the evolution or displaying first evolution of critical patients). Also, the *notification and alert service* (UL1) is used to remind the patient to enter data and to alert the clinician of any detected problem. Additionally, the *p2p communication service* (UL2) is especially critical for cases of home confinement due SARS-CoV-2.

Finally, it is mandatory to include security and privacy (S1), which is developed in the *authentication and authorization service* in order to assign roles to registered users, within a hierarchy (patient, clinician type and supervisor). This simplifies patient management by ensuring users access only to needed data (clinicians only access their patients and a supervisor manages the patients assigned to each clinician).

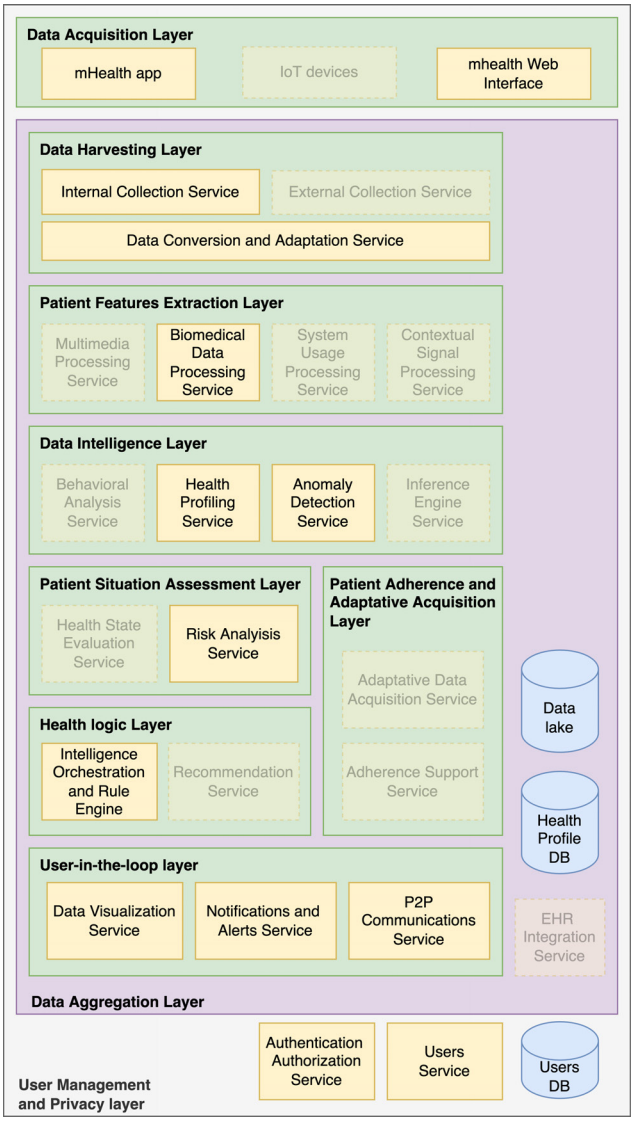


Figure 2.14: Proposed multipurpose mHealth architecture.
In the figure, the boxes corresponding to those services that are not used have been faded (lighter dim color).

2.2.4.3 Step 4. mHealth system implementation

To conclude the presented methodology, the instantiated architecture (depicted in Figure 2.14) has been implemented with the help of the proposed guidelines. In this sense, RESTful communication has been applied to the frontend and mobile applications, as well as for the majority of backend module communication. For databases, MongoDB object-based store has been used. Regarding notifications, the choice is Firebase for mobile apps and WebSockets for browser-based frontend notifications. Backend services have been developed ad-hoc for the proof-of-concept implementation in NodeJS platform. As a result, the web and mobile (both for Android and iOS) interfaces shown in Figure 2.15 are available to patients and practitioners. The system was deployed for patient follow-up with a total of 38 enrolled patients, as discussed in [Yus+20].

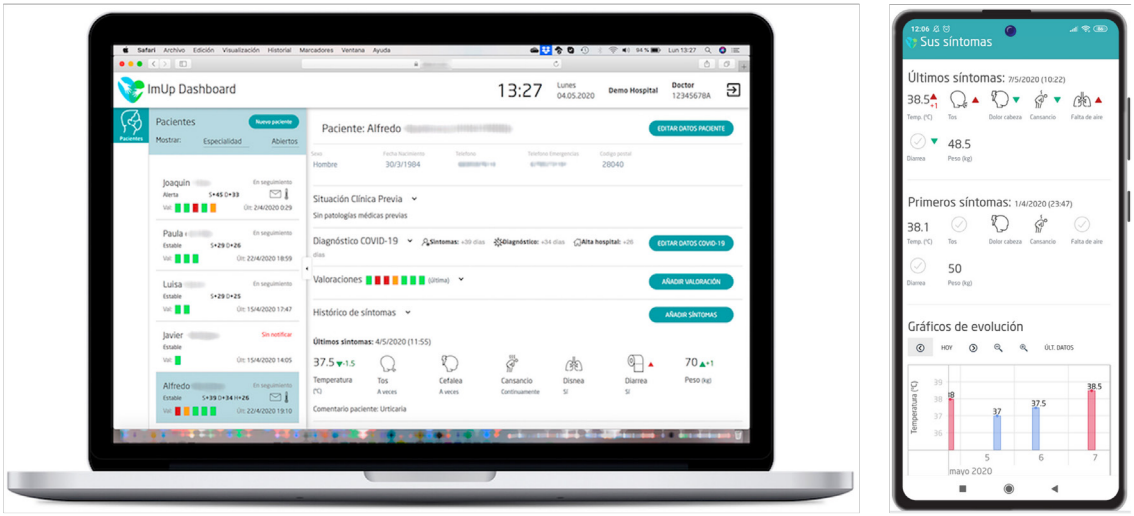


Figure 2.15: IMUP application as an example of the proposed mHealth fusion architecture

2.2.5 Conclusion and limitations

Smart territories can leverage citizens' smartphones to improve healthcare provision through mHealth applications. In this regard, the literature review in Section 2.2.1 has identified a lack of standardized structure and components in the design of mHealth systems. Current proposals identify basic services for data acquisition, processing or visualization. However, they do not identify specific funcional components related to health needs which makes it difficult for the developer to instantiate the architecture and identify reusable components.

As an alternative, Section 2.2.3 has proposed a comprehensive architecture that arises from combining a bottom-up functional analysis approach with insights from existing literature on mHealth systems, complemented by a customized adaptation of the JDL data fusion model (in Section 2.2.2). The architecture consists of eight modular data fusion layers with customizable microservices, tailored to specific system requirements. Within those layers, AI modules can leverage data to enrich the capabilities of both patients and healthcare professionals.Additionally, two complementary layers are incorporated to ensure efficient data aggregation and privacy protection. In order to guide developers in implementing the architecture, an instantiation methodology has been discussed including algorithms and software guidelines.

The validation of the proposal in Section 2.2.4 has been conducted through a relevant use-case (i.e.: SARS-CoV-2 patients remote monitoring app) to provide evidence of its feasibility and effectiveness in delivering the required health logic. The approach is thought to lead to cost-effective developments to monitor citizens healthcare in a pervasive way. Also, thanks to its modularity, the architecture enables solutions with better functional extensibility and simplified system maintenance. Another advantage is that it facilitates the development of horizontal systems capable of extending the use of the system to tackle evolving comorbidities in a cohesive manner, instead of the current silo-oriented approach.

On limitations and future work, further evidence could be gathered from more advanced

use-cases (e.g.: track multiple diseases) and with a larger number of users (to assess scalability issues). Also, while security mechanisms are conceptually integrated into the proposed architecture, protection against cyberattacks should be further considered together with compliance protocols with existing data privacy laws in different countries. Finally, the integration of mHealth with EHR is usually bypassed to accelerate system development, making clinicians to work with different systems and duplicating their work (e.g., data input in each system). Therefore, bidirectional integration with external data providers, such as clinical systems, is still a relevant open issue.

——— * * * ———

Chapter summary

Instrumentation of smart territories through the Internet of Things paradigm enables real time monitoring. In essence, monitoring systems are Data Fusion and Resource Management systems that process and integrate information from different sensors and dynamically manage sensors to improve the overall monitoring performance. This section has presented two novel data fusion architectures to deliver cost-effective, ubiquitous monitoring.

In Section 2.1 , the usage of a fleet of drones for wide area surveillance missions have been explored. The usage of AI algorithms to automate drone operations usually hinders human supervision and operations explainability. To this end, the first contribution of this section is a DF-RM architecture that structures the problem in such a way that it is possible to know which task is being executed by each drone and to visualize the automated ground target detections. Also, in order to test the architecture safely and in a lawful manner, a digital twin platform has been presented as a testbed for indoor flying.

In Section 2.2 , the usage of smartphones as a monitoring asset has been described. mHealth applications allow to deliver healthcare services at any time and in any place through patients' smartphones. However, these applications are usually silo-based and lack standardisation to reduce development and maintenance costs. In this sense, a conceptual model considering mHealth systems as data fusion chains have been presented to detect common features across applications. Based on it, a modular, micro-service architecture with AI modules has been defined as a comprehensive framework to implement assorted mHealth use-cases.

Chapter 3

Fair Unmanned Traffic Management: smart services to enable drone operations

As discussed previously in Section 1.2 and exemplified in Section 2.1 , the usage of drones unlocks a myriad of novel applications including ubiquitous monitoring and novel services (e.g.: urban air mobility, parcel delivery...) in smart territories. This will happen to such an extent that UAS are expected to become predominant in low-level airspace. Yet, for this development to take place, a competitive drone services market must be fostered. In this sense, authors in [SES16] argue that additional advances should be considered in three fields:

- **Regulatory advances** are needed to gradually allow more advanced aerial operations. Moreover, this must be done while maintaining the necessary safeguards to maintain safety and considering social concerns (e.g.: privacy).
- **Technological advances** are required to improve unmanned aircraft robustness and reliability. Also, the development of Dectect And Avoid (DAA) systems, new technologies for the air-ground radio link, and enhanced cybersecurity will help to scale up commercial drone usage.
- **Novel traffic management solutions** are needed to enable a seamless integration of manned and unmanned aviation. Indeed, as discussed in [SES18], new air traffic control systems should implement a shared, safe airspace usage by all aircraft.

Focusing on the latter aspect, current Air Traffic Management (ATM) systems are highly dependent on the human factor (e.g.: air traffic controllers, verbal communication between actors) which is not easy to make compatible with the inherent automation of drone operations. Furthermore, they would be unable to scale to cover the foreseen high number of unmanned operations. Consequently, it is necessary to develop new control systems, denoted **Unmanned Traffic Management (UTM) systems**, that are better adapted to the new requirements imposed by unmanned operations, mainly: flexibility, scalability, and automation.

This chapter explores these UTM systems considering them as intelligent and distributed

services that interconnect and coordinate different actors within the smart territory (e.g.: drone operators, autonomous aircrafts, authorities, Counter UAS (C-UAS) systems) to ensure safe, efficient and innovative aerial operations. They represent the cornerstone of the drone ecosystem as they will become a comprehensive tool to facilitate widespread and heterogenous (manned and unmanned) aerial operations over the same airspace. Thus, as identified in Figure 3.1 , **UTM systems are themselves a smart service for drone operations**.

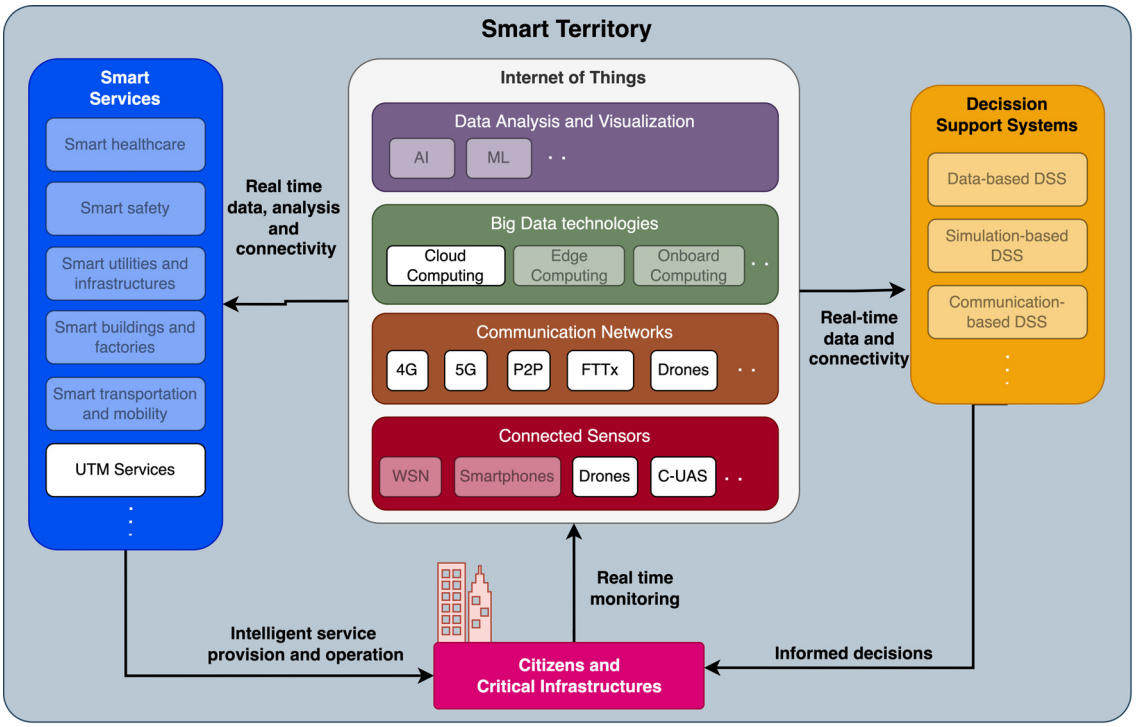


Figure 3.1: Technological elements required for deploying a UTM system in a smart-territory
Relevant technological elements are highlighted over the full architecture

To that end, the chapter is organized as follows. First, Section 3.1 presents UTM systems with a focus on European developments and deployment status. Based on this review, a series of existing challenges and ongoing research topics are identified. These needs are aligned with this chapter contributions. In particular:

- The need for modeling and simulation tools to streamline the testing and validation process of these systems, which will be covered in Section 3.2 .
- The need of managing congestion and ensuring fair airspace access, through a novel strategic authorization process, as discussed in Section 3.3 .

3.1 An introduction to Unmanned Traffic Management systems

Even if slightly different concepts for UTM systems exist around the globe (i.e.: the FAA view is described in [Fed20]), the present review on UTM is focused on the European point of

view for obvious proximity reasons.

3.1.1 An introduction to U-Space: the European UTM concept

In 2016, European member states recognized the potential of a common Drones market through the Warsaw declaration ([EAS16]). From that point, Single European Sky ATM Research Joint Undertaking (SESAR) was tasked of developing a common UTM concept for the whole Union which would be named the U-Space. In this sense, [SES17] summed up the main objectives of that ecosystem:

- Ensure the safety of both aircraft (manned and unmanned) and people on the ground. This applies to both rural and urban environments and to all kinds of airspace.
- Provide a system that must be scalable, flexible and easily adaptable, so that it can respond to changes in demand, technology used and type of operation. All this, while ensuring integration with manned aviation.
- Enable control of areas with high density of operations with automated drones.
- Ensure equitable airspace access for all users.
- Minimize development, deployment and operation costs by making use of pre-existing standards or technologies (GNSS or mobile networks) so that the provision of the service does not represent a barrier in the operators' business model.

Based on these objectives, **the U-space can be defined as a set of digitalized and highly automated services, designed to support access to airspace for a large number of drones in a safe and efficient manner**.

3.1.2 U-space: operational view

To that end, the U-Space operation guarantees safety by enforcing and monitoring UAV separation in two phases:

- **a pre-flight, or strategic, phase** which relies on the prior declaration of flight operations that must be authorized before flight. Operation definition, usually known as flight plan or u-plan, is done through a series of 4D volumes to be reserved for that operation: portions of 3D airspace occupied during a given time. Then, separation is enforced a-priori by ensuring that submitted operations are free of 4D intersection.
- **a in-flight, or tactical, phase** which relies on real-time position reports to monitor aircraft separation, generating alerts in case of loss of separation, non-compliance with the approved flight plan and/or other constraints. Also, tactical deconfliction services are expected in the future to dynamically ensure separation.

Thus, from the point of view of an operator who wants to fly an operation, the U-space would work as described in [CS20]:

1. The operator registers into the U-space ecosystem while also indicating its drones' information (e.g.: identification, aircraft dynamics).

2. The operator submits a u-plan authorization request specifying the aerial mission it wants to perform. To this end, the U-Space ecosystem provides flight plan definition tools that collect relevant information such as weather forecasts, ground obstacles, flight restrictions (e.g.: Notice To Airmen (NOTAM) , no-fly zones) and may also include mission-related trajectory prediction models.
3. As the u-plan is submitted, the U-Space verifies compliance with regulations, airspace requirements (e.g.: restricted areas, maximum flight density, minimum aircraft equipment) and that the requested flight plan does not conflict with other previously authorized flight plans (using a First Come-First Served priority). In case of conflict, alternative u-plans (e.g.: delay the mission, modify the route...) may be provided to the operator to ensure that flights are strategically deconflicted. Additionally, some flights may require manual clearance by another entity, such as a flight in the vicinity of an airport that should be approved by an Air Traffic Controller (ATCO).
4. If the flight plan is accepted, the operation can be activated shortly before take-off. Throughout the flight, the operator periodically forwards relevant information to the U-Space. This includes information such as drone position, speed, and operational status. Additionally, the U-space can also receive positioning information from ground surveillance networks and manned aviation information from the ATM system.
5. With this information, the U-Space ecosystem can track aircrafts in real time providing situational awareness to all airspace actors. In parallel, the ecosystem provides monitoring, alert and separation enforcing capabilities:
   - drones' trajectories are checked against their approved u-plans for conformance, yielding alerts in case of deviation.
   - drones' separation is checked in real time, generating alerts and advisory avoidance maneuver when drones are in proximity or in collision risks.
   - drones' compliance with flight restrictions is monitored.
6. Finally, the u-space ecosystem is also able to handle non-nominal situations such as emergencies, priority flights...

3.1.3 U-space: services view

From a logical point of view, the U-Space operation is articulated through a set of U-Space services: encapsulated logical functions that partially fulfill U-Space objectives. This approach allows a flexible, modular and scalable implementation of the ecosystem. Moreover, and as specified in [SES18], this also enables a gradual roll-out of the ecosystem in four different phases. Relevant services and phases are depicted in Figure 3.2 and discussed in the following subsections.

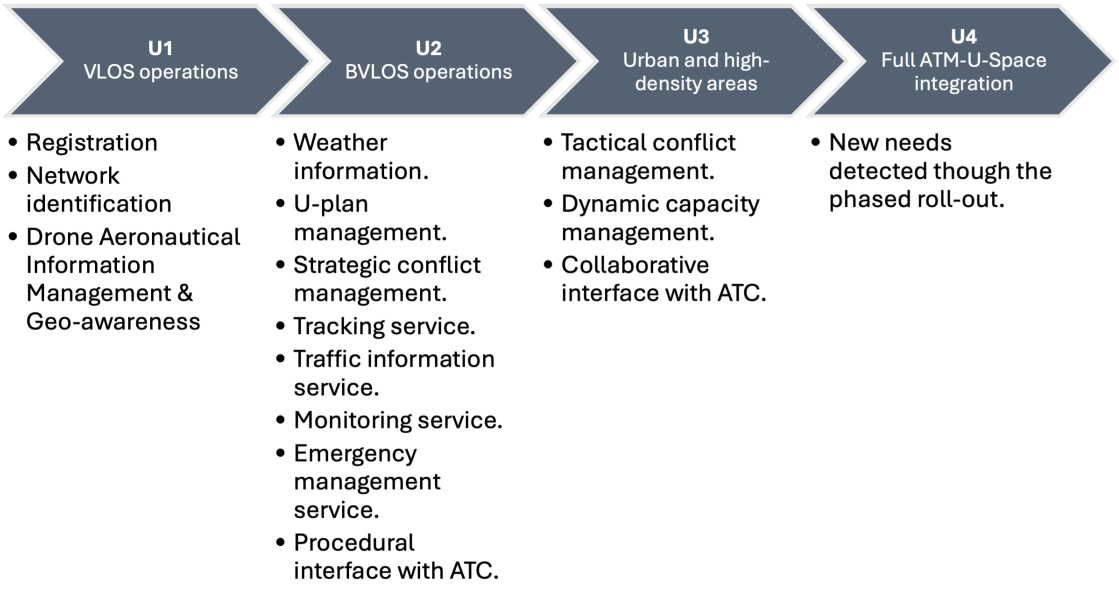


Figure 3.2: U-space services and expected deployment phases.

3.1.3.1 U1 services

The first phase (U1), which was initially expected to be deployed as early as 2019, consists of basic services to register operators, inform them of restricted areas and identify in-flight drones. This initial deployment intended to enable VLOS and Extended Visual Line Of Sight (EVLOS) operations in low-traffic areas through 3 services:

- The **electronic registration service**, which maintains a database with the information operators and their associated drones and pilots, so that this information can be accessed from other services.
- The **network identification service**, which allows to identify any airborne drone and its associated pilot/operator. This service relies in each drone periodically forwarding to the U-space ecosystem information such as its position, associated mission, and their unique identifier in a process which can also be known as telemetry forwarding.
- The **Drone Aeronautical Information Management** and **geoawareness services**, which manages and makes available to operators information of applicable no-fly zones (e.g.: prisons, military bases) as well as aeronautical information such as NOTAMs.

3.1.3.2 U2 services

The second phase (U2) deploys services that enable drone operations management in real time and introduces interconnection with ATM systems. With the new features introduced in this phase, which should have been rolled out by 2022, it would be possible to perform BVLOS flights on a routine basis.

In the pre-flight phase several new services are considered. On the one hand, current and

future weather information is incorporated through the **weather information service**. Working in conjunction with this service is the **u-plan preparation service** that helps operators to define and model their aerial operations through valid u-plans.

On the other hand, services to enforce a priori separation are introduced. For this purpose, the u-plan management service is in charge of orchestrating the authorization process by maintaining a data base of u-plans. This service is also known as **Flight Authorization Service** and manages the u-plan state through its lifecycle from its authorization request to the activation of such request before flight. It is also expected to monitor that the authorization always remains valid. Then, the strategic conflict management service works at the request of the previous service and aims to check for strategic conflicts (i.e.: 4D volume intersection in u-plans, implemented in the **Strategic Conflict Detection (SCD) service**) and to help the operator find alternative flight options in case of conflict (**Strategic Conflict Resolution (SCR) service**).

In the flight phase, drone's positioning information from different sources is incorporated through the **tracking service**. Particularly, the following sources can be considered:

- **Network identification**. As already discussed, this consists in receiving periodic position reports, typically relayed to cloud-based U-space services over public communication networks.
- **ATM surveillance** information interchange to obtain manned aircraft location. This information is typically derived from the integration of measures from a network of Secondary Surveillance Radars (SSR, recently of the Mode S type), Primary Surveillance Radars (PSR), Wide Area Multilateration (WAM) or Multilaterarion (MLAT) systems, and Automatic Dependent Surveillance (ADS-B) stations.
- **UAS Non-Collaborative Surveillance networks**, which may detect drones performing operations that have not been authorized and declared with the U-Space ecosystem (non-collaborative or rogue drones). This information is key in some use cases, such as detecting unauthorized flights over critical infrastructures or real-time tactical conflict detection with collaborative drones or with manned aircraft. The prototypical example of this type of sensors are radar systems but visual, acoustical or RF techniques are also used; as it will be further discussed in Section 3.1.6 . One or several of these sensors can be integrated as part of a C-UAS system, typically used by law enforcement agencies to detect, track and even neutralize (through jamming or kinetic effectors) drones.
- **UAS Collaborative Surveillance networks**. UAS positions are derived from messages sent by the aircraft themselves through an ad-hoc broadcast network. Contrary to their non-collaborative counterpart, collaborative sensors require the presence of compatible equipment on the aircraft (an RF emitter) that, being active, periodically and automatically transmits their kinematic states and some additional state information. Examples of these types of sensors include ADS-B, FLARM or direct remote identification, which will be detailed in Section 3.1.5 .

The tracking service associates and filters measurements coming from these sources to maintain a continuous and identifiable track for each aircraft, that can be distributed to relevant actors

through the traffic information service. Based on this information, the **monitoring service** is in charge of associating each of the tracks to a flight plan and to a drone and providing alerts when required, increasing the operational awareness of the actors. Also, the **emergency management service** is included, which allows the transmission to the stakeholders of alerts generated by the operators regarding flight contingencies: loss of control of the drone, loss of ground-to-air link...

Finally, the necessary connections with the ATM systems are established. Specifically, a **procedural interface with ATC** to enable unmanned traffic in controlled airspace zones or in the vicinity of airports with prior authorization from ATC.

3.1.3.3 U3-U4 services

The third phase (U3), which was initially scheduled for 2027, focuses on ensuring safety in complex and high traffic density areas, especially in urban areas.

In this regard, the **tactical conflict management service** ensures in-flight conflict prediction and resolution to maintain separation between aircraft so that drones can receive flight plan modifications in the air. To do so, the Tactical Prediction Service assesses airspace situation in real time and predicts possible separation losses for which the Tactical Resolution Service proposes potential deconfliction solutions. The **dynamic capacity management service** is also added, allowing the definition of traffic density thresholds by zones and modifying the flight plan authorization process to ensure that these are not exceeded. Regarding the connection with ATM systems, the **interface with ATC is reinforced and becomes collaborative**, sharing aircraft information between the two systems in real time.

Finally, in the last phase (U4) planned for 2035, integration with ATM systems and manned aviation is expected to be completed, in addition to achieving the highest level of automation of all processes.

——— * * * ———

In light of the previous collection of services, the U-space constitutes a system of systems (i.e.: services) that enables routine drone operations while interfacing with existing ATM systems, and authorities. In other words, the U-space is a technological ecosystem in which multiple actors and systems digitally interact in real time with the ultimate goal of ensuring the safety of all aerial operations.

3.1.4 U-space: architecture view

From an architectural point of view, all previous services could be provided by a single entity in the manner of ATM systems. This would simplify interactions between the various services and ensure greater control by the authorities. However, in keeping with the liberalizing and competitiveness-enhancing spirit of the European institutions, the U-space is built as a federated set of coordinated services that exchange information through interoperable formats to form the complete ecosystem [SES19] . Thus, the services described above can be provided by multiple entities, as long as all of them interoperate to meet the requirements of the

U-space. On the other hand, there are some services that, for security or privacy reasons (e.g., Air Navigation Service Provider (ANSP) interface, registration), must be provided centrally and usually by the authorities. Therefore, the ecosystem is articulated around two main entities:

- **U-Space Service Provider (USSP)** which are independent, usually commercial, systems providing a subset of the previous services directly to operators. To provide these services, USSPs may need weather or terrain information that they can obtain from external actors known as Suplemental Data Service Providers (SDSPs).
- **Common Information Service Provider (CISP)** a unique, centralized entity providing relevant common information (e.g.: airspace areas, manned traffic information) to all involved USSPs. This service is optional, and a member state may choose to provide the information directly to each USSP.

This way, there may be multiple USSPs working in the same airspace volume. As a result, they will have to coordinate and exchange information with each other in order to ensure that traffic does not conflict. At the same time, there may exist a central system (the CISP) that provides common aeronautical information to all of them. Moreover, an extension of the CISP role is being considered in some countries to enable data exchange between USSPs and provide additional functionalities. Thus, three architectural approaches are being considered, as depicted in Figure 3.3 :

- **Decentralized architecture**. USSPs coordinate through direct communication between them, supported by a distributed Discovery and Synchronization Service (DSS, as proposed in [AST19] and [AST22]). A CISP is not considered, and common information is made available to all USSPs.
- **Direct exchange architecture with CISP**. As in the previous case, USSP coordination is made through direct communication facilitated by a DSS. Yet, a CISP considered to provide common information to all USSPs. As discussed in [Fed20], this approach is consistent with the UTM concept developed by the FAA in the United States of America, with the CISP being called Flight Information Management System (FIMS) and USSPs being called UAS Service Suppliers (USS).
- **Indirect exchange architecture with CISP**. USSPs coordination is managed through the CISP that provides indirect information exchange capabilities between USSPs. At the same time, the CISP is still a single and reliable source of all common information. This approach is consistent with the one used in the DOMUS project [SES20] or in the Spanish U-Space development.

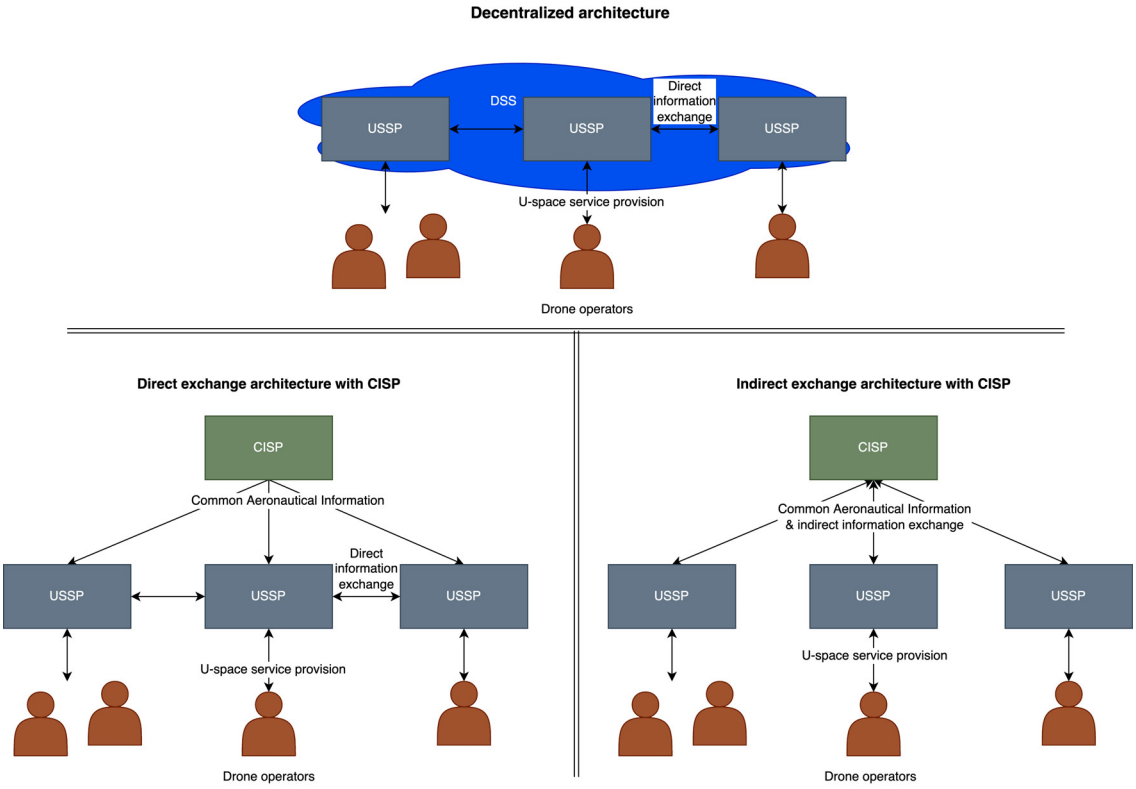


Figure 3.3: U-space architecture alternatives.

3.1.5 U-space surveillance information: collaborative identification

As already introduced, the operation of tactical U-Space services depends on the availability of real-time positioning information which can be obtained from collaborative UAS surveillance protocols. In this sense, the already discussed network identification process relies con public communication networks (e.g.: 4G, 5G) to periodically forward telemetry information to USSPs. In addition, European drone regulations ruled in 2019 that all UAS over 250 g must have the means to provide direct remote identification functionality (unless physically tethered). Particularly, drones must be able to perform *"real time [...] direct periodic broadcast using an open and documented transmission protocol [...] in a way that can be received directly by existing mobile devices within the broadcasting range"* [Eur19]. This broadcast in a neighbouring geographical area consists of a unique (manufacturer-provided) UAS serial number, a UAS operator registration number, the geographical position of the drone with its height above the take-off point, the course and speed of the drone, its emergency status and the geographical position of the operator.

Similarly, a regulation enacting the need of broadcasting remote identification has been recently endorsed in the United States [Fed21b]. In that case, UAS must broadcast (themselves or using a remote identification broadcast module) the following information: UAS identification, latitude/longitude, altitude, and velocity of UAS, latitude/longitude and altitude of Control Station, emergency status and a time mark. Regulation mandates that information broadcasting must be done using a protocol compatible with existing personal wireless devices.

Initially, due to the quick emergence of drones and the lack of purpose-built technologies, traditional aviation surveillance protocols such as ADS-B or FLARM have been adopted (and in the latter case adapted) to fill the gap in collaborative unmanned surveillance. However, some authors argue that this approach may not be suitable as long-term solution due to the limited available bandwidth and the high drones' density expected in the coming decades [Gut+17]. Moreover, regulation from the USA also prohibits the use of ADS-B out as a means of meeting remote identification requirements. Regulators argue that its usage shall be limited (with explicit authorization) when large UASs are operating in controlled airspace.

Therefore, a standardized and purpose-crafted solution for UAS collaborative surveillance is required. To this end, technical standards such as ASTM F3411-19 [AST19] or AD-STAN prEN 4709-002 [ASD21a] have been developed and are under adoption. The following subsections will overview in detail each technology, as discussed by this author and others in [Bes+22b].

3.1.5.1 ADS-B

ADS-B [EUR09a] is the acronymous of Automatic Dependent Surveillance-Broadcast. It is a surveillance technology based on periodic broadcast of on-board navigation (typically GNSS) technologies position estimation. The system is "Automatic" because it does not require any interrogation signals from the ground, and it is "Dependent" because the quality of the position detected depends on the sensors installed on the aircraft. Since the signal is broadcasted, it can also be received by nearby aircrafts to provide situation awareness and allow self-separation, without the intervention of other entities. Apart from the benefits directly derived from its technical features (specifically, high accuracy and integrity), one of the other main benefits is the reduced cost of the system: ground stations are significantly cheaper than primary and secondary radar systems used by air traffic control.

ADS-B relies on two avionics components aboard each aircraft: a high-integrity satellite navigation source (i.e., a certified GNSS receiver) and a datalink (the ADS-B unit). Several categories of equipment (i.e.: A0-A3) are considered with different transmission power and sensitivity requirements. The system may work on two different carrier frequencies: 1090 MHz which is the default frequency and 978 MHz, preferred for flying altitudes below 18.000 feet with the aim of reducing congestion.

The typical information provided by the ADS-B is the identification of the aircraft, the current position (longitude and latitude), altitude and velocity. Messages (called "squitters") are typically broadcasted between 0.5 and 2 seconds. The packets have a size of 56 bits or 112 bits. Short squitters are packets of 56 bits which only contain 8 bits of packet information, 24 bits of aircraft identification and 24 bits of parity. Long squitters (112 bits) have 8 bits of packet information, 24 bits of aircraft identification, 56 bits of data and 24 bits of parity. Since the packets are only 56 or 112 bits, a lot of effort has been devoted to compressing as much information into the packets as possible. No medium access protocol is implemented, packets can interfere, but the very short messages (max 120 μs), the low frequency of broadcast permitted (max 2 Hz) and the large number of receivers reduce the probability of collision and increase the probability of reception. ADS-B is currently susceptible to attacks and security issues, as it does not consider any authentication and encryption features. Data fusion and

other techniques may be implemented to reduce the impact of these issues. A summary of ADS-B protocol parameters is depicted in Table 3.1 .

General features	
Communication type	Broadcast, message based
Transmitter location	UAS
Physical layer	
Transmission frequency (MHz)	1090 (978 preferred below 18.000 feet)
Maximum transmission power (dBm)	A0: 48.5-51.5
	A1/A2: 51-54
	A3: 53-56
Maximum physical bit rate (kbps)	1000
Medium Access Protocol	None. The packets can interfere, alleviated deploying a large number of distributed receivers
Modulation	Pulse Position Modulation (PPM)
Message codification	
Message types	Short squitter (56 bits) for identification, Long squitter (112 bits) for identification + data
Information encoding resolution	high-integrity GNSS
Message frequency	0.5 Hz (up to 2 Hz)
Other features	
Authentication	No
Encryption	No
Error detection	Yes
Error Correction	Yes

Table 3.1: ADS-B protocol parameters

Nowadays, ADS-B is incorporated in most manned aircrafts and the operators are encouraged to install ADS-B equipment. It is a good surveillance method for the flights over areas not covered by traditional radar. In the context of UASs, ADS-B transceivers on small UAS are no longer limited by payload and power capabilities (e.g., the system described in [uAv20] has a size of 50x25x17 mm and a weight of 20 g), and they represent promising opportunities for the regulated operation of small UAS, at a reduced cost. However, the use of ADS-B on UAS is still under study. As already introduced, the lack of a medium access protocol could represent a limitation for UAS, where a huge number of aircrafts can potentially fly in a relatively reduced space. In [Gut+17] and [DM17], authors analyze this scenario through several simulations, coming to the conclusion that it will be probably necessary to balance the demand of air space occupation to provide a good safety level to all the aircrafts. In [SSC13], a similar scenario is studied, also considering the current regulatory practices of the U.S. National Airspace System, authors finally report that ADS-B should be used along with some other system, especially optical sensors for collision avoidance.

3.1.5.2 FLARM

Traditionally, Flight Alarm (FLARM) was a traffic awareness and collision avoidance technology designed for the needs of light aviation (i.e.: gliders, helicopters...), similar to ADS-B. Based on this technology, an electronic identification system was released in 2018, known

as “FLARM UAS Electronic ID” [FLA18]. This new protocol broadcasts a secure electronic identification and 3-D position of a given UAS via a radio frequency digital message. Similarly, the ADS-L (ADS-Light) protocol has been proposed in 2022 as a FLARM-compatible solution to provide air to ground electronic conspicuity [EAS22b].

A FLARM system consists of a GNSS receiver for localization and timing, a processor to manage the protocol and a RF physical transmitter. The transmitter broadcasts the protocol messages over the unlicensed 868.4MHz band using a digital modulation scheme. Carrier-Sense Multiple Access is used for medium access listening for other transmissions on the channel and deferring the transmission a random time if it has been used. Therefore, packet collisions and transmission delays might occur.

Two different messages are to be transmitted over this physical layer: UAS identification messages and operator identification messages; indicating the identity and position (longitude, latitude and MSL/WGS84 altitude) of the UAS and its operator respectively at a given time. By default, messages are transmitted once every 3 seconds. The protocol uses the manufacturer serial number as the drone identification.

Messages consist of a preamble, sync word, a partially encrypted payload with the message information, a CRC code for integrity and an optional message signature for UAS authentication. Using this last field, the protocol describes a public-private key mechanism to ensure transmitter (UAS) identity using a registration service for verification. Table 3.2 , on the next page, summarizes the main technical features of the protocol.

Multiple commercial products exist implementing FLARM electronic identification protocol (often along with other technologies described also in this Section 3.1.5). For instance, the Atom UAV [FLA21] dongle from FLARM includes a FLARM transceiver, ADS-B receiver and ASTM F3411-19 emitter. Droniq HoD [Dro20] also implements a FLARM transceiver along with ADS-B reception and telemetry transmission over LTE. Commercial ground receivers (sensors) also exist such as GBSAS stationary from Droniq [Dro21] with up to 20km range for FLARM and 100km range for ADS-B .

General features		
Communication type	Broadcast, message based	
Transmitter location	UAS	
Physical layer		
Transmission frequency (MHz)	868.4	
Maximum transmission power (dBm)	14	
Maximum physical bit rate (kbps)	50	
Medium Access Protocol	CSMA without collision detection (random initial wait of 0-1000 ms, random wait after transmission detection of 15-150 ms)	
Modulation	2-Gaussian Frequency Shift Keying (GFSK)	
Message codification		
Message types	UAV eID message, operator eID message	
Message format and size (in bytes)	Preamble (4B) + Sync (3B) + Payload (24B) + CRC (2B) + Optional signature	
Information encoding resolution	Timestamp	2 s
	Latitude	0.000025º
	Longitude	0.000025º
	Altitude	1 m (MSL or WGS84)
Message frequency	0.33 Hz (or 1 Hz if position deviation is greater than 30m)	
Other features		
Authentication	Optional with external unspecified registration service	
Encryption	Symmetric encryption (same key for all devices)	
Error detection	Yes	
Error correction	No	

Table 3.2: FLARM protocol parameters

3.1.5.3 Remote identification

Existing regulations prompted the need to develop open and standardized broadcast transmission protocols that can be received using mobile devices. This limits the possible physical layers to mainly Wi-Fi and Bluetooth (although protocols such as LoRa have also been explored [Muj+21]). Currently, ASTM F3411-19 [AST19] remote ID standard is already established mainly inspired on US regulations. In parallel, ASD-STAN developed the prEN 4709-002 standard inspired on European regulations but seeking compatibility with ASTM standard [ASD21b]. On the other hand, IETF drip group [Car+21a] is also developing a remote id protocol leveraging existing internet protocols.

Focusing on the ASTM standard, broadcast of identification messages is performed using existing broadcast frames (usually used as advertisements to establish connections) within the Bluetooth (legacy or 5.x LE) and Wi-Fi protocols. The idea is to use the payload of these frames to transmit a short ad-hoc id message thus eliminating the need to extend physical protocols. Two main compulsory message types are considered: a static one broadcasted each 3 seconds with drone identity, and a dynamic one broadcasted at least each second with UAS location. Additionally, the protocol also provides for a set of optional messages that can be used for operator identification, operator location transmission, transmitter authentication. Table 3.3 summarizes the main technical specifications of the ASTM standard.

General features			
Communication type	Broadcast, message based		
Transmitter location	UAS		
Physical layer			
Physical protocol	Legacy BT	BT 5.x long range	WiFi
Physical frame/message supporting remote id payload	Beacon broadcast message	Extended advertisements (with FEC for x4 range)	Service discovery frame based on Neighbor Awareness Networking
Physical frame overhead [B]	23	35	21
Transmission frequency (MHz)	2400 (channels 37,38,39)	2400 (all BT channels)	2400 (ch. 6) 5800 (ch. 149)
Maximum transmission power (dBm)	+5	+5	+11 (2400 MHz) +4 (5800 MHz)
Medium Access Protocol	No medium access protocol.		Non-persistent CSMA
Modulation	Gaussian Frequency Shift Keying (GFSK)		Orthogonal Frequency-Division Multiplexing (OFDM) with QAM symbol mapping.
Message codification			
Message types (only compulsory)	Basic ID message (UAS identification) Location message (UAS position)		
Message format and size (in bytes)	Header (1B) + Message data (24B) [payload of aforementioned frames]		
Information encoding resolution	Timestamp	0.1 s	
	Latitude	10e-7º	
	Longitude	10e-7º	
	Altitude	0.5 m	
	Speed	0.25 m/s if speed<0.5 knts. 0.75m/s otherwise	
Message frequency	Basic ID message: at least 0.33 Hz Location message: at least 1 Hz		
Other features			
Authentication	Optional with external unspecified registration service		
Encryption	No		
Error detection	Physical layer dependent (Yes)		
Error correction	No	Yes (convolutional FEC)	Yes

Table 3.3: Remote ID protocol parameters
MAC information for BT and WiFi has been extracted from [Gha+18] and [Cam+15] respectively

It is expected that new drones directly implement its own means to fulfill remote id requirements (e.g.: DJI statement in [Sch21]). For instance, some DJI [Str20] and Parrot [Par21] drones already support some remote identification capabilities via software update and repurposing the C2 channel. Meanwhile, dongles or broadcast modules are to be used for this purpose. Some commercial examples include Aerobits idME+ [Aer23] or ScaleFlyt [Tha21] from Thales, both compatible with ASTM standard. Sensors/receivers for Remote Id seem to be still under development. For instance, a proof-of-concept smartphone-based receiver is proposed in [ope25]. Some companies commercial announcements publicizes remote id receivers (e.g.: [Rad21], [Ded21]) but with no technical details that allow to assess their technological maturity.

3.1.6 U-space surveillance information: non-collaborative sensors

Although collaborative surveillance technologies discussed in previous subsection offer enhanced accuracy, greater detection performance, and aircraft identification capabilities; non-collaborative sensors are critical for security and contingency purposes. In fact, the protection of critical infrastructures from hostile drones must always rely on non-collaborative sensors since it does not seem reasonable to expect rogue drones to share their position. Similarly, collision avoidance through tactical deconfliction services also benefits from the existence of additional sources of information. This set of sensors work by detecting physical perturbances caused by drone presence and have been widely studied in review articles such as [WLS21], [Bes+22a], [Kha+22] and [Sei+24]. Mainly, five types of sensors are described in literature and are commercially available: radar based sensors, acoustic sensors, RF sensors, optic sensors, and integrated sensors.

The gold-standard for detecting and tracking manned aircraft is the use of **radar techniques** as they offer good performance independent of atmospheric conditions and day and night operation. However, a direct transfer of manned aviation radar techniques is not feasible due to drone's small size (with radar cross sections below 0.01 m^2 [SN21]) and their low speed. In this sense, authors of [WLS21] cite two approaches to adapt active radar techniques: utilize higher frequency bands (above 6GHz, K-band) and the usage of spacial diversity antennas (i.e.: MIMO techniques) to achieve leaner beams. With these adaptations, authors in [Yan+21] demonstrate a MIMO system capable of detecting small drones up to a range of 5 km. Also, in [ZB20], authors propose a FMCW radar at 24 GHz capable of detecting pico drones. Another viable technique is the utilization of passive radars that take advantage of signals of opportunity (e.g.: cellular signals, DVB signals) to illuminate the target [Tan+25]. For instance, authors of [Dan+19] propose the usage of LTE downlink frequency to successfully track drones. Overall, there are numerous commercial examples of the use of radar technologies for drone detection, such as the ART Midrange 3D radar [ART20] that achieves an omnidirectional coverage of up to 5000 m with a range determination error of less than one meter. Another example is the Ranger R8SS-3D radar [FLI22] that operates in X band, reaching an instrumental range of up to 8000 m and frequent positioning updates (up to 4 Hz)

The usage of **acoustic sensors** has also been reported in the literature as a possible, yet limited (short range, affected by weather), way to detect and track UAS [WLS21]. In this sense, an analysis of the audio spectrum is usually performed to detect an airborne UAS from audio captured in a wide area [Kim+17]. The application of ML algorithms has also been explored to generate UAS acoustic fingerprints that allow to identify specific UAS models [RPH19]. By integrating several audio sources through an array of spatially distributed microphones it is also possible to track UAS trajectories over time analyzing the time delay of arrival (TDOA) differences to each sensor within the array [Cha+18].

In addition to detecting drones from their noise emissions, it is also possible to analyze their electromagnetic emissions through **RF sensors**. Indeed, controlling drones remotely usually requires setting up a point-to-point communications link: the C2 channel. In this sense, a possible approach is to use Software Defined Radios (SDRs) to passively analyze RF spectrum. For example, in [Moh+23] the spectrogram of an RF signal is analyzed using YAMNet: a neural network oriented to audio waveform analysis; to perform fingerprinting and classify

the type of drone with 96% accuracy. Another approach consists in sniffing packets from a known communication protocol (e.g.: WiFi). Authors in [Bis+18] demonstrate that it is possible to perform a statistical analysis of the periodicity and duration of the Wifi-exchanged packets reaching an accuracy of 70% in drone detection and classification. Extending this RF channel sniffing approach, the authors of [Aou+23] propose a system that is able to decode telemetry messages sent over a WiFi channel to extract the position of drones. Finally, RF receivers with spatial diversity can also be used to localize drone's position thanks to TDOA measurements. This approach is discussed in [MS18] achieving localization errors of about 50-75 meters. As for commercially available solutions, the AARTOS X7 system [AAR21] combines drone protocol decoding and a direction finding antenna array to achieve 4-6° of azimuth accuracy and 3 m in rage accuracy with detection ranges of at least 2 km. Also, the Aeroscope system uses a C2 channel sniffing and decoding approach to track drones manufactured by DJI with a range of up to 50 km [DJI17].

Finally, **optical sensors**, either in the visual or infrared spectrum, allow to detect drones thanks to image segmentation and classification algorithms [Els+21]. While traditional image segmentation algorithms can be used as in [SSH21], the current trend is the use of computer vision techniques based on neural networks. For instance, authors in [Par+17] make a comparison of different convolutional neural network architectures to perform drone detection, with the YOLO model performing better than its counterparts. Some of the disadvantages of this approach are the need of having a direct line of sight with the drone (which limits coverage) and dependence on night/day. As with previous sensors, multiple cameras can be jointly used to improve coverage and accuracy. In this sense, [Sri+21] proposes a multi-camera system that is able to reliably track multiple drones from a range of at least 100 m. For night conditions, thermal cameras can be used as an alternative to visible spectrum cameras, as employed in the comercial system described in [HGH20].

The previous paragraphs have individually presented the fundamentals of drone detection systems using radar, acoustic, RF and visual sensors. Now, Table 3.4 presents a comparison analyzing their sensitivity to environmental conditions changes, their detection range, cost, and provided information. It can be concluded that an **integration of different sensor types** can help overcome their individual shortcomings and enhance detection and tracking results. For instance, in [Hen+23] acoustic sensors are used to provide early, wide-area UAS detection capabilities so that visible sensors can be used in a second phase for refined tracking and identification. Similarly, in [Abd+23] a radar sensor is enhanced with cameras to reduce false alarms (e.g.: avoid detecting birds as targets). Another alternative is found in [Jov+21] where a multihypohtesis tracking system fuses information from three passive sensors: a RF sensor to capture the uplink and downlink communications of the UAV, an acoustic sensor searching for the rotor noise and a passive radar system using the cellular network. Although the development of non-collaborative drone sensors is still in progress (as it will be identified in Section 3.1.7), the integration of multiple sensors is the most used approach in commercial systems. In this sense, Dedrone [Ded23] commercializes a fusion system that integrates RF sensors, radars and cameras. A similar sensor suite is available in the DroneSentry solution [Dro19b] which allows tracking drones up to a 5 km range.

Sensor Type	Operational Conditions	Range	Cost	Measurements
Active radar	• Affected by hostile interferences	• Long range (> 5 km)	• High economic cost. • High power cost.	• Range • Azimuth • Elevation (in some cases)
RF	• Affected by RF interferences.	• Medium range (1-2 km)	• Low economic cost.	• Azimuth • Identification information
Acoustic	• Affected by weather conditions. • Affected by noise interference.	• Short range (< 500 m)	• Low economic cost.	• Azimuth. • Identification information.
Camera	• Affected by weather conditions. • Affected by illumination conditions.	• Medium range (1-2 km)	• Low economic cost. • High computational cost.	• Azimuth • Elevation • Identification information.

Table 3.4: Comparison of Non Collaborative drone detection technologies

3.1.7 U-space: deployment status and challenges

A thriving drone market that provides services to smart territories depends on the deployment of the U-Space ecosystem as only then it will be possible to safely extend the use of drones and reduce flight limitations. Therefore, building on the of U-Space conceptual framework described in the previous sections, the European Commission together with European union Aviation Safety Agency (EASA) have developed an enabling regulatory framework to enforce U-Space deployment. Thus, Commission Implementing Regulation 2021/664 [Eur21] together with a set of associated Acceptable Means of Compliance/Guidance (AMC/G) [EAS22a] lay out the rules for the initial U-Space deployments by member states. In this sense, this initial regulatory package only mandates a subset of U1-U2 basic services (i.e.: network identification, geo-awareness, flight authorization and traffic information services) with two optional ones (i.e.: conformance monitoring and weather service), in order to support low density, low risk traffic.

Information from 2022 in [EUR22] shows a slow and uneven deployment of U-Space services by member countries. For example, only 10% of countries had deployed the network identification service in 2021, with just 41% of countries expecting to have it implemented by 2025. Particularly, in the Spanish case, the deployment of the CISP system enabling the national U-Space ecosystem was scheduled for 2023. However, as of today, this system is not yet operational [ENA21]. In fact, the latest version of the European ATM roadmap [SES25] delays the certification of the first USSP to 2025, and the full deployment of U1-U2 services (initially planned for 2022) to 2035.

In view of the above, it seems necessary to analyze the **causes of this delay**, and even to identify what lines of research are needed to promote the development of U-Space systems. In this sense, [But24] argues that the technological capacity for U1-U2 services is already available. Indeed, SESAR projects have already demonstrated the required technological capabilities in 2022, as discussed in [SES22]. On the contrary, the former article discusses that some blockers for the U-Space roll-out are inconsistent implementation of European regulation

across member states, and the complexity of the certification process of USSPs/CISPs. Also, it argues that there is a lack of real demand due to regulatory requirements not meeting operator's economical needs.

Another blocker is cited in [Eur22a] pointing to tests and demonstrations complexity. This is further aggravated by a lack of demonstration facilities and the regulation complexity of tests operational approval. In this sense, SESAR projects such as U-elcome and BURDI intend to foster the uptake of U1 and U2 U-space services through demonstration of early U1 and U2 U-space of several locations. Another possible approach to streamline the validation of U-space services would be the utilization of advanced simulation tools, as already discussed by this author in [Car+21b].

Regarding **future research** to enable advanced U-Space services, Table 3.5 depicts current SESAR projects and identifies ongoing research topics. Besides the effort to foster real deployments and demonstrators, current research topics are:

- developing dynamic capacity management techniques to allow high-density operations,
- advancing in the integration of U-Space and traditional ATM systems,
- measuring and evaluation UAS usage social concerns together with ensuring an equitable airspace access,
- extending current U-Space concept of operations to cover Innovative Air Mobility (IAM) solutions,
- integration of AI tools

Research topic	Project	Project description
Separation and Dynamic capacity management.	SAFIR-Ready	Dynamic capacity management, detect and avoid algorithms, m2m communication. Use case applied to transport and infrastructure inspections.
	SPATIO	Strategical and tactical conflict resolution considering dynamic capacity management and fairness issues.
Social acceptance and fairness	MUSE	Social and environment impact of UAS operations to citizens (noise and visual pollution).
	ImAFUSA	Also covers social acceptance of UAS operations in terms of environmental impact, safety and socioeconomic impact.
Integration with ATM	ENSURE	Definition of a common interface and services for the integration of U-Space and ATM.
Innovative Air Mobility	OperA	Innovative Air Mobility (piloted air taxi and unmanned cargo) in real life air traffic conditions.
	EUREKA	Vertiport management (arrival and departure processes, network flow...) for Urban Air Mobility
	EALU-AER	Integration and demonstration project in Ireland to check the feasibility of eVTOL operations.
Real deployments and demonstrators	BURDI	Practical implementation of U-Space concept in Belgium and the Netherlands.
	U-Elcome	Industrialization, implementation and roll out of U1 and U2 to demonstrate TRL8 maturity.
Artificial intelligence usage	AI4HyDrop	Incorporation of AI-based tools into U-space to help managing the scale and complexity of future drone operations.

Table 3.5: Ongoing SESAR research topics.

Another existing challenge is the reliability of the U-Space ecosystem. In order to ensure the safety of air and ground operations, it is necessary for both UAS and U-Space services to have as few failures as possible. Only this way it would be possible to maintain high social acceptance. As for UAS systems, authors in [Vil22] analyze the reliability of military UAVs. A statistical analysis shows that the Mean Time Between Failures (MTBF) of unmanned aircraft is at least an order of magnitude worse than in manned aviation. Similarly, for commercial drones, a MTBF of 1 failure per 100-150 hours is cited in [SOR15]. Even if these tentative figures are not robust enough (partly due to the emerging state of the U-Space ecosystem and lack of standardized accident reporting), the failure rate is expected to be higher than for commercial aviation. Thus, to analyze the implications of these failures, methodologies such as SORA [JAR24] exist to help implement mitigation measures with the aim of ensuring that UAS operations do not pose a greater risk than manned operations. Some of those mitigation barriers can be implemented through the usage of U-Space services. In this case, and to this author's knowledge, there does not exist an analysis on the effectiveness of these barriers or their reliability. However, the use of web technologies in the design of the U-Space system or the use of public communications networks suggests that its performance may be lower than that of its manned counterpart. For all these reasons, reliability analysis is an open research gap. Finally, and also related with drone's social acceptance, the need of developing counter-drone technologies to protect citizens and critical infrastructure from malicious and non-collaborative drone usage has been identified in [Eur22a].

In line with some of these needs, the next sections will discuss the thesis contributions related to the development of U-Space systems in two critical areas:

- proposing a comprehensive simulation system to streamline the validation of U-space services (Section 3.2),
- incorporating fairness and congestion management criteria in the strategic authorization phase through a token-based authorization framework (Section 3.3).

3.2 Modeling and simulating the U-space ecosystem

As pointed out in the previous section, a common way to evaluate the overall behavior of UTM systems is to perform live experimentation in real, complex scenarios. However, it is not possible to perform real tests at the scale that UTM systems are expected to operate, with hundreds or thousands of aircraft flying simultaneously. This would entail a very high economic cost. Also, tests would be very complex as they are subject to meteorological variability, to the availability of equipment and personnel, and to the human factor. Furthermore, actual tests involving validation of safety in critical and emergency situations can lead to serious risk for equipment and people. For example, it is needed to evaluate the collision detection and avoidance algorithms between drones and manned aircraft, and this could require putting the aircraft at serious collision risk, which would not be acceptable.

As an alternative, the utilization of advanced simulation tools to streamline U-Space validation services has been discussed by this author in [Car+21c] and others in [Oos+21]. In this sense, simulation techniques enable performing reproducible, scalable, and exhaustive testing while reducing costs and evaluation time. As conceptually depicted in Figure 3.4 , simulation frameworks can model the whole U-Space ecosystem including entities such as operators, pilots, drones, GCS, surveillance and communication networks, ANSPs; as they interact in a given environment (e.g.: urban topology, weather).

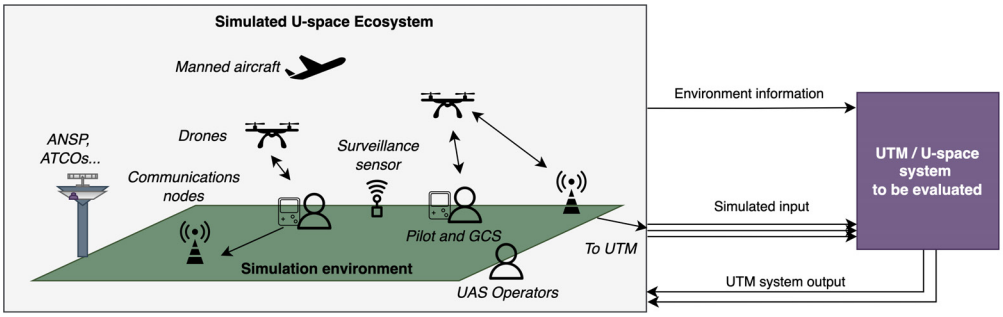


Figure 3.4: Entities considered in U-Space simulation platforms

The objective of these simulation frameworks is to simulate the set of input signals required to excite an UTM system while also responding (in a closed loop) to the outputs it provides. This may include simulating u-plan authorization requests, telemetry messages of in-flight operations or incoming tracks from surveillance systems, etc. It may also require responding to both strategic and tactical deconfliction results, traffic information or conformance alerts.

In this sense, Section 3.2.1 explores existing simulation solutions for modeling and simulating the U-Space ecosystem. As a result of this analysis, a knowledge gap has been detected regarding the simulation of human behavior both in the strategic (i.e.: operator) and tactical (i.e.: pilot) phases. Also, existing systems usually disregard the simulation of surveillance networks (both collaborative and non-collaborative) as information sources to UTM systems. To this end, Section 3.2.2 will present USPASIM: a comprehensive simulation platform covering all relevant entities in the U-Space system. Then, in the context of this simulator, Section 3.2.3 to Section 3.2.7 propose novel simulation models to cover both literature gaps. Particularly, the proposals on this section are based on the contributions of this author to

the following publications, as identified below. Finally, Section 3.2.8 describes a use-case execution that demonstrates the validity and usefulness of the proposed models.

Supporting publications

- **[Rap+23a]** - Daniel Raposo, David Carramiñana, Juan Besada, and Ana Bernardos. "A Realistic UAS Traffic Generation Tool to Evaluate and Optimize U-Space Airspace Capacity". In: *18th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2023)*. Cham: Springer Nature Switzerland, 2023, pp. 24–33. DOI: 10.1007/978-3-031-42536-3_3 where this author's contributions have been at the specification and design of the different traffic generation models, and their integration within the proposed simulation framework.
- **[Bes+22b]** - Juan A. Besada, David Carramiñana, Luca Bergesio, Ivan Campaña, and Ana M. Bernardos. "Modelling and Simulation of Collaborative Surveillance for Unmanned Traffic Management". In: *Sensors* 22.4 (Jan. 2022), p. 1498. DOI: 10.3390/s22041498 where the author has worked in designing, implementing, integrating and testing the proposed models for ADS-B, FLARM and Remote Id.
- **[Bes+22a]** - Juan A. Besada, Ivan Campaña, David Carramiñana, Luca Bergesio, and Gonzalo de Miguel. "Review and Simulation of Counter-UAS Sensors for Unmanned Traffic Management". In: *Sensors* 22.1 (Jan. 2022), p. 189. DOI: 10.3390/s22010189 where the author has peformed development, integration and validation activities for the proposed non collaborative sensor models.

3.2.1 Review on simulation systems for the U-space ecosystem

As depicted in Table 3.6 at the end of the subsection, there are numerous simulation proposals in the literature oriented to modeling some of the previously defined entities within the U-Space ecosystem. Starting with **drones modeling**, there exist a wide variety of systems that are usually focused on modeling aspects such as drone navigation and flight dynamics. For instance, an analysis of the existing commercial tools can be found in in [Hen+18] or [MBJ19] which discuss tools such as X-plane, Flight Gear, JMavSim or Microsoft AirSim. These applications allow to realistically simulate and control the trajectory of a single drone as it flies in a synthetic three-dimensional environment that can be visualized by the user. Multi-drone simulation frameworks are also available as in [RMC15], which presents a web architecture designed to simulate mission planning, execution and monitoring of multiple UAVs within a virtual environment that can contain no-flight zones.

Besides contemplating a basic flight trajectory simulation engine, some literature considers the effect of external factors on drones' flight. For instance, in [Xue+18] a high fidelity aircraft dynamics model is used to consider the effect of wind, energy consumption and possible GNSS errors. Drone's flight path may also be affected by autonomous processes/safeguards within the drone. In this sense, the previous article also considers simulating onboard sensor and autonomous collision avoidance algorithms within a Detect and Avoid system. In other cases, decision-making is made manually by the pilot. To model this, authors of [Fro+21]

propose a comprehensive simulation platform that considers contingency responses for UASs in emergency situations (e.g.: loss of engine, loss of power, loss of link, loss of GNSS).

Also affecting drones' behavior is communications integrity in the C2 channel between the drone itself and its associated **Ground Control Station**. For instance, as a drone moves away from a communications node, connectivity might be interrupted prompting the drone to halt its operation and return to a known location. In this sense, tools such as QgroundControl [Dro19a] or MAVProxy are devoted to modeling the mission planning process and the behavior of a GCS. Particularly, this last tool allows simulating the C2 link and message exchange between the GCS and the drone using UDP and MAVLink protocols. GCS simulation is also considered in [Al-+19] where a wide set of commands (e.g.: take-off, go to waypoint, hover...) can be sent to drones over a simulated C2 channel. This proposal uses the Unity platform and considers elements such as models aircraft specification, navigation, control and sense and avoid systems. C2 link coverage is also modelled in [Zha+19] where a scenario that geographically positions communication nodes (to be used by drones) can be defined.

It is also worth noting that some systems allow interfacing a real GCS with a simulated environment. This approach is of particular interest for pilot training missions or for testing conflict avoidance algorithms. In this sense, DJI Assistant 2 app [DJI19] allows using a physical DJI drone controller to pilot a single virtual drone within 3D simulation environment. This capability is also available in [Fro+21] which even allows the execution of hybrid simulations with a complex virtual environment and real drones. Human-in-the-loop simulation is also considered in [JD22] where a human operator can manually modify the default drone behavior (i.e.: its trajectory) to avoid conflicts.

Regarding **human intervention** in the mission definition stage, the scenario definition process is not usually automated and requires a human operator to define the missions to be simulated. Only in the case of [Oos+21] automatic traffic generation utilities are presented for a urban delivery use-case. Similarly, systems modelling operator decisions within the U-space strategic phase (e.g.: flight replanning process) are not found in the literature.

On **communication networks**, besides C2 communications which have been already covered, Vehicle to Vehicle (V2V) wireless links as well as wired communications between different U-Space systems should be considered. Indeed, the operation of UTM systems heavily depends on having ubiquitous, reliable and low latency connectivity between all actors. To this end, in [Mar+17] authors propose a tool focused on simulating UAV flight and control together with mobile communications between UAVs, and in the C2 channels. The platform federates two existing simulators to do so: x-Plane Flight simulator to incorporate UAV dynamics and OMNet++ network simulator to model communications. The simulator enables measuring network metrics such as RSSIs, throughput, package losses... Also, in this author's previous work [Car+21c], communications are simulated considering delays and throughput limitations.

As previously discussed, the U-Space system can rely on **surveillance networks** in order to provide real-time monitoring capabilities. To this end, both collaborative (e.g.: ADS-B) and non-collaborative technologies (primary radars) may be used. As depicted in the comparative table, modeling of such networks is often overlooked. In this sense, in [Fro+21] a network of ADS-B and primary/secondary radars can be simulated together with the subsequent data

fusion process. Also, in [Car+21c] this author has previously proposed a simple and general model to create simulated tracks from drone trajectories by adding noise and an ideal circular sensors' coverage.

To support the simulation of all these entities, it is necessary to simulate the **environment** in which they operate. In this sense, it is common to have a terrain model or the topography of an urban environment to provide realistic flight results. This is the case of [Wen+24] which relies on a city roads network to build aerial corridors to organize drone traffic. Moreover, the definition of different geofences or no-fly areas is considered together with weather models.

So far, contributions that allow to simulate different actors of the U-Space ecosystem have been discussed but their integration with the U-Space or ATM systems has not been considered yet. By making this distinction, the number of available tools is reduced. In the case of ATM systems, Bluesky [HE16] is a popular open-source, high performance air traffic simulator envisioned to test traditional ATM systems. Also, ESCAPE simulator [EUR09b] by EUROCONTROL may be used to test airspace designs or ATCOs training. Another alternative is TAAM which is a fast-time, end-to-end comercial simulation tool [Jep25].

Focusing on simulation models **integrated with U-space services**, a set of tools was proposed in [Hom+16] as a result of NASA's developments to investigate UTM systems. It allows defining drone trajectories as a list of waypoints and simulating drone movements in real time, in order to generate position report messages (telemetry) that are injected into a UTM systems. Also, the simulator allows manual control to introduce contingencies in the trajectories so that alert generation may be evaluated. In [Mil+19], authors propose a simulation framework built using the Gazebo and ROS frameworks so that the interactions of GCS and drones with the U-Space ecosystem can be tested both in strategic and tactical (e.g.: emergency management) phases. Similarly, in [Fre+22] authors take advantage of the Bluesky simulation platform to integrate it into a U-Space ecosystem. To do so, the simulation platform mimics the whole lifecycle of a u-plan from its strategic phase (i.e.: SCD, Demand and Capacity Balancing (DCB) also considering terrain information) implemented through a purpose-built block; to its tactical phase, simulated making use of Bluesky. Autonomous algorithms to update u-plans so that drones respond to contingencies (landing in close alternative points) and avoiding tactical conflicts (by changing the trajectory) are considered. Also, U-Space-ATM integration can be tested by injecting live ADS-B data.

Finishing with the two most comprehensive platforms, Airbus commercial simulation tool U-Sim [Air22] has been used by NASA and Eurocontrol to test operational concepts. The tool allows simulating vehicles and operators, including flight planning and impacts of navigation error and other error sources on vehicle performance. Moreover, it includes a reference implementation of UTM services. Unfortunately, due to its commercial nature, no additional information is available supporting these claims. Finally, authors of [Oos+21] propose SKALE: a modular UTM simulator that mimics an environment in which a large number of virtual drones fly. Particularly, the simulator allows modeling the environment: topography, no-fly areas and weather conditions. Over this scenario, drones controlled by virtual GCSs are modeled to simulate UAS flights considering both strategic and tactical interactions with a UTM systems. ATM traffic can also be included in the simulation platform. Data exchange between different agents is enabled thanks to network simulator that models both wired and

wireless networks causing delays.

	Literature proposals (author and year)														
	Rodriguez-Fernandez et al.2015	Homola et al. 2016	Hoekstra et al. 2016	Marconato et al. 2018	Xue et al. 2018	Mousa et al. 2019	Zhao et al. 2019	Millan-Romera et al. 2019	Carramiñana et al. 2021	Frontera et al. 2021	Oosedo et al. 2021	Airbus 2022	Fredmon et al. 2022	Wen et al. 2024	**Aggregated**
Aircraft simulation															
↦ Basic trajectory simulation	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100,00%
↦ Effect on trajectory of GNSS errors					✓				✓			✓			21,43%
↦ Effect on trajectory of weather					✓				✓		✓				21,43%
↦ Pilot decissions simulation						✓				✓					14,29%
↦ Onboard sensors simulation (e.g.: DAA)					✓										7,14%
↦ Integration with real drones										✓		✓			14,29%
GCS simulation															
↦ Basic GCS simulation		✓		✓		✓	✓	✓	✓	✓	✓		✓		64,29%
↦ Simulation of C2 channel				✓		✓	✓		✓						28,57%
↦ Integration with real GCS										✓					7,14%
Operator simulation															
↦ Basic operator behavior											✓	✓			14,29%
↦ Automatic traffic generation											✓				7,14%
↦ Operator decisions simulation															0,00%
Network simulation															
↦ Mobile communication networks				✓	✓	✓	✓		✓		✓				42,86%
↦ Fixed communication networks									✓		✓				14,29%
Surveillance simulation															
↦ Collaborative surveillance simulation									✓	✓					14,29%
↦Non-Collaborative surveillance simulation									✓	✓					14,29%
Environment simulation															
↦ Ground/topography simulation		✓			✓			✓	✓		✓		✓	✓	50,00%
↦ Weather simulation					✓	✓			✓		✓				28,57%
↦ Airpsace definition simulation	✓		✓		✓			✓	✓		✓	✓	✓	✓	64,29%
Integration with U-Space/UTM system															
↦ In strategic phase		✓						✓			✓	✓	✓		35,71%
↦ In tactical phase		✓						✓			✓	✓	✓		35,71%
Integration with ATM systems															
↦ Injection of ATM traffic			✓							✓			✓		21,43%
Non functional capabilities															
↦ Real time simulation			✓			✓		✓	✓	✓	✓	✓			50,00%
↦ Fast time simulation			✓		✓					✓		✓			28,57%
↦ Federated simulation				✓										✓	14,29%
↦ Open source availability			✓												7,14%
Coverage	8,00%	20,00%	24,00%	20,00%	36,00%	28,00%	16,00%	28,00%	52,00%	40,00%	52,00%	36,00%	28,00%	16,00%	

Table 3.6: Comparison of U-Space simulation literature proposals

Overall, the previous analysis (summarized in Table 3.6) shows that there is a wide variety of tools that can be used to test U-Space systems. However, most of the tools focus on the simulation of the drone's flight or on the environment in which the simulation takes place. Therefore, they **leave two gaps in the existing knowledge**:

- Modeling the operator's actions which includes planning u-plans with operational significance, reacting to the strategic phase responses of the U-Space system and being able to replan its operations. This lack of human behavior models also extends to the pilot's actions on the tactical phase (e.g.: flight control error, contingency responses). This gap will be covered in Section 3.2.3 and Section 3.2.4 .
- Improved modeling of surveillance sensor networks that provide additional information to U-Space tracking services. Models for both collaborative and non-collaborative sensors will be discussed in Section 3.2.5 to Section 3.2.7 .

3.2.2 USPASIM: U-SPAce ecosystem SIMulation framework

Although the contributions on this section focus on modeling the behavior of operators, pilots and surveillance networks (aligned with the previously identified knowledge gaps), interactions between these entities are highly coupled and even dependent on other entities. Indeed, drone detections by surveillance sensors depend on drones' trajectories over time which, in turn, depend on decisions taken by operators and pilots. Therefore, to analyze the impact of these proposed models at the U-Space level, it is necessary to have a comprehensive simulation system. Such a system must contemplate all the actors involved in the U-Space ecosystem and considers their interactions in both the strategic and tactical phases.

In this sense, the proposed models are integrated in **U-SPAce ecosystem SIMulation framework (USPASIM)**, which represents an evolution of the one proposed by the author in [Car+21c]. This simulator follows an agent-based approach. Under this paradigm, computer architectures or models are articulated around the concept of agents, which are computer systems that can take autonomous actions in a given environment in order to meet a set of design objectives. In this sense, as discussed in [Woo09], agents are usually characterized by having:

- **autonomy**, agents do not require invocation or intervention of humans or other agents to act.
- **reactivity**, agents perceive the environment they are placed on (e.g.: the physical world, a virtual world, a collection of agents...) and act accordingly.
- **proactiveness**, agents can also take the initiative implementing goal-directed behavior based on their objectives.
- **social ability**, agents can interact with other agents (usually through the exchange of messages) which may cooperate by coordinating their actions to meet both individual and organization-level goals. In such cases, usually referred as Multi-Agent Systems (MASs), the inherent bottom-up approach is of special interest as inter-agent coordination can lead to the appearance of emergent behaviors (e.g.: self-organization in swarms of agents).

Moreover, agents can also exhibit intelligent behavior by incorporating that incoporating learning (e.g.: through AI models) or reasoning and decision-making processes (such as the belif-desires-intentions model) [Băd+11].

From a practical point of view, an agent-based approach may be used to implement general-purpose complex distributed systems (as discussed in [Jen01]). To this end, the literature review in [Wro+23] showcases the existence of several multi-agent development platforms such as JADE (based on JAVA language) or SPADE (for the Python ecosystem). These frameworks provide an environment to manage agents' lifecycle (i.e.: creation, activation...). They also offer an abstract agent to easily define agents' behavior and interactions over interoperable and distributed message exchange mechanisms.

In addition, the agent-based approach can be used to model and simulate socio-physical systems. In this case, it is usually referred as **Agent Based Model (ABM)**. This technique

relies on identifying a set of relevant entities within the modelled system. Then, each entity is abstracted as an agent for which an state, set of behaviors and relationships (both with regard to other agents and the environment) are defined. As a result, the overall system-level behavior is obtained by aggregation of those autonomous agents' interactions. As discussed in [Wro+23], frameworks for ABMs are also commercially available, including NetLogo (oriented to the simulation of natural and social systems), Repast suite (with a focus on large-scale distributed simulations) or Mesa (extendable, low-overhead Python library).

Coming back to the simulation of the U-Space ecosystem, ABM is a suitable technique as each U-Space entity (e.g.: pilots, drones...) can be easily mapped into an agent with clear responsibilities and expected interactions. Moreover, the ability to analyze emergent behavior (provided that interdependent interactions between agents are modeled) is critical to find safety and stability flaws on complex systems, such as U-space services. Hence, within the USPASIM system, different agents together with their coordination and dependence interactions are considered to model the U-space ecosystem. The identification of these agents and their expected interactions is based on the state of the art in Section 3.1 , and detailed in the forthcoming sections.

As for the implementation of the USPASIM platform, the aforementioned frameworks enabling agent architectures have not been used. On the one hand, frameworks add programming overhead and conceptual rigidity that, while useful for large-scale industrialized projects, may not be appropriate or flexible for the objectives of this section (providing models of different entities of a u-space system). Another argument against their use is that the models use legacy algorithms developed in other languages. Moreover, agile integration with web protocols (e.g.: HTTP, MQTT) is necessary for interaction with real U-Space systems. For these reasons, and ad-hoc platform has been developed (nevertheless, migrating models to a standardized framework is still feasible in the future). Therefore, the USPASIM platform is articulated around **different services which encapsulate thematically related agents**. As depicted in Figure 3.5 , these services are:

- **Operator simulation service**. This module encapsulates `Operator` agents which are tasked of generating drone traffic (specification of a series of u-plans) and of performing strategic phase interactions with U-Space services to authorize that traffic.
- **UAS simulation service**. `Pilot, GCS and Drone` agents are considered in this module to realistically simulate the drone command and control process and the resulting trajectories. At the same time, tactical phase interactions with the U-Space system are simulated here, including telemetry reporting and traffic information or conformance alerts reception. The simulation of these three agents is performed in a closed loop, so that reaction actions for contingencies (e.g.: low battery, loss of communications link, tactical alert...) can be modeled. In this sense, both pilot actions and autonomous drone responses are considered and may result in a change of the originally planned trajectory.
- **Surveillance simulation service**. Aircraft detection through a surveillance network is modeled in this module by means of `non-collaborative sensor` agents (which may model radars, acoustic sensors or cameras), `collaborative emitters` (which

model drones broadcasting information in ad-hoc networks), `collaborative receivers` (which model the reception of collaborative information in ADS-B, FLARM o Remote ID systems) and `fusion nodes` (which model the aggregation of detections from different sensors and track forwarding to the U-Space system).

- **Network simulation service**. Information exchange between the previous agents and the U-space system through simulated communication networks is modelled in this module. To that end, `mobile communication nodes` (both public communications and point to point links) are considered together with `wired communication nodes`.
- **ATM simulation service**. This module enables injecting simulated manned traffic through a set of source agents.

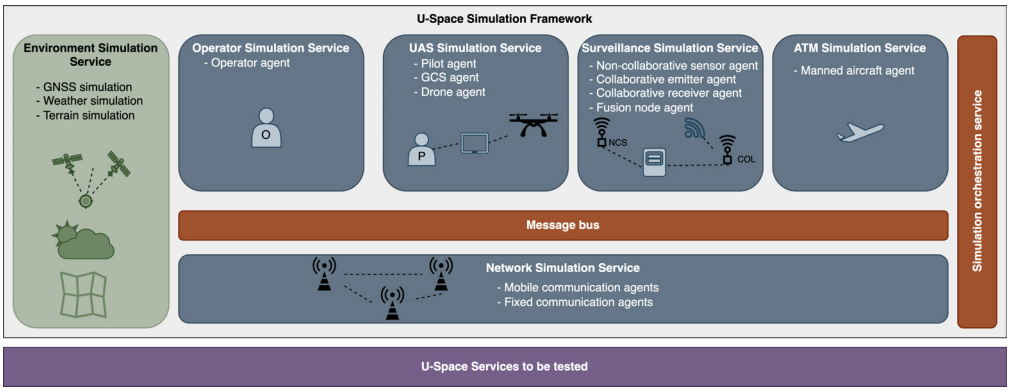


Figure 3.5: USPASIM architecture

In parallel, to equip the simulation with contextual information, the **environment simulation service** provides simulated navigation, terrain and weather information. For instance, this information is considered by the drone agent to include the effects of navigation degradations or adverse weather conditions (e.g.: strong winds) in the simulation of the drone trajectory. In addition, a set of U-space services must be connected to the simulation environment for experimentation, diagnosis and validation. To enable testing of only a subset of services, or to evaluate U-space policy alternatives, the simulation environment has a **reference implementation of U-space services** that can replace the missing ones.

Finally, in order to enable a synchronized simulation, two additional services are contemplated. First, a real-time pub-sub **message bus** enables a seamless information communication between different distributed services (and agents). For instance, to enable drone detection from surveillance sensors, the status of each drone is periodically reported through the message bus so that it can be used by the Surveillance Simulation service. Similarly, simulated networked messages between agents and U-Space services are also forwarded through this bus into the Network Simulation service. Second, the **simulation orchestration service** allows for the configuration of a simulation scenario in the different services and coordinates its synchronized execution by enabling an interface to start and end the simulation.

By default, the simulation is carried out in **real time** to allow its simple interconnection with operational U-Space systems and even to enable the realization of hybrid simulations in which some of the agents are replaced by a human participant or real hardware platforms (e.g.: a

real GCS or drone). However, the system does allow for **fast-time simulation of strategic phase interactions** so that tests can be made on authorization rates, traffic densities...; which will be needed later in Section 3.3 and Section 4.2 .

In short, a complete simulation system is considered in which realistic models for some of the agents involved will be detailed in the following sections (Section 3.2.3 to Section 3.2.7). For other agents, simplified models or models already found in the literature are used. In this sense, the simulation of communications (including the C2 link) is based on what is described in [Car+21c]. The simulation of drone trajectories is based on the synthetic trajectory engine described in [Bes+18], which in turn is based on the multirotor trajectories specification language defined in [Fro+19].

3.2.3 Modeling human behavior in the strategic phase: operator agent

As previously introduced, `Operator` agents represent real drone operators seeking to perform a set of missions for the achievement of their business objectives (e.g.: monetary income or social objectives in the case of state-affiliated operators such as police). To perform these missions, each operator has a fleet of drones and a set of associated pilots. The operator's role is to define the missions to be performed and to assign them to each pilot and drone. In this sense, each `Operator` carries out two processes, which have been depicted in Figure 3.6 :

- **Traffic generation**. First, the objective is to automatically (i.e.: stochastically) define a set of drone operations which are to be performed by the operator's pilots and fleet of drones. The key contribution is a process able to generate mission descriptions that are operationally meaningful for different mission types. This process is described in Section 3.2.3.2 .
- **U-plan definition and authorization process**. Then, for each of the generated operations, a u-plan must be created and an authorization must be requested to the USSP. Communication with the USSP services is handled through the network agents to simulate the effect of possible degradations in the network. As a result of these interactions with the USSP's services (mainly u-plan preparation service and u-plan authorization service), a set of authorized u-plans can be communicated to **Pilot** agents to simulate each flight. This process is discussed in Section 3.2.3.3 .

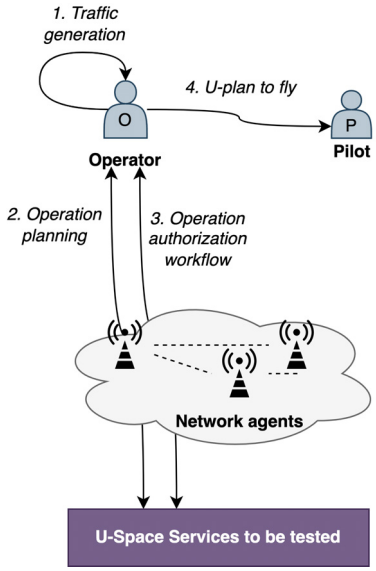


Figure 3.6: Operator agent interactions

3.2.3.1 Operator behavior specification

Before explaining the two processes simulated within the `Operator` agent, it is necessary to identify different features that can affect the behavior of the agent and that must be parameterized. In this sense, operators may be classified according to:

- **Types of missions to be performed**. Depending on its missions' characteristics, the observed operator's drone traffic will follow different patterns that will be considered in traffic generation.
- **Operator geographical extent**. Operators may be simulated that concentrate their operations in an area or certain corridors (e.g.: urban operators). On the contrary, other operators may perform operations scattered in wide areas (e.g.: rural operators).
- **Operator size**. The size of the operator determines the number of aircraft and pilots available and consequently the number of flights to be generated in a scenario.
- **Operator profitability**. If economic factors are introduced into the flight authorization processes (as it will be done in Section 3.3), operators with different levels of income per flight (€/flight) should be simulated to analyze the possible impact of this feature.
- **Operator responsibility**. In general, operators act in a responsible and fair manner by reliably informing the USSPs about their operations and respecting the decisions of the USSPs. However, there are other operators who may act in a self-interested manner or even operate in a non-collaborative or rogue manner without respecting the outcome of the authorization process.
- **Flight submission time**. Finally, and although it may be related to the type of operation, an operator may have a behavior related to how far in advance it submits their operations to be authorized. As it will be discussed later in Section 3.3 , this can have a major impact on the flight authorization rate.

Taking these characteristics into account, the behavior of an operator can be characterized by the **parameters** defined in Table 3.7 .

Parameter	Description
Identifier	Identifier of the operator agent.
Drones	List of drones owned by the operator and used to fly his missions. For each drone, a reduced kinematic model (i.e.: drone type, minimum and maximum speeds, maximum endurance) is included to optimize the assignment of drones to each type of mission.
Pilots	List of pilots associated to the operator who will be assigned to an operation to conduct it.
Traffic patterns	A list of traffic patterns together with values for their associated parameters which are used for mission-aware automatic flight generation. This includes the number of flights to be generated for each traffic pattern, which should be related to the operator size.
Geographical extent.	A bounding box defining the area in which operator's traffic will be generated.
Income by flight	An interval (minimum and maximum income [€/flight]) defining how profitable operator flights are.
Operator responsibility	A flag indicating whether the operator is fair/responsible or whether it is non-collaborative/self-interested.
Flight submission time	An interval (minimum and maximum time before take-off) indicating when u-plans are submitted to the U-Space system.

Table 3.7: Operator agent specification parameters

3.2.3.2 Automatic mission-aware traffic generation

As described in Section 1.2 , different operators may perform different types of drone missions (e.g.: logistics, precision agriculture, mobility...), which determine the shape of the u-plan (extent, heights...) needed to perform the mission. In this sense, authors in [Rap+23b] affirm that realistic traffic scenarios are critical to ensure the validity of simulated results. From this fact and the requirement of automating tests to make them scalable, the need of developing tools for the automatic generation of drone traffic is clear.

In this sense, the proposed idea is to identify common flight patterns across different type of drone missions. Then, a statistical specification of each pattern is made to represent the shapes and flight distribution under that pattern. From that specification, it is possible to implement stochastic traffic generators that generate a set of flight plans that follow the desired pattern. As a result, and considering that these traffic patterns can be mapped back to each type of mission, complex and realistic traffic scenarios can be easily defined as an aggregation of stochastically generated flights derived from a set of user-selected mission types for each operator.

Starting with the identification of common flight patterns, several types of drone missions

have been already discussed in Section 1.2 . A discussion on the main types of flights for each application can be found below based on the information in [Sha+19] and the author's operational knowledge:

- **Industry and construction**: two main types of flights can be considered depending on the type of infrastructure to be monitored/inspected. In the case of roads or power lines, flights follow linear trajectories along the infrastructure. However, for mines or buildings, flights follow arbitrary trajectories along the infrastructure that can be modeled as delimited flights within an area where the drone follows variable trajectories. Multirotor drones are used in shorter missions where maneuverability is key, whereas fixed-wing drones are utilized for linear inspections.
- **Logistics**: in the case of inter-warehouse transport tasks, the trajectories followed by these drones tend to be a series of linear trajectories among static positions (warehouses). Another possibility consists of last-mile delivery, where variable points are brought in (delivery locations). Multirotor or eVTOL drones may be used in this application depending on the required endurance.
- **Mobility and Transportation**: flights describe longer trajectories, ideally following a straight line between two points (vertiports). However, due to conflicts with other flights or restricted areas, they may need to change their trajectory. For this reason, eVTOL drones are used for these types of applications, as they can perform longer flights.
- **Agriculture**: drones dedicated to precision agriculture perform flights inside a delimited area (crop fields) in which they usually follow very regular trajectories, generally sweeping over it so they can perform their monitoring, spraying, or sowing tasks in a homogeneous way across the entire terrain. Multirotor drones are typically employed in this task.
- **Security and Defense**: like in the previous case, these flights occur in a delimited area, following regular trajectories to cover an area of interest (for searching/exploring the area). However, in this case, flights may start outside the interest area as it may be quicker or unreachable by ground. Additionally, another modality involves a drone called "captive" that remains practically static with the mission of monitoring a specific area.

From the previous mission-driven analysis and considering the point of view of the type of u-plan to be defined in the U-Space ecosystem, six common flight patterns can be extracted (as depicted in Figure 3.7).

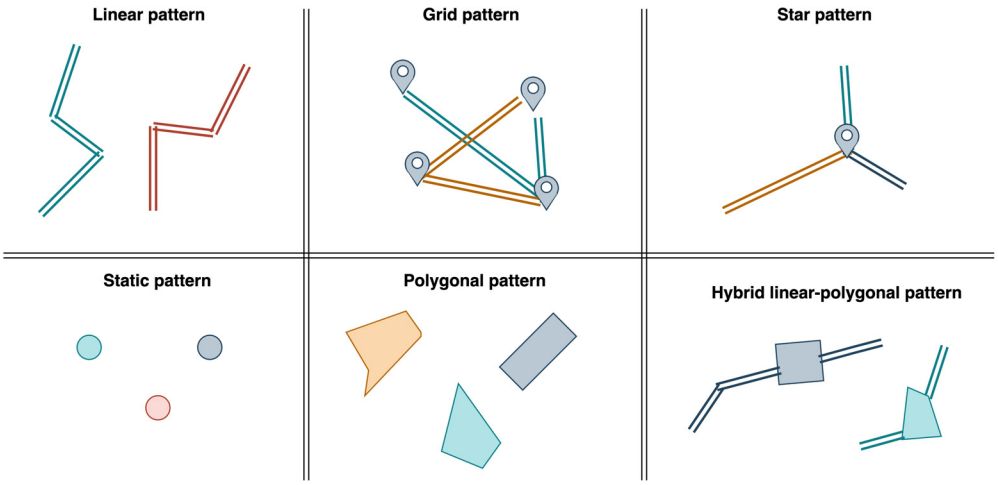


Figure 3.7: Considered flight patterns to model u-as operations

For each flight pattern, a random traffic generator has been implemented that, given some input parameters (which are detailed in Table 3.8), produces a set of conforming u-plans. Generated operations are represented using a u-plan data model that extends the usual volume-based approach by also contemplating the existence of linear flight phases (a list of waypoints). Thus, each u-plan would consist in a combination of arbitrary shape 4D-volumes and linear shapes. Unmanned operations representation through different u-plan data models will be later described in Section 4.2 . As proposed by this author and others in [Rap+23a], considered mission-aware generators are:

- **Linear pattern**. To consider linear missions in industry and construction domains, this pattern represents linear flight paths through a set of arbitrary waypoints. To this end, the associated traffic generator is based around an adapted version of the random linestring generation algorithm in [Tur15a]. The algorithm generates u-plans of random length and number of waypoints within a provided bounding box (aligned with the `Operator` geographical extents parameterization).
- **Grid pattern**. Logistics and mobility missions between a grid of fixed, well-known locations (which can represent vertiports) are represented by this pattern. Given a list of these locations, the associated generator creates linear u-plans passing through a subset of that locations, landing in them to drop off passengers or cargo, and taking off to start the flight to the next location.
- **Star pattern**. Representing last-mile delivery missions, this pattern consists in round trip linear flights between a fixed, well-known waypoint and an arbitrary waypoint within an area of interest. The generator relies on a bounding box confined random point generation algorithm.
- **Static pattern**. Captive-drone operations in security or advertising can be modelled by this pattern which consists in a drone hovering at a given altitude. The output u-plan generator produces circular 4D volumes in random locations to confine the quasi-static operation of the drones through a random mission duration.

- **Polygonal pattern**. Agriculture or surveillance missions in which the exact trajectory to be followed by the drone is not known or is too complex to be detailed at flight plan level are represented by this pattern. Thus, the pattern generates flight plans consisting of a single polygonal area of arbitrary shape. Random shape generation is based on the algorithm described in [Tur15b] that allows to control the size of the polygons and their number of vertices.
- **Hybrid linear-polygonal pattern**. Finally, by combining the linear and polygonal patterns it is possible to represent operations in secluded areas which require a linear segment to arrive/depart from an arbitrary polygonal operational area.

Traffic pattern	Required parameters	Applicable mission types
Linear pattern	• Generation area bounding box. • Minimum and maximum flight length [km]. • Minimum and maximum number of segments. • Maximum rotation angle [º]	• Inspection of linear industry/construction infrastructure
Grid pattern	• List points [lat, lng] within the grid. • Minimum and maximum number of visited points by flight.	• Inter-warehouse logistics • Urban Mobility between vertiports
Star pattern	• Central point [lat, lng] of the pattern. • Generation area bounding box.	• Last-mile delivery
Static pattern	• Generation area bounding box. • Minimum and maximum flight duration.	• Static surveillance in security/defense. • Advertising.
Polygonal pattern	• Generation area bounding box. • Maximum radial length [km] • Minimum and maximum number of vertices.	• Inspection of non-linear industry/construction infrastructures. • Agriculture. • Moving surveillance in security and defense. • Recreational flights
Hybrid linear-polygonal pattern	• Generation area bounding box. • Maximum radial length [km] • Minimum and maximum number of vertices. • Minimum and maximum flight length [km]. • Minimum and maximum number of segments. • Maximum rotation angle [º]	• Search and rescue in secluded areas (security and defense)

Common parameters
- Number of flights.
- Traffic pattern time interval (min time, max time).

Table 3.8: Parameters needed to model traffic patterns and associated mission types.

Once the flight geometry has been generated, flights are distributed homogeneously over the duration of the scenario to set their take off time and authorization submission time (based on the operator parameterization). In addition, a drone and a pilot must be assigned to each mission ensuring that they can perform it (i.e.: that there are no time overlaps, that the drone endurance is sufficient for the mission duration...). Finally, flight generation process also considers assigning to each flight parameters such as the priority of the flight, or the economic benefit obtained per operation that will be useful in other sections of the thesis.

The above patterns make it possible to **easily generate a large number of flights**

following realistic patterns. Yet, there are occasions when, in order to test some functions of the U-Space ecosystem, it is necessary to force the occurrence of non-realistic patterns. For example, previous patterns generate operations that may or may not be in conflict depending on chance. Nevertheless, to evaluate conflict detection algorithms (both strategic and tactical), it is necessary to ensure that the generated operations are in conflict. Furthermore, for load testing in the tactical phase, it may be necessary to ensure that all in-flight operations conflict at the same instant of time. Similarly, congestion management algorithms require scenarios where high demand hotspots are created to verify if the algorithm is able to avoid congestion. In this sense, the following additional traffic patterns are also considered:

- **Conflict pattern**. This pattern guarantees the existence at least one conflict between pairs of aircraft. For this purpose, the random generation of flights is not performed on a flight-by-flight basis but in pairs. First, the geographical point where the conflict is due to occur is randomly selected. From this point, linear flight phases before and after this point are generated with the previously described algorithm, and for each flight in the pair. Finally, and considering a kinematic model of the drone, a take-off time is assigned to each aircraft ensuring that they temporally coincide at the conflict point.
- **Hotspots pattern**. This pattern generates high flight density around a set of user selectable hotspots. For its implementation, the previous linear pattern is reused by varying the size of the flight generation area. First, a background, low-density traffic is generated throughout the area of interest. Then, for each hotspot, linear flights circumscribed to a small area around them are generated.

3.2.3.3 Modeling human interactions with U-space strategic phase services

Once the traffic generation phase is completed, the `Operator` agent carries out the simulation of the strategic phase interactions with the U-Space services. These interactions are described in the sequence diagram in Figure 3.8 and consist in three sequential processes:

1. **U-plan definition/preparation**. In the previous phase, each operation has been defined at a high-level indicating flight geometry and take-off time. However, the operation of U-space services requires a detailed definition of the u-plan including the passing time intervals for in each phase of the flight, an altitude specification... For this purpose, USSPs might provide a u-plan preparation service that allows refining the u-plan definition. These services include terrain models to perform height conversions and avoid collisions with the terrain, aircraft dynamic models to predict transit times, and data model transformation utilities. From the simulation point of view, this process simply consists of initiating the interaction with this u-space service and processing its response which will consist of a complete and valid u-plan.
2. **U-plan submission for authorization**. The next interaction that needs to be simulated is the operator's u-plan authorization request. Again, from the simulation point of view, this only consists in sending the corresponding request to the USSP (through the Flight Authorization Service) at the defined submission time and to process the response. The outcome of this process may be:
    - The u-plan is directly authorized without changes as no issue has been found by

the USSP (through the SCD service). In this case, the authorized operation is communicated to the `Pilot` agent for in-flight simulation. This is also the case if the authorization decision is delayed in time (e.g.: if it requires manual approval by an ATCO or authorities)

- The u-plan is not authorized but a valid alternative is provided by the USSP (through the SCR service, if available). The alternative u-plan is resubmitted for authorization and provided to the `Pilot` agent, if accepted.
- The u-plan is not authorized, and no alternative is provided. In this case, the operator agent can start the following deconfliction process.

3. **U-plan deconfliction process**. This process consists in emulating the actions of a human operator that tries to find a solution to the conflicts detected in its operation. In this sense, by means of the algorithms described below, modifications are made to the u-plan to avoid the conflicting zones, and the operation is sent back to the USSP so that it may be authorized. If the alternative operation is accepted, it is forwarded to the pilot. Otherwise, the lifecycle of that operation is finished.

In addition to the nominal behavior described above, an `Operator` can be parameterized as self-interested or non-collaborative. This means that it does not respect the outcome of the authorization process and fly the operation regardless of whether it has been accepted or rejected. In that case, the u-plan is directly forwarded to the `Pilot` agent.

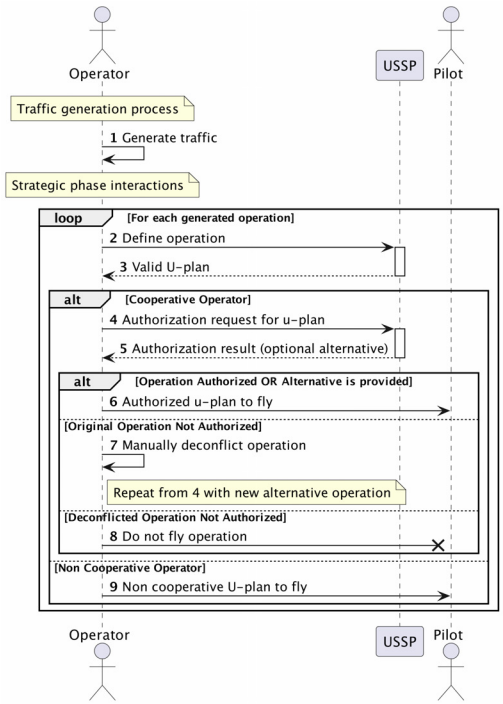


Figure 3.8: Sequence diagram representing modeled interactions in the strategic phase.

Finishing with the elements modelled in the `Operator` agent, let us consider **simulated deconfliction capabilities**. The alternative generation process by a human may be formulated

as a combinatorial optimization problem that can be solved by an algorithm. For this purpose, let us assume:

- A flight plan, fp, is to be deconflicted. The u-plan is characterized by:
  - A time duration of $fp_{duration}$
  - N detected conflicts.
- Each conflict can be resolved individually by applying one strategy from a set of M available strategies: $\{0, S_1, S_2, ..., S_M\}$, 0 meaning that no strategy has been applied. Globally, the variable S is defined to indicate which alternative has been selected for each conflict:

$$S = [s_1, s_2, s_3, ..., s_n, ...s_N], \;\; s_n \in \{S_1, S_2, ..., S_M\} \tag{3.1}$$

  so that s_n represents the strategy used to solve conflict n.
- The operator's decision-making process for preferring one alternative over another can be modeled as a cost function dependent on the original flight plan and the alternatives applied for each flight plan:

$$c(S, \;\; fp): \quad S \rightarrow \mathbb{R} \tag{3.2}$$

- For a solution to be valid, it must respect the original take-off and landing waypoints and must avoid previously detected conflicts while avoiding incurring in new conflicts.

Given that, the deconfliction process is formulated as minimizing the cost function by adjusting the selected strategies subject to the original flight plan restrictions and that conflicts are eliminated:

$$\begin{gathered} \min_{S} c(s, fp) \\ \text{subject to: no conflicts and original fp restrictions} \end{gathered} \tag{3.3}$$

Under this conceptual umbrella, 3 possible conflict resolution strategies (S set) are considered and depicted in Figure 3.11 :

- **Time resolution**, which consists in delaying the u-plan take-off time to avoid conflict in time.
- **Height resolution**, which consists in changing the u-plan height around the conflicting point to avoid the conflict in height.
- **Route resolution**, which consists in changing the u-plan route to go around the conflict, avoiding a 2D intersection.

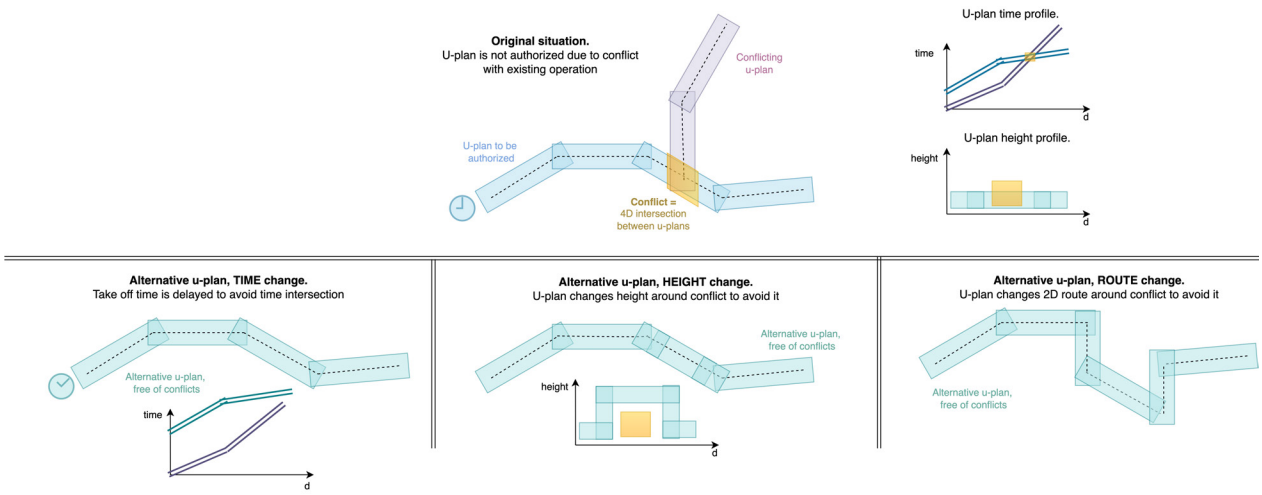


Figure 3.9: Alternative deconfliction strategies.

For the cost function, it is assumed that the operator's objective is to make the alternative operation as similar as possible to the original operation. Therefore, the cost function is defined in terms of the change in flight time:

$$c(S, fp) = |fp_{duration} - fp^{S}_{duration}| \tag{3.4}$$

where fp^S represents the u-plan where the strategies defined in S have been applied. This cost function can be tuned in order to prioritize the use of one strategy over others. Indeed, by adding a cost penalization each time an undesired strategy (from the operator-point of view) is used, it is possible to consider operator preferences in the deconfliction process.

As for the resolution of the combinatorial problem, it is mapped into a graph search problem. To this end, the implemented algorithm iteratively builds a tree by gradually solving each conflict (tree level) through the application of the different strategies (nodes from the same level). Thus, within the graph, each of the nodes represents a flight plan containing a partial solution of the problem with some of the conflicts already solved with a combination of strategies determined by the path from the root to that node. If a partial solution does not meet the restrictions, it is removed from the tree. The tree is gradually expanded (i.e.: new conflicts are solved) using a graph search algorithm (i.e.: A*) that considers the cost function of each partial solution to select which nodes are expanded until an optimal solution is found. It is possible to find an optimal solution due to the fact that the value of the proposed cost function for a partial solution never overestimates the value of a full solution (the current conflict resolution strategies implementation ensures that the u-plan duration is never reduced).

Although the presented model does not aim to provide an optimal solution to the deconfliction problem, a literature review reveals other possible algorithms that could have been implemented. For instance, graph search is used in [Raz+18] to find a conflict-free path. In this case, rather than choosing from a range of alternatives to avoid conflicts, a trajectory is defined directly by selecting the drone's maneuvers between two points. This way, conflict resolution could be addressed as a path planning problem with UAV-avoidance constraints,

as discussed in [AK20]. In this sense, authors of [Liu+23a] propose to project the desired flight path and conflicting u-plans into a hexagonal grid and to perform integer programming to find a viable solution. These solutions could be applied individually by each `Operator` provided that they have knowledge of conflicting u-plans.

Going beyond existing European regulations, strategic conflict resolution could also be addressed through direct coordination between the operators involved in a conflict. Following this idea, a negotiation between operators should be established performing coordinated changes to each operation until a free of conflict solution is reached. This approach could also be modeled in the simulator through interactions between the different agents. In essence, this would involve framing the problem as a collaborative path planning problem, as reviewed in [Zha+20].

3.2.3.4 Operator model execution results

In this subsection, a simple simulation scenario exemplifies the operation of an `Operator` agent. In this sense, the example considers an operator which is expected to carry out four types of aerial missions over the city of Soria:

- Logistic distribution missions between four warehouses,
- Last-mile delivery missions between the said warehouses and neighbourhoods in the city's outskirts,
- Infrastructure inspection missions scattered throughout the city,
- Precise agriculture missions in the rural areas close to the city.

The execution of these missions has been parameterized in the `Operator` agent by means of 4 traffic generation patterns as described in Section 3.2.3.2 . As a result of the traffic generation process, a set of u-plans has been generated, as depicted in Figure 3.10 where colour coding allows to differentiate traffic generated from each of the patterns. The figure shows how the proposed model allows to generate varied traffic and to superimpose different types of patterns in order to generate realistic mixed scenarios.

After the traffic generation process, the `Operator` agent performs the interactions with the U-Space system described in section Section 3.2.3.3 . In this case, it is of particular interest to analyze the flight deconfliction process which is illustrated in Figure 3.11 . It depicts the authorization process for two of the flights defined above. First, the green u-plan is submitted and thus authorized. Then, as the orange u-plan is submitted, a conflict is detected with the green u-plan (in red in subfigure a). To solve this conflict, the `Operator` agent autonomously deconflicts the operation with a route-based strategy and resubmits it (subfigure b). After this process, no conflict is found and both u-plans can be flown.

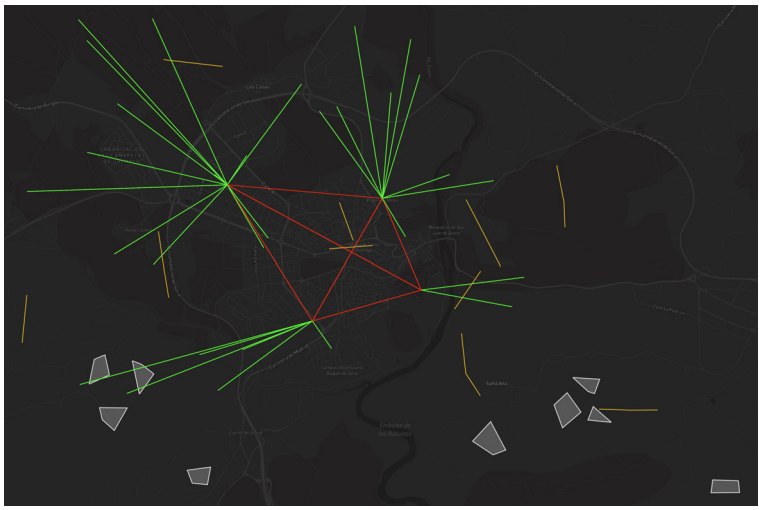
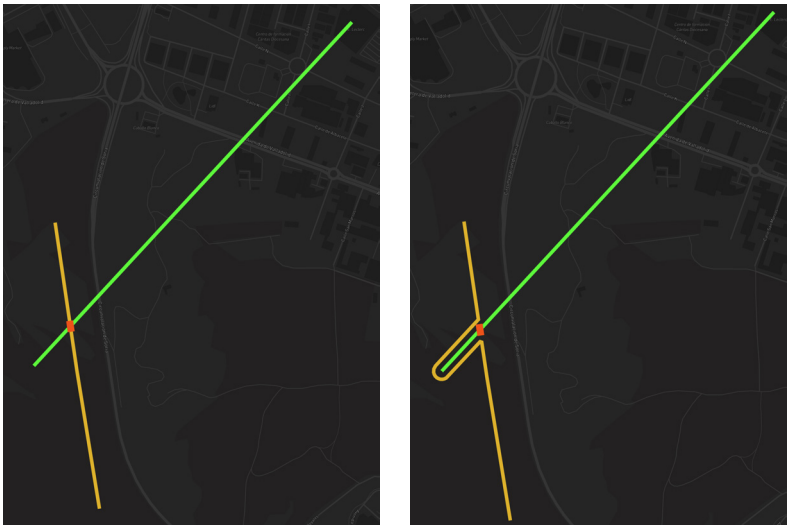


Figure 3.10: Example flight scenario generated by superposing flight patterns.
Four traffic patterns have been aggregated: a) grid pattern for a logistic mission, shown in red; b) star pattern for last-mile delivery, shown in green; c) linear pattern for inspection missions, shown in orange; d) polygonal pattern for agriculture missions, shown in gray



(a) Original u-plans. **(b)** Deconflicted u-plans.

Figure 3.11: Example of simulated deconfliction process within the Operator agent.
Pre-existing authorized u-plan is depicted in green. Submitted u-plan under analysis is shown in blue. Detected conflicts are shown in red.

3.2.4 Modeling human behavior in the tactical phase: pilot agent

Flight-time simulation of aircraft control and human contingency management is part of the functionality included in the `Pilot` agent. Therefore, this agent takes the operation defined by the `Operator` agent in the strategic phase (see Section 3.2.3), and generates the appropriate commands to manage its execution. To this end, and as shown in Figure 3.12 , the `Pilot` agent interacts in real time with the `GCS` agent, which serves as a gateway for sending orders to the `Drone` agent, that finally simulates the indicated trajectory. In parallel, it also receives drone status feedback and contextual information from the USSP, so that it can react to unforeseen situations and perform a closed-loop control of the aircraft. Two processes have been considered within this agent:

- **Closed-loop decision engine**. This process, which will be described in Section 3.2.4.1 , simulates pilot's decision-making in order to command an appropriate sequence of orders for the safe completion of all flight phases (i.e.: take-off, en-route and landing).
- **Trajectory determination and piloting model**. While the operator-defined u-plan should be used as a guideline to determine the drone's trajectory, it does not specify it completely and is subject to the pilot's actions. Thus, this process (detailed in Section 3.2.4.3) includes the guidance of the aircraft considering the complete specification of the aircraft's trajectory together with flight-control errors.

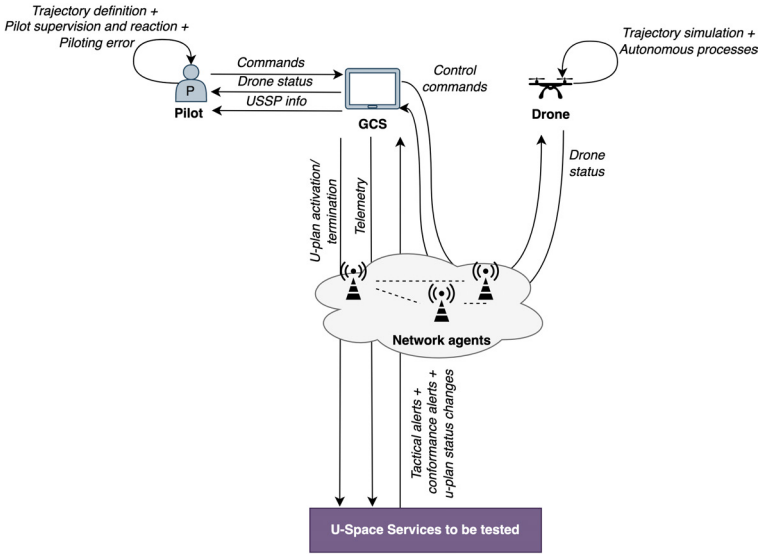


Figure 3.12: Pilot agent interactions

In order to tune the behavior of those proceses within the `Pilot` agent, the following **parameters** are considered:

- Assigned drone operation, as defined by the `Operator` agent.
- Assigned `GCS` and `drone` agents, with a 1 to 1 relationship between the three agents.
- Collaborative / non-collaborative behavior. That is, whether the `Pilot` performs tactical interactions with the USSP system and complies with received instructions and alerts.

- Flight control mode and associated flight control errors. As described in next subsections, either a manual or autonomous flight control can be simulated.
- Optionally, an actual trajectory specification that overrides the operator-defined operation to simulate non-nominal situations and contingencies.

3.2.4.1 A closed-loop pilot decision engine

Aircraft's control by the `Pilot` agent must be tightly linked to the GCS capabilities regarding available information and expected command sequencing. In this sense, the `GCS` agent of the USPASIM platform implements a control interface inspired in the MAVlink protocol (i.e.: a lightweight, widely supported messaging protocol for the C2 channel) [Mei09]. In particular, it supports the subset of commands and status messages described in Table 3.9 . These commands consider two possible flight control modes: an automatic one based on mission loading (i.e.: a list of waypoints) with autonomous flight execution aircraft, or a manual mode with pilot guidance (whose simulation will be described in the next section). Hence, the control engine detailed below should be aligned with this concept of flight operations, tipycally used in commercial drone flights.

Simulated C2 message	Origin	Description
START_ENGINES	GCS	Turn on drone engines and start transmitting telemetry information.
TAKE_OFF	GCS	Take-off and hover in air until further commands are received.
LOAD_MISSION	GCS	Loads the operation to be flown as a set of waypoints to be visited by the drone.
START_MISSION	GCS	Starts the execution of pre-loaded mission from the first waypoint.
PAUSE_MISSION	GCS	Pauses the execution of current mission and hovers in air until further commands are received.
RESUME_MISSION	GCS	Continues the execution of the current mission from the closest waypoint.
CHANGE_FLIGHT_MODE	GCS	Changes flight simulation mode between AUTOMATIC (low piloting error) or MANUAL (high piloting error, mimicking human control).
RETURN_TO_HOME	GCS	Halts current mission execution and implements an immediate Return To Home maneuver.
LAND	GCS	Lands drone from current position.
STOP_ENGINE	GCS	Stops drone engines and halts telemetry information reporting.
TELEMETRY	UAS	Status message with position, attitude, speed, attitude and autopilot state information.

Table 3.9: Simulated messages considered in the GCS-drone C2 channel.

The proposed `Pilot` agent decision engine is articulated as a finite state machine which is described in Figure 3.13 to Figure 3.15 . Each state is linked to an action to be performed

by the pilot throughout a flight's life cycle. Actions may consist of waiting for an event to occur (e.g.: receiving activation from the USSP), monitoring the outcome of a process (e.g.: periodically checking the status of the drone for unexpected variations), commanding an order to the GCS (e.g.: taking off the aircraft), or making an advanced decision in response to a contingency (e.g.: tactical collision avoidance). Transitions between states depend on the results of previous actions: the occurrence of external events (e.g.: reception of flight clearance or alerts from the USSP), changes in the status of the drone (e.g.: completion of the take-off operation) or the response received to the issued GCS commands (which may be successful or a failure).

With all this in mind, Figure 3.13 describes the states related to **flight initialization**. To start, the decision engine waits to receive the operation defined by the `Operator` (`AWAITING_OPERATION` state) and the corresponding authorization by the USSP (`AWAITING_FLIGHT_AUTHORIZATION`). Subsequently, the trajectory to be flown is specified (as further explained in Section 3.2.4.3) and the mission is loaded in the drone (`COMMAND_OPERATION_LOAD`). Once the take-off time defined by the operator is reached (`AWAITING_START_TIME`), the activation of the operation is requested (`REQUEST_ACTIVATION`), the engines are started (`COMMAND_START_ENGINES`) and the take-off is commanded (`COMMAND_TAKEOFF`). The state machine considers the possibility of collaborative operations (which respect the result of the authorization and activation process) as well as non-collaborative ones (which either do not respect it or do not carry it out).

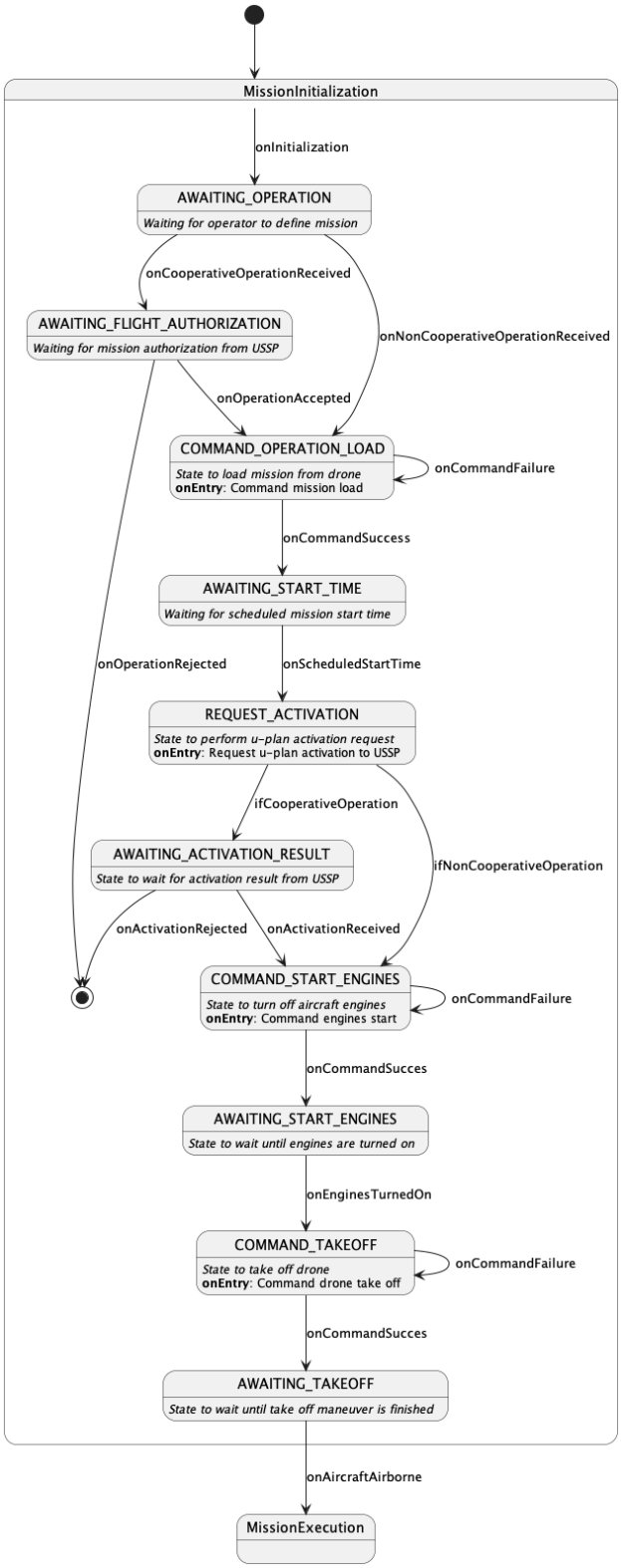


Figure 3.13: Pilot decision engine finite state machine for mission start-up

Once the aircraft is airborne, `Pilot` agent control is described by the finite state machine in Figure 3.14 . In it, the agent sets the flight mode (either automatic or manual control) and commands the start of the previously loaded mission (`COMMAND_MISSION_START`). As the scheduled flight proceeds, the pilot monitors the drone's status and the information received from the USSP. As a result of this process, the following events may occur during the **mission execution** phase:

- A **non-conformance alert** is received from the USSP. Due to the effects simulated in Section 3.2.4.3 , the aircraft trajectory may deviate from the authorized flight plan. As a consequence, the U-space system may generate a non-conformance alert. The `onConformanceAlertAcknowledged` event and successive states model the pilot's reception of such an alert and its reaction to it. As a simple reaction, the `Pilot` agent commands to pause the current mission to avoid moving further away from the authorized operation. Then, it seeks a solution by returning to the nearest point within the authorized u-plan, and resumes the operation from there.
- A **tactical alert** is received from the USSP. In case of loss of separation between two aircraft (usually one of them is a non-compliant, rogue aircraft) the U-Space system can send tactical alerts to warn the pilot of the imminent risk (Tactical Conflict Prediction service). The model also monitors these alerts and reacts accordingly (states after `onTacticalAlertAcknowledged` event). For simplicity, the reaction is to pause the ongoing operation and wait for the conflicting aircraft to move away (i.e.: the tactical alert disappears) before continuing the operation. Alternatively, conflict avoidance algorithms could be implemented (a description of alternatives can be found in [REH20]) or even the instructions received from a Tactical Resolution Service (if available) could be adopted.
- An **authorization revocation** is received from the USSP. The emergence of police or emergency operations (special operations in European regulatory jargon) may result in the operation's authorization being revoked while in-flight. This situation is contemplated in the `onMissionRevokedAcknowledged` event and successive states that prompt an immediate halt of the ongoing mission and command an immediate aircraft landing.
- An **autonomous drone response** is detected. Usually, drones are programmed with a set of contingency actions that are automatically executed in cases such as the loss of C2 communications link, low battery level or loss of the navigation system. These usually consist of a Return To Home (RTH) action or an immediate landing. The considered `Pilot` model is able to detect these actions (`onAutonomousRTH` and `onAutonomousLanding` events) and monitor their execution to ensure a safe operation termination.
- Successful **termination of the programmed mission**. Finally, it is necessary to consider (`onMissionExecutionFinished` event) the nominal case in which the programmed mission has been successfully executed and it is necessary to proceed to land and terminate the operation as will be described in the next paragraph.

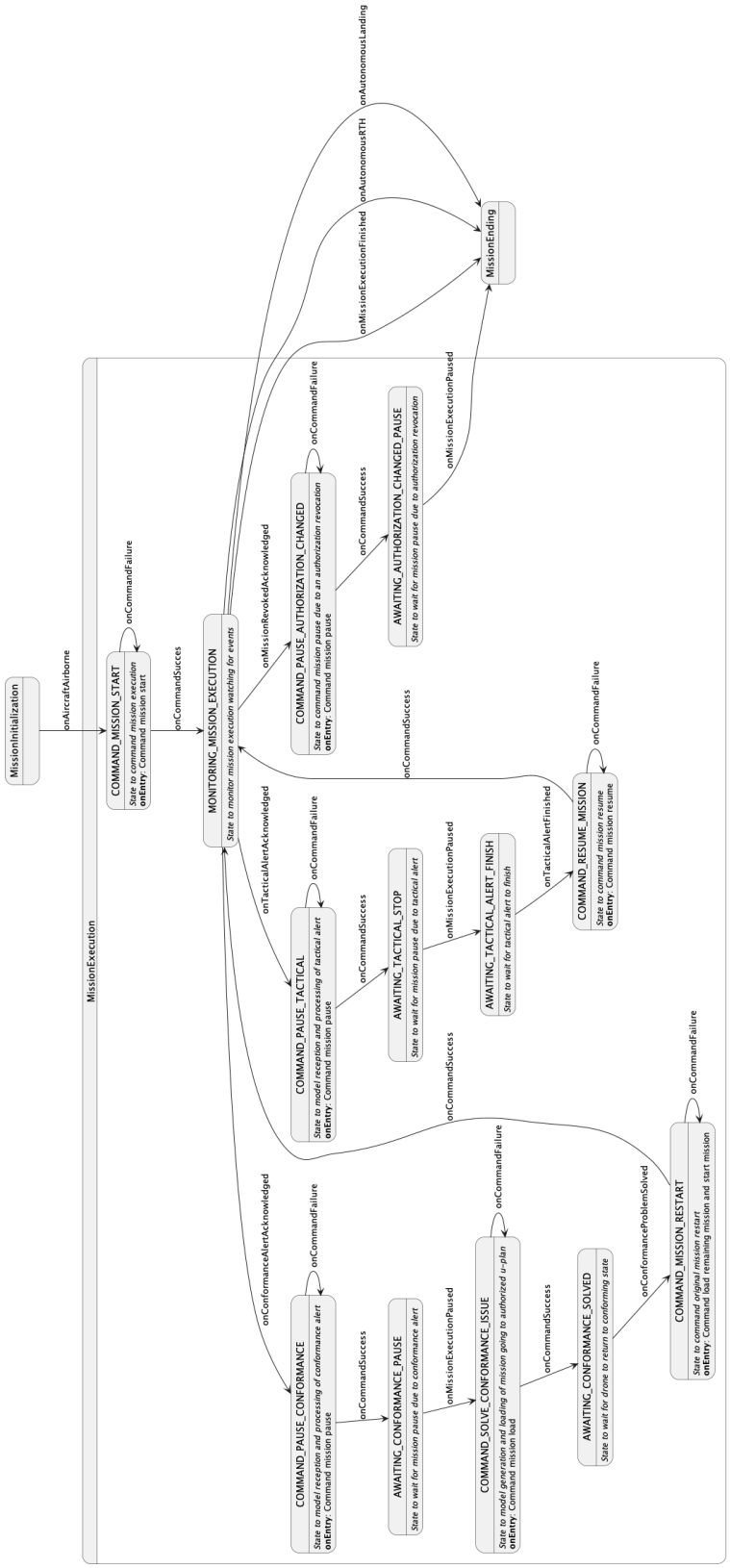


Figure 3.14: Pilot decision engine finite state machine for mission execution

To complete the flight operation lifecycle, Figure 3.15 describes the finite state machine charged of managing the **operation termination**. To this end, the aircraft is commanded to land (`COMMAND_LANDING`), engines are shut down (`COMMAND_SHUTDOWN`) and the USSP may be informed of the end of the operation (`REQUEST_TERMINATION`).

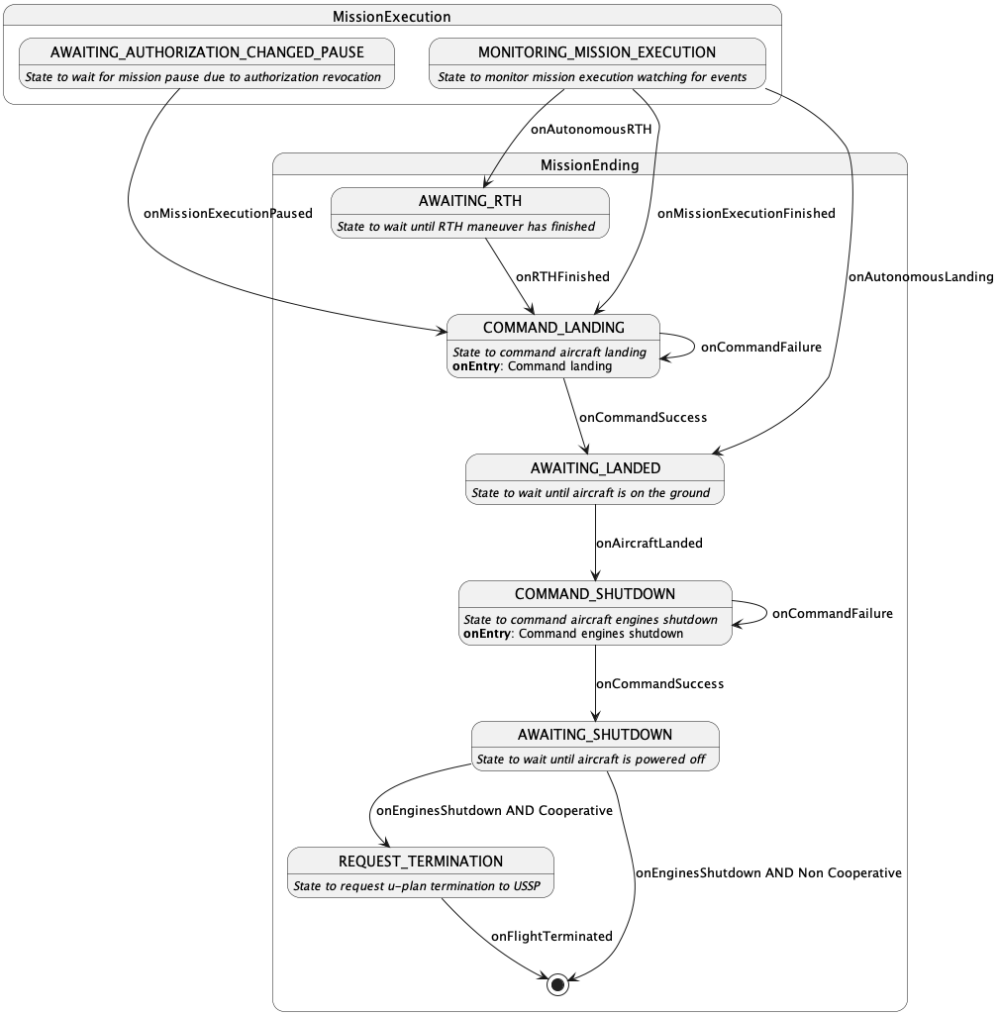


Figure 3.15: Pilot decision engine finite state machine for mission termination

The described model allows to simulate the usual actions of a pilot controlling a drone. Moreover, it does so in accordance with the usual C2 messaging protocols. Consequently, with minimal adaptations, it could even be connected to a commercial GCS software (e.g.: Ardupilot Mission Planner [Ard13]) for the control of a real drone integrated with the USPASIM simulation environment.

3.2.4.2 Trajectory determination and manual control errors

While the state machines in the previous section allow modeling the high-level operation of the aircraft, it is also necessary to **determine in detail the trajectory to be followed by the aircraft**. In this sense, the trajectory must follow the u-plan defined by the `Operator`

agent. However, as explained in Section 3.2.3.2 , a flight plan consists of a set of linear flight phases (list of waypoints to be visited) and polygonal flight phases. While the former flight phases completely specify the flight path, the later only define the flight envelope and it is up to the pilot to decide on the most appropriate path to achieve the mission. Therefore, and as discussed in [Rap+23a], it is necessary to model the linear trajectory defined by the pilots within these polygonal flight phases. For this purpose, Figure 3.16 depicts four alternative flight paths to be performed within polygonal flight phases:

- **Radial flight**. This flight path consists of a spiral flight from the center of polygon to its edges maintaining a minimum distance separation between passes (usually related with an on-board sensor coverage). The objective of this alternative is to visit the whole polygon starting from a known location. This makes it particularly suitable for search and rescue missions where the last position of the search object is known. It can also be applied for surveillance missions.
- **Sweep flight**. In this case, a sweep is performed with equispaced parallel lines, which makes it suitable for precision agriculture applications, although it can also be used for the previous use cases.
- **Random flight**. In recreational or infrastructure inspection flights, drones follow a changing and difficult to predict flight pattern. Consequently, for the purposes of simulating the U-Space ecosystem, this can be viewed as a random flight pattern (i.e.: list of random waypoints) circumscribed in the polygon.
- **Static flight**. Finally, in advertising or surveillance operations with captive drones, the trajectory followed by the drone consists in positioning itself at a given height and hovering there.

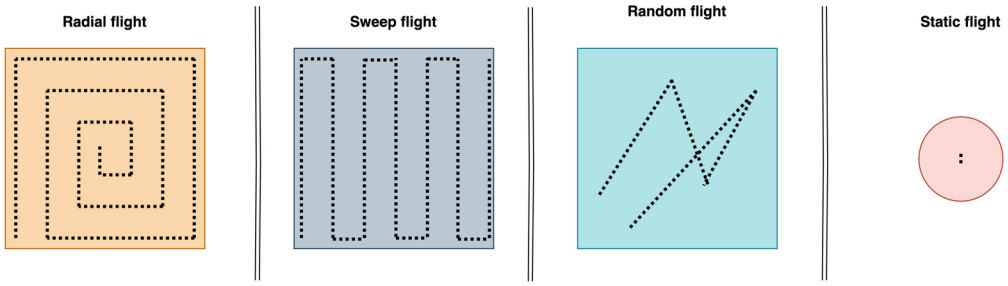


Figure 3.16: Alternative flight paths in polygonal flight phases

By applying the above flight paths together with the u-plan linear flight phases, it is now possible to completly specify the drone trajectory as a set of waypoints. As a result, the determined drone mission can be loaded into the GCS and simulated within the USPASIM framework. Alternatively, in order to allow the simulation of non-nominal situations, the previous flight path (derived from the u-plan) can be completely or partially replaced by another user-defined specification named `actual trajectory`. This path allows to change the original operation (e.g.: change take-off time, change waypoints, change altitudes...) to force the occurrence of compliance alerts or tactical conflicts.

Finally, the `Pilot` agent also considers possible **control errors** that deviate the real trajectory

of the aircraft from the ideal trajectory. For this purpose, position and velocity errors are modeled as a Gaussian stochastic process with exponential correlation. Starting from a random variable representing the pilot's acceleration control error ($a_n \sim \mathcal{N}(0, \sigma_a^2)$), it is possible to compute speed and position errors in each dimension as:

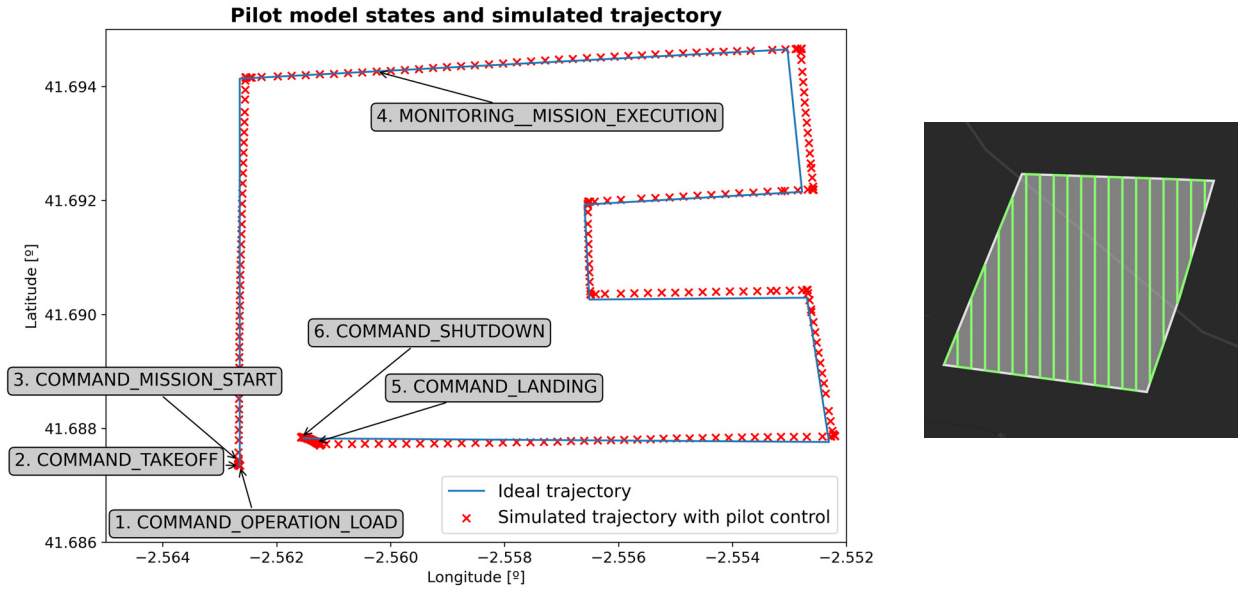
$$v_{error}[n] = e^{\frac{-T}{\tau_v}} \cdot v_{error}[n-1] + T \cdot a_n \tag{3.5}$$

$$x_{error}[n] = e^{\frac{-T}{\tau_x}} \cdot x_{error}[n-1] + T \cdot v_{error}[n] \tag{3.6}$$

where $v_{error}[n]$, $x_{error}[n]$ represent the speed and position error to by applied at time instant n. T is the time difference between two time instants, and τ_v, τ_x are the time correlation constants that allow to adjust the behavior of the random walk. Tuning these parameters (i.e.: σ_a^2, τ_v, τ_x) allow representing different pilot behaviors and control modes (e.g.: automatic vs manual). In any case, these errors are to be added to the ideal kinematic state of the drone as it follows the defined flight path.

3.2.4.3 Pilot model execution results

Finally, in Figure 3.17 two simple examples are given to demonstrate how the `Pilot` agent model described above works. On the one hand, Figure 3.17a shows the operation of the `Pilot`'s decision engine by displaying the most relevant states of the finite state machine throughout the flight (for clarity just a subset of states are shown), thus determining the simulated trajectory. This way, it can be seen how, at the initial phases, the model goes through the states that command the mission load and take-off, Then, as the mission progress it is monitored and, at the end of the mission, landing and engine shutdown are commanded. Likewise, the subfigure shows the effect of the pilot's manual control error as the ideal trajectory (in blue) differs from the simulated trajectory (in red). On the other hand, Figure 3.17b exemplifies the trajectory determination process for a polygonal flight phase. In this sense, the reserved polygonal airspace (in white) is covered by a sweep pattern (shown in green).



(a) Linear flight phase. **(b)** Polygonal flight

Figure 3.17: Example of simulated flights as directed by the models in the `Pilot` agent. Subfigure 3.17a shows a linear flight phase. Annotations depict relevant state changes throughout the trajectory. Both ideal (blue) and simulated trajectory with pilot error (red) are shown. Subfigure 3.17b shows a polygonal flight phase in white and the generated flight path (a sweep flight) in green color.

3.2.5 Modeling collaborative surveillance sensors

As already introduced, several collaborative surveillance technologies are available: ADS-B, FLARM, Remote ID. However, all systems consist of two different elements: an onboard transmitter and a ground-located receiver or sensor. This will be modeled by two analogous agents: the `collaborative emitter` and the `collaborative receiver`; interacting with other agents within the considered simulation framework, as shown in Figure 3.18 below.

Each simulated drone could have one or more associated `collaborative emitter` agents in charge of broadcasting its position with a different protocol. On the other hand, each `collaborative receiver` will be adjusted to operate with one of the surveillance protocols. As a result, it will be able to receive information from several drones as long as they have the corresponding emitter and are within its range.

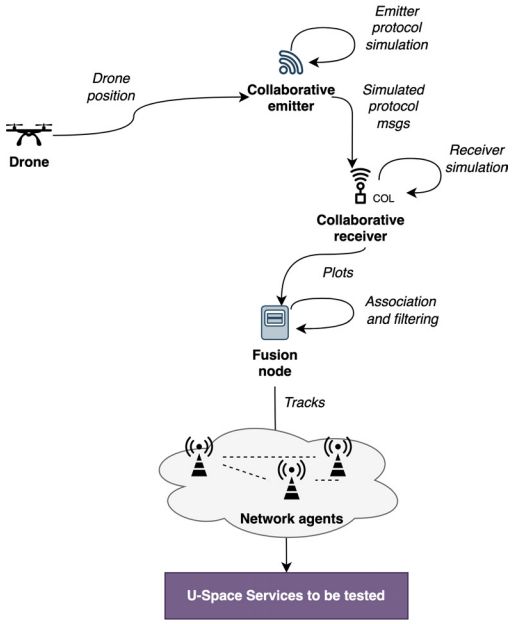


Figure 3.18: Considered flight patterns to model uas operations

From the detailed description of the protocols made in in Section 3.1.5 , it can be concluded that in order to simulate the message transmission and reception process of each protocol, it is necessary to consider several subprocesses that are depicted in figure Figure 3.19 . Regarding the **`Emitter` agent**, which will be discussed in Section 3.2.5.1 , its operation consists in obtaining information of the drone positioning, and encoding it using different resolution and message structures, which are dependent on the type of surveillance protocol. Then, two types of messages can be generated, either periodically or due to protocol-defined events such as important position changes. These messages may contain only UAS identification information, or identification plus encoded positioning/kinematic information. Finally, protocol messages are broadcasted over a simulated RF physical layer for which a Medium Access Control (MAC) protocol is modeled.

As for the **`Receiver` agent**, further described in Section 3.2.5.2 , it collects simulated messages sent by collaborative emitter agents. To this end, the physical layer characteristics and environmental conditions are considered to determine the receiving range and the signal quality. Indeed, as a signal is propagated and received it suffers attenuation, interferences, presence of noise, etc. which may result in digital information degradation (bit errors, burst errors...) at the receiver (sensor). Also, concurrent access to the shared RF medium can also affect the receiving capacity and should also be modelled. Moreover, even if the digital signal has been successfully detected by the receiver, it does not guarantee that corresponding message can be decoded. This will depend on the number of transmission-generated errors (which depends on signal quality, interferences, etc.) and the capacity of the protocol to correct such errors, that should be considered. Finally, the receiver will process the received information and generate plots reports that will be sent to a fusion node agent for further processing (as it will be later described in Section 3.2.7).

As discussed in the author's published contribution [Bes+22b], a top-down approach has been

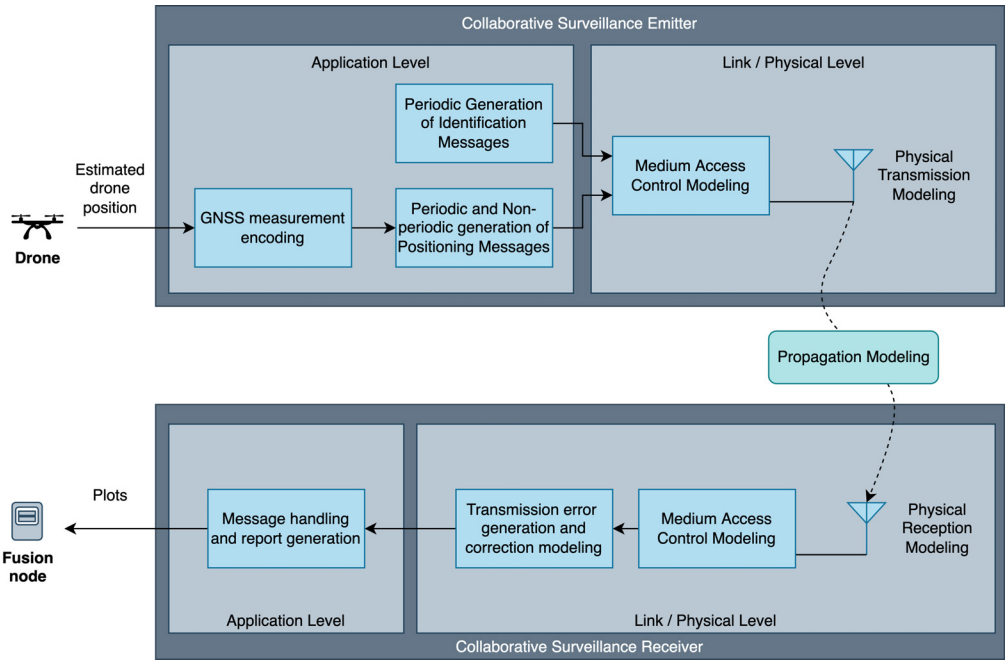


Figure 3.19: Considered flight patterns to model uas operations

followed to model these two agents. First, a general model that considers all the elements described above and shown in Figure 3.19 is proposed. In this sense, a series of parameters and functions will be defined that will depend on the protocol implemented by each agent. This will be noted by the superscript *protocol*. Subsequently, this general model can be particularized for each of each surveillance protocols. Particularly, the following cases are considered and described in Section 3.2.5.3 : ADS-B, FLARM, Remote ID with BT legacy physical layer, Remote ID with BT ER physical layer and Remote ID with WiFi physical layer.

3.2.5.1 Collaborative emitter agent

As previously introduced, the simulation model for `collaborative emitter` agents considers the following elements:

- **Drone position encoding** which determines which information is sent (e.g.: FLARM does not include speed information whereas Remote Id does) and the resolution of each element within a measure. Position information would be obtained from the drone agent associated with each emitter.
- **Message generation frequency** and other conditions that trigger messages, which determines the number of messages of each type to be generated and their timing.
- **Medium Access Control**. Depending on each of the protocols, emitters may be required to sense the RF medium before transmitting to avoid collisions. This can yield to transmission delays and depends on the number of emitters within the area.
- **Physical transmission**. The emitter RF module propagates a signal with the message

in a specific frequency, from the drone location and with a given transmission power and directionality.

In order to **parameterize** such an agent, it is necessary to define the drone to which it is associated and the type of protocol it implements.

Drone Position Encoding

To model the information forwarded by each protocol and the resolution used to forward that information, a list ($P^{protocol}$) of the information elements included by each protocol can be defined as follows:

$$P^{protocol} \subseteq P \tag{3.7}$$

$$P = \{timestamp,\ longitude,\ latitude,\ altitude,\ speed\} \tag{3.8}$$

Where P is the set of all possible state information variables provided by the drone or GNSS system and *protocol* denotes the surveillance system the list refers to.

Then, for each element of the said list or subset, a minimum encoding resolution ($r_p^{protocol}$) must be defined, understood as the minimum change of value representable with the protocol message encoding:

$$r_p^{protocol}\ ,\quad \forall\ p\ \in P^{protocol} \tag{3.9}$$

Given this resolution and the drone reported value for a variable, it is possible to encode the information forwarded by a given protocol as follows:

$$m_p^{encoded} = \left\lceil \frac{m_p^{drone}}{r_p^{protocol}} \right\rceil \cdot r_p^{protocol}\ ,\qquad \forall\ p\ \in P^{protocol} \tag{3.10}$$

Where $\lceil \cdot \rceil$ represents rounding to the nearest integer and m_p^{drone} refers to the drone-measured variable or state in its original precision. This encoding is only performed to the state variables forwarded by the protocol.

Message generation

Once the information has been encoded, it is necessary to generate the message to be broadcasted and simulate its transmission frequency. The proposed model considers two types of broadcasted messages, in accordance with contemplated protocols:

- **Position report message**. This message is generated with a $f_{position}^{protocol}[Hz]$ frequency and includes the encoded position information ($m_p^{encoded}\ \forall\ p\ \in\ P^{protocol}$) and drone identification information. Additionally, some additional conditions might be defined in each protocol to trigger a new position message sending

- **Identification message**. This optional message is generated with a $f^{protocol}_{identification}[Hz]$ frequency and only includes drone identification information (typically the drone serial number although other identifiers are under discussion).

As these processes are periodic and decoupled from the drone status update rate, forwarded information may be outdated. Moreover, MAC protocols may introduce additional delays. To mitigate this delay, some protocols limit the longevity of the forwarded information ($m^{drone}_{time} - \ sendingtime$) to an amount of time $T^{protocol}_{max\ age}[s]$. If this condition is not met, messages are discarded.

As for the message size, it is assumed that the payload of both messages is of equal length: $L^{protocol}_{payload}\ [B]$, measured in bytes ($B$). This simplification is accurate in most of the reviewed protocols as they used a fixed-length frame. Also, it is considered that each message has an overhead of $L^{protocol}_{overhead}\ [B]$ (to include both MAC and PHY levels frames overhead), for a total length of $L^{protocol}_{payload} + L^{protocol}_{overhead}\ [B]$.

Medium Access Control

When multiple users concurrently access a unique shared RF channel and transmit simultaneously, information may be corrupted (situation known as collision) depending on the users' geographical location and timing. Medium Access Control (MAC) techniques and rules are used to minimize these collisions.

The simplest technique consists in listening the channel for other transmissions before transmitting. However, even when listening for other transmissions, collisions might occur due to the propagation delay or geographical distribution of nodes. Let us imagine a situation (depicted in Figure 3.20) in which a principal emitter (emitter 1) wants to transmit information to a receiver in presence of other emitters (emitter 2 and 3). In the figure, the sensing ranges (distances from which they can sense other emitters transmissions) of each node are depicted. The emitters can be classified in four different sets and subsets:

- Emitters that are within sensing range of the main emitter, $S^{emitter}$. These are the emitters for which the main emitter may avoid collisions in some cases (depending on timing because of propagation delay), for all the possible receivers.
- Emitters that are within sensing range of the receiver, $S^{receiver}$. These are all the emitters that can cause collisions in the receiver when concurrently sending information. Two subsets of emitters can be identified here, depending on its relationship with the main emitter:
  - Emitters sensed by the main emitter, for which collisions may be avoided to some extent for this receiver, $S_{sensed} = S^{emitter} \cap S^{receiver}$
  - Emitters not sensed by the main emitter (known as hidden nodes), $S_{hidden} = S^{receiver} - S_{sensed}$, and for which collisions cannot be avoided in any case.

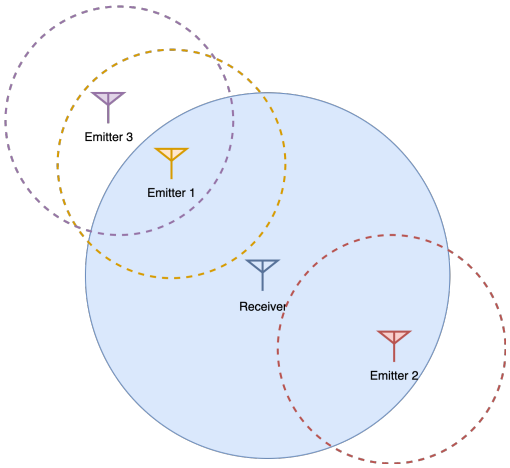


Figure 3.20: Relative visibility of emitters and received in a shared radio medium.
Emitter 2 is a hidden node of emitter 1. Emitter 1 detects and tries to avoid messages from emitter 3. This avoidance would not be needed as emitter 3 is out of the range of the receiver.

Some of the considered surveillance systems use the Carrier Sense Multiple Access (CSMA) MAC protocol [Har02]. The main idea behind this protocol is the previously described "listen before transmit" concept. Some variations of the protocol include a random wait before retrying a transmission (non-persistent CSMA) after the channel is sensed busy, while others transmit as soon as it is free (persistent CSMA). Other surveillance protocols do not implement any explicit MAC protocol: nodes access the channel in a completely unsynchronized manner; this is called the Aloha access mechanism [Abr77]. Therefore, the proposed MAC model must be able to adapt to this variety of MAC protocols. From this discussion, some ideas can be extracted:

- If a CSMA protocol is implemented, there is a probability that the channel is sensed as occupied, and this probability depends among other things on the traffic generated by the nodes in the $S^{emitter}$ set:

$$P_{busy} = f\left(S^{emitter}, \text{traffic characteristics, distance between nodes in } S^{emitter}\right) \quad (3.11)$$

- If a non-persistent CSMA protocol is used, a random wait in the range $(0,\ T_{backoff}\ [s]\ ]$ must be used before sensing the channel again.
- When using a CSMA protocol the collision probability in the receiver is reduced but only for those nodes included in S_{sensed} :

$$P_{collision\ known\ nodes} = f(S_{sensed}, \text{traffic characteristics, distance between nodes in } S_{sensed}) \quad (3.12)$$

- In any case, unknown known nodes (to the emitter) may exist (all nodes if using Aloha) which will yield to collisions with a higher collision probability:

$$P_{collision\ hidden\ nodes} = f(S_{hidden},\ \text{traffic characteristics, distance between nodes in } S_{hidden}) \quad (3.13)$$

These ideas are reflected in the model MAC effect model depicted in Figure 3.21 , which consists in the following sequence:

1. Messages are handled one by one in arriving order, meaning that if a package is waiting for a free channel, the following package waits in a queue.
2. Channel usage is simulated using P_{busy}. If the channel is busy, a random wait can be simulated before sensing again the channel (i.e.: generating a new binary decision). Otherwise, the sending continues.
3. On the receiver end, collisions are simulated using $P_{collision\ known\ nodes}$ and $P_{collision\ hidden\ nodes}$ (by generating random binary decisions with that probabilities). It is assumed that a collision completely invalidates the message (this may not be true in real systems due to possible decoding of overlapping messages in some cases), and it is therefore discarded.

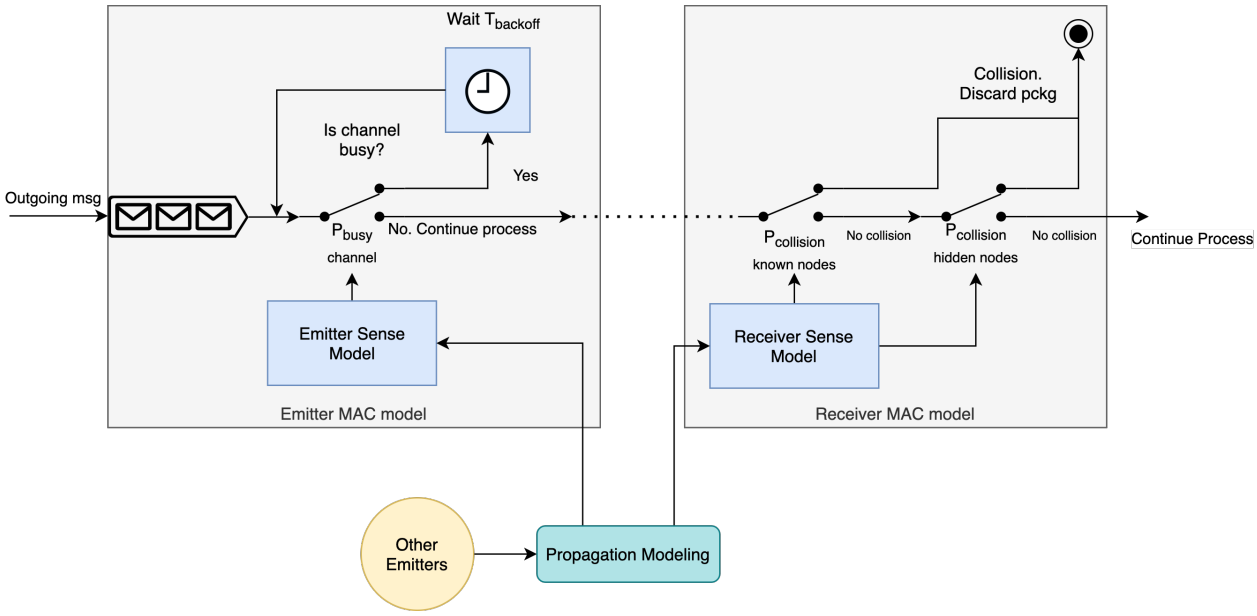


Figure 3.21: MAC model for an emitter and a receiver.
The figure is a block diagram representing the different processes and information flow within the model. In general, each block represents a process or function within the model with arrows representing dependencies or the information flow. Additionally, the clock block represents a synchronous/timed process, and switches represent a decision which depends on the stated condition or probabilistic term. The initial block in the figure represents a message queue for FIFO processing.

The model assumes that each UAS surveillance protocol utilizes an independent communication channel. That means that communications from two different systems do not interact with each other (even though some systems share bandwidth) and that no other external users use that bandwidth. This approximation greatly simplifies the analysis but might reduce the validity of RemoteId and FLARM models, using ISM bands in which other traffic (e.g.: BT,

WiFi...) exist. Considering the previous assumption, the probabilities defined in the model (P_{busy}, $P_{collision\ knownnode}$, $P_{collision\ hidden\ nodes}$) will be specified for each of the surveillance systems in Section 3.2.5.3 depending on the MAC protocol they use. These probabilities will depend on the number of emitters of each of the defined sets ($S^{emitter}$, S_{sensed}, S_{hidden}) which can be dynamically computed in simulation time using the emitters and receivers' position and the physical transmission model.

As for the traffic in each independent channel, some common notation and assumptions (based on [KT75]) can be introduced. It is assumed that the traffic generated by all surveillance emitters can be modelled as an independent Poisson source (this hypothesis is valid for large number of drones, as no time synchronization is expected between them). As previously stated, it is also assumed that all packets are of constant length and that discarded packages are not retransmitted. Under these conditions, an aggregate traffic source for the nodes in each set of nodes S can be considered. This source will have a mean packet generation rate of:

$$\lambda_S^{protocol}\ \left[\frac{pckt}{s}\right] = N_{emitters\ in\ S} \cdot \left(f_{position}^{protocol} + f_{identification}^{protocol}\right) \tag{3.14}$$

Finally, the normalized traffic (G) generated by this source offered to the independent channel can be computed as follows.

$$G_S^{protocol} = \lambda_S^{protocol} \cdot T_{pckt}^{protocol} \tag{3.15}$$

This traffic is the one that the sources generate, not the one that will be finally carried over the network. The real throughput will be lower due to the collisions effect. In the previous equation, $T_{pckt}^{protocol}$ $[s]$ is the packet transmission time which depends on the channel physical capacity or physical bit rate, $R^{protocol}$ $[bps]$, and can be computed as:

$$T_{pckt}^{protocol}[s] = \frac{8 \cdot (L_{payload}^{protocol} + L_{overhead}^{protocol})}{R^{protocol} \cdot FEC^{protocol}} \tag{3.16}$$

where $R^{protocol}$ is the gross bitrate measured at the physical layer, not the achievable throughput at application level considering protocol overhead and encoding redundancy, and $FEC^{protocol}$ is the coding rate used by the protocol in those cases with Forward Error Correction (FEC). That is, $FEC^{protocol}$ is the ratio of actual information over the total bits transmitted due to the additional redundancy included.

3.2.5.2 Collaborative receiver agent

After the message is sent by the emitter agent, it is processed by the receiver agent, which in accordance to what has already been mentioned considers:

- **RF Propagation**. Depending on the propagation environment (urban/rural) and frequency, different attenuations and noise levels are expected.

- **Physical reception**. The receiver requires a minimum signal level to detect the emitter message which determines the receiving range.
- **Medium access control**. Due to the possibility of collisions when concurrently using the RF medium and depending on the MAC protocol, packets may be discarded by the receiver. This element has already been discussed in the previous subsection.
- **Transmission Error Generation and Correction**. The transmission errors are to be simulated within this module considering the digital signal quality. Depending on the capacity of the protocol to correct such errors, messages may also be discarded here by the receiver.
- **Message handling and plot generation**. Receiver may perform additional data processing with the incoming messages such as information filtering. The frequency for which plots are forwarded to fusion nodes is also considered.

In order to **parameterize** such an agent, it is necessary to define its geographical location and the type of collaborative protocol it implements.

Physical transmission, propagation and reception

Although complex physical transmission models could be proposed to simulate messages transmission and reception, this detailed approach is not needed for the purpose of the models. Indeed, the interest is not to reliably reproduce in detail the internal operation of collaborative surveillance systems, but to provide a statistical model that allows to simulate their outputs in order to analyze their impact on U-Space systems.

Therefore, a go-no go model is proposed based on the power balance of the radio link assuming free space losses and omnidirectional antennas. To this end, a transmission power is defined for the emitter depending on the protocol: $P_{tx}^{protocol}[dBm]$. Likewise, the transmission frequency will be defined for each system as $f_{tx}^{protocol}[GHz]$. Each network node also requires an associated position (drone position in the case of the emitter and surveillance sensor position for the receiver) so that the distance between each emitter and receiver can be computed: $d_{e-r}[km]$. With this information it is possible to compute the power in the receiver as [ITU19]:

$$P_{rx}[dBm] = P_{tx}^{protocol} - 20\log d_{e-r} - 20\log f_{tx}^{protocol} - 92.45 \tag{3.17}$$

To decide if a message is received it can be compared with the receiver sensitivity $S_{rx}[dBm]$, defined as the minimum power required to detect an incoming signal. Likewise, emitters also have an associated sensitivity, $S_{tx}[dBm]$ in those cases where a "listen and transmit" (such as CSMA) procedure is used. Therefore, for a signal to be received it must fulfill:

$$P_{rx} \geq S \tag{3.18}$$

Where S is either the emitter or receiver sensitivity as appropriate. However, attenuation is not the only effect suffered by the signal. Notwithstanding the effect of interferences, the antenna receives the signal with an added level of noise. Assuming an Additive White

Gaussian Noise (AWGN) channel, the signal quality can be estimated by the $\frac{E_b}{N_0}$ metric which is a normalized version of the SNR for digital signals. The former can be computed as:

$$\frac{E_b}{N_0} = \frac{P_{rx}[W]/R_{protocol}}{N_0 \left[\frac{W}{Hz}\right]} \tag{3.19}$$

N_0 is the noise power spectral density and can be computed from the overall noise factor of the system, $F_{eq}^{protocol}[dB]$:

$$N_0 = k \cdot T_0 \cdot 10^{\frac{F_{eq}^{protocol}}{10}} \tag{3.20}$$

Where k is the Boltzmann constant ($1.38 \cdot 10^{-23}\ J/K$) and $T_0 = 290\ K$ is the reference temperature. The overall noise factor of the system is dependent on the transmission frequency (different noise sources are predominant for different frequencies) and the receiver quality [ITU21]. The proposed model is summarized below in Figure 3.22

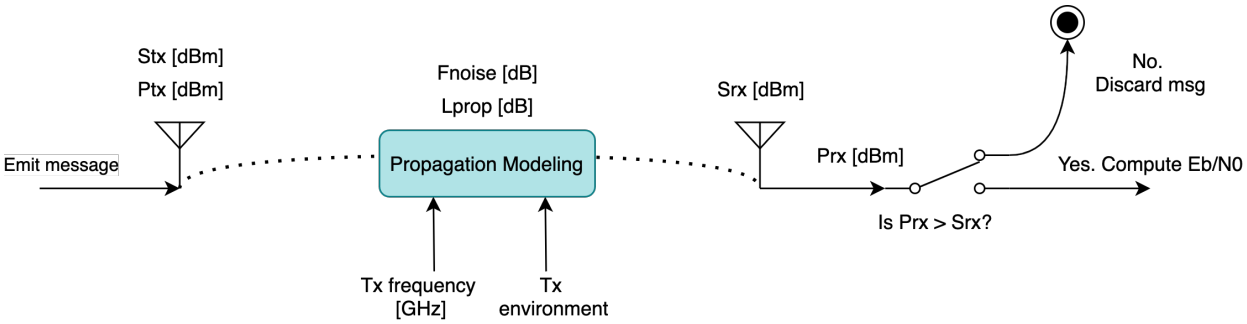


Figure 3.22: Physical description model

The figure is a block diagram representing the different processes and information flow within the model. In general, each block represents a process or function within the model with arrows representing dependencies or the information flow. Add tionally, switches represent a decision which depends on the stated condition or probabilistic term.

Transmission errors and error correction

Additive noise in a digital signal manifests as a set of bit errors that might be recovered with error correction techniques. Otherwise, bit errors might yield a message as undecodable. Although burst errors might also appear due to interferences or multipath effects, only the effect of isolated bit errors will be considered. For a given modulation scheme, which is different for each protocol, it is possible to derive theoretical expressions relating the E_b/N_0 metric with the mean number of expected errors or Bit Error Rate (BER). Thus, it exists a function $b^{protocol}$ such as:

$$BER = b^{protocol}\left(\frac{E_b}{N_0}\right) \tag{3.21}$$

Once this mean error rate is computed, it is possible to statistically simulate the number of errors in a message. To do so, we consider a Bernouilli experiment with BER probability

repeated for each bit of the message (of length $L_{payload}^{protocol} + L_{overhead}^{protocol}$ $[B]$). Overall, the number of errors in the message can be obtained with a binomial distribution:

$$N_{errors} = \mathcal{B}\left(8 \cdot (L_{payload}^{protocol} + L_{overhead}^{protocol}),\ BER\right) \tag{3.22}$$

As all considered protocols include an error detection algorithm, the probability of undetected errors is assumed to be negligible. Therefore, packages with errors (or for which errors cannot be corrected) will be discarded and will not result in corrupted decoding of data. Although some error correction algorithms such as convolutional FEC correct errors within a bitstream and not in an isolated manner, we model the error correcting capacity of a given protocol as a maximum number of correctable errors: $N_{max\ errors}^{protocol}$. Therefore, a message is decodable if: $N_{errors} < N_{max\ errors}^{protocol}$; otherwise, it is discarded. This process is depicted in Figure 3.23 .

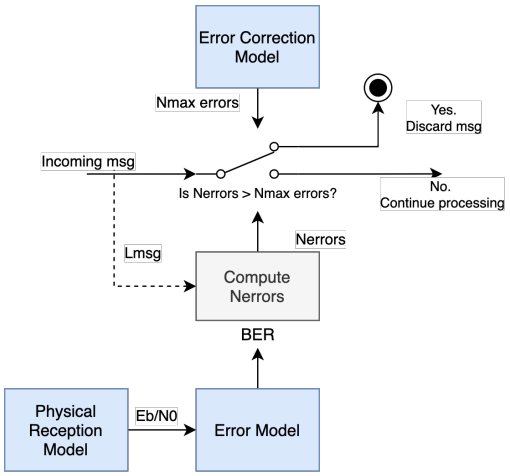


Figure 3.23: Transmission errors and error correction model.
The figure is a block diagram representing the different processes and information flow within the model. In general, each block represents a process or function within the model with arrows representing dependencies or the information flow. Additionally, switches represent a decision which depends on the stated condition or probabilistic term.

Receiver message handling and report generation

Finally, once a successful message reception is simulated, messages are used to maintain an up-to-date list of detected drones and its associated positions. This information: a set of plots, is then periodically reported to a `Fusion node` agent with a frequency of $f_{report}^{protocol}$ $[Hz]$. After further processing in that node to create a list of tracks, drone positioning information is forwarded to the U-Space ecosystem through the Network Simulation service as depicted in Figure 3.18 at the beginning of this Section.

3.2.5.3 Model specification for existing surveillance systems

In this subsection, the general models proposed in the previous subsections for the `Emitter` and `Receiver` agents are particularized for each of the surveillance systems/protocols identified in Section 3.1.5 : ADS-B, FLARM, Remote ID (with different physical layers: BT legacy, BT ER, and WiFi). In this sense, Table 3.10 specifies the value needed for each of the

tuning/configuration parameters that have been previously described. These parameters are extracted from the technical specification of each protocol and, in its absence, uses typical values.

Model Parameter	ADS-B	FLARM	RemoteID-BT legacy	RemoteID - BT ER	RemoteID-WiFi
P	{timestamp, longitude, latitude, altitude, speed}	{timestamp, longitude, latitude, altitude}	{timestamp, longitude, latitude, altitude, speed}		
$r_{timestamp}$ $[s]$	0.0078125	2	0.1		
$r_{longitude}$ [º]	0.0000215	0.000025	10e-7		
$r_{latitude}$ [º]	0.0000215	0.000025	10e-7		
$r_{altitude}$ $[m]$	0.0047625	1	0.5		
r_{speed} [m/s]	0.2315	N/A	0.75		
$f_{position}[Hz]$	0.5	0.33	1		
$f_{identification}[Hz]$	N/A	N/A	0.33		
$L_{payload}^{protocol}$ $[B]$	14	24	25		
$L_{overhead}^{protocol}[B]$	0	9	23	35	21
$T_{max\ age}^{protocol}[s]$	N/A	1	1		
$T_{backoff}$ $[s]$	N/A	0.15	N/A	N/A	0.01
$R^{protocol}[bps]$	1e6	50000	1e6	125e3	24e6
FEC	1/6	1	1	1/4	1/2
$P_{tx}[dBm]$	48.5	14	5	5	11
$f_{tx}[GHz]$	1.09	0.8684	2.4	2.4	2.4
$S_{rx}[dBm]$	-72	-95	-70	-82	-85
S_{tx} $[dBm]$	-72	N/A	N/A	N/A	-85
$F_{eq}[dB]$	10	10	10	10	10
$N_{max\ errors}$	7	0	0	7	7
$f_{report}[Hz]$	1	1	1	1	1

Table 3.10: Particularisation of the collaborative surveillance model for each protocol.

ADS-B specification

In addition to the table parameters, it is necessary to adjust other elements of the models such as MAC protocols and digital encodings. In this regard, ADS-B does not use any MAC control. Within commercial ADS-B deployments for manned aviation, collisions are avoided using spatial diversity, however in this case receivers work in an isolated way. Thus, the probability of collision is that of an Aloha channel, as per [KT75]:

$$P_{busy} = 0; P_{collision\ known\ nodes} = 0 \tag{3.23}$$

$$P_{collision\ hidden\ nodes} = P_{collision\ ALOHA} = 1 - e^{-2G_{Sreceiver}} \tag{3.24}$$

Regarding the BER model, ADS-B uses PPM modulation. In [Pen+12], the relationship between the BER and the E_b/N_0 is defined as:

$$BER = \frac{1}{2} e^{-\frac{1}{2} \cdot \frac{E_b}{N_0}} \tag{3.25}$$

FLARM specification

As for the FLARM standard, it defines CSMA as the MAC protocol. A channel is sensed busy by a given emitter if any other emitter in $S^{emitter}$ transmits in the previous $T_{prop} + T_{pckt}$ seconds. This probability can be computed as (see [KT75] for a detailed analysis):

$$P_{busy} = 1 - e^{-(1+a)G_{Semitter}} \tag{3.26}$$

where a is the normalized propagation delay to the furthest node in each set:

$$a = \frac{T_{prop}[s]}{T_{pckt}\ [s]}; T_{prop}\ [s] = \frac{d_{max}\ [m]}{c}; c = 3 \cdot 10^8 \frac{m}{s} \tag{3.27}$$

The collision probability in the receiver for CSMA-aware emitters can be obtained as:

$$P_{collision\ known\ nodes} = 1 - \frac{e^{-aG_{S_{sensed}}}}{G_{S_{sensed}}(1+2a) + e^{-aG_{S_{sensed}}}} \tag{3.28}$$

Finally, the collision probability for the hidden nodes is that of the Aloha channel:

$$P_{collision\ hidden\ nodes} = 1 - e^{-2G_{S_{hidden}}} \tag{3.29}$$

Regarding the modulation in FLARM protocol (GFSK), as per [LMC12] the BER-E_b/N_0 relation is also:

$$BER = \frac{1}{2} e^{-\frac{1}{2} \cdot \frac{E_b}{N_0}} \tag{3.30}$$

Remote ID specification

For the Remote ID protocol with BT physical layer, no MAC procedure is used [Gha+18], as is the case in ADS-B. Therefore, the collision probabilities depicted for ADS-B in Equation 3.23 and Equation 3.24 are also applicable here. With respect to the BER, GFSK modulation is usually used. Therefore, the expression defined in Equation 3.30 for FLARM is also used here.

Finally considering the Remote ID protocol with WiFi physical layer, CSMA is used according to [Cam+15]. Thus, the expressions derived for FLARM in Equation 3.26 to Equation 3.29 are to be used here. For the BER computation, [CT12] proposes a simplified expression for the QPSK-OFDM modulation used in WiFi:

$$BER = \frac{1}{1 + \frac{T_p}{T_p + T_g} \cdot \frac{E_b}{N_0}};\ T_p = 0.0032;\ T_g = 0.0008 \tag{3.31}$$

3.2.5.4 Collaborative protocol models execution results

To finish with the collaborative surveillance protocols, a simple simulation scenario is proposed in order to verify their operation and to visualize the main simulated effects (e.g.: range, measurements periodicity and collisions due to RF medium access). To this end, 5 `collaborative receiver` agents are placed in a flight field, one for each protocol considered in Section 3.2.5.3 . Then, 50 random linear flights are simulated over this area using the USPASIM framework. Each drone has been equipped with the corresponding `collaborative emitter` agent so that they can be detected by all sensors.

Figure 3.24 shows the plots received by each of the considered sensors so that the different effective ranges of each protocol can be observed. Indeed, FLARM and ADS-B can cover greater areas, whereas RemoteID-based protocols only cover the proximity of the sensor. In addition, if the geographical proximity between the different plots is analyzed, the reporting frequency of each protocol can be verified. Thus, as expected, the Remote-Id protocol plots are closer than in the case of FLARM or ADS-B due to the higher reporting frequency of the first ones. Gaps can also be observed in the trajectories that correspond to missing plots due to collisions or low E_b/N_0.

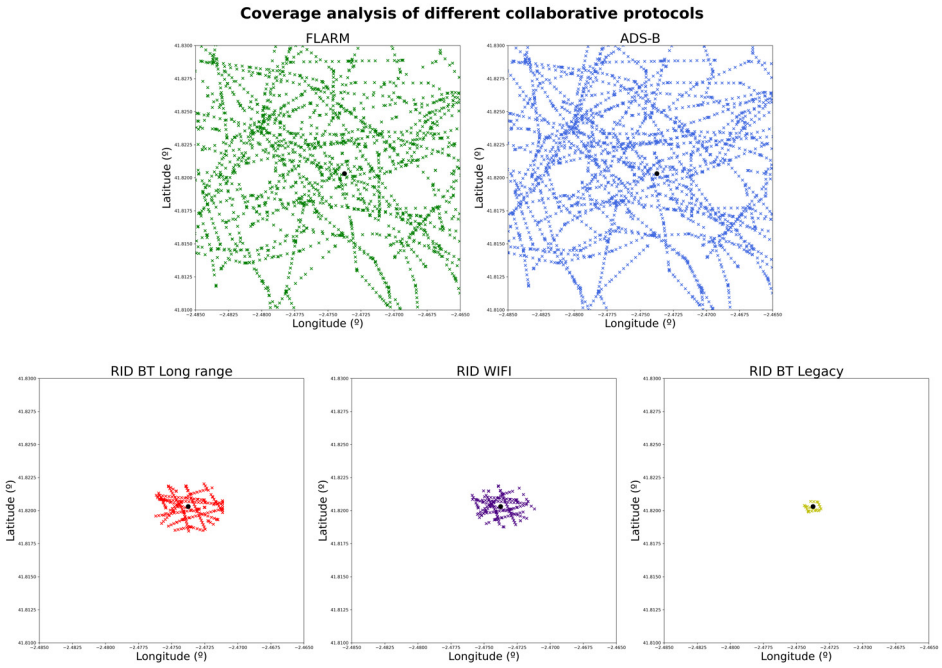


Figure 3.24: Coverage analysis of collaborative protocols under proposed model
Crosses in each plot represent the position of surveillance messages that have been correctly received by each sensor. Sensor location is depicted as a black point in the center of the graph. All subplots share the same dimensions and aspect ratio.

The collisions effect can be analyzed more precisely in Figure 3.25 . The graph shows the number of collisions measured as a function of the number of flights operating simultaneously. In general, as the number of detected drones increases, so does the number of collisions. It might be expected that the performance of ADS-B would be worse than FLARM as it does not use a MAC protocol. However, channel capacity is far greater in ADS-B protocol yielding

to a less occupied channel. The same effect can be observed by comparing RemoteID and Bluetooth's physical layers: the long-range version achieves greater distances at the expense of a lower data rate yielding to higher channel occupancies for the same number of drones. The lower range of the RemoteID protocol is also prone to fewer collisions than FLARM or ADS-B, as for a given drone density it will detect fewer drones. Overall, the results in this subsection demonstrate that the proposed model works as expected.

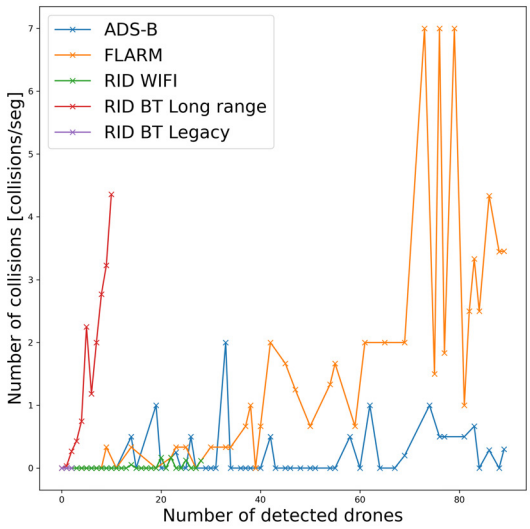


Figure 3.25: Collision analysis for each collaborative protocol under the proposed model

3.2.6 Modeling non-collaborative surveillance sensors

Collaborative surveillance technologies may be complemented by non-collaborative technologies that detect drones by means of the disturbances they generate in the environment. Therefore, sensors such as radars, cameras or acoustic sensors have been modeled within the USPASIM framework. As shown in Figure 3.26 , the modeling of these sensors is based on the definition of the `Non-collaborative sensor` agent. This agent starts from the known position of all aircraft and simulates the detection and position determination process according to the particularities of each type of sensor. As an output, it periodically provides a list of plots to the associated `fusion node` for filtering and distribution to the U-Space system.

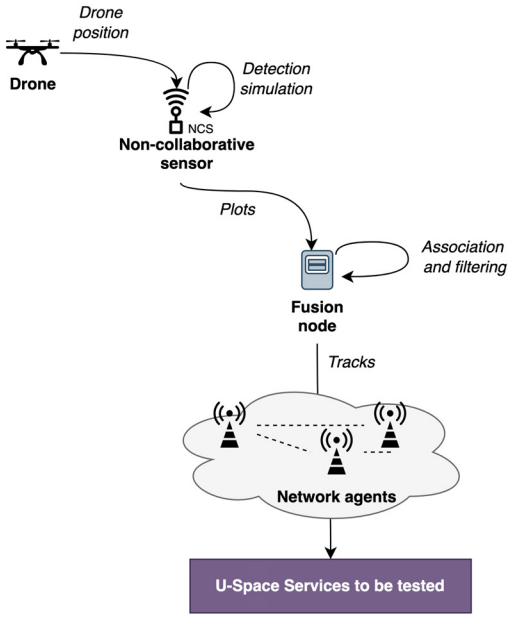


Figure 3.26: Interactions of non collaborative sensor agents with other agent's in USPASIM

As with collaborative technologies, it is possible to propose a general conceptual model of the processes to be simulated within the `non-collaborative sensor` agent, irrespective of the type of sensor. This general model is displayed in Figure 3.27 . Sensors **periodically scan** the environment in order to sense drone-generated perturbations (i.e.: electromagnetic scattering in radar, noise sources for acoustic sensors or light sources for visual sensors). Scanning can be omnidirectional or have **geometrical restrictions** (e.g.: focus on a specific azimuth sector or elevation range). Natural disturbance reception depends on their propagation from their source (i.e.: the radar or the drone itself) to the sensor. Thus, a **physical transmission and reception model** is required to consider phenomenons such as physical transmission, propagation attenuation and sensor's reception capabilities. Furthermore, the reception of the disturbance does not occur in isolation but in a real environment and is therefore subject to noise and other interferences. Consequently, an Signal to Noise Ratio (SNR) can be measured as a figure of merit of how easy it is to discriminate the disturbance caused by the drone from the environmental background level. Based on this value, the sensor will have a **stochastic detection process** that will discriminate whether a disturbance really corresponds to a drone. If so, the sensor will determine the position of the target (or the orientation relative to the sensor position). In this **measurement generation process**, the sensor model must encompass for measurement errors and inaccuracies.

In addition, during the detection process, it is possible that an environmental disturbance is mistakenly marked as a real target. This is known as a false alarm and should also be simulated. To do this, it is necessary to **estimate the number of false alarms** generated by the sensor, which will be related to the detection threshold set by the sensor. Subsequently, it will be necessary to randomly generate them within the range of the sensor and to add them to the output plots that are periodically sent to the `fusion node`.

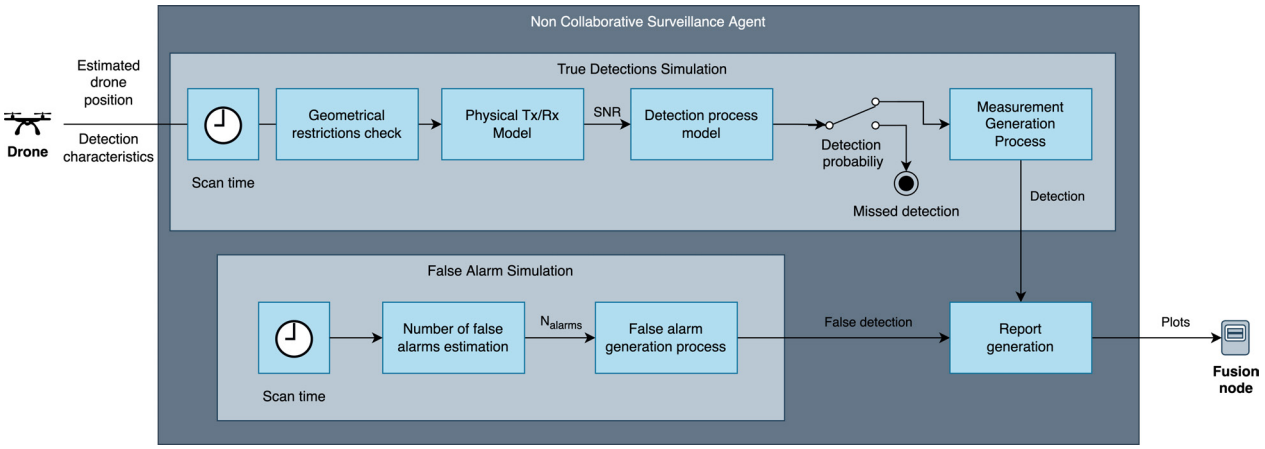


Figure 3.27: General model of a non-collaborative sensor

For the purposes simulating the U-Space ecosystem, a detailed simulation model for each of the previous processes is not required. Instead, it is just needed to roughly estimate the sensor's ranges along with the position determination errors. To this end, representative models are extracted from the literature, adapted to drone detection and integrated into the USPASIM framework. Specifically, Section 3.2.6.1 particularizes the previous general model for the simulation of active radars, proposing equations for each process and suggesting typical values for radar parameters. Then, Section 3.2.6.2 provides pointers to available literature that could be adapted to integrate other types of sensors (i.e.: passive radars, MIMO radars, EO/IR sensors...). Finally, Section 3.2.6.3 exposes some execution results to assess the proposed radar model.

3.2.6.1 Active radar sensor model

For the simulation of an active radar sensor, a monostatic (i.e.: transmiter and receiver are colocated) radar will be considered. The model is characterized by the following **parameters**, summarized in Table 3.11 :

- Radar's measurements consists of range, azimuth and elevation information.
- Radars emits a power of P_{tx} for a maximum theoretical range of R_{max}. Also, the radar only emits in an azimuth coverage between θ_{min} and θ_{max}.
- Both transmission and reception antennas consist in a single beam with a parameterizerable beamwidth in azimuth and elevation (i.e.: $\theta_{3dB}^{az/ele\ -\ tx/rx}$) which yield to G_{tx} and G_{rx} gains.
- Radar parameters are adapted to drone's detections: integration times (T_{int}), exploration times (T_{scan}) in the order of one second and range resolution (Δ_{dis}) in the order of meters.
- The radar is able to eliminate clutter through doppler filtering.. Also, the detector is assumed to be Cell-Avering Constant False Alarm Rate (CA-CFAR) with a false alarm probability of p_{fa} and N cells.

- Free-space propagation losses are assumed to be predominant. Moreover, a system losses parameter L_S is also considered to model feeder losses, CFAR detection losses, doppler filtering...
- Thermal noise is assumed to be predominant and aggregated in a single F_a noise factor.

Parameter	Description	Typical / example values	Unit
R_{max}	Instrumental (maximum) range of the radar.	10	[km]
$\theta_{min}, \theta_{max}$	Azimuth coverage. Range of azimuth angles that are explored by the radar.	(-180, 180)	[º]
f	Central transmission frequency of the radar. Also expressed as the radar's wavelength $\lambda = c/f$	10	[GHz]
P_{tx}	Radar transmission power	500	[W]
Δ_{dis}	Distance resolution. Obtained from the radar's bandwidth (B) as $\Delta_{dis} = \frac{c}{2B}$	10	[m]
$\theta_{3dB}^{az-tx}, \theta_{3dB}^{ele-tx}$	Azimuthal and elevation width of the transmission pattern. To be represented in radians to compute antenna gain.	360	[º]
$\theta_{3dB}^{az-rx}, \theta_{3dB}^{ele-rx}$	Azimuthal and elevation width of the reception pattern. To be represented in radians to compute antenna gain.	2 6	[º]
p_{fa}	False alarm probability.	0.00002	N/A
T_{int}	Dwell / integration time.	0.06	[s]
T_{scan}	Exploration time. i.e.: time required to scan the whole coverage area.	1	[s]
F_a	Radar receptor noise factor.	4	[dB]
L_s	Aggregated radar power losses. To be converted to natural units in the radar equation.	5	[dB]

Table 3.11: Parameters of active radar sensor model and example representative values
Example values have been extracted from C-UAS radar in [HEN20]

Following the general modeling scheme outlined in Figure 3.27 , the process of simulating detections would be repeated periodically every T_{scan}. At each scan, the process would start by filtering all possibly detected targets based on the sensor's coverage (both in terms of range and azimuth).

Physical Tx-Rx model

Then, for each target within theoretical coverage, the usual radar equation is considered to compute the Signal To Noise (SNR) ratio at the receiver. As described in [Bar04], it may be

computed as:

$$SNR = (P_{tx}G_{tx}) \left(\frac{\sigma}{4\pi d_{d-r}^2} \right) \left(\frac{G_{rx}\lambda^2 T_{int}}{(4\pi)^2 d_{d-r}^2 L_S} \right) \left(\frac{1}{N_0} \right) \tag{3.32}$$

where the first term is related to the radiated power of the radar, the second term with the fraction of powered reflected by the drone target, the third term with the power received by the radar, and the final one with the noise level. Reflected power depends on each target's RCS (σ) which is usually in the range of $0.01 - 0.1\ m^2$ and the one way drone-radar distance (d_{d-r}).

Transmiter and receivers gains can be approximated from the beamwidth as follows (with an ideal antenna efficiency ($\eta = 1$), although other more realistic values could be considered):

$$G_{tx} \approx \frac{4\pi}{\theta_{3dB}^{az-tx} \cdot \theta_{3dB}^{ele-tx}} \tag{3.33}$$

$$G_{rx} \approx \frac{4\pi}{\theta_{3dB}^{az-rx} \cdot \theta_{3dB}^{ele-rx}} \tag{3.34}$$

Also needed in the radar equation is the noise power in natural units which can be derived from the system noise factor (where k is the Boltzmann constant and T_0 is the Earth temperature which is typically 290 K):

$$N_o = k \cdot T_0 \cdot 10^{F_a/10} \tag{3.35}$$

Detection process model

Once the SNR of a target at the receiver is computed, it is possible to assess whether it is detected by the radar or not. To this end, it is assumed that the target behaves as a Swerling I target (as usually considered to model aircrafts) and that the noise residual has a Gaussian distribution. Under these conditions, the detection threshold (α) of the CFAR with a false alarm rate of P_{fa} can be derived as follows [GK88]:

$$\alpha = (P_{fa})^{-1/N} - 1 \tag{3.36}$$

where N is the number of considered cells. From the threshold, the individual detection probability of a given target (P_{det}) can be computed as:

$$P_{det} = \left(1 + \frac{\alpha}{1 + SNR} \right)^{-N} \tag{3.37}$$

Therefore, target detection is dependent on the result of the following random experiment in which a random uniform number is compared against the previous detection probability:

$$U(0,1) < P_{det} \tag{3.38}$$

Measurement generation process

For those aircrafts that have been successfully detected, a noisy measurement is to be generated around the true target position. To this end, a zero-mean gaussian noise is added in the radial and tangential directions from the radar's point of view. This implies a change from Cartesian coordinates (in which the real aircraft position is known) to spherical coordinates. By assuming that the SNR dependent measurement error is predominant, the standard deviation of each error distribution can be computed as described in [Bar04] for optimal range, azimuth and elevation estimators:

$$\sigma_d = \frac{\Delta_{dis}}{1.63 \cdot \sqrt{2 \cdot SNR}} \tag{3.39}$$

$$\sigma_{azi} = \frac{\theta_{3dB}^{azi-rx}}{2 \cdot \sqrt{SNR}} \tag{3.40}$$

$$\sigma_{ele} = \frac{\theta_{3dB}^{ele-rx}}{2 \cdot \sqrt{SNR}} \tag{3.41}$$

False alarm generation process

To conclude, the generation of false detections (false alarms) should be considered. In this sense, the number of false alarms in each scanning cycle can be modeled as a binomial random variable of mean, N_{alarms}:

$$N_{alarms} = p_{fa} \cdot \frac{R_{max}}{\Delta_{dis}} \tag{3.42}$$

As for the generation of the false measurements itself, random positions are randomly generated within the radar's theoretical coverage (i.e.: limited by $R_{max}, \theta_{min}, \theta_{max}$).

3.2.6.2 Other sensor models

The above model for a monostatic active radar can be extended to consider MIMO radars or passive radars as described in [Bes+22a]. For MIMO radars with multiple transmitters, the received power should be computed by averaging the received power from each of the transmitters. The individual power could be obtained from Equation 3.32 considering the bistatic situation to compute the distance. Also, detection is performed with incoherent pulse integration. Thus, the detection probability expression changes as explained in [DR04].

For passive radars, the radar equation should be adapted to consider the effect of interference from the direct opportunity signal-radar path and the radar's capability to eliminate it (related to the antenna sidelobes gain). In this case, elevation information cannot be obtained and the measurement accuracy expressions should be changed for those in [Mal19].

Acoustic sensors could also be modeled by adapting the general model. In this case, instead of electromagnetic powers the model should consider acoustic powers, adapt to the sound propagation conditions (e.g.: atmospheric attenuation, divergence attenuation...) and to

the directionality of the microphones. However, no closed expressions for the calculation of detection probability or measurement errors have been found in the literature. A similar adaptation would be possible in the case of EO/IR sensors taking into account the received light and the contrast between the drone and the environment, following the points made in [KNT13]

3.2.6.3 Non collaborative surveillance models execution examples

As in the previous cases, a simple simulation scenario is used to verify the effects modeled within the `non collaborative surveillance` agent with the active radar model. The scenario consists of simulating the flight of a drone directly approaching the sensor site from a distance greater than 7 km. The trajectory followed by the aircraft is drawn in a dashed line in Figure 3.28 where the sensor position is depicted as a solid black circle. As expected, the plots detected by the sensor (blue crosses) show that at the beginning of the trajectory (far from the sensor) the detections are discontinuous and very noisy. In addition, a set of scattered plots throughout the coverage area appear, corresponding to generated false alarms.

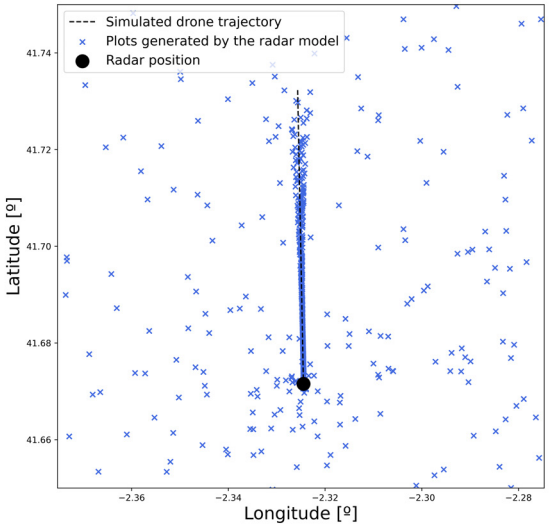


Figure 3.28: Plot distribution analysis of a drone detected by radar sensor.

Finally, Figure 3.29 shows an analysis of angular and range errors relative to the target-sensor distance. Errors follow the typical distribution as defined by the inverse relationship with the SNR square root.

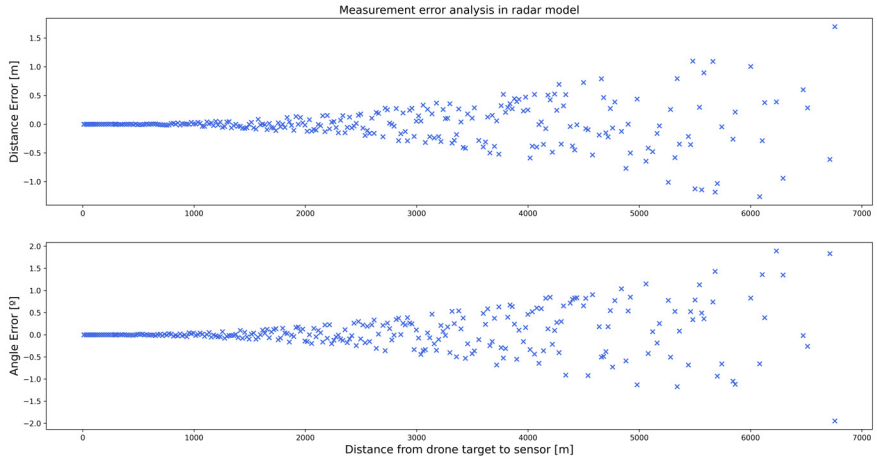


Figure 3.29: Distance and error analysis of drone plots as detected by modeled radar sensor.

3.2.7 Modeling surveillance networks

Models presented in Section 3.2.5 for collaborative protocols and in Section 3.2.6 for non-collaborative sensors provide isolated detections or measurements of aerial targets, so far named plots. However, for the information to be consumed by the U-Space system, it is necessary to ensure information continuity and stability through time in a process known as **tracking**. This process associates consecutive sensor plots for the same target into tracks: an integrated and time-continuous cinematic report of a given aerial target. In addition, as part of tracking, information filtering algorithms are applied that are able to smooth errors and eliminate biases.

Moreover, Section 3.1.6 has identified the benefit of integrated surveillance sensors that fuse information from different type of sensors (e.g.: an optical sensor integrated with a radar). In this case, a process known as **association** is needed, which relates tracks from different sensors belonging to the same target. Then, associated tracks are fused together to form a single multi-sensor track per target. In this way, tracks from a network of sensors can be integrated to provide a single source of information to the U-Space system.

Simulation of the tracking, association and information fusion processes are carried out in the **`Fusion Node`** agent depicted in Figure 3.30 . This agent, of which there may be several instances to consider different sensor networks or simulate integrated sensors, periodically receives plots from a set of linked sensors. Then, it performs the aforementioned processes and provides as an output a set of tracks that fuse the received information.

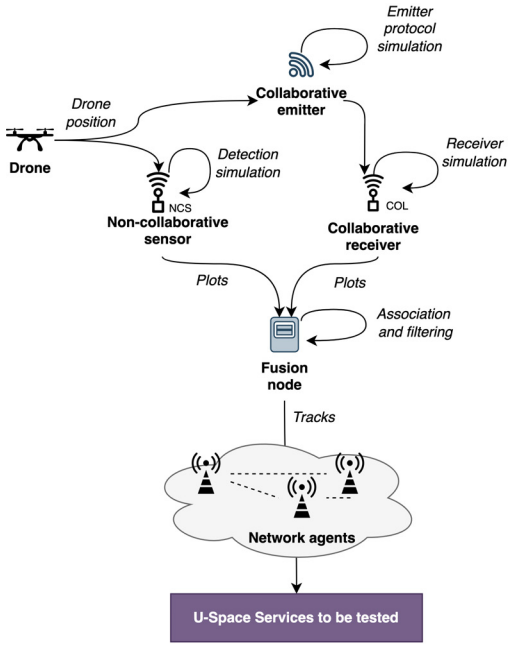


Figure 3.30: Interactions of surveillance node agents.

As explained above, the sequence of processes carried out in the `Fusion Node` agent are those described in Figure 3.31 . In a simplified way, two criteria have been considered in the temporal correlation and track association processes:

- Distance criterium. If the information source is not collaborative, a distance criterion can be used to associate a plot to its nearest track. Other more advanced approaches such as the Probabilistic Data Association Filter in [BDH09] might also be used.
- Identification criterium. If the information source is collaborative, plots can be linked by means of the corresponding drone identifiers that are sent within the collaborative protocol messages.

As for the plots tracking, the filtering scheme proposed in [Cam20], which consists in the usage of Kalman filters, has been used. Finally, for the track fusion stage, bayesian ponderation has been implemented [BWT11].

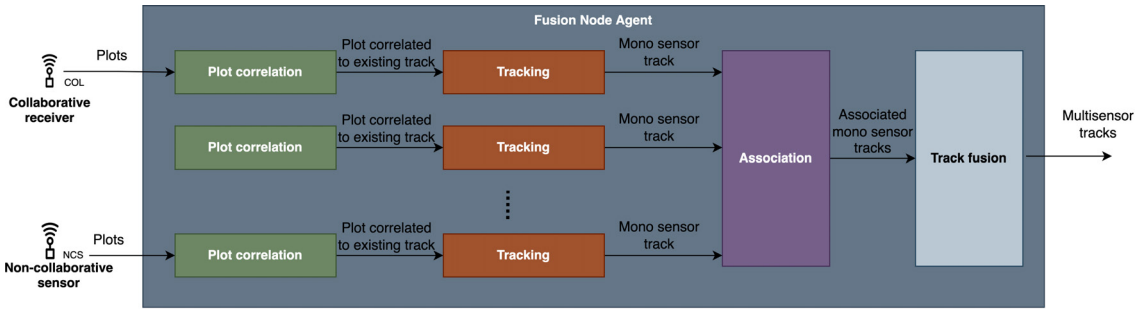
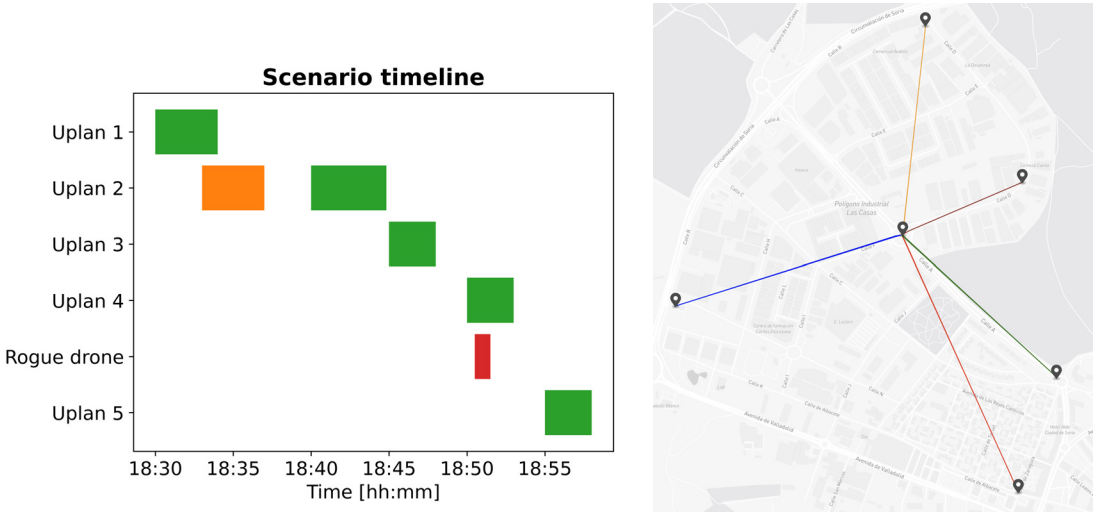


Figure 3.31: Fusion node agent processes

3.2.8 USPASIM execution example

As a summary, this section has identified the need of providing simulation tools to validate and develop U-Space systems. In this sense, the USPASIM framework has been presented in Section 3.2.2 . Next, contributions have been developed to model two elements that are not usually considered in the literature. Particularly:

- Human intervention in both the strategic phase (with the `Operator` agent described in Section 3.2.3) and the tactical phase (with the `Pilot` agent described in Section 3.2.4).
- Surveillance networks with both collaborative (Section 3.2.5), non-collaborative sensors (Section 3.2.6), or an integration of them (Section 3.2.7).



(a) Flights schedule **(b)** Trajectories of authorized flights

Figure 3.32: Simulation scenario description
In scenario timeline, green areas represent the maximum duration of accepted u-plans. Orange represents u-plans with a detected conflict that have been deconflicted. Red represents non-authorized operations.

As an example of these contributions' functionality, the simulation scenario depicted above, in Figure 3.32 , is proposed. It consists of:

- A delivery operator which is to perform 5 last-mile delivery operations over the city of Soria, under supervision of the U-Space ecosystem (i.e.: collaborative operator). Timing and trajectory information of the automatically generated flights is depicted in Figure 3.32 . There, it can be observed how flights are round trip from a vertiport to the delivery location. Flights are manually controlled by human pilots.
- A recreational operator intends to fly in the same area as the delivery operator. Unaware of existing regulations, the operator does not interact with the U-space ecosystem neither in the strategic phase (i.e.: it does not submit a u-plan for authorization), nor in the tactical phase (i.e.: it does not forwards telemetry information). The schedule of this operation is depicted in red in the timeline displayed in Figure 3.32a
- The operational area is surveilled by a collaborative Remote-ID BT long range receiver

and a non-collaborative radar sensor. The position of both sensors is the same as the departure point of the delivery operations. Delivery operator's drones are equipped with a Remote-ID BT Long Range emitter.

Once the scenario has been defined (using the automatic traffic generation models described in Section 3.2.3.2), it is simulated in both the strategic and tactical phases. In the former phase, the `Operator` agent corresponding to the delivery operator submits flights for authorization. Authorization results are depicted in the USSP's interface showcased in Figure 3.33 . In accordance with the scheduled timeline and following the regulation-mandated First Come-First Served priority, the original definition for *flight_2* has been denied due to a conflict with *flight_1*. However, an alternative operation has been provided by the USSP delaying the execution of the flight. This alternative has been resubmitted by the operator and finally accepted. Overall, 5 operations by the delivery operator have been accepted. Also, it can be observed how there is no record of any additional operations by the recreational operator.

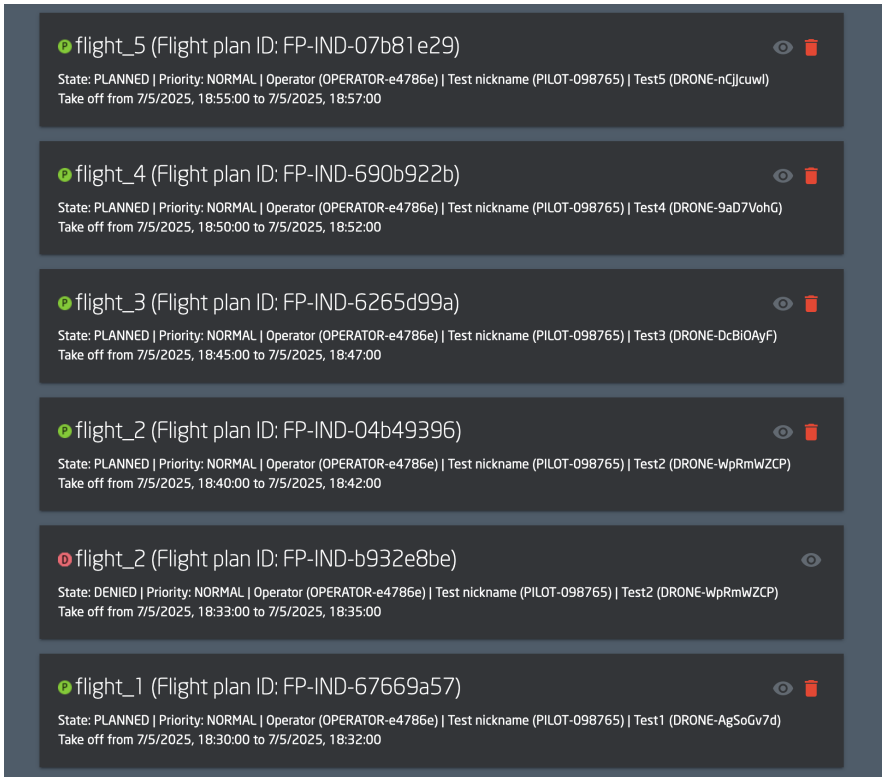


Figure 3.33: Authorization results as depicted in the connected USSP.
Note in the left side of each card the authorization result. Also, note how two *flight_2* entries are visible due to the deconfliction process

In flight time, USPASIM framework reproduces aircrafts trajectories as a result of the human control model in `Pilot` agents, as explained in Section 3.2.4 . In parallel, the U-Space system is able to track all collaborative aircraft thanks to the simulated network identification process (i.e.: telemetry forwarding). Reported aircraft positions have been depicted in Figure 3.34 . There, it can be seen how aircraft trajectories slightly deviate from the authorized u-plan due to the manual control error simulated in the platform.

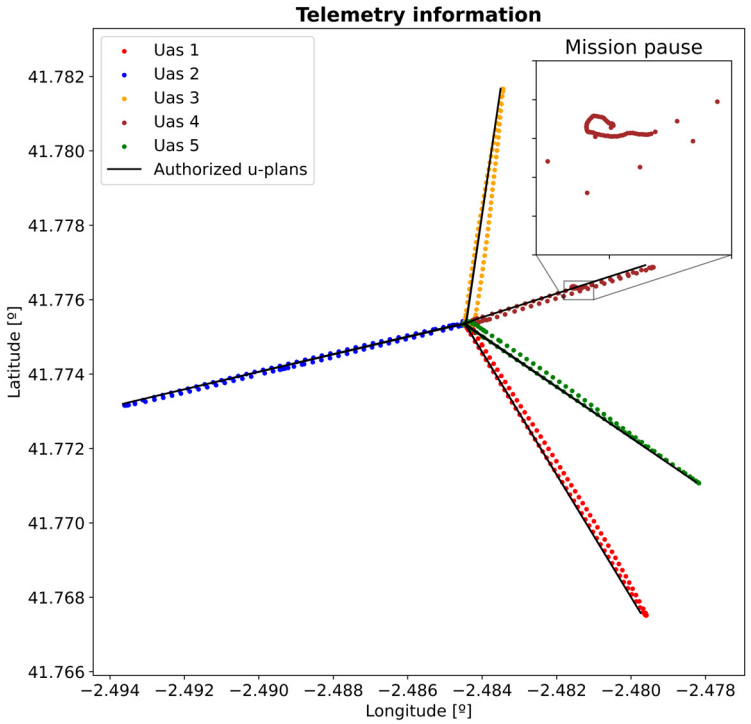


Figure 3.34: Telemetry information forwarded to the USSP by collaborative drones.
Each colored point corresponds to a position report by an aircraft (color code). Black lines represent the expected accepted u-plan.

Besides the network-id protocol, positioning information is also received through the surveillance network formed by `Receiver` and `Emitter` agents for the Remote Id protocol, and a `Non-collaborative sensor` agent for the radar. Tracks forwarded by these agents are depicted in Figure 3.35 . In the figure, different coverage patterns are observed for each technology (e.g.: remote id protocol is not able to track the whole flights) as a result of the models described in Section 3.2.5 and Section 3.2.6 . Also, it can be noted how radar measurements are noisy and deviate from the real aircraft trajectory depicted in the previous figure. Moreover, false alarms detections from the radar sensor can also be observed.

Integrating a non-collaborative sensor allows the U-space ecosystem to track rogue aircraft (e.g.: the recreational operator flight). Indeed, in the zoomed inset of Figure 3.35 detections that do not match with the authorized operations are observed. This way, it is possible to predict a loss of separation between a delivery aircraft (i.e.: u-plan 4) and the recreational rogue operation. As a result, the u-space system sends tactical alerts to nearby operations informing them of the hazard. Within the simulated framework, the `Pilot` agent controlling the collaborative aircraft reacts to the alert in a close-loop to avoid a collision. Following the models described in Section 3.2.4.1 , mission execution is paused until the alert is cleared. This is observed in the zoomed inlet of Figure 3.34 , where a set of very close positions are reported, corresponding with the aircraft hovering with a slight drift.

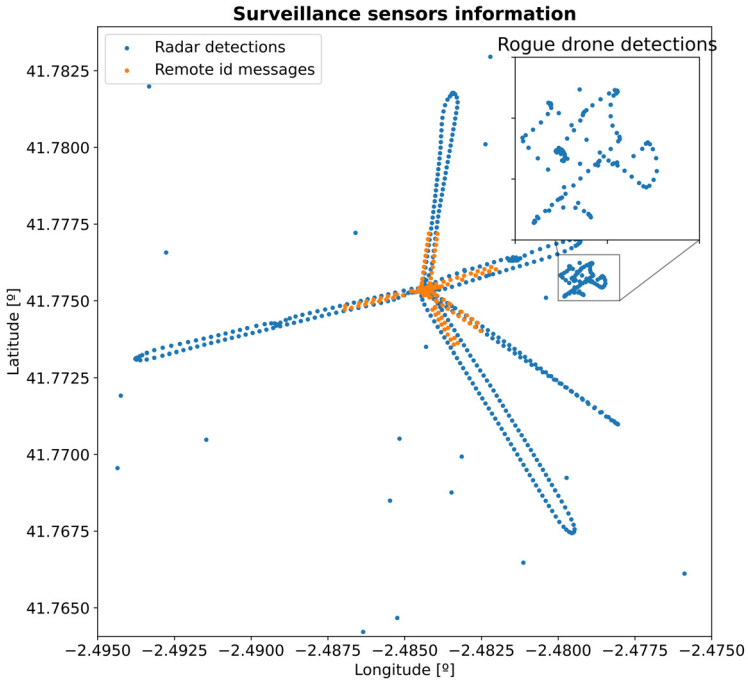


Figure 3.35: Tracks forwarded to the USSP by surveillance sensors.

Overall, the example has demonstrated the operation of the proposed agents and how the integration of this section's contributions into a comprehensive simulation framework helps to **realistically reproduce complex closed-loop simulation scenarios**. In addition to the evaluation of existing u-space systems, the presented contributions will also allow to evaluate future u-space services including the integration of surveillance networks or tactical conflict resolution algorithms.

3.3 Managing fairness and congestion: a token-based framework for U-Space authorization

As previously explained in Section 3.1 , U-space separation management currently implies that an airspace 4D volume cannot be simultaneously reserved for two UAS operations. Therefore, from the operators' point of view, **airspace is a contested resource** whose access is necessary to achieve their business plan. Additionally, European regulations also stipulate that available capacity must be allocated to operators on a First Come-First Served (FC-FS) basis [Eur21] [EAS22a]. This term implies that those operators who request the usage of a given airspace volume (through the submission of a u-plan) in the first place, are those who are authorized to fly. While this flight authorization approach may be valid for the time being when airspace is uncongested, it will generate problems in the future when operators start to compete to access congested airspace. It is therefore necessary to analyze the implications of this approach and propose suitable alternatives.

In Section 1.2 , the heterogeneity of unmanned aerial operations has been described. As a result, UAS operators' needs can be very different depending on the characteristics of their operations [SFD22]. In this sense, some operators can schedule their operations in advance (e.g.: agricultural inspection) while others cannot do so until just before the moment of take-off (e.g.: urban delivery operations). As a result, with a FC-FS authorization process, the former operators could monopolize airspace usage over the later operators. Additionally, each operation yields a value for the operator depending on its criticality for its objectives or business logic (e.g.: a monetary value). Accordingly, some operations can be subject to changes (i.e.: route changes, delays) at little cost to the operator (e.g.: parcel delivery) while others have strong restrictions on when and where they must be performed (e.g.: photoshoot of a relevant event). These operators' needs are also not considered in the regulation-mandated FC-FS approach and can generate inequalities in airspace access.

One of the objectives to be considered in the design of U-Space systems is to ensure that **access to airspace is "equitable and fair" for all operators**, irrespective of their different needs [COR23]. In this sense, fairness may be understood as the virtuous state in which the welfare of each operator is maximized subject to capacity constraints and the needs and limitations of the other operators [Sac+20]. From all of the above, it is evident that the current logic of airspace access (i.e.: FC-FS), where the first operator to report the operation is the one who gains access to the airspace, does not meet this fairness criterion. This means that operators can take advantage of the FC-FS logic to authorize a greater number of operations or being subjected to fewer deconfliction changes, such as delays. In fact, simulated studies [EEM20] show that when operators with different behaviors (e.g.: different flight submission times) are introduced into the existing U-Space systems, the achieved fairness is reduced.

Authors in [SFD22] also argue that the existing authorization mechanism is not appropriate. As a solution, they discuss a series of requirements that should be considered in an alternative protocol: efficiency (airspace should be used efficiently to avoid congestion), fairness (for which different notions of what is fair can be selected by the regulator), simplicity (operators should easily report their preferences hiding the complexity of the underlying fairness optimization logic), incentives (operators should be encouraged to truthfully communicate their preferences.)

and scalability (the mechanism must support the expected high number of operations).

From a practical point of view, fairness issues in airspace access will appear hand in hand with airspace congestion. Demand and capacity management protocols are expected to set congestion thresholds and propose measures that homogenize geographical and time distribution of traffic to avoid hotspots [DAC22]. Also, politically imposed constraints related to social (e.g.: privacy) or ecological aspects (e.g.: noise pollution) are to be expected. Ideally, an alternative authorization mechanism should also be able to deal with these aspects in conjunction with fairness considerations.

Overall, it seems reasonable to conclude that **a new airspace access control mechanism is needed to address the challenges of the future drone ecosystem** (i.e.: fairness, capacity management, regulatory constraints) while ensuring elements such as efficiency, scalability, and simplicity. In this sense, the literature review in Section 3.3.1 indicates that there does not exist a comprehensive authorization framework that considers fairness and is easily adapted to European regulations. As a solution, this section proposes an enabling mechanism that supports enforcing fairness criteria based on:

- The definition of an alternative strategic flight authorization / deconfliction workflow that is based on the current European regulations and proposes a gradual evolution by implementing a deferred and multi-priority authorization scheme.
- The definition of a token-based framework that guides the authorization process enabling operators to state the expected value of each operation while authorities/U-space services can establish temporal, geographical and operator-linked incentives to implement different notions of fairness, congestion, or other constraints.

The proposed mechanism, as well as its two possible versions (i.e.: basic token authorization and geographically-aware token authorization) and its integration into the U-Space ecosystem are discussed in Section 3.3.2 .

Regarding the fairness notion (i.e.: what is fair) to be enforced, the literature review showcases different conceptions. In this sense, it is not the objective of the section to decide what that fairness notion (and its associated incentives/implementation) should be. Instead, the focus is on providing U-space stakeholders (i.e.: national regulators, USSPs and operators) with an easily implementable and transparent framework that allows the implementation of a broad set of protocols to optimize different notions of fairness or social restrictions. Thus, the proposed framework allows to jointly manage fairness and congestion while enabling operators to express their operation preferences in a privacy-aware way. To prove this, Section 3.3.3 describes how the framework can be particularized to implement different notions of fairness. Moreover, feasibility and effectiveness of the proposal is tested on a set of representative simulation experiments carried out with an adapted version of the USPASIM framework which is described in Section 3.3.4 . Finally, Section 3.3.5 lays out and discusses obtained results.

The proposals on this section are a thoroughly extended version of the proposal of this author on the following publication.

Supporting publications

- **[Car+24b]** - David Carramiñana, Juan A. Besada, José M. Molina, and Ana M. Bernardos. "Scalable U-Space Authorization Workflow Enabling Fairness and Capacity Management". In: *SESAR Innovation Days 2024*. Rome, Italy, 2024 where the author is the main contributor of the work including the design of the alternative authorization workflow, the integration of fairness criteria and the validation set up.

3.3.1 Review on fairness and capacity management methods

The flight authorization process in U-Space can be considered as resource allocation problem in which 4D airspace volumes are reserved to authorized operations. In this sense, fair resource assignment has been previously studied in domains such as communication networks [SHI+14], cloud computing [SK19] or even healthcare [Lan+17]. When addressing fairness in such systems, two topics should be contemplated: **measuring fairness** (i.e.: what is considered fair) and algorithms/protocols to implement **fair resource allocation**.

3.3.1.1 Fairness metrics

On the first topic, different notions of fairness are possible such as min-max or proportional fairness [BFT11]. In the first case, an allocation is fair as long as an increase of resources for one participant does not result in a decrease for another participant with a smaller resource share. That is, the utility of the worst user is to be maximized. On the contrary, in proportional fairness, assignments are made to ensure that each individual gets a proportional share of their total possible allocations. This way, a compromise is reached between system efficiency and egalitarian fairness (i.e.: max-min fairness). To measure if these fairness notions (or others) are met, fairness metrics must analyze the variability in the distributed resources among the set of demands [BT20]. In this sense, and given a magnitude that measures how many resources have been allocated to each individual, simple statistical measures over that magnitude such as the standard deviation can be used to represent resource distribution (i.e.: the higher the standard deviation the less fair the allocation has been). More complex metrics can also be derived to obtain a normalized figure of merit such as the ratio of the geometric and arithmetic mean [dVG09] or the Gini coefficient.

Focusing on measuring fairness in UTM systems, a thorough discussion can be found in [Sac+20] which warns that defining a single airspace usage fairness notion is difficult due to different benefit perceptions across stakeholders (mainly operators). For instance, in [Chi+21a], if a FC-FS allocation is assumed to be fair (which is debatable as it has been previously discussed), the number of arrival reversals (i.e.: the deviation of the arrival order vs. FC-FS) can be considered as a fairness metric. In [Chi+23b] and [TX23] , fairness is studied in a UAM setting where operations are delayed (either in ground or in air) to maintain separation. As a result, incurred time delays are used as a metric to compare how different operations are treated. Literature also discusses how fairness can be measured at an operator level by measuring the overall delay across all operations such as in [Chi+23a].

3.3.1.2 Fair resource allocation

As for the second topic (i.e.: ensuring fairness), fair resource allocation usually consists in solving an optimization problem (e.g.: mixed integer linear programming) to which fairness constraints are added, as in the case of [BA15]. However, in systems with scalability or real-time requirements (e.g.: communication networks or job allocation in computers) the resolution of these problems is intractable, and one has to resort to approximate models and heuristics. Thus, techniques such as round-robin scheduling in processors might be used and result in max-min fairness under certain conditions [Hah91]. Also, techniques such as fair queuing or token buckets are used in networks to provide differentiated traffic allocation for different groups (even achieving different qualities of service) [Dor20]. Focusing on UTM systems, existing literature on fairness management is scarce and can be grouped in three different approaches: **market-based, based on optimization techniques and distributed mechanisms**.

Authors in [MBB21] perform an economic study on the possible adoption on a money-based market that is similar to road pricing but managing airspace usage. They argue that a pay-per-usage (i.e.: operators pay a fee depending on the airspace occupation of each flight) scheme is equitable and efficient and should be adopted. Continuing with proposals based on a market supported by money, the author of [Sko18] concludes that portions of airspace should be delimited and allocated exclusively to some operators based on an auction process. Thereby, operators would not pay per flight, but for the right of unlimited and unique usage of some air corridors. A technical implementation of an airspace auction is also proposed in [LMP24]. Authors articulate pre-departure strategic deconfliction through bids on the set of airspace cells required for a flight-plan. Within the framework, airspace price and bids are shared between operators through a common, centralized price heatmap. In [SFD22], authors conceptually discuss the possibility of introducing a market with an artificial currency (or token) that would be distributed to operators based on how their behavior aligns with the regulator's notion of fairness. Such point-based system is also proposed in [DAC22] to decide which flight plans are subjected to changes in a congestion management protocol. However, no comprehensive proposal in terms of the strategic flight authorization (i.e.: considering flight plan authorization timing, services interactions...) is discussed in any of the previous cases. Similarly, no discussion is made regarding how such a scheme could be technically implemented and whether it is possible to distribute such schemes within the U-Space ecosystem.

In addition to the above examples that articulate airspace access as a market, there exists another set of authors that propose introducing optimization techniques within the authorization loop to simultaneously solve congestion and fairness issues. In this lane, authors of [Chi+20] propose the usage of an adapted version of ATFM (Air Traffic Flow Management, a congestion management algorithm envisioned for conventional aviation) which consists in formulating a mixed-integer programming problem to assign delays to UAS aircraft in order to ensure separation between them. The proposed algorithm handles simultaneously a batch of operations and tries to ensure their separation while maximizing different fairness metrics. This work is further refined in [Chi+21b] by introducing operator preferences (e.g.: delay sensitivity) into the ATFM formulation. A rolling horizon implementation is also proposed to allow real-time flight submission. In both cases, the optimization process is performed in a

centralized way which might not be consistent with current U-Space trends where decisions are taken in a decentralized manner by different USSPs. Similarly, authors in [TX23] and [TXI22] also address the problem of fairly and efficiently assigning delays to balance demand and capacity. In this case, both centralized and decentralized processing is explored. However, as in the case of network resource allocation, scalability concerns are cited as a limiting factor of optimization-based techniques.

Another set of deconfliction and capacity management strategies are explored in [Chi+23a] [Chi+23c], which constitute the most exhaustive proposal found in the literature. In this case, the operation deconfliction problem is projected onto a grid and reduced to deconflicting aircraft movements between adjacent cells. To this end, they establish a priority queuing system that regulates access to each cell and a distributed algorithm that defines which aircraft can switch cells. The mechanism is privacy-aware, decentralized and studies different prioritization mechanisms (e.g.: random, delay-based, auctions-based where operators can define their priority). While it is shown that fairness is achieved (in terms of delays induced in each operation), the proposed mechanism is oriented to short term deconfliction (almost tactical) as it does not allow to deconflict a complete route (strategic deconfliction). Thus, it does not seem suitable for the current European regulatory paradigm where flights must be fully authorized before take-off.

3.3.1.3 Congestion management techniques

To finish the literature review on the topics covered in this section, **congestion management techniques** in UTM should also be explored. Again, available literature is scarce. Some of the already cited proposals attempt to solve this problem by means of optimization mechanisms [TX23] and [TXI22]. Also, in the DACUS project [DAC22] a new central entity (Dynamic Capacity Management service) has been proposed to jointly analyze the full flight picture and change u-plans specification to avoid the formation of hotspots.

3.3.1.4 Proposal novelties and comparison with existing literature

From the review on fairness metrics, it can be concluded that there are several notions of fairness in UTM systems (e.g.: price-based, FC-FS, delay based...). In this sense, the authorization framework that will be proposed in the next section is not limited to a given notion but allows to implement different fairness policies (as it will be discussed in Section 3.3.3). In addition, all the fairness metrics proposals discussed above are derived from UAM applications where operations are dynamic and can be subject to small delays at low cost to the operator. However, as explained previously, in the U-Space ecosystem there are a wide variety of operators' needs, with operations unable to be subjected to delays. Therefore, literature metrics might not be directly applicable to the considered use-case. As a result, in Section 3.3.5 the flight authorization rate will be used instead as a fairness metric for the purposes of validating the proposed framework.

As for fairness-enabling authorization frameworks in UTM, there does not exist a comprehensive proposal jointly covering all the following requirements:

- It enables managing fairness (i.e.: surpasses current regulation-mandated FC-FS ap-

proach) and congestion issues.

- It is scalable (in terms of number of managed aircraft). Thus, optimization-based alternatives should be avoided.
- It is distributable between different entities to comply with the CISP-USSP functionality distribution mandated in European regulations. Thus, alternatives operated from a central entity should be avoided.
- It can be adapted to the European regulation-mandated authorization workflow in which flight plans are submitted over time and require a full authorization before flying.
- It is flexible in order to allow the implementation of different fairness notions.
- It enables operators to state their operation preferences.

In this sense, the proposal on this thesis surpasses existing literature by proposing a comprehensive authorization framework based on a **non-monetary market** that is well-defined and integrated within an **extension of the European regulation-mandated authorization workflow**. Moreover, it enables implementing **different fairness policies** (including some of the mentioned ones in the existing state-of-the-art) and jointly considering fairness and congestion management. Furthermore, the framework is **scalable** and authorization decisions can be taken in a **decentralized manner by USSPs**, also considering **operator preferences**.

3.3.2 Token-based authorization framework

3.3.2.1 Deferred and multiple-priority authorization workflow

While the mandated FC-FS authorization scheme facilitates the implementation of the deconfliction workflow, it prevents the adoption of other protocols (e.g.: authorizing operations from different operators with different priorities...) addressing the previously identified fairness and congestion challenges. It can be argued that these protocols can be applied once an authorization has been provided to the operator. However, this would imply making changes to already approved operations in a way that is not transparent, nor predictable from the operator's point of view. Also, the inherent fairness problems in the FC-FS approach should also be addressed.

As an alternative, the proposed framework embraces the idea in [COR23] that **delays the definitive flight authorization** until sufficient information is available to apply those protocols. This proposal would require regulatory changes but does not imply a total paradigm shift as other proposals in the literature would entail (e.g.: flights are still authorized in advance; authorization responsibility remains with the USSPs...). In this case, when an operator submits a flight authorization request, the flight plan is only registered into the U-space system and considered as part of the expected traffic. Thus, an authorization is not issued immediately as mandated by regulation. Alternatively, the operator could be provided with an authorization probability estimate that considers both existing information and an estimation of future traffic patterns and expected measures/changes to instill some certainty and transparency into the process.

Later on, when most possibly conflicting flight plans have already been submitted, it is feasible to simultaneously perform strategic deconfliction (i.e.: to select which u-plan should access the contested airspace) between them, irrespective of when they were submitted and considering additional deconfliction protocols. This situation occurs at an agreed decision time relative to each flight plan take off time, denoted **RTTA (Reasonable Time To Act)**. After $t_{take\ off} - RTTA$, the flight plan is either approved or rejected and is not subject to any modifications except in the case of abrupt changes in airspace conditions (e.g.: new geofences/no-fly zones) or special operations (as required by regulations [EAS22a]). This difference between the adopted approach and regulation is depicted in an example scenario in Figure 3.36 where different flights may be approved depending on authorization workflow and applied protocols. In the figure three u-plans are competing to operate in the same 4D volume and must be deconflicted. Timelines are provided for each flight indicating when the u-plan is submitted and when an authorization decision is taken. Comparatively, FP1 is approved following a FC-FS approach but may be rejected in favor of FP2 in the second case due to the application of a given prioritization protocol.

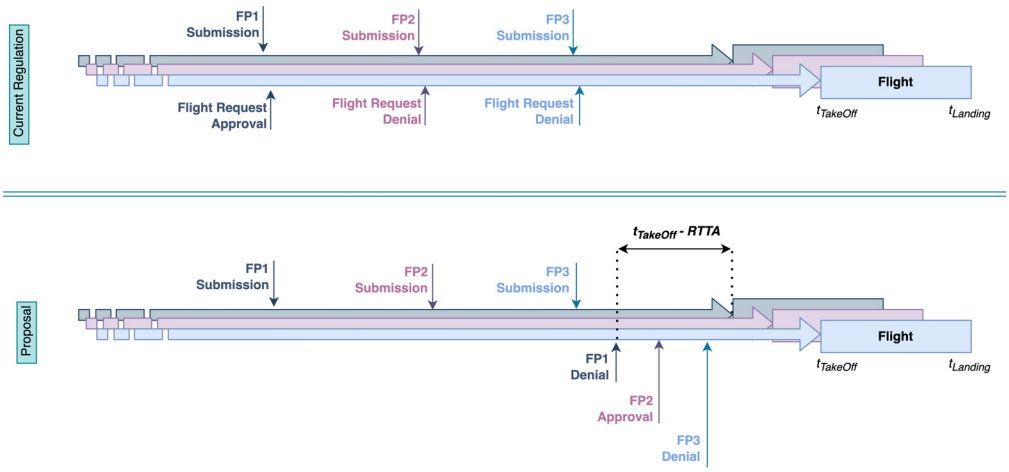


Figure 3.36: Comparison between the regulation mandated FC-FS approach and the adopted deferred authorization scheme.

Squares indicate flight duration while the previous horizontal arrow represents the u-plan lifecycle from submission, to authorization result and take-off. All arrows are aligned in time. Events are indicated as vertical arrows.

Besides breaking the implicit unfairness of the FC-FS approach, the deferred authorization scheme also enables adopting other deconfliction protocols. In this regard, and notwithstanding the special operations already mentioned, the proposed framework introduces a **system of priority classes in the authorization process**. Again, the proposal is inspired by the current European regulation, extending the authorization priorities currently in place. The proposal would not apply to special operations as defined in regulation (e.g.: medical flights, law enforcement flights) that should be authorized separately with priority over the rest of non-special operations. For the later, through the proposed prioritization framework, the operator can easily inform the U-Space system of the criticality or perceived value of each operation so that this information can be implicitly considered within the deconfliction process.

In this prioritization system, a u-space operator (denoted op_p, where OP is the set of all U-space operators):

$$op_p \in OP \stackrel{\text{def}}{=} \{op_1, \ldots, op_p, \ldots, op_P\} \tag{3.43}$$

must submit flight plans together with their selected **priority class** (denoted pc_i). Multiple *priority classes* can be considered:

$$pc_i \in PC \stackrel{\text{def}}{=} \{pc_1,\ pc_2, \ldots, pc_i, \ldots, pc_N\} \tag{3.44}$$

where PC is the set of all considered priority classes (N classes). Within the set, each class has less priority than the previous one:

$$priority\{pc_i\} > priority\{pc_{i+1}\} \tag{3.45}$$

This means that, when checking for conflicts (i.e. detecting contested 4D volumes), **only those flights in the same class or classes with higher priority than the target flight will be considered as possible conflicting flights**. This can be understood in terms of flight plans being entered into a pool of "virtual priority sets" only conflicting with those in sets of higher or equal priority. Continuing with the example of Figure 3.36 and as shown in Figure 3.37 , FP2 and FP3 have been submitted in priority class pc_1 while FP1 has been submitted in pc_2. This explains why FP1 has been denied even if considered earlier for authorization.

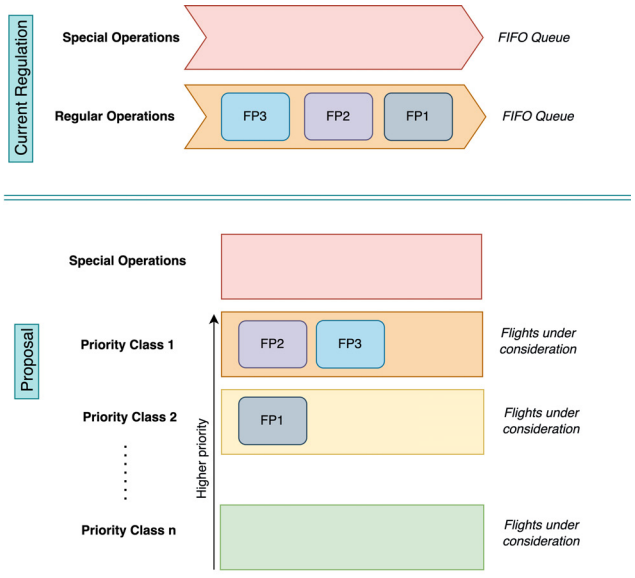


Figure 3.37: Comparison between the single priority mandated approach and the adopted multi-priority approach.

FP1, FP2 and FP3 are the same operations than the ones depicted in Figure 3.36 . In the current regulation approach, FIFO queues are shown and u-plans are ordered by submission times. In the proposed approach, U-plans are depicted in their assigned priority class.

Within the same priority class, an additional prioritization strategy is required. Multiple strategies could be followed: a random selection based on an identifier, a FC-FS prioritization based on the submission time, a FC-FS prioritization based on the take-off time... In keeping with the spirit of deferred authorization (i.e.: irrespective of the submission time), the later strategy is adopted for in-class deconfliction. Also, take-off time is constrained by mission requirements (e.g.: when the customer wants to operate) and thus is somewhat immune to possible abuses by the operators. Following this approach, in the example, FP2 is authorized over FP3 as it has an earlier take-off time. However, authors recognize that it could also be argued that a FC-FS based on submission time would be advisable to incentivize operators to submit operations as soon as possible. This approach would enable U-Space to perform reactive congestion management by promptly issuing dynamic capacity balancing measures (as proposed in the DACUS project [DAC22]) but would retain a fairness bias.

3.3.2.2 Basic Token Authorization (BTA) workflow: token-controlled access to priority classes

In the previous proposal, if operators were to act fairly and not driven by self-interest, those operations with the highest perceived value (i.e. those that have been entered in the first priority classes) would be mostly authorized. However, nothing prevents operators from acting selfishly and declaring all operations as having high value (high priority class). This case generates a "tragedy of the commons" situation in which the proposed mechanism loses its raison d'être (it degenerates into an unprioritized access to airspace). Therefore, it seems necessary to establish a **gatekeeper mechanism to limit access to each of the priority classes**. This mechanism would apply to all priority classes except for special operations [23]that should be managed separately.

In this sense, we propose the concept of scarce "**flight tokens**" that are distributed between operators and used to access a given priority class. The underlying philosophy is to transfer the scarcity of limited airspace into the access mechanism of the prioritization classes so that abusive behavior is avoided. As tokens are limited, operators must manage them according to their business logic in order to properly transmit the criticality of their operations to the U-Space system. In essence, tokens and their associated priorities involve a preference sharing system that maintains the privacy of the operators' business logic.

Further elaborating on this idea, each flight token (which may be individually denoted as tk) can be considered as a tuple with the following properties: an identifier (denoted tk_{id}), an assigned priority class (tk_{pc}), a value (tk_{value}) , a pledge state ($tk_{pledged}$), an expiration date ($tk_{expiration}$) and an assigned operator (tk_{op}):

$$tk = (tk_{id},\ tk_{pc},\ tk_{value}, tk_{pledged}, tk_{expiration}, tk_{op}) \tag{3.46}$$

Flight tokens have the following characteristics:

- They are atomic and unique. A token cannot be subdivided to optimize its usage, and two tokens cannot have the same identifying index ($tk_{id} \in \mathbb{N}$).
- They are scarce to effectively restrict access to the priority classes:

$$tk_{id} < N_{max\ tokens} \tag{3.47}$$

where $N_{maxtokens}$ is the maximum number of tokens that can be issued. This restriction can be relaxed for comparison purposes, allowing infinite tokens so that the mechanism degenerates into a non-scarce token assignment problem.

- They are linked to a priority class for which they can grant access which is represented as:

$$tk_{pc} = pc_i; \qquad pc_i \in PC \tag{3.48}$$

which denotes that token tk is only assigned to priority class pc_i.

- They have an intrinsic, limited value in terms of the amount of airspace occupation (i.e.: 3D volume for a certain amount of time) they can insert into a priority class. This value, denoted TK_{value}, is constant and equal for all tokens and must be decided before implementing the proposed authorization scheme.

$$tk_{value} = TK_{value} \ \in \ R^{+} \ \left[\frac{m^3 \cdot s}{token}\right] \tag{3.49}$$

It signifies that TK_{value} tokens are required to access 1 $m^3 \cdot s$ of airspace.

- They are single use: they can only be pledged once to access a priority class. This is defined by the following token property:

$$tk_{pledged} = \{0, 1\} \ \ where \ \begin{cases} 0; \ if \ unpledged \\ 1; \ if \ pledged \end{cases} \tag{3.50}$$

- They expire at a given date, $tk_{expiration}$, to avoid token harvesting by operators.
- They are uniquely assigned to an operator (op_p):

$$tk_{op} = op_p; \qquad op_p \in OP \tag{3.51}$$

which denotes that token tk is assigned to operator op_p. In principle, tokens are non-transferrable, but exchange mechanisms between operators could be ideated in the future.

Periodically, each operator (op_p) is assigned a set of individual tokens (denoted T_{op_p}) distributed among the different priority classes:

$$T_{op_p} = T_{op_p}^{pc_1} \cup \ T_{op_p}^{pc_2} \cup \ T_{op_p}^{pc_i} \ldots \cup T_{op_p}^{pc_N} \tag{3.52}$$

$$T_{op_p}^{pc_i} = \{tk \ \mid \ tk_{op} = op_p \ \wedge \ tk_{pc} = pc_i\} \tag{3.53}$$

where $T_{op_p}^{pc_i}$ represents the set of tokens assigned to operator op_p for priority class pc_i. The total number of tokens assigned per operator (i.e.: the cardinality of the T_{op_p} set: $|T_{op_p}|$) and the distribution of tokens per priority class (i.e.: the cardinality of the $T_{op_p}^{pc_i}$ set: $|T_{op_p}^{pc_i}|$) can be decided by policymakers and regulators based on different policies that will be described in Section 3.3.3 . These policies can enforce different fairness notions by adequately distributing tokens between operators in a non-uniform manner. This way, some operators may be incentivized over others depending on their characteristics.

When an operator op_p wants to submit a flight plan (denoted fp) for authorization in priority class pc_i, it must also pledge a set of previously unpledged tokens, denoted $T_{pledged\ for\ fp}$), to cover that flight. Using the token notation, this set can be defined as:

$$T_{pledged\ for\ fp} = \{tk|\ tk \in T_{op_p}^{pc_i}\ \wedge\ tk_{pledged} = 0\} \tag{3.54}$$

The number of required tokens to submit a given flight (fp) is denoted as the flight **Token Cost** (TC_{fp}). It is related to the occupation of the flight plan and the fixed token value. As previously stated, all authorized flight plans require the reservation of a set of 4D boxes around their expected trajectory to ensure their separation in case of navigation errors, piloting errors... The sum of the temporal-spatial volume (i.e.: 3D volume multiplied by the time occupation) occupied by each 4D box constitutes the airspace occupation of that flight, denoted: o_{fp}. From it, and given the airspace occupation right granted by a single token (defined by the TK_{value}), the token cost of a flight can be computed as:

$$TC_{fp} = \left\lceil \frac{o_{fp}}{TK_{value}} \right\rceil \in \mathbb{N}\ [number\ of\ tokens] \tag{3.55}$$

Thus, at submission, the U-space system must ensure that the operator has enough tokens remaining (i.e.: not-pledged tokens) to generate the $T_{pledged\ for\ fp}$ set. That is, the cardinality of the pledged tokens set ($|T_{pledged\ for\ fp}|$) must be equal to the token cost of the flight (TC_{fp}). As a result, the flight will enter the desired priority class and be subjected to the authorization process. Additionally, the subset of submitted tokens will become pledged and could not be used again:

$$|T_{pledged\ for\ fp}| = TC_{fp} \tag{3.56}$$

$$tk_{pledged} = 1, \quad \forall tk \in T_{pledged\ for\ fp} \tag{3.57}$$

Overall and as depicted in Figure 3.38 , the mechanism discussed so far introduces a gate at each priority class that can be surpassed using a virtual currency: flight tokens. As tokens are limited, it avoids that an operator monopolizes the use of airspace. Moreover, as the consumption of tokens is linked to the flight plan occupation, it also incentivizes an efficient use of airspace. This token-based mechanism will be referred as the **Basic Token Authorization** (BTA) workflow.

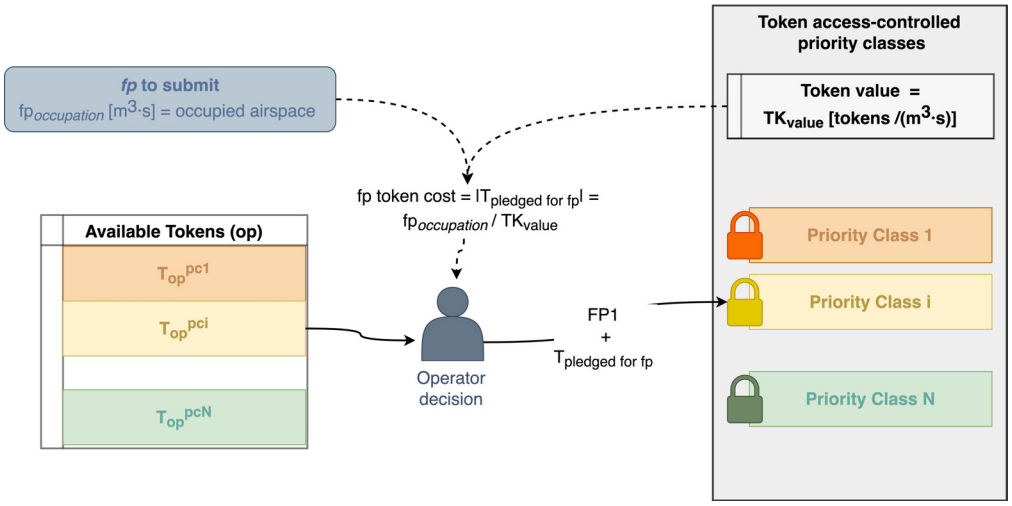


Figure 3.38: Graphical representation of the token-controlled access to priority classes in the BTA workflow.

Tokens (on the left) are assigned to operators (on the center) that use them to enter a u-plan (on the top) to a given priority class (on the right) subject to the computed token cost.

3.3.2.3 Geographically-aware Token Authorization (GTA) workflow: introducing temporal and geographical incentives

The previous BTA protocol contemplated a flight Token Cost that was geographical and time independent. This implies that the location and take-off time of a flight plan do not affect the number of tokens required to fly, only the required airspace occupation. However, airspace capacity is not uniform as it is affected by UAS zones restrictions (i.e.: geofences/geocages) or CNS (i.e.: Communications, Navigation and Surveillance) capabilities. Similarly, traffic demand also follows geographical and temporal patterns that should be considered. In this sense, it is possible to extend the token cost computation to consider this heterogeneity by implementing geographical and temporal incentives. This extension will result in a new authorization protocol named **Geographically-aware Token Authorization** (GTA) workflow.

To do so, a new element is to be introduced: the **Airspace Cost** (**AC**). This parameter consists in a 3-dimensional grid that evolves over time and can be operator-dependent. It can be interpreted as a matrix of weights/multipliers affecting the flight Token Cost computation. Through these multipliers, different portions of the airspace will have a different cost in terms of the number of tokens required to occupy that airspace portion. Mathematically, the **AC** matrix can be defined as follows:

$$\mathbf{AC}(\mathbf{t},\ \mathbf{op_p}) = [ac(t, op_p)_{x,y,z}]_{0\leq x\leq x_{max},\ 0\leq y\leq y_{max},\ 0\leq z\leq z_{max}} \tag{3.58}$$

$$\begin{gathered} ac(t, op_p)_{x,y,z} \in [0, \infty); \\ x,\ y,\ z \in N; \\ 0 \leq t < t_{max},\ t \in N \end{gathered} \tag{3.59}$$

So far, the airspace occupation of a flight plan has been considered as a single aggregated value (denoted o_{fp}). However, to calculate the number required of tokens with the 4D-dependent airspace cost, it is needed to project the flight plan occupancy into the same temporal-spatial grid, obtaining a finer view on which areas of airspace are used and when. Thus, we define a new version of this variable, denoted **geographic occupation** ($\mathbf{O_{fp}}$) of flight plan fp:

$$\mathbf{O_{fp}(t)} = [o_{fp}(t)_{x,\ y,\ z}]_{0\leq x\leq x_{max},\ 0\leq y<y_{max},\ 0\leq z<z_{max}} \tag{3.60}$$

$$\begin{gathered} o_{fp}(t)_{x,y,z} \in\ R^{+}\ with\ units\ of\ m^3 s \\ x,\ y,\ z \in N; \\ 0 \leq t < t_{max},\ t \in N \end{gathered} \tag{3.61}$$

where x, y, z are the indexes of the matrix (spatial grid). Each element of the matrix, denoted $o_{fp}(t)_{x,y,z}$, indicates how much airspace (in $m^3 \cdot s$) the flight plan fp is occupying in the (x,y,z) cell of the grid within the t time interval.

Now, if both matrices are multiplied in an element-wise manner and added across all dimensions, it is possible to obtain a **weighted version of the previous Token Cost** ($TC_{fp}^{weighted}$). This new figure considers that, depending on the area or time instant, more tokens will be required to occupy the same volume of airspace, as a function of the incentives defined in the Airspace Cost. It can be computed as:

$$TC_{fp}^{weighted} = \left\lceil \frac{\sum_{t=0}^{t_{\max}} \sum_{\forall x,y,z} \mathbf{AC}(t,\ op) \circ \mathbf{O_{fp}}(t)}{TK_{value}} \right\rceil \quad [number\ of\ tokens] \tag{3.62}$$

Where $\circ$ represents the element-wise multiplication across dimensions x, y, z of the AC and O_{fp} matrices, so that each occupation element/cell is weighted by the corresponding airspace cost. From the aggregated weighted occupation figure obtained (numerator in Equation 3.62), the token cost can be derived from the token value: TK_{value}. The computation of this geographical, temporal and operator-aware token cost is represented in Figure 3.39 .

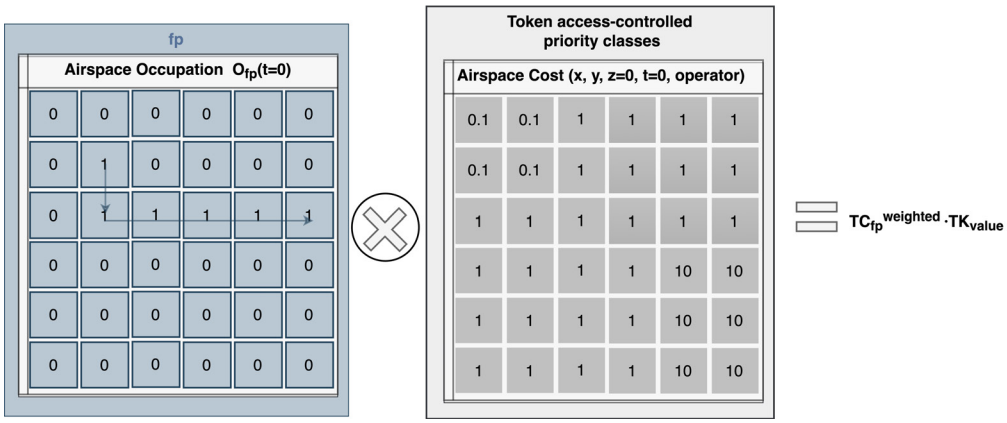


Figure 3.39: Representation of the weighted token cost computation.
Both the airspace occupation (from the u-plan represented as arrows) and the airspace cost matrices are depicted with example values (Time and height evolution are disregarded for easier representation). The token cost can be obtained by multiplying both matrixes element-by-element (which is represented by the circled multiplication character).

Once the required number of tokens has been obtained considering both the geographical occupation and the geographical incentives, the gated access to the priority queues would work as already explained above for the BTA workflow. That is, the number of pledged tokens (cardinality of $T_{pledged\ for\ fp}$) must be equal to the weighted token cost of the flight:

$$|T_{pledged\ for\ fp}| = TC_{fp}^{weighted} \tag{3.63}$$

3.3.2.4 Integration on UTM/U-space architectures

So far, the proposed authorization workflow and its two versions (i.e.: BTA and GTA) have been described without referring to its implementation within the U-space ecosystem. As mentioned, the proposed mechanism goes beyond the current European regulation. Therefore, its implementation would require some regulatory changes but not a paradigm shift. This section explores how the proposal could be implemented and distributed between the two main U-Space systems: the Common Information Service Provider (or a central platform that extends such a role), CISP, and U Space Service Providers, USSPs (see Section 3.1.4 for more information on these systems).

In this regard, a new centrally-deployed service extending the CISP core functionalities should be adopted: the **Authorization Policy Service (APS)** (as depicted in Figure 3.40). This service would be responsible of implementing a given fairness policy by setting the airspace cost and distributing tokens among the operators of each USSP. This process can be based on optimization techniques or heuristics, grounded on relevant information coming from different sources: historical flight data, capacity and demand estimates, possible markets established for token management... The APS operates periodically (i.e.: daily), and far in advance so that it is decoupled from the dynamic nature of the flight authorization process. The centralization of this service is justified since it is linked to the authorities/policy-makers activities, allowing them to introduce incentives to govern the U-space authorization process.

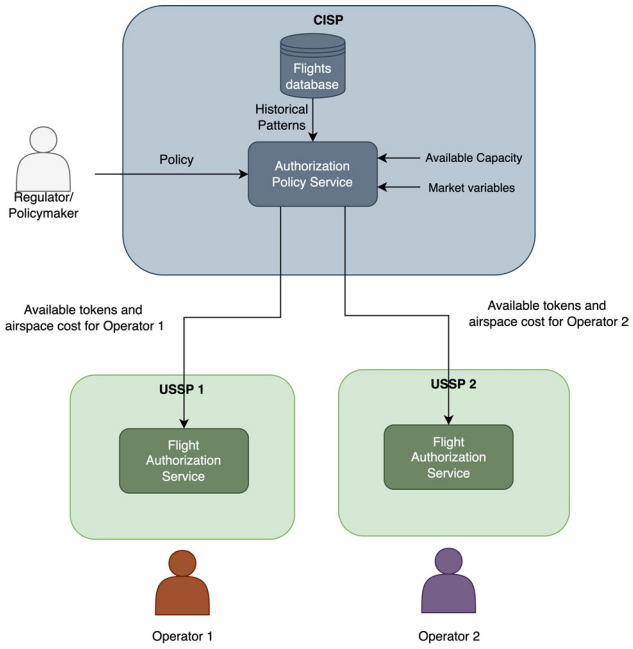


Figure 3.40: Authorization Policy Service interactions.

Once the tokens have been distributed a-priori (i.e.: before flights are submitted), USSPs can operate in a distributed way, without intervention from the APS. This is demonstrated in Figure 3.41 which shows the sequence of interactions (i.e.: the numbered steps) through this process in a scenario where two operators submit a flight plan in different priority classes across different USSPs. When a flight is submitted (interaction 1), the operator would express the criticality of its operation through the selection of the desired priority class subject to token scarcity. Optionally, the functionality of the USSP's u-plan preparation service (i.e.: as defined in [COR23], to assist operators in defining a u-plan for their operations) could be extended to include the calculation of the number of tokens required for an operation so that operators can make informed decisions. Then, within each USSP, the Flight Authorization Service (FAS) would be charged of managing operator's tokens (already distributed by the APS), enforcing that enough tokens are pledged to access the desired priority class (interaction 2).

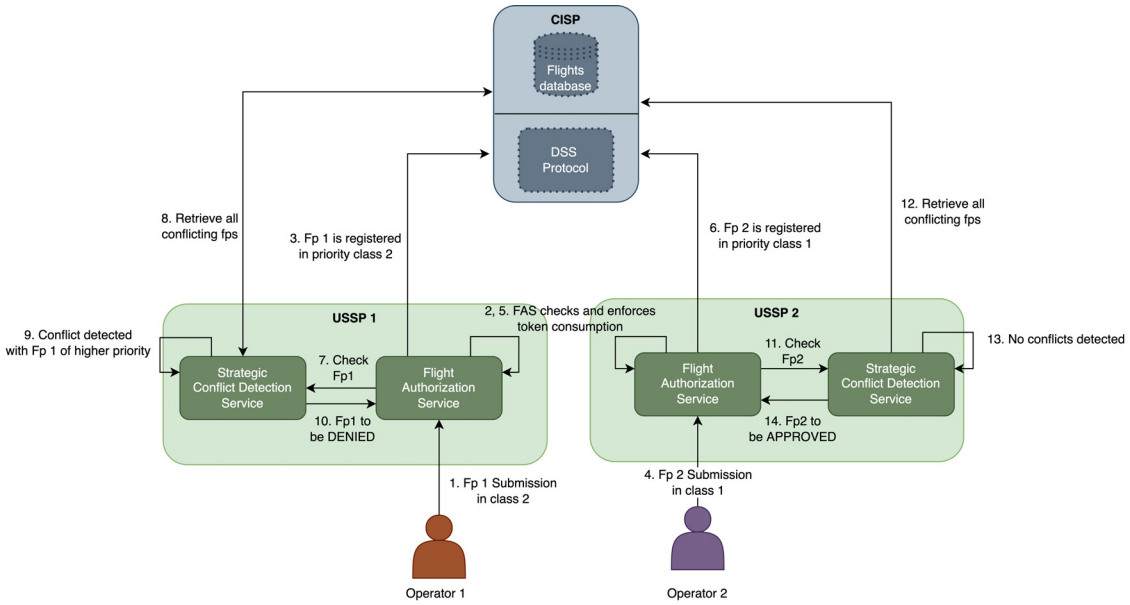


Figure 3.41: Sequence diagram of interactions between U-Space services when implementing the proposed deconfliction framework.

Interactions shall be understood sequentially as depicted by the number in each step. A centralized flights database in the CISP (or centralized platform) or a decentralized DSS protocol may be used to exchange flight information.

Next (interaction 3), the submitted flight plan could be registered in a common centralized Flights Database (FD), together with its associated priority class. Alternatively, a decentralized flight information exchange protocol could also be used to directly exchange flight information between USSPs assisted by a Discovery and Synchronization Service (DSS, as proposed in [AST19]). An adapted, extended version of existing protocols (mainly, ASTM F3548-21 [AST22]) would be needed to contemplate priority classes and additional flight plans states. Even if all u-plans are to be registered and shared across USSPs, this information must not be shared with operators for privacy reasons and to avoid abuses of the proposed authorization system.

At $t_{take-off} - RTTA$, the Strategic Conflict Detection Service (SCD) would retrieve (interaction 8) all possible conflicting flight plans (i.e.: already approved flights and flights under consideration in higher priority classes) so that conflicts can be checked (interaction 9) and a final authorization decision can be individually taken by each USSP (interaction 10).

In conclusion, both BTA and GTA workflows can be performed without resorting to direct intermediation between USSPs to deconflict flights while considering fairness and/or capacity issues. The APS encapsulates all policy management complexity avoiding in-the-loop optimizations for each authorization request. In this sense, **tokens serve as a preemptive gatekeeper that can be managed individually by each USSPs once tokens are distributed**. Then, as prioritization information is registered in a centralized DB or exchanged with a distributed protocol, each USSP can also autonomously take authorization decisions. Overall, the discussed integration details show that the proposed authorization approach is viable and easy to implement into the real U-Space ecosystem while providing high scalability (due to its distribution between different systems).

3.3.3 Policies for token distribution and airspace cost setting

In the previous section, a flexible strategic authorization framework has been proposed based on priority classes, scarce flight tokens to control the access to those classes and a heterogeneous airspace cost to introduce geographical incentives into the authorization process. The proposal makes it possible to incorporate into the same framework the operators' preferences (communicated through the selection of a priority class), flexible fairness enforcement (with the choice of policy by regulators, that will be described in this section) and even congestion management. This framework is **policy-agnostic in the sense that it allows enforcing multiple fairness, congestion management and, in general, prioritization strategies**; which would be configured in the APS. Two variables are key in implementing these policies (as shown in Figure 3.42): the token distribution between operators ($|T^{pc_i}_{op_p}|$, both in BTA and GTA) and the airspace cost ($AC(t, op_p)$, in GTA). The former allows to incentivize certain operators by assigning them more tokens or using a heterogeneous distribution of tokens between priority classes. In turn, the later enables creating geographical and temporal incentives, as already explained.



Figure 3.42: Representation of different policy objectives and encoding within the token-based framework

Figure exemplifies how the policy objectives (operator-level or geographical incentives, on the left) can be implemented through token distribution and airspace cost (on the right). Note how operators to be incentivized (in green) are assigned more tokens and how areas to be disincentivized (in red) have a higher airspace cost. Numeric values are examples.

Although **it is out of the scope of this thesis to determine an optimal fairness or congestion management-enabling policy**, some policies will be explored to prove the usefulness and flexibility of the approach. Those policies are: A) unlimited token distribution, B) scarce, uniform token distribution, C) scarce, non-uniform token distribution D) pay-per-token distribution, E) pay-per-airspace distribution, F) congestion management and other space-based incentives. A straight-forward, non-optimal version of each policy will be implemented in Section 3.3.4 and Section 3.3.5 for validation purposes. For simplicity and ease of understanding, these preliminary implementations of each general policy are

constrained to 3 priority classes (pc_i *with* $i \in 1, 2, 3$, which will be denoted $HIGH$, $MEDIUM$ and LOW respectively) and will also be described in the following subsections. Again, the proposed implementations are not intended to provide an optimal and complete solution to the fairness problem, but rather to explore the possibilities of the enabling mechanism discussed in the previous section.

3.3.3.1 Unlimited token distribution policy

As a starting point for the implementation of the proposal, a trivial policy would consist in not constraining the access to the priority classes. Thus, relying on the operator's good faith in declaring each operation value (i.e.: selecting the appropriate priority class). To do so, an unlimited number of tokens of all types could be distributed to all operators while keeping the airspace cost homogeneous (i.e.: using the BTA version of the proposal):

$$\left|T_{op_p}^{pc_i}\right| = \infty, \qquad \forall pc_i \in PC,\ op_p \in OP \tag{3.64}$$

where the previously established notation is followed in which $T_{op_p}^{pc_i}$ represents the set of tokens for priority class pc_i assigned to operator op_p, being $|T_{op_p}^{pc_i}|$ the number of tokens in such set. In the simplified implementation for validation, this means that all operators receive an unlimited number of tokens for both LOW, $MEDIUM$ and $HIGH$ priority classes:

$$\left|T_{op_p}^{pc_{LOW}}\right| = \left|T_{op_p}^{pc_{MEDIUM}}\right| = \left|T_{op_p}^{pc_{HIGH}}\right| = \infty \tag{3.65}$$

Tokens in this case would serve to inform the system of operators' benefit/expected criticality of each operation but would not limit the number of flights each operator can submit. For instance, let us assume two operators: one with 100 flights and one with 10 flights; for whom, according to their business logic 50% of the total operations are critical and the remaining can be labelled as having low value. Following this unlimited distribution of tokens, the first operator could submit 50 operations in the high priority class (pc_{HIGH}) informing the U-Space of their high value. Similarly, the second operator would only submit 5 high priority operations. These operations would have a high probability of being authorized over the rest of operations submitted to pc_{LOW} priority class.

With this approach, the intrinsic unfairness of the FC-FS approach would be solved but operators with dissimilar characteristics could yield to an unfair distribution of flights. As the previous example exposes, operators submitting a higher number of u-plans would receive a higher number of authorizations than low-traffic operators. More importantly, as already discussed, greedy operators (i.e.: those who do not declare faithfully operation preferences) could take advantage of the policy by submitting all flights to the higher priority classes monopolizing airspace access (i.e.: the aforementioned tragedy of the commons).

3.3.3.2 Scarce, uniform token distribution policy

To avoid a tragedy of the commons, scarce token distribution policies can be implemented. In this case, a limited number of tokens of each priority class is distributed to each operator.

In principle, the number of tokens assigned to high priority classes should be lower than the number assigned to low priority classes. By doing so, token scarcity is used as a signal to inform operators of the relative value (i.e.: higher authorization rates) of each priority class. Thus, the operator is forced to analyze comparatively the utility of each operation and decide which ones to prioritize. It would be expected that those operations with higher utility should be assigned a higher priority class subject to token scarcity. Mathematically:

$$\left|T_{op_p}^{pc_i}\right| < \left|T_{op_p}^{pc_{i+1}}\right|, \quad \forall pc_i \in PC \tag{3.66}$$

A possible token distribution that meets this condition is one in which, in each priority class, all operators obtain the same number of tokens:

$$\left|T_{op_p}^{pc_i}\right| = \left|T_{op_{p+1}}^{pc_i}\right| = k_{pc_i}, \qquad \forall \quad op_p \in OP \tag{3.67}$$

The number of tokens for a priority class (i.e.: k_{pc_i}) should be related to the maximum available airspace capacity and the relative advantage that the priority class gives for airspace access (i.e.: how likely it is that a flight plan will be authorized in that priority class). These values could be obtained with an optimization by simulation process or based on historical statistics. Moreover, if the opportunity to access the airspace is to be guaranteed at all times (albeit entirely unprioritized), the number of assigned tokens in the lowest priority class (i.e.: pc_N) should be unlimited:

$$\left|T_{op_p}^{pc_N}\right| = \infty, \forall\ op_p \in OP \tag{3.68}$$

Following on with the example in Section 3.3.3.1 , let us know assume that each operator receives 5 tokens for the high priority class ($k_{pc_{HIGH}} = 5$), 5 for the medium priority class ($k_{pc_{MEDIUM}} = 5$) and an unlimited number in the low priority class ($k_{pc_{LOW}} = \infty$). Also, for simplicity, let us assume that each flight requires just one token for its submission. With this distribution, the operator with 100 flights would have to use its business logic (expected value of each operation) to distribute its 50 critical flights among the 3 priority classes, as it does not have a sufficient number of tokens. This way, it would be prevented to monopolize airspace. Contrary, the second operator would be able to submit all of its critical flights in the high priority class. Indeed, with this uniform distribution policy, small operators can be benefited over large operators (i.e.: those with a high number of submitted flights). The former can enter a higher proportion of flights in the higher priority classes and therefore secure a higher authorization rate.

In particular, this policy has been implemented with the following assumptions. First, the number of tokens to be distributed in total ($N_{max\ tokens}$) is defined as the number of tokens required to authorize the maximum number of concurrent flights in an airspace (i.e.: fp_{max}, the airspace capacity). To compute it, the average number of tokens required per flight (i.e.: $\overline{TC_{fp}}$) is estimated from historical data, obtaining:

$$N_{\max tokens} = \overline{TC_{fp}} \cdot fp_{\max} \tag{3.69}$$

These tokens are equally divided between the number of operators (i.e.: $|OP|$). Additionally, it has been assumed that the high priority class gives a 50 % higher chance of authorizing a flight. And thus, that the number of tokens for each class must be inversely proportional to this advantage. Therefore, the implemented policy is:

$$k_{pc_i} = \begin{cases} \frac{1}{3} \cdot \frac{\overline{TC_{fp}} \cdot fp_{\max}}{|OP|} , & if\ pc_i = HIGH \\ \frac{2}{3} \cdot \frac{\overline{TC_{fp}} \cdot fp_{\max}}{|OP|} , & if\ pc_i = MEDIUM \\ \infty , & if\ pc_i = LOW \end{cases} \tag{3.70}$$

With this uniform distribution policy, small operators can be benefited over large operators (i.e.: those with a high number of submitted flights). The former can enter a higher proportion of flights in the higher priority classes and therefore secure a higher authorization rate.

3.3.3.3 Scarce, non-uniform token distribution policy

Alternatively to uniform token distribution, tokens can be allocated non-uniformly based on an operator's past behavior. A possibility would be establishing a virtue-points system (such as the one proposed in [DAC22]) in which those operators whose behavior has been reliable, safe, efficient... are rewarded with more tokens over non-conforming operators. Another viable approach would consist in allocating tokens based on an operator's traffic share. This way, large operators could have assigned a higher proportion of tokens to support their historical traffic so that they authorization probability can be similar to those of small operators.

Assuming the existence of an allocation function (denoted f) that depends on an operator historical behavior and condenses the non-uniform distribution logic for each priority class (pc_i), this policy can be formalized as:

$$\begin{gathered} BTA\ workflow\ with: \\ \left|T^{pc_i}_{op_p}\right| \neq \left|T^{pc_i}_{op_{p+1}}\right| = f\left(op_{p+1}\ past\ behavior, \quad pc_i\right), \qquad pc_i \in PC, \quad op_p \in OP \end{gathered} \tag{3.71}$$

In the validation implementation, the later allocation function is implemented by distributing tokens proportionally to the flight share of an operator (i.e.: historical number of flights relative to total number of flights, denoted $share_{op_p}$). Moreover, the distribution of tokens per priority class and the total number of allotted tokens is consistent with the previous policy defined in Equation 3.69 . Therefore, the implemented function is as follows:

$$f\left(op_p, \quad pc_i\right) = \begin{cases} \frac{1}{3} \cdot N_{\max tokens} \cdot share_{op_p} , & if\ pc_i = HIGH \\ \frac{2}{3} \cdot N_{\max tokens} \cdot share_{op_p} , & if\ pc_i = MEDIUM \\ \infty , & if\ pc_i = LOW \end{cases} \tag{3.72}$$

where non integer token values should be rounded to the nearest integer. Following on with the previous example, let us assume that the number of tokens to be distributed is 20 ($N_{max\ tokens} = 20$). If the large operator submits 100 operations and the small operator

submits 10, their historical number of flights would be approximately: $share_{op_{large}} = 0.941$, $share_{op_{small}} = 0.09$. As a result, the small operator would receive just one token for both HIGH and MEDIUM priority classes, whereas the large operator would receive 6 for the HIGH priority class and 12 for the MEDIUM priority class. This way, both operators would be able to submit a similar proportion of their critical flights in the higher priority classes (i.e.: 2 flights over 5 critical and 20 flights over 50 critical) and have similar authorization rates irrespective of their size.

3.3.3.4 Pay-per-token distribution policy

Following on with policies based on token distribution (based on the BTA alternative), a pay-per-use approach such as the one defended in [MBB21] can also be implemented with the proposed framework. This would require assigning a monetary value to the flight tokens of each of the priority classes (denoted $tk_{price}(pc_i)$). Thus, the number of tokens in each class ($|T^{pc_i}|$) could be unlimited as the scarcity notion is already included in the token price Additionally, to signal the different value of each priority class, those classes of higher priority should have a higher economic cost:

$$\begin{gathered} BTA\ workflow\ with: \\ |T^{pc_i}| = \infty,\ \forall\ pc_i\ \in PC \\ tk_{price}(pc_i)\ [euros]\ \ |tk_{price}(pc_i) > tk_{price}(pc_{i+1})\ \forall\ pc_i\ \in\ PC \end{gathered} \tag{3.73}$$

Setting the token price for each priority class would be a responsibility of the regulators that need to consider the available airspace capacity and the profitability of drone operations so that airspace is economically accessible. They would also need to consider that this policy can introduce inequalities if operators' marginal income (revenue per flight) is heterogeneous, as those operators with lower operating costs and more expensive operations would be incentivized. On a positive side, since the cost in number of tokens is related to airspace usage, less efficient operators would be disincentivized as they would need to pay a higher amount of money.

From the operators point of view, instead of allocating a number of tokens in advance, they would have to pay to obtain the desired number of tokens to submit their operations. Depending on the revenue and cost per operation, operators may pay (represented as function h) for one priority class or another according to their business logic, while also avoiding economic losses. This operator behavior will be further explained in Section 3.3.4.2 .

$$\left|T_{op_p}^{pc_i}\right| = h\left(op_p\ business\ logic\right) \tag{3.74}$$

3.3.3.5 Pay-per-airspace distribution policy

Moving on to geographic incentives (i.e.: GTA workflow), the proposed framework can also implement a fairness policy such as the one described in [Sko18] that grants exclusive (or preferrable) fly access of some corridors/airspace areas to bid-winning operators (pay-per-airspace). To model this, an operator-dependent airspace cost (previously denoted as

$ac(t, op_p)_{x,y,z})$ would be used.

Operators which have paid to use certain area (e.g.: $op_{reservation}$ in Equation 3.75) would have a null airspace cost within the reserved grid cells (a set denoted as $A_{reserved} = \{(x, y, z)\dots\}$). Conversely, the rest of operators would be assigned an infinite cost (or a high cost if flight is non-exclusive but disincentivized) in the same areas. Non reserved cells would have a cost of 1, allowing for their contested access by all operators:

GTA workflow with:

$$\text{ac}(t,\ op_{reservation})_{x,y,z} = \begin{cases} 0, & if\ (x,y,z) \in A_{reserved} \\ 1, & otherwise \end{cases}$$

$$\text{ac}(t,\ op_p)_{x,y,z} = \begin{cases} \infty\ , & if\ (x,y,z) \in A_{reserved} \\ 1, & otherwise \end{cases} \quad ; \quad op_p \neq op_{reservation} \tag{3.75}$$

Thereby, the former operators could fly without limits (without spending tokens) in their reserved area, while the latter would have to dedicate a very high number of tokens to do the same (or even been impeded to fly). The proposed framework allows more flexibility than the literature proposal, as it would allow a hybrid approach considering both a mix of reserved and free-flight zones (with an homogeneous nominal cost) or even allowing the access to reserved cells to other operators but with a very high cost. Token allocation for this policy can be performed with any of the previous policies (Section 3.3.3.1 to Section 3.3.3.4). Particularly, in the implemented proof-of-concept a scarce, uniform token distribution has been used.

3.3.3.6 Congestion management policy

Finally, the heterogeneous airspace-cost (GTA workflow) can also be used as a congestion management protocol. Historical data allows to estimate which airspace areas are more congested, modelling it as a congestion grid. From the detailed, grid-discretized version of a flight plan occupation ($\mathbf{O_{fp}(t)}$, defined in equation Equation 3.60), the congestion in a grid cell can be measured as the sum of the occupation in that cell for all approved flight plans:

$$congestion(x, y, z) = \sum_{\substack{\forall\ authorized\ fp, \\ \forall\ t}} o_{fp}(t)\ _{x,y,z} \tag{3.76}$$

Thus, cells that are occupied by many flights will have a high congestion value. Then, this value can be translated into an airspace cost matrix that deters operators from accessing congesting areas. In this sense, a function c could assign low airspace cost to areas with low congestion, whereas congested areas would have a higher cost:

$$\begin{aligned} &\text{GTA workflow with:} \quad \text{ac(t, op}_p)_{x,y,z} = c(x,y,z) \\ &\text{Function c is s.t.:} \quad \text{if congestion}(x_1,y_1,z_1) > \textit{congestion}\,(x_2,y_2,z_2) \implies \\ &\qquad ac\,(t,\ op_p)_{x_1,y_1,z_1} > ac\,(t,\ op_p)_{x_2,y_2,z_2} \end{aligned} \tag{3.77}$$

As an example of such a function, in the test implementation, the relationship between the congestion of a cell and its corresponding airspace value is as follows:

$$c\,(x,y,z) = \begin{cases} 5 & if\ RC(x,y,z) \geq 0.8 \\ 4 & if\ 0.6 > RC(x,y,z) > 0.8 \\ 3 & if\ 0.4 > RC(x,y,z) > 0.6 \\ 1 & otherwise \end{cases} \tag{3.78}$$

where $RC(x,y,z)$ is the relative congestion of a cell related to the maximum congestion of all cells (i.e.: $\frac{congestion(x,y,z)}{max(congestion(x,y,z))}$) . This way, operators wanting to operate in congested areas would require a higher number of tokens (again, distributed following any of the previous token-distribution policies described for the BTA workflow in Section 3.3.3.1 to Section 3.3.3.4), being compelled to avoid such areas (if possible). Overall, this policy would enable to pre-emptively avoid congestion by incentivizing the use of low-traffic areas and disincentivizing the use of high-traffic areas, all while avoiding tight restrictions.

A similar approach can be followed to manage other u-space aspects with a geographical component such as noise limits, social impacts... In fact, the framework could also be used to code the constraints of geofences.

3.3.4 Experimental setup

In order to assess the validity of the approach and to show the proposed framework capability to successfully implement different authorization management policies within the strategic deconfliction phase, three different simulated validation scenarios have been envisioned and executed using USPASIM (the U-Space ecosystem simulator proposed in Section 3.2.2).

In each scenario, different operator behaviors are considered (as it will be discussed in Section 3.3.4.1) together with alternative token allocation and airspace cost-setting policies (among the ones identified in Section 3.3.3). Then, scenarios are simulated using an extended version of the simulator where the proposed authorization framework has been implemented and integrated into pre-existing, regulation-mandated services. Further details on the implemented extensions are found in Section 3.3.4.2 . Finally, the metrics described in Section 3.3.4.3 are used to analyze the impact of the proposed framework in terms of fairness and congestion.

3.3.4.1 Scenario generation

Each validation scenario consists in a series of runs (i.e.: different random realizations of the scenario generation process for the same input parameters) to perform Montecarlo simulation. Within each run, `Operator` agents with different behaviors are considered. This allows to

compare the policies' effect in different types of operators. In accordance with the operator behaviors that have been proposed in Section 3.2.3.1 , a combination of the following behaviors has been considered:

- Operators that concentrate their operations in an area / corridors or operators with sparse operations. Areas in which operations are generated are randomly selected ensuring that there are no overlap between operators.
- Operators with different levels of traffic: large operators (each submitting 100 or 150 flights) or small operators (each submitting 50 flights).
- Operators with different levels of income per flight: high income (randomly selected between 1500 and 10000 € per flight) or low income (from 100 to 6000 €/flight).
- Operators that act fairly or that are self-interested (i.e.: do not faithfully report operation preferences if not conveyed to do so).
- Operators that submit their operations early (far from the take-off time, randomly selected for each flight from 3 to 6 hours before) or operators that submit them late (close to the take-off time, half an hour to 4 hours before.).

Drone traffic for each operator (i.e.: the set of flight plans that will be submitted to the U-Space) is generated considering a selection of the previous behaviors and the traffic patterns described in Section 3.2.3.2 . This will determine the number of submitted flights, their geographical flight distribution... The reader should recall that in the traffic generation step within `Operator` agents two relevant properties are also stochastically assigned (from a statistical model) to each flight plan: the take-off time and the submission time (i.e.: how far in advance before take-off the flight is submitted); from . Related to this, a common RTTA value of 20 minutes is assumed for all operators. Finally, the preference or relevance of each flight for the operator is modeled in economic terms, assigning an income value to each flight (denoted inc_{fp}, income of a given fp).

3.3.4.2 Adaptations to the U-Space ecosystem simulator

In order to execute each of the Montecarlo runs within an scenario, the following adaptations have been made to the `Operator` agent model and to the reference U-Space service implementation (both USSPs services and CISP services) of the USPASIM platform.

Operator agents

In order to submit flight plans ($fp_f \in FP_{op_p} = \{set\ of\ fps\ of\ operator\ op_p\}$) according to the proposed framework, `Operator` agents must autonomously assign a priority class to each flight considering the authorization framework restrictions (i.e.: available tokens and airspace cost) and their business logic. After each u-plan submission, Operator agents wait for an authorization result at $t_{takeoff} - RTTA$. In case a conflict is detected by the USSP's services, no deconfliction action (e.g.: submission of an alternative operation) is taken by the Operator agent and the operation is set as rejected (i.e.: it cannot be flown).

As previously discussed, operator preferences (i.e.: the perceived value or welfare of having each operation authorized) are understood in economic terms. Therefore, operators seek to

have as many operations authorized. To this end, operators shall be efficient in flight tokens consumption and operations may be rerouted in some cases (e.g.: congestion policy) to avoid high airspace cost areas. This is possible by converting the flight rerouting problem into a path search problem in which the weight of each node is related to the airspace cost. The A* algorithm is used to search optimal paths (that minimize the token cost) subjected to the original flight plan origin and destination.

Once operations are rerouted, flight tokens are assigned to each operation. As the operator value/welfare function is related to economic benefit, operators assign priority classes to each flight plan so that their total benefit is maximized. In those cases where the number of tokens is scarce (e.g.: pay per token policy), this is done by formulating and solving a mixed integer linear optimization (MILP). The objective function to be maximized is the overall benefit considering an income expectation for each operation depending on its assigned priority class (which determines if that income is finally realized, i.e..: if the operation is flown). In this sense, it is assumed that if operations within a certain priority class (pc_i) are approved with probability $auth_{rate}^{pc_i}$, the total expected revenue from those operations will be the cumulative income multiplied by that probability. The defined MILP problem seeks to maximize this expected benefit subject to policy constraints (e.g.: available tokens, prices...). In general, this results in operations which yield higher incomes at a lower token cost to be given higher priority classes.

Particularly, the following MILP is used to decide which tokens to buy for each operation within the pay per token policy (thus implementing previous g function). In this formulation, the decision variable is denoted as d_{fp_f} which encodes the decision taken for fp_f. At the same time, each variable consists in four possible values ($d_{fp_f}^{pc_{HIGH}}$, $d_{fp_f}^{pc_{MEDIUM}}$, $d_{fp_f}^{pc_{LOW}}$, $d_{fp_f}^{NO\ FLIGHT}$) that can only take {0, 1} values. Respectively, they represent whether a $HIGH$, $MEDIUM$, LOW priority class should be selected or if the flight should be avoided. Also, according to previous nomenclature, the token cost of a flight is denoted as TC_{fp_f}, and the price of a token in each of the classes is expressed as: $tk_{price}\left(pc_{HIGH}\right)$, $tk_{price}\left(pc_{MEDIUM}\right)$, $tk_{price}\left(pc_{LOW}\right)$. From this variables, the expected revenue of a flight can be defined as:

$$\begin{aligned} expected\ income_{fp_f} = \ inc_{fp_f} \ \cdot (auth_{rate}^{pc_{HIGH}} \cdot d_{fp_f}^{pc_{HIGH}} + auth_{rate}^{pc_{MEDIUM}} \cdot d_{fp_f}^{pc_{MEDIUM}} + \\ auth_{rate}^{pc_{LOW}} \cdot d_{fp_f}^{pc_{LOW}} + 0 \cdot d_{fp_p}^{NO\ FLIGHT}) \end{aligned} \tag{3.79}$$

$$\begin{aligned} \cos t_{fp_f} = \ TC_{fp_f} \cdot (tk_{price}\left(pc_{HIGH}\right) \cdot d_{fp_f}^{pc_{HIGH}} \\ + k_{price}\left(pc_{MEDIUM}\right) \cdot d_{fp_f}^{pc_{MEDIUM}} + \ tk_{price}\left(pc_{LOW}\right) \cdot d_{fp_f}^{pc_{LOW}} + 0 \cdot d_{fp_f}^{NO\ FLIGHT}) \end{aligned} \tag{3.80}$$

$$expected\ revenue_{fp_f} = expected\ income_{fp_f} - cost_{fp_f} \tag{3.81}$$

where Equation 3.79 defines the expected income as a function of the selected priority class, Equation 3.80 defines the expected economic cost of purchasing the required tokens depending on the priority class and Equation 3.81 computes the difference to obtain the expected revenue. Finally, the MILP is defined as:

$$\max \sum_{\forall fp_f \in FP_{op_f}} expected\ revenue_{fp_f} \tag{3.82}$$

$$Subject\ to: d_{fp_f}^{pc_{HIGH}} + d_{fp_f}^{pc_{MEDIUM}} + d_{fp_f}^{pc_{LOW}} + d_{fp_f}^{NO\ FLIGHT} = 1;\ \forall fp_f \in FP_{op_p} \tag{3.83}$$

$$expected\ revenue_{fp_f} \geq 0; \qquad \forall fp_f \in FP_{op_p} \tag{3.84}$$

where Equation 3.82 formulates the objective function (i.e.: maximization of the total revenue), the restriction in Equation 3.83 guarantees that only one decision is taken and the one in Equation 3.84 ensures that the operator does not incur in losses.

USSP services

USSP services manage the authorization workflow, detect conflicts between submitted operations and take a final authorization decision. Particularly, the reference Flight Authorization Service (FAS) implementation has been extended to consider both:

- the regulation-mandated FC-FS (which will serve as a baseline),
- and the proposed workflows (i.e.: BTA and GTA) with deferred authorization and multiple priority queues (N=3, denoted LOW, $MEDIUM$ and $HIGH$ priorities for better legibility).

The existence of special operations is not contemplated. The reference Strategic Conflict Detection (SCD) service is also used. It detects conflicts by checking for 4D volume intersections between the flight plan under analysis and other possible conflicting flight plans (considering the different priority classes if needed). In absence of conflicts, a flight plan is approved. However, as a limitation, Strategic Conflict Resolution is not used. Thus, conflicting u-plans are automatically rejected without considering the effect of a possible operation replanning to avoid conflicts (i.e.: strategic deconfliction). From the results point of view, this implies that the obtained authorization data will correspond to the worst case scenario and that better authorization rates may be actually obtained as a result of operation changes due to the deconfliction process.

CISP services

Finally, the implemented CISP (Common Information Service Provider) system consists in a centralized entity extending the role of the CISP by including data exchange and fairness management features. Thus, considered services include a centralized Flight Database (FD) and the Authorization Policy Service (APS) functionality. USSP services register in the CISP's FD all submitted flight plans and update them accordingly once a final authorization decision is taken. Token distribution and airspace cost setting is performed here.

To demonstrate the flexibility of the proposed framework, the 6 policies detailed in Section 4 can be simulated. As a reminder:

- the unlimited tokens distribution policy (hereby denoted `UNLIMITED`, and described in Section 3.3.3.1),
- the scarce tokens policy with uniform distribution among operators (i.e.: `SCARCE_UNIFORM`, explained in Section 3.3.3.2),
- the scarce tokens policy with non-uniform distribution based on a proportional assignment based on historical flight patterns (i.e.: `SCARCE_PROPORTIONAL_FLIGHTS`, depicted in Section 3.3.3.3)
- the pay-per-token policy (i.e.: `PAY_PER_TOKEN`, laid out in Section 3.3.3.4),
- a pay-per-airspace policy based on the GTA workflow with a scarce, uniform number of tokens per operator (i.e.: `PAY_PER_AIRSPACE`, proposed in Section 3.3.3.5),
- a congestion management policy also based on the GTA workflow with a scarce, uniform number of tokens per operator (i.e.: `CONGESTION`, presented in Section Section 3.3.3.6).

Implementation details for each policy have already been commented in the referenced subsections. Moreover, the CISP implementation is also able to work with a FC-FS authorization workflow for comparison purposes.

3.3.4.3 Evaluation metrics

Once results are obtained (in terms of which priority classes have been assigned to each flight, which flights are finally authorized...) several metrics may be used for their analysis. As previously discussed in Section 3.3.1 , several metrics exist to assess fairness in UAM use cases. However, it can be argued that those are not applicable to the U-Space ecosystem and this validation set-up in particular since operations are not rerouted/delayed but either authorized/rejected. Also, it is not the objective of this thesis to propose a single metric to measure fairness or congestion comprehensively. Therefore, for the purpose of validation, fairness among different operators will be measured by analyzing the **flight authorization rate** (or authorization probability) which measures the number of authorized flights ($N_{authorized}$) over the total number of submitted flights ($N_{submitted}$):

$$auth_{rate} = \frac{N_{authorized}}{N_{submitted}} \quad \in [0,1] \tag{3.85}$$

This figure may also be computed on a per operator basis or using temporal, income or priority class filters. By applying those filters it will be possible to compare the authorization rates obtained by different operator groups and assess if they are treated equally or not, thus assessing if a given notion of fairness is achieved.

Given a saturated airspace, its **occupation efficiency** can be measured globally in terms of the number of operations that can be flown. The more missions that can be flown, the more efficient the use of airspace. Without reaching the saturation point, airspace use efficiency can be measured from the average airspace occupation of approved flight plans (i.e.: $E(o_{fp})|_{fp\ is\ approved}$). Thus, a more efficient use of airspace will result in a lower occupation per flight plan.

Finally, **airspace congestion** can be measured by analyzing airspace occupation from a geographic and temporal perspective, which is possible using the congestion grid already defined in Equation 3.76 .

3.3.5 Results and discussion

Using the previous experimental setup, the three scenarios identified in Table 3.12 have been simulated. In each one, the FC-FS workflow is used as a baseline to compare it against the proposed workflow and policies, showing how they allow implementing different notions of fairness. The first scenario (Section 3.3.5.1) demonstrates how the proposed, deferred authorization workflow manages to break the FC-FS bias when authorizing flights. Also, it showcases how the BTA workflow can be used to implement different fairness policies that benefit certain operator behaviors and how operators can inform their preferences through an appropriate selection of priority classes (in order to optimize their benefit). In turn, the second scenario (Section 3.3.5.2) explores the GTA workflow and proves how geographical incentives can be used to benefit local operators. Finally, the third scenario (Section 3.3.5.3), presents how the proposed framework enables to jointly manage fairness and congestion, allowing for preemptive demand and capacity balancing.

Each scenario consists of 100 Montecarlo runs to ensure statistical soundness of the results. Within each run, 16 operators are simulated with a total of 1600 flights over 24 hours, in an area of around 500 km^2. Different operator behaviors are considered in each case to analyze how they are treated by the selected policy.

Scenario	Tested operator behaviors	Tested workflow and policies
1	A combination of: fair/greedy, early filler/late filler, low flights/high flights, low income/high income operators.	*FC-FS, UNLIMITED, SCARCE_UNIFORM, SCARCE_PROPORTIONAL_FLIGHTS, PAY_PER_TOKEN*
2	Local (area-based) operators vs. general operators.	*FC-FS, PAY_PER_AIRSPACE*
3	Not Applicable	*FC-FS, CONGESTION*

Table 3.12: Considered scenarios description

3.3.5.1 Scenario 1: token allocation evaluation

This validation scenario aims to demonstrate the implementation of different fairness policies using exclusively the token distribution logic (BTA workflow). For this purpose, the same

scenario has been run for baseline (FC-FS) operation and for five different policies (as shown in Table 3.12).

A first element to verify is whether the proposed mechanism reduces or eliminates the dependence between the probability of authorization and the flight plan submission time. This is explored in Figure 3.43 that depicts the authorization rate for flights submitted at different times before take-off. It can be seen how in the FC-FS case (or FC-FS, in dark blue) the authorization probability strongly declines as flights are submitted closer to the take-off time (i.e.: with less time in advance). Conversely, and as desired, in the rest of the simulated cases (which implement the proposed deferred authorization workflow) this dependency is almost eliminated, as demonstrated by the horizontal lines.

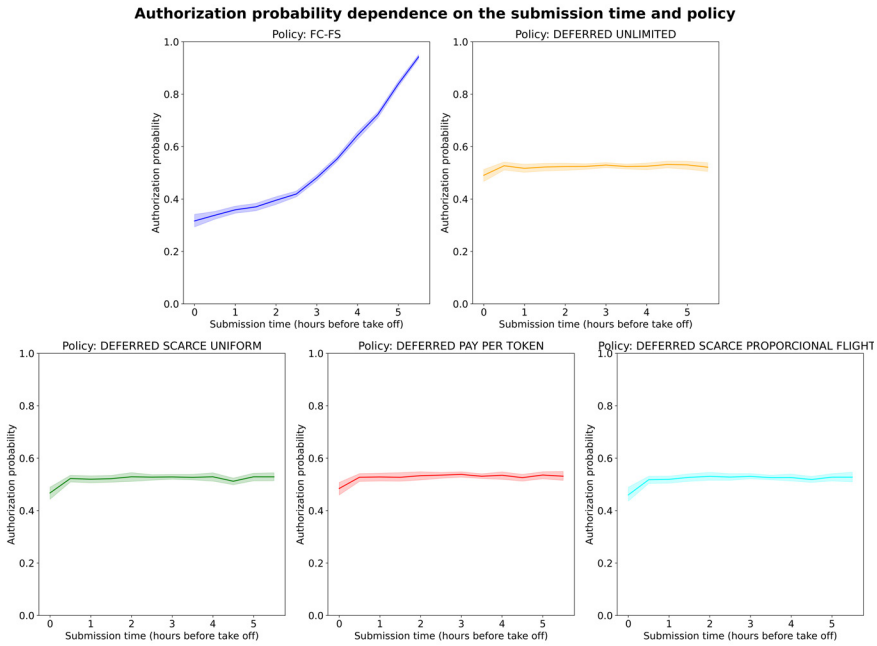
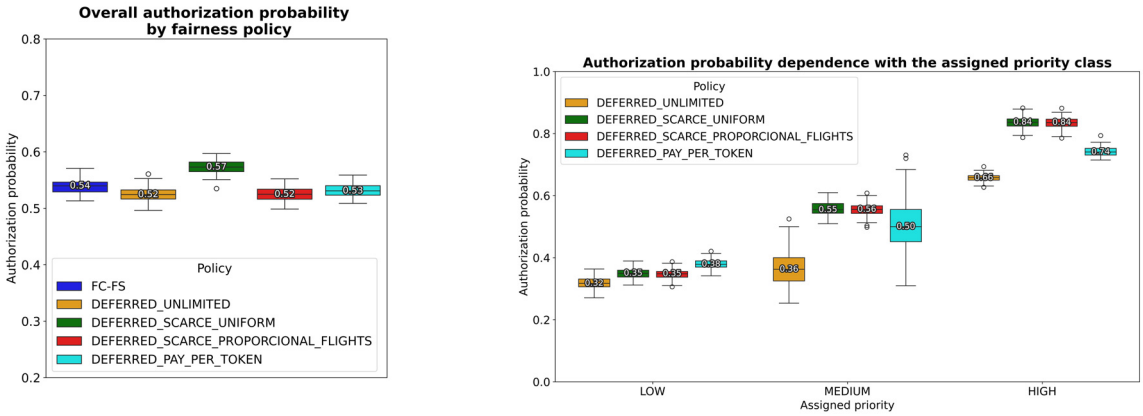


Figure 3.43: Scenario 1 results. Analysis of the dependence of the flight authorization rate and the submission time.

Flight authorization rate is represented by the authorization probability in the y axis. Submission time is measured as the number of hours before takeoff in the x axis. Color coding indicates the different policies that have been tested. Colored lines indicate the mean value across the 100 montecarlo runs, decolored areas represent a confidence interval of 99.7%

This fairness improvement is achieved without significantly impacting the overall authorization rate. This can be assessed in Figure 3.44a where the average authorization probability remains almost constant regardless of the chosen workflow and policy. Another advantage of the proposed authorization mechanism is that operators can inform the U-Space system of their flight priorities and these preferences can be considered in the authorization process. In this sense, Figure 3.44b shows how operations submitted to different priority classes are treated differently in terms of authorization success. It can be seen how if an operator informs the system of the criticality of an operation by assigning it a high priority class, significantly higher approval probabilities are obtained. Conversely, low priority operations obtain lower authorization probabilities. This behavior is consistent notwithstanding the selected token

distribution policy. In addition, since the utility of each operation has been aligned with its economic value, it would be expected that those operations with a higher income would be more likely to be approved. To this end, the income and authorization probability dependency has been depicted in Figure 3.45 . Indeed, it is shown that contrary to the FC-FS approach where it is not possible to encode each operation value (i.e.: as demonstrated by the obtained horizonal line), those flights which yield higher incomes are authorized more (as evidenced by the increasing trends in the rest of policies).



(a) Impact on overall authorization probability

(b) Impact of selected priority class on authorization probability

Figure 3.44: Scenario 1 results. Comparison by token allocation policy

Subfigure 3.44a depicts the overall authorization probability (considering all operators) for each of the tested policies (including the FC-FS baseline). Subfigure 3.44b shows the authorization probability stratified by the priority class a flight plan has been submitted to (x axis). In both subfigures, boxplots are used to represent the statistical distribution of the values across the different Montecarlo runs. Median values are depicted within each boxplot. Color coding is used to differentiate the different policies.

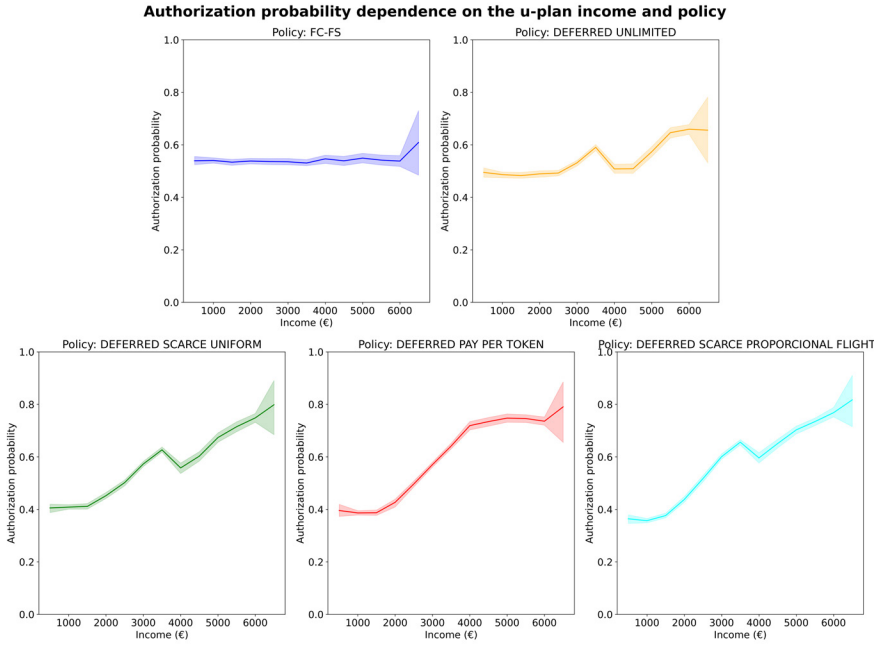


Figure 3.45: Scenario 1 results. Analysis of the dependence of the flight authorization rate and the operator perceived criticality/preference for each operation.

Authorization probability is shown in the y axis. Operator preferences are modelled from an economic point of view as the received income, depicted in the x axis. Color coding indicates the different policies that have been tested. Colored lines indicate the mean value across the 100 montecarlo runs, decolored areas represent a confidence interval of 99.7%

So far it has been demonstrated that the proposed authorization mechanism breaks the FC-FS dependency, allows operators to express their preferences and does so without affecting the overall authorization rate. The following figure (Figure 3.46) allows to analyze the impact on efficiency. To do so, for each of the tested policies within the proposed framework, it depicts the average airspace occupation per authorized operation compared against the occupation in the FC-FS workflow. It can be seen how the proposed mechanism together with an appropriate policy can also slightly increase the efficiency of the U-Space systems (i.e.: relative values are below 1). This happens for token distribution policies that limit the number of tokens (i.e.: `SCARCE_UNIFORM`, `SCARCE_PROPORTIONAL_FLIGHTS`, and indirectly for the `PAY_PER_TOKEN` case). In those cases, operators try to maximize their profit by prioritizing those flights that yield a higher revenue with a lower consumption of tokens. Thus, they are implicitly selecting the most efficient operations. Efficiency improvements are small since the duration of operations is fairly homogeneous. However, if operators with different levels of efficiency were introduced, it would be expected that more efficient operators would be able to authorize a greater number of operations.

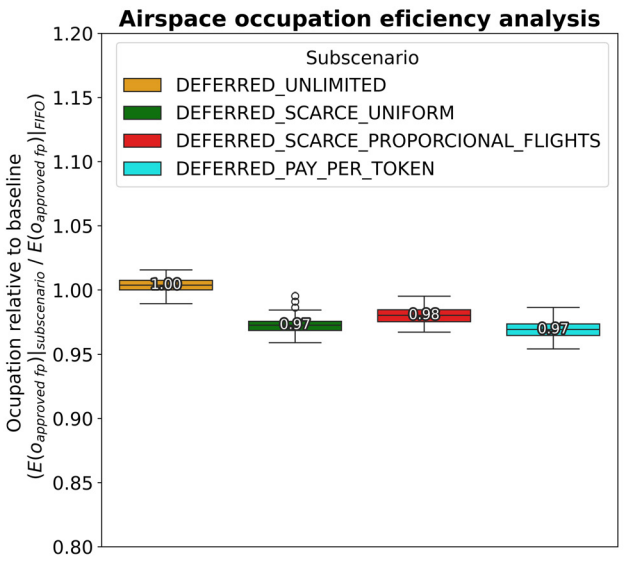


Figure 3.46: Scenario 1 results. Analysis on the impact on efficiency of each implemented policy. Efficiency (in x axis) is measured as the mean occupation per u-plan relative to the occupation in the reference scenario (FC-FS). A value of 1 indicates that there is no change in efficiency, whereas a value below it indicates that there is an efficiency improvement. Results for different policies are shown in different colors. Boxplots are used to represent the statistical distribution of the values across different Montecarlo runs. Median values are depicted within each boxplot.

Finally, Figure 3.47 allows to showcase the flexibility of the proposed authorization workflow in implementing different fairness notions. In it, the relative authorization advantage obtained by operators with different behaviors (color legend) is shown for different workflows (FC-FS) and token distribution policies. For example, in the regulation mandated FC-FS case, operators who send flight plans early get an authorization rate that is on median 1.66 times than those who send operations later. As previously demonstrated and also shown in this graph, this advantage is eliminated if the proposed framework is implemented regardless of the fairness policy. Nevertheless, the choice of policy by regulators is crucial as depending on it, either some behaviors are incentivized over others, or fairness is obtained. For instance, if the number of available tokens is not limited (i.e.: `UNLIMITED`), those operators who do not faithfully report the criticality of their trades are benefited. However, even if tokens are limited to avoid the latter, other dependencies with the number of flights (if tokens are evenly distributed, i.e.: `SCARCE_UNIFORM`) or with operators' revenues (if a price per token is introduced, i.e.: `PAY_PER_TOKEN`) may be introduced. In the former case, small operators can take advantage of having a larger number of tokens per submitter flight to enter flights to higher priority classes. In the later, operators with higher incomes can pay for more tokens than low income operators. Overall, a scarce token distribution based on the historical flight patterns of operators (i.e.: `SCARCE_PROPORTIONAL_FLIGHTS`) seems to eliminate these biases and reaches fairness. This statement assumes fairness as obtaining a similar authorization rate for all operator cohorts. However, other notions of fairness could also be considered and analyzed.

In sum, scenario results show that the **proposed framework meets its expectations in terms of FC-FS bias reduction, preference sharing and flexibility**. Also, by

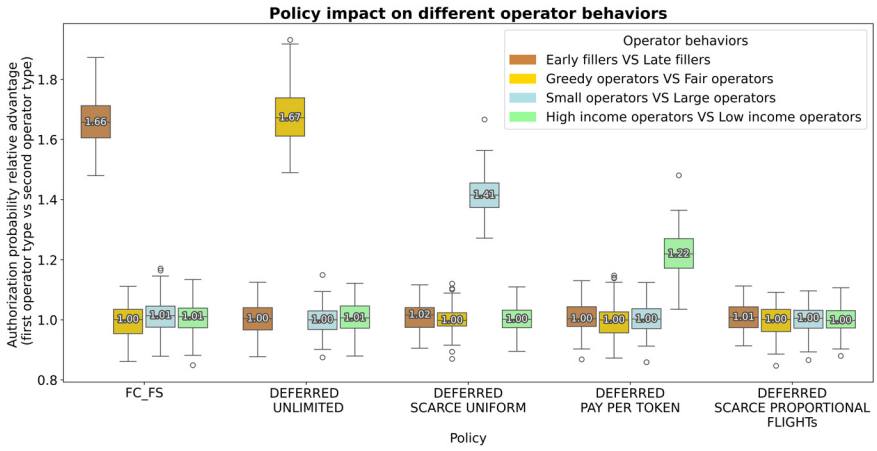


Figure 3.47: Scenario 1 results. Analysis on the impact of policy to different operator behaviors. Behaviors are grouped in comparison pairs that are depicted in different colors. The authorization probability for each behavior within the pair has been divided to get the relative advantage each behavior obtains (shown in y axis). The relative advantage is computed as follows: authorization rate of first group in pair divided by authorization rate of second group in pair. Therefore, a value of 1 indicates that no advantage is obtained. A value above 1 indicates that operators with the first behavior in each pair are benefited over operators with the second behavior. The comparison is made for different policies (in the x axis). Boxplots are used to represent the statistical distribution of the values across different Montecarlo runs. Median values are depicted within each boxplot.

implementing different fairness policies through the distribution of tokens (BTA workflow) it is possible to incentivize different behaviors or types of operators over others. Therefore, it can be concluded that the proposed mechanism allows to increase the fairness of the system given an appropriate choice of policy. Moreover, it does so without compromising safety since the conflict detection process ensures that authorized operations are free of 4D intersection, as also does the FC-FS mandated authorization process.

3.3.5.2 Scenario 2: geographical incentives evaluation

The second scenario (whose results are shown in Figure 3.48 and Figure 3.49) is aimed at showing the implementation of incentives per operator based on the cost of airspace (GTA workflow). In this case, two types of operators are simulated: some that fly mainly in limited geographical areas (area operator) and others that operate throughout the entire area under study (general operator) are simulated. Following the airspace reservation policy (i.e.: `PAY_PER_AIRSPACE`), the former are able to reserve their area of operation.

As detailed in Section 3.3.3.5 , airspace reservation is encoded through an operator-dependent airspace cost. This cost is shown as a 2D heatmap in Figure 3.48 . It depicts how operator B, of the "area" type, has a very low airspace cost (green grid cells) in the reserved area that concentrates its flights. In the remaining areas it has a nominal (gold cells) or high cost (orange cells) depending on whether the area is reserved by other operators. On the other hand, operator A, of the "general" type, can only operate at normal cost in a reduced set of zones because they have been reserved by other operators and it is disincentivized to operate in those zones.

The implementation of these geographical incentives has an impact in the selection of priority classes per operator that is shown in Figure 3.49a . It can be seen how area type operators

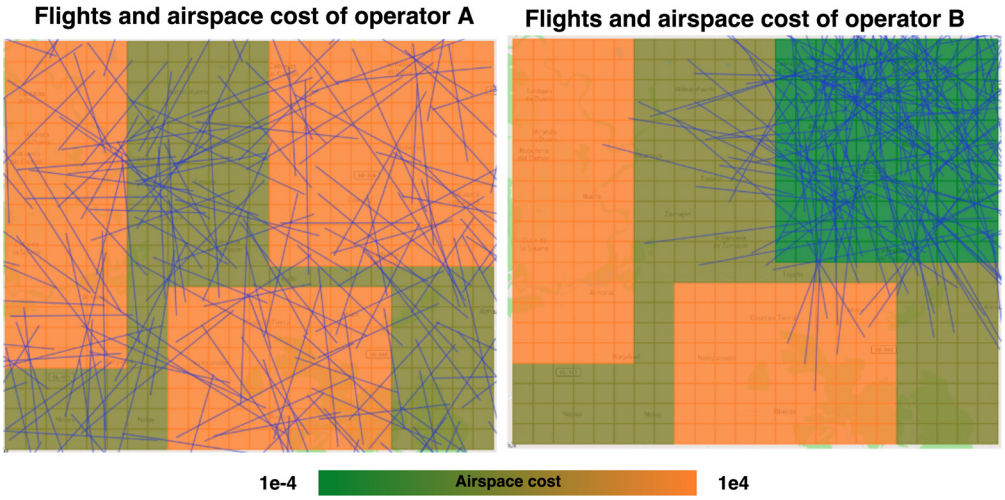
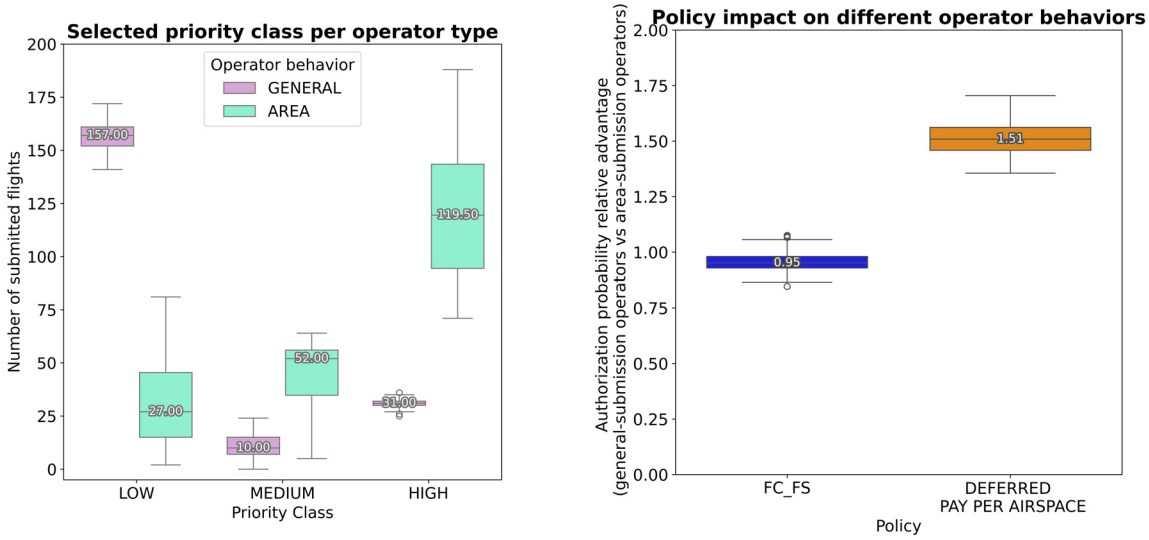


Figure 3.48: Scenario 2 results. Comparison of airspace cost comparison per operator type
Figure depicts the airspace cost grid for two different operators as a heatmap (A operator, depicted on the left, has a general distribution of flights and B operator, depicted on the right, has a local distribution of flights).

can send most of their operations with high priorities (as they have a very low airspace cost in the areas where they operate). On the contrary, general operators have to limit themselves to low priority classes as they do not have reserved portions of airspace.

As a result of this operator-dependent priority class distribution, different authorization priorities are obtained for each operator type. Figure 3.49b depicts the relative authorization rate of area operators over general operators for the baseline (i.e.: FC-FS) and `PAY_PER_AIRSPACE` policies. It can be seen how geographical-based incentives manage to prioritize area-based operators over general operators. The former manage to authorize on median 51% more flights than the general operators.



(a) Analysis on priority class selection per operator type

(b) Analysis on authorization advantage per operator type

Figure 3.49: Scenario 2 results. Comparison per operator type.
Subfigure 3.49a depicts the number of flights (y axis) that have been submitted to each priority class (x axis) by operators with two different behaviors (color code). Subfigure 3.49b depicts the authorization probability advantage obtained by general operators over area operators (y axis) for the two tested workflows (x axis). A value of 1 indicates that no advantage is obtained whereas a value above 1 indicates that area operators are benefited over general geographic distribution operators. In both subfigures boxplots are used to represent the statistical distribution of the values across different Montecarlo runs. Median values are depicted within each boxplot

3.3.5.3 Scenario 3: congestion management evaluation

Finally, the third scenario (whose results are depicted in Figure 3.50 to Figure 3.52) shows how the proposed framework enables to simultaneously enforce fairness and perform demand-capacity balancing. Indeed, thanks to the use of geographic incentives (i.e.: GTA workflow with `CONGESTION` policy) it is possible to preemptively avoid congestion. To implement this policy, the aggregate demand of all flight plans considered in a simulation run is used to estimate congestion in different cells of a geographical grid. In this sense, Figure 3.50 , left subfigure, showcases the congestion estimate as a heatmap and reveals the existence of several flight hotspots (orange colored areas). This estimate is used to set high airspace costs in those cells that are congested (as detailed in Section 3.3.3.6).

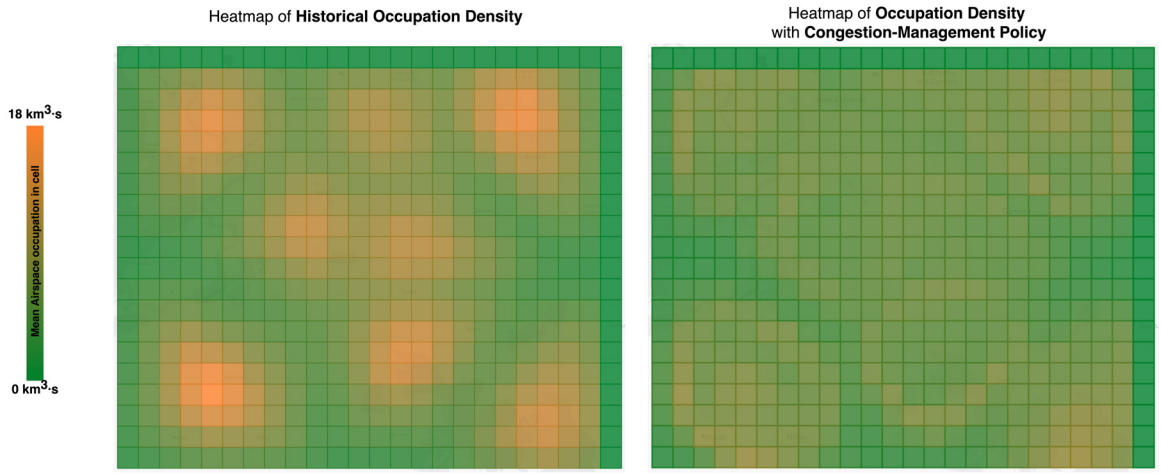


Figure 3.50: Scenario 3 results. Geographical comparison of airspace congestion without and with the implemented policy

Figure on the left shows the historical average airspace congestion in the area under analysis. Congestion values are color coded as a heatmap with high congestion values depicted in orange. Subfigure on the right depicts the average airspace occupation per cell (also as a heatmap, in the same scale than the previous subfigure) when applying the proposed congestion management policy

As a consequence, operators are compelled to avoid those zones by rerouting (as explained in Figure 3.3.4.2) if it is compatible with their business logic (i.e.: if they are able to reroute their operations). As a result, drone occupation density is dispersed throughout the whole area. This is supported by subfigure Figure 3.50 (right subfigure) which depicts the resulting occupation heatmap after applying the congestion management policy. It clearly shows that the previous orange hotspots have been dissolved. Statistically, subfigure Figure 3.51 estimates a probability density function for the airspace occupation metric. In it, the reference distribution (i.e.: FC-FS, in blue) presents a tail with high occupation values (corresponding to the cells with hotspots) while the distribution with the proposed protocol (in red) concentrates the occurrences in lower occupation values.

However, this reduction in airspace occupation per cell could be due to a significantly lower flight authorization rate. For this purpose, Figure 3.52a compares the authorization probability in the FC-FS case and with the proposed mechanism. Although there has been a reduction in the authorization rate when applying the congestion management mechanism, this change is small and could be explained by the increase in total occupation shown in Figure 3.52b . Indeed, by having to reroute operations to avoid congested areas, operations are longer and new conflicts may arise, which explains the change.

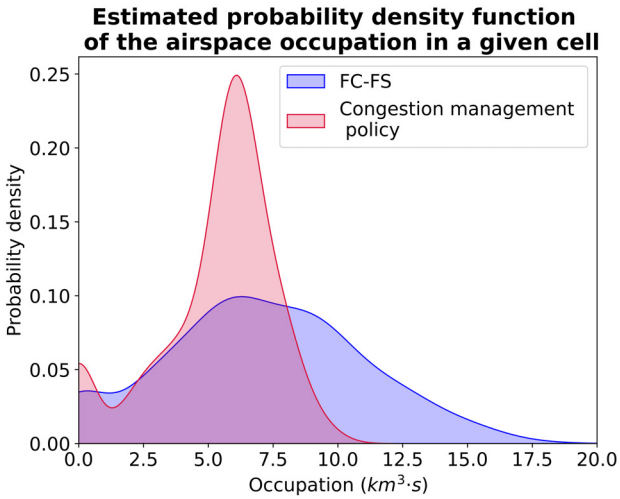
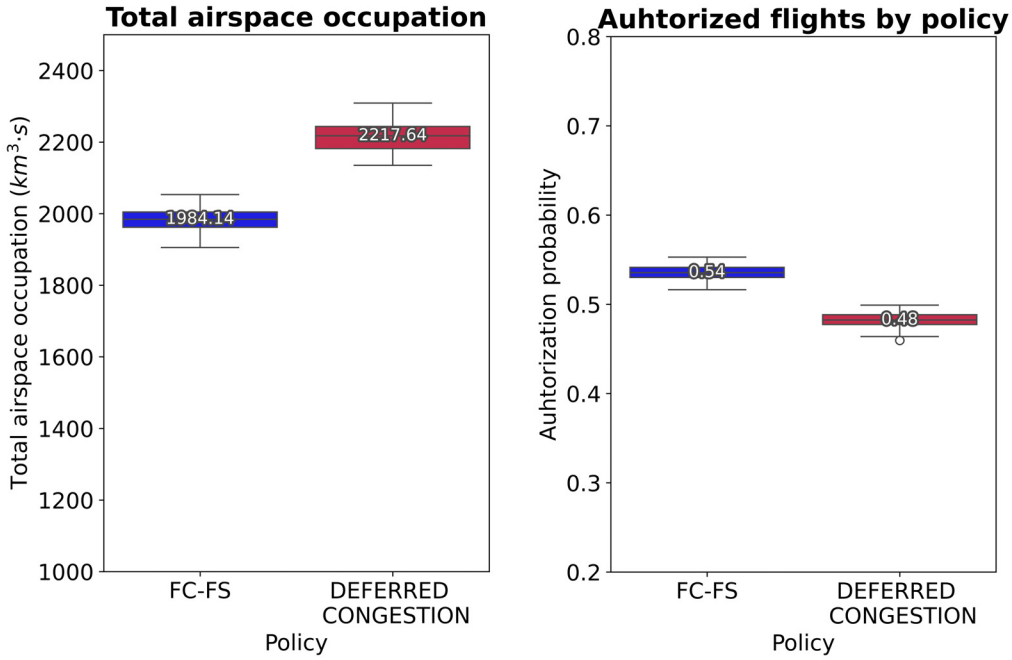


Figure 3.51: Scenario 3 results. Statistical comparison of airspace occupation without and with the implemented policy

Analysis of airspace cell occupation (x axis) across all runs. A probability density function (y axis) has been estimated with data from all Montecarlo runs using a Kernel Density Estimation method. Color coding compares the results for the FC-FS case and the congestion management policy.



(a) Subfigure compares the overall authorization probability (y axis) for each of the two tested policies (x axis)

(b) Subfigure depicts the overall airspace occupation (overall sum of the occupation of all authorized flights, shown in y axis) in each policy (y axis)

Figure 3.52: Scenario 3 results. Occupation and authorization rate analysis.

In both subfigures boxplots are used to represent the statistical distribution of the values across different Montecarlo runs. Median values are depicted within each boxplot

3.3.6 Conclusion

As the number of drone operations increases in smart territories, it is expected that issues related to inter-operator fairness and congestion management, among others, will arise. However, the strategic flight plan authorization process currently set out in the European regulation is not adequate to deal with these issues. Existing literature (discussed in Section 3.3.1) only deals with this problem broadly, without proposing a feasible comprehensive technical framework that is somehow compatible with European regulations. Also, scalability issues are overlooked in some cases. As an alternative, **this section has proposed an efficient, fair, simple and scalable alternative authorization workflow to address this problem**. The contribution on this section not only goes beyond the state of the art by enabling a novel, scalable technical framework; it also embraces some pre-existing proposals (i.e.: pay-per-fly, reserved air corridors) unifying their implementation into a single framework.

In Section 3.3.2 it has been discussed how the proposed alternative is articulated around a deferred authorization phase based on the determination of a time (RTTA) in which all operations in potential conflict are evaluated simultaneously. In addition, it introduces different priority classes so that operators may guide the u-space system by indicating their preference in each operation according to their business logic. Access to each priority class is controlled by a set of scarce tokens that are distributed among operators. Each operation sent to the U-Space system must be accompanied by a sufficient number of tokens related to airspace occupancy in a way that promotes efficiency and the correct declaration of priorities. Furthermore, the number of tokens also depends on the occupied area and when it is occupied, so that temporal and geographical incentives can be established.

This comprehensive logic allows for the joint implementation of a broad set of fairness and capacity management policies (e.g.: homogeneous and heterogeneous distribution of tokens, assigning an economic value to tokens or to airspace, and congestion disincentivization) as the ones described in Section 3.3.3 . Its technical feasibility has been demonstrated by integrating the proposal into the USPASIM simulation framework (previously presented in Section 3.2.2) that has served as a test bed to compare the previous policies with the regulation-mandated FC-FS approach. Furthermore, the proposal allows its implementation in distributed U-Space systems as it decouples authorization decisions from each of the USSPs, making it a scalable system as well. Finally, the proposed mechanism is simple and transparent from the operator's point of view, abstracting him from the underlying logic in charge of fairness management through the figure of the token. In short, the proposed framework meets the desiderata previously described in [SFD22] for an alternative authorization system.

On **limitations and future work**, the validity of the proposed framework has only been explored from a formal and technical point of view. The opinions of the regulators and the operators themselves on its possible practical implementation should be gathered as a future work. On the other hand, even if different fairness policies have been identified and implemented, a selection of an optimal policy for real u-space systems has not been performed. In this sense, future work could consist in implementing a token allocation policy based on a Mixed Integer Lineal Programming optimization model, assisted by simulation results to estimate the authorization rates obtained in each priority class. Also, Reinforcement Learning models trained within the simulated U-Space environment could yield optimal token

allocation policies. Even if not mandated by regulation, another limitation of the study is that Strategic Conflict Resolution services have not been considered. Future studies should analyze how it interacts with the proposed token-based authorization framework, possibly requiring limitations on what changes are allowed on deconflicted operations. Another open point to be analyzed is the robustness and fault tolerance of the framework in the face U-Space communication failures or attempts by operators to abuse it. Finally, the applicability of such an approach to Urban Air Mobility schemes with ground infrastructures (i.e.: vertiports) should be explored. The study of these gaps as future work will bring the proposal closer to its possible implementation in U-space systems.

——— * * * ———

Chapter summary

Within smart territories, smart services take advantage of real time information and pervasive connectivity to improve services provision or even enable novel applications. In this context, drones are expected to enable smart services such as monitoring, delivery and mobility services. Yet, for this to be possible, the U-Space ecosystem is needed as the main enabler to safely and efficiently operate UAVs. As presented in Section 3.1 , this ecosystem consists of a set of interconnected and automated systems that provide services to drone operators, ANSP and authorities (among others). Therefore, U-Space can be considered in itself as a smart service that make it possible to control airspace usage by maintaining aircraft separation a priori (before flight, through strategic deconfliction services) and during flight (services involved in the tactical phase).

One of the problems encountered in the deployment U-Space services is the difficulty in testing and validating their operation. In this sense, the first contribution of this chapter (described in Section 3.2) has presented a simulation framework that models the entities involved in the ecosystem. Particularly, the contribution focuses on the simulation of human actions by operators and pilots, together with the modeling of surveillance networks (with collaborative and non-collaborative sensors).

Looking ahead, airspace is going to become a scarce and coveted resource for operators. It is therefore necessary to ensure its efficient and fair use. For this purpose, Section 3.3 has addressed the problem of equity between operators. Thereby, this section's contribution consists in a new strategic flight authorization framework that allows managing fairness and congestion through token-gated authorization priorities and a set of geographical incentives.

Overall, these contributions will help the development of smart-territories supported by commercial drone operations coordinated through the U-space ecosystem.

Chapter 4

Simulation-based decision-making in smart territories

Informed decision-making relies on data, computational models and expert knowledge to build a better understanding of the decision problem including relevant decision alternatives, their outcomes and risks. Also, it allows reflecting on past decisions analyzing if the desired outcomes have been met and whether undesirable second-order effects have been generated. Overall, such an approach accelerates decision-making processes and improves decisions effectiveness while rationalizing the use of economic and human resources.

In this context, a Decision Support System (DSS) is an interactive computer-based tool that assist decision-makers in making informed decisions. As previously discussed, DSSs constitute one of the technological pillars of smart territories that help take informed decisions at different levels:

- **Operational/tactical decisions**, which relate to short-term decisions on the usual operation of a system. An example of this type of decision would be the response implemented in a given infrastructure to react and adapt its operation once a threat has been materialized.
- **Strategic decisions**, which relate to medium/long-term decisions and goals. In this case, a strategic decision would be an investment that could increase the future capability of a smart-territory's asset.
- **Regulatory decisions** that define the legal framework that governs all entities and systems within a smart territory, thus outlining constraints and conditions on their operation.

To this end, DSSs can take advantage of different techniques, which are aligned with the classification in Section 1.1 :

- **Visualization and data analysis**. At a first level, a DSS can simply consist of a dashboard that allows to visualize information on a smart-territory so that the decision-maker can analyze a given situation and detect possible interdependencies. Also, data analysis algorithms may be included to detect patterns in gathered data. In this sense,

the mHealth platform proposed in Section 2.2 is an example of a DSS that helps taking operational decisions based on data. In general, IoT platforms provide real-time data acquisition and analytics that can be fed to a decision-making process, thus serving as a data-based DSS.

- **Prediction and simulation**. Going a step further, estimation algorithms can help to extrapolate a given situation in order to predict future events. Thereby, decision-makers can anticipate decisions, avoiding undesirable situations or increasing the decisions' effectiveness. With such models, it is possible to create simulation tools that consider comprehensively various complex interrelationships within a territory. This provides the decision maker with a flexible environment in which the outcomes of different scenarios may be evaluated (i.e.: what-if analysis), and decision alternatives impact optimized.

Moreover, it is also possible to integrate simulation capabilities with real-time data and analysis. This is the case of **digital twins** that, as previously discussed in Section 2.1.4 , allow a seamless interoperation between simulated and real entities. From a decision making point of view, this is useful as it would allow automating models' parameterization and calibrating them with real data. This way, it would be possible to quickly react to time-sensitive problems. Also, by integrating physical actuators into a city's digital twin, it would be possible to automatically implement decisions. Overall, digital twins represent a high integration level across the different technological pillars of smart territories. In digital twins, monitoring technologies and DSSs would be combined synergistically to deliver efficient smart services. Thus, digital twins would fulfill the overall conceptual architecture for smart territories identified in Section 1.3 .

From the above classification, this chapter is going to explore the usage of simulation models for decision-making purposes at operational, strategic and regulatory decision levels:

- In Section 4.1 , operational and strategic decision-making is considered to perform risk-informed planning in smart territories. To this end, a hybrid simulation framework is proposed to model interdependent Critical Infraestructures (CIs) comprehensively. As a validation, the framework has been instantiated in a novel use case that considers relationships between healthcare and ICT infrastructures. The model is used in a decision scenario to showcase how it can provide useful metrics at different aggregation levels to compare decision alternatives. Thus, it is demonstrated that the proposed framework constitute a useful DSS to plan and operate smart infrastructures.
- Section 4.2 leverages the USPASIM framework (proposed in Section 3.2) to inform regulatory-level decisions in the U-Space ecosystem. In this sense, the section discusses how unmanned aerial operations can be represented through different data models that impact on airspace usage efficiency and computational requirements. Thus, in order to enable a high density of operations, regulators might need to adopt a new u-plan format. This decision can be based on the insights provided in this section through simulation experiments.

4.1 A hybrid simulation framework for operational and strategical resiliency enhancing decision making

As briefly hinted in Chapter 1 , smart territories are complex systems composed of multiple inter-reliant domains that include social and economic interactions, governance networks and CI systems [MNS16]. Even if this complexity makes smart territories poles of economic prosperity, sustainability and innovation, it also makes them especially **vulnerable to hazards whose impact can propagate across different infrastructures**.

In recent years, modern societies have experienced the occurrence of risks that have demonstrated the wide-ranging consequences that they entail. For instance, the SaRS-COV-2 pandemic has generated unprecedented disruption in cities [NO20]. In addition to the obvious cost in human losses, it changed citizens' mobility, work, and consumption habits; with repercussions that are still visible today. In a short period of time, the imposition of confinements meant a shift from public life to the virtual world with the consequent effect on communications and electricity distribution networks. Examples of compounding threats were also experienced with systematic cyberattacks affecting collapsed hospitals [MS21]. Also, recent conflicts have prompted multiple examples of cascading failures across the electrical grid, water distribution systems, heating systems and mobility systems [Ass22]. These examples prove that infrastructures are often not sufficiently well prepared to cope with existing natural and deliberate threats. As a result, there exists a trend to create **resilient cities and territories**, which can be understood as human settlements that are able to absorb, adapt and recover from the effects of a threat without generalized chaos or harm [UN-18].

Enhancing the resiliency of territories requires a comprehensive risk analysis process that can be used to guide the planning decision process. However, this process is challenging due to the complexity of understanding and modeling infrastructures' operation. In fact, as already explained, to ensure the effectiveness of the planning decisions and make optimal use of available resources, these processes cannot be guided by the intuitions of decision-makers but must be data-driven. In this sense, **DSSs can help in the planning process by providing helpful insights and metrics based on real data or reliable simulation models, contributing to territories resilience** [Rez+23].

As it will be discussed in Section 4.1.1 , some existing challenges in DSSs are the difficulty of providing a comprehensive view of interdependent CIs operation and offering risk-related useful metrics to the decision makers. To mitigate this gap, Section 4.1.2 proposes to analyze a **smart territory as a system of intelligent and measurable systems**. Territories are composed of several interdependent CIs that operate within a geospatial context and are driven by citizens behaviors. Based on these ideas, a conceptual model that provides a holistic, formal view of the territory is suggested. This theoretical contribution defines the different critical systems based on a literature review. Additionally, as a novelty, it proposes to consider a social, urban landscape and environment systems to encapsulate the modeling of citizen behaviors and geospatial interactions. Then, the section proposes to identify interdependences between different systems through influence diagrams. In a smart territory, the operation of each system can be monitored in real time thanks to IoT devices. In line with the data fusion models presented in Chapter 2 , these metrics can be aggregated at different levels

(i.e.: entity, system, and city levels) to assess the operation of the territory and guide the data-driven decision-making process.

As previously introduced, an important feature of modern DSSs is their ability to provide predictability using simulation models. In this sense, Section 4.1.3 builds upon the previous conceptual all-encompassing model of a smart-territory to propose an implementable **simulation framework consisting in a hybrid approach that integrates agent-based modelling and network-based modelling**. Agent Based Models (ABMs) enable studying the emerging behaviors of a territory while network-based modelling allows to easily articulate the interdependences between different entities. Thus, a territory is modelled as a set of autonomous agents (i.e.: entities within the city) each composed by a set of subagents and encompassing stochastic elements. Within an agent, each subagent operates in a different network (i.e.: critical system) modelling dependencies among a single system. Then, inter-system dependencies are modelled within the agent by defining the interactions between the different subagents. The proposed hybrid framework eases the modelling of complex interdependences as it can be implemented using object-oriented-programming enabling component reusability. Moreover, this structure also allows to perform simulator federation which facilitates the integration of pre-existing simulators. Also, as a valuable output for decision-makers, the model naturally provides metrics at different aggregation and explainability levels.

By fusing the conceptual model and the hybrid simulation framework, this section aims to provide a valuable DSS tool that can assist decision-makers in taking informed decisions aimed to enhance infrastructure resiliency. The tool allows data-driven decision making and includes explainable elements to compare decisions. This practical application is explored in Section 4.1.4 where the **proposed simulation model is instantiated to enhance healthcare systems resiliency to compounding threats** (including pandemics and cyberattacks). Therefore, the implemented model considers interdependences between the healthcare and ICT domains. This serves as a validation of the proposed hybrid simulation framework showing how it can easily model different critical infrastructures within a city. Figure 4.1 showcases the proposed DSS in the context of the smart territory conceptual architecture.

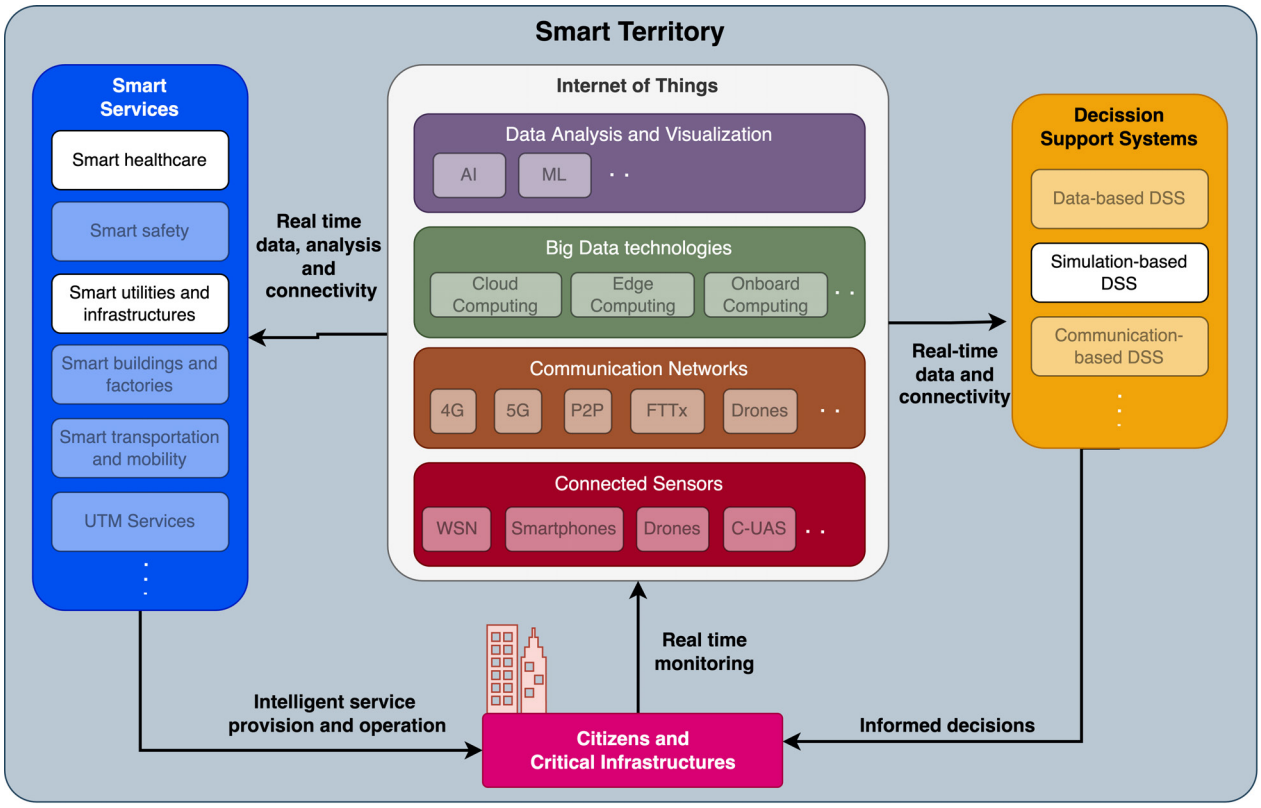


Figure 4.1: Technological elements in the proposed decision-support system
Relevant technological elements are highlighted over the full architecture

Finally, in Section 4.1.5 , a decision-making use case is presented to exemplify the operation of the DSS tool. In this context, and inspired on the SARS-COV-2 situation, the decision-maker must take alternative resiliency-enhancing decisions to mitigate the impact of and a pandemic and concurrent cyber-attacks. The objective of the proposed use-case is two-fold. First, it allows to assess the validity of the instantiated healthcare-ICT simulation model through a literature-based methodology. Second, the scenario proves how the proposed framework can be integrated into a **risk-informed decision-making process and provide useful information to the decision-maker for both operational and strategic decisions**, constituting a powerful DSS.

The proposals and discussion on this section are based on the contributions of this thesis's author to the following publications.

Supporting publications

- **[Car+24a]** - David Carramiñana, Ana M. Bernardos, Juan A. Besada, and José R. Casar. "Towards Resilient Cities: A Hybrid Simulation Framework for Risk Mitigation through Data-Driven Decision Making". In: *Simulation Modelling Practice and Theory* 133 (May 2024), p. 102924. DOI: 10.1016/j.simpat.2024.102924, where the author is the main contributor to a conceptual model to represent smart cities, has designed a hybrid simulation framework that can be used as part of a DSS for risk-informed decision-making, and has validated the proposal.
- **[Car+23]** - David Carramiñana, Ana M. Bernardos, Juan A. Besada, and José R. Casar. "Enhancing Health Care Infrastructure Resiliency Through an Agent-Based Simulation Methodology". In: *2023 International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*. Sept. 2023, pp. 1–4. DOI: 10.1109/ICT-DM58371.2023.10286957, where the author is also the main contributor proposing a model that considers healthcare system interdependences with ICT infrastructures to enhance their resiliency.
- **[Car+25a]** - David Carramiñana, Ana M. Bernardos, Juan A. Besada, and José R. Casar. "Enhancing Healthcare Infrastructure Resilience through Agent-Based Simulation Methods". In: *Computer Communications* 234 (Mar. 2025), p. 108070. DOI: 10.1016/j.comcom.2025.108070, which extends the previous conference paper with new contributions of this author related to improved healthcare systems modeling.
- **[Alb+23a]** - César Alberte, David Carramiñana, Ana M. Bernardos, and Juan A. Besada. "Managing Pandemics Through Agent-Based Simulation: A Case Study Based on COVID-19". In: *18th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2023)*. Cham: Springer Nature Switzerland, 2023, pp. 185–195. DOI: 10.1007/978-3-031-42529-5_18, where the author has helped in the design of the multiagent disease propagation model and supervised its implementation.

4.1.1 Review on decision-support systems for smart, resilient territories

According to [Cam15], threats affecting cities and territories are diverse and can range from natural catastrophes, adverse climatic events or pandemics to financial crises, political instability, hybrid conflicts, terrorism or technological failures (e.g.: cyberattacks, infrastructure failures). In fact, multiple of these threats can materialize simultaneously (known as compounding threats) or even as a set of interrelated and sequential threats (known as cascading threats) is also possible [Wel+22]. Once a hazardous event occurs, it can have a negative impact (i.e.: a series of consequences) on critical systems depending on their vulnerability. Moreover, the occurrence of a threat does not impact only one part of the territory. Due to interdependencies, failures can spread between systems (i.e.: cascading effects) disrupting the territory as a whole, and even generating long-term effects. The intersection of a threat, its

associated likelihood and its impact on a system can be denoted as a risk [RH20].

To implement a resiliency-enhancing planning process, it is important to assess territories' risks: identifying and analyzing hazards and threats to determine their likelihood and consequences. In some cases, analyzed risks can exceed acceptable risk thresholds and urban planning must take appropriate actions to reduce that risk. In this sense, risk mitigation measures can be taken to eliminate or reduce their impact on the territory [God03]. When multiple alternative mitigation actions are possible, a risk-informed decision-making process can be used to compare the different decision options based on risk information. Moreover, within the resilient territory context, **risk mitigation must be understood as a distributed process between different infrastructures that can collaboratively implement actions to benefit the whole settlement**.

As discussed in the introduction of this section, performing risk-informed decision making within smart territories and cities is a demanding task. Previous studies have already proven that DSSs can be used to enhance urban resilience [Rez+23]. In order to position this section contributions, this subsection gathers an analysis of existing tools in the literature. An ample literature review has been previously performed in [Ins+12] showing DSS with varied characteristics:

- considering different **geographical levels** (e.g.: national level [Bus+05] or city level [Elv+21]),
- targeting **different audiences and decision horizons** (strategic [Kam+20] or tactical decisions [Ott+17]),
- providing **different views on the city/territory** (e.g.: a comprehensive interconnected view [Bia17] or only a subset of systems [Ros+20]),
- encompassing **different stages of the decision process** (e.g.: only risk assessment [Elv+21] or supporting the full decision-making [Ros+20]),
- and **generating different types of insights** (e.g.: qualitative, such as CARVER2 described in [Ins+12], or quantitative metrics [Bus+05]).

Overall, authors of [Ins+12] and [Elv+21] conclude that there is a gap in how to approach disruption and disaster situations in cities that would require further investigation to improve existing DSSs. First, in many cases the proposed tools are generalizations of previous domain specific (only one or some systems covered) tools/methodologies. However, these generalizations might not be able to cover all interdependencies and other system-specific knowledge within complex systems of systems. Thus, a comprehensive framework would be preferrable. In fact, this holistic approach would allow tackling risk mitigation from a resilience point of view via coordination and balancing between multiple networked infrastructures. In that line, authors in [Dia+22] also defend that the **decision-makers should be presented with a systematic DSS (both in risk assessment and planning) to help them make decisions about which actions, plans and policies to implement to make resilient cities**. In that regard, to provide a global view of the city encompassing its complexity, resiliency assessment systems are expected to include the evaluation of all city entities (including all sectors: economic, social, physical, and institutional) and to consider

their interconnections.

Authors in [Rez+23] argue that **there is a need to provide spatial representations of data that helps decision makers to better understand situations**. In that sense, GIS techniques can provide spatial context to data, helping decision makers to understand the problem and evaluate alternatives in a more comprehensive way. Alternatively, explainable indicators or advanced dashboarding techniques can also be used to help decision-makers to interpret DSS results.

Reviews also show that **DSS tools are expected to provide predictability via city simulation models that encompass stochastic components**. The models are anticipated to generate qualitative and quantitative metrics to support multicriteria decision making techniques. Metrics can help decision makers to evaluate and compare alternative options for risk reduction and response. In that sense, another existing gap is related to **the need of providing compatible performance and impact metrics among different sectors** (e.g.: electricity grid and healthcare system). Even though the use of aggregated economic or social impact figures and the use of scoring systems has already been explored, additional research along this line would be beneficial.

In order to provide predictability, it is necessary to model the different systems of the city taking into account their complexity and interconnections. **Modeling and simulation of critical infrastructures** has been well studied in literature as discussed in reviews [Ouy14] [ST16] [Wel+22]. Traditionally, several methods have been used, as described in Table 4.1 . Particularly, **agent-based modelling and system-dynamics methods** are well suited to model CIs as complex systems either following a bottom-up approach in which individual entities and interactions are modeled, or following a top-down system view respectively. Agent-based techniques have been already discussed and used in Section 3.2.2 .

City Modelling Approach	Description	Advantages/Disadvantages
Empirical techniques	Historical data and expert experience is used to identify and statistically model frequent patterns such as failures, interdependencies	These methods are affected by a reporting bias and cannot provide good estimates for new unknown types of disasters
Agent-based modelling	CIs are considered as complex adaptive systems (CAS) that can be modelled following a bottom-up approach. Here, the behavior of the complex system emerges from the interaction of many individual, autonomous, generally simple agents. Agents can represent individuals, organizations, or even abstract entities, depending on the context of the modeled system.	They can capture all types of interdependencies between systems but are difficult to calibrate and validate.
System dynamics methods	A top-down approach is used to analyze CIs as CAS. Each system is modelled by a set of variables representing their state that evolve over time by flow rates between systems, usually modelled with differential equations. The dependences between systems are modelled by a series of diagrams that govern causal relationships and product flows.	Stablishing the differential equations and diagrams might be challenging, requires expert-knowledge and they cannot analyze component level behavior
Economic theory approaches	They analyze CI dependencies through models of economic relationships analyzing the interchange of economic goods between them	Not all systems and relationships (e.g.: weather) are observable from an economic point of view.
Network-based methods	Relationships are described by networks/graphs where nodes represent CIs and edges model physical and relational connections. Over these networks, the flow of goods and commodities can be simulated. They can also serve as a basis for point of failures analysis.	They might not be suitable for the representation of isolated CI complexity.
Hybrid-based methods	Integrate modeling approaches in order to be able to organize the relationships among systems or entities, while also being able to show their internal dynamics.	They achieve a balance between the representation of micro and macro dynamics, albeit at the expense of elevating the complexity of the design model.

Table 4.1: Comparison of modeling techniques for smart-territories decision support systems.

Each of the traditional approaches captures a different perspective of interdependent CI (e.g.: geographical relationships can be well explained with a network approach but can be difficult to represent with a systems dynamics approach) and require different resources (in terms of data and computational cost). As a result, **integrated or hybrid approaches** (also included in Table 4.1) are also found in literature complementing several techniques to provide a comprehensive view on critical infrastructure operations [Ouy14]. In general, hybrid systems in the literature seem to integrate two modelling approaches. First, network-based modelling is used to organize and model interdependencies between system or entities. Then, a second approach (either, system dynamics, economic-based modelling or ABM) is used to model the operation of each of the nodes within the network. For instance, network-based modelling is used in [Her+17] to organize interdependencies between entities of different systems. Then, a system-dynamic approach is followed to model each entity through a set of state machines with associated differential equations. A gap in this method is that logical and geographical dependencies are not covered. In [ZP11], a generalized framework is described based on the usage of an economic-based approach on top of a multilayer network. Geographical and inter-system dependencies are modelled through the network while the detailed simulation is based on a goods flow model.

Focusing on CI within cities and smart territories, similar techniques are used due to the importance of infrastructure networks within them. An agent-based approach is used in

[LP21] to model electrical vehicles charging within a smart city. The transportation system and the electrical grid are divided into a set of agents: vehicles, charging stations... Then, the inner operation of each agent is individually model along with their interactions. An agent-based method is also explored in [MMR18] [RW25] with the novelty of introducing disruptions (i.e.: hazardous events) and the possibility of intelligent agents implementing corrective measures to adapt to them. Hybrid approaches are also used here, as in [DPS14] where a multi-layered agent-based model is introduced. In this hybrid proposal, agents are grouped in different layers that represent abstract organizations with specific goals. This allows modelling inter-system and intra-system dependencies at different levels. Overall, even if abstract modelling frameworks are proposed, the particularization of such frameworks in the context of a city or broad smart territory is not explored in a comprehensive way.

Taking into account this literature review, the introduction of the hybrid approach for smart territories modeling proposed in the following subsections addresses significant shortcomings in existing methods. One of them is **the absence of a conceptual model** rooted in systems theory that can effectively guide the development of simulation frameworks. Unlike conventional approaches, the **proposed hybrid model makes possible simulation federation** (or co-simulation), a technique often overlooked in the urban simulation field. This method involves creating a comprehensive simulation model by seamlessly integrating various off-the-shelf system-specific simulators. The innovative aspect lies in the reduction of the modeling effort required, a distinct advantage over traditional methods. This streamlining of the modeling process not only enhances efficiency but also ensures a more realistic representation of diverse systems within a smart territory, leveraging the strengths of domain simulation while ensuring versatility to include specificities. Additionally, **the hybrid model is specifically designed to focus on decision-making aggregation through layered indicators**, a critical aspect often neglected in prior models. By emphasizing this dimension, the hybrid approach can result into improvements over existing city modeling methods, showing a more comprehensive and efficient tool for urban simulation.

4.1.2 A conceptual model of smart territories critical infrastructures

Following on the need identified in Section 4.1.1 to have a comprehensive approach to understand and deal with the complexity of cities, a conceptual model is presented in this section to consider smart territories as set of interconnected systems. Moreover, to enable data-driven decision-making in the context of a smart territory (where IoT enables real-time infrastructure instrumentation) measurements at different levels are explored to assess territories' performance. This section lays out the general, theoretical, formal model that will be used in Section 4.1.3 to define a hybrid simulation framework that supports risk-informed decision-making.

Territories are composed of multiple physical elements (e.g.: parks, roads, transport stops...) and infrastructures (e.g.: electricity, water...) that are used daily by citizens. From an abstract point of view, these elements, including the citizens, can be thought as autonomous entities that make up cities. As previously introduced, these entities interact with each other. In fact, the high-level territory behavior and operation emerges from these low-level

interactions. Following on this idea, an **smart territory can be understood as a Complex Adaptive System** (CAS) [Sen17]. Under this paradigm, cities are dynamic systems in which multiple entities relate to each other over time, showing non-linear and adaptative behaviors. Besides, city's entities also exhibit emergent and organized behavior consisting in the formation of larger entities or operations (i.e.: other subsystems) at different scales. These subsystems, in turn, also relate and interact to each other, forming new systems. Overall, and as discussed in [DK09] [LP21], a **smart territory can be seen as a "system of systems"** where the proper functioning of the whole system (e.g.: the city) emerges and depends on the functioning of a set of interrelated systems (i.e.: each of the essential services), each of them composed of a set of autonomous entities.

Although territories' systems are also formed in the domain of human interactions and in the governance bodies of cities, infrastructure systems providing basic services (e.g.: energy and water distribution, waste management, transportation, health systems, digital and communication systems...) are essential in the functioning of modern settlements [DK09]. In general, these infrastructures are known as **Critical Infrastructures (CI)** which, paraphrasing the definition provided in [US 13], are those systems or assets that are so vital to the functioning of a city that their incapacity or destruction would have a debilitating impact on security, economy, health or safety. Building upon the previous "system of systems" idea and focusing on risks, the smart territory can be analyzed as a system of interrelated critical infrastructures. Therefore, to build a conceptual smart territory model under this paradigm, three questions must be discussed:

- Which systems/critical infrastructures make up the smart territory? What entities make up each of the systems? (this will be analyzed in Section 4.1.2.1)
- How are these systems interconnected through a series of dependency relationships? (this will be studied in Section 4.1.2.2)
- How the operation of each system and the overall settlement can be measured? (as discussed in Section 4.1.2.3)

4.1.2.1 Critical infrastructures in smart territories

Regarding the first question, there is not a single definition of which infrastructure systems should be considered critical: each country defines those most relevant to its national security. However, an analysis of CI definitions from different countries (as the one depicted in Table 4.2) can help identify which critical infrastructures are relevant for smart territories.

Considered critical infrastructure sectors in each country						Proposed systems in the city		
	United States	Germany	Spain	United Kingdom	European Union		Aggregated Sectors	Example of entities within sector
	Nuclear reactors, materials and waste	-	Nuclear	Civil nuclear	-		Electricity grid	Power plants, power grid, control elements, consumers, prosumers...
	Energy	Energy	Energy	Energy	Energy		Gas & Oil distribution	Gas and oil pipelines, storage plants, oil distribution trucks, distribution networks, gas stations, consumers...
	Water and wastewater systems	Water	Water	Water	Wastewater		Water management	Dams, water wells, treatment plants, water pipes, valves, consumers...
	Dams				Drinking water			
	-	Municipal waste disposal	-	-	-		Waste management	Waste trucks, recycling plants, dumpsters, landfills...
	Communications	Information Technology and Telecommunication	ICT	Communications	Digital infrastructure		ICT	Fixed or mobile communication networks, data centers...
	Information technology							
	Transportation systems	Transport and traffic	Transportation	Transport	Transport		Transport & mobility	Vehicles, roads, traffic control systems, underground systems...
	Chemical	-	Chemical	Chemical	-		Food & Goods	Agricultural areas, factories, wholesale market, logistic centers, shopping malls, convenience stores, consumers...
	Commercial facilities	-	-	-	-			
	Critical manufacturing	-	-	-	-			
	Food and agriculture	Food	Food	Food	Producing processing and distribution of food			
	Financial services	Finance and insurance	Finance	Finance	Financial market		Banking & Finance	ATMs, bank offices, stock exchanges...
					Infrastructure Banking			
	Healthcare and public health	Health	Health	Health	Health		Healthcare	Hospitals, clinics, mHealth systems...
	Emergency services	-	-	Emergency services	-		Emergency services	Police departments, fire department, public works departments...
	-	Media and culture	-	-	-		Education, media & culture	Schools, high schools, universities, cultural centers...
	-	-	Research	-	-			
	Government facilities	State and administration	Administration	Government	Public administration		Government	City council, lawmakers, regulations...
	Defense industrial base	-	-	Defense	-		Not applicable	
	-	-	Space	Space	Space		Not applicable	
							Social System	Citizens, households...
							Urban Landscape	Facilities, houses, streets...
							Weather	Weather, pollution...

Table 4.2: Declared critical infraestructures by country and proposed city systems.
United States data is from [US 13]. Germany data is from [Fed21a]. Spain data is from [Spa11]. United Kingdom data is from [Nat23] . European Union data is from [Eur22b]

The energy sector appears as a critical infrastructure in all the analyzed countries. Specifically, the **electricity sector** is ubiquitous in modern societies, enabling the operation of all types of processes and systems (e.g., transportation, manufacturing, communications, etc.). Therefore, its interdependencies with the rest of the CIs to which it provides energy must be carefully considered. This sector includes not only the distribution and consumption processes of energy but also its production through different power plants, including nuclear energy. Continuing in the energy sector, **gas and oil distribution** is also critical for transportation or heating. However, the nature of these distribution networks is different from the electrical networks due to the possibility of storage and non-networked distribution. Hence its modeling as an independent critical system.

Another of the fundamental service networks is **water management**, including drinking water purification, distribution, and wastewater treatment. Likewise, the **waste management** system ensures the sanitation of cities through street cleaning and waste removal and processing. All infrastructures require control and communication systems to ensure their stability and real-time operation. Therefore, **information and communication technology (ICT) systems** also play a critical role within cities, going beyond citizen-to-citizen communications. Similarly, the **transportation and mobility system** also have strong dependencies with other systems by enabling the distribution of goods and citizens mobility. This system includes both road traffic in the city and collective transport systems such as railroads or bus networks.

Services provision is a fundamental part of the economy of cities. This importance is reflected in the existence of the critical **food and goods sector**, which groups critical infrastructures in different countries related to the production and distribution of both food and manufactured goods, including commercial facilities, and the provision of final services to citizens. Underlying this, the **banking and finance sector** enables monetary transactions between different citizens or companies and provides financing and securities.

The SaRS-COV2 pandemic has demonstrated the importance of **health infrastructures**, which is recognized in the CI repository of all countries. Included in this system are hospitals, primary care clinics, or mHealth services such as the ones previously described in Section 2.2 . Similarly, **emergency services** such as police, fire departments and essential public workers help to maintain order and safety for people and property. In this way, they ensure a safe ecosystem in which all other infrastructures are deployed. They are also essential for restoring the city's functioning in the event of an emergency or disaster.

Some countries identify the **media and research** as critical sectors. Although, the education, media and culture sector that may seem less relevant in the daily operation of cities, it reports public announcements in emergency events and has long-term impact in the city. In fact, this system is essential to educate future citizens and to build the city of the future through research. Finally, the **government sector** establishes the conditions under which city activities are carried out through legislation and the promotion of projects. Other sectors such as the defense and space sectors are relevant at the country level, but do not have an immediate impact on the operation of cities and are therefore not usually considered.

Although the critical infrastructure systems that make up smart territories have already been identified, it is also necessary to model other aspects of the territory. The demand of each

of the previous systems arises from the daily patterns and social interactions of the citizens of the settlement. Therefore, the proposed conceptual model extends the previously defined systems to also consider a **social system**. This system models the behavior of citizens (i.e.: daily movements and activities), their interactions between them (e.g.: households, social gatherings) and with other systems (e.g.: electricity consumption). Similarly, CI systems are also modified and affected by the geospatial context they are deployed on. This includes the physical form of the city/territory (e.g., its streets, parks) and environmental elements (e.g., weather, pollution). Therefore, to complete the formal model that makes up the territory as a system of systems, the proposal also includes a system to model the **urban landscape** and another to consider **the environment**.

As for **drones**, they are a transversal enabler that may improve the operation of the several of the previously identified CIs. As discussed in Section 1.2 , drones may be used for inspecting power lines as part of the electricity grid. Also, ground surveillance missions can be performed as part of emergency services, as enabled by the architecture proposed in Section 2.1 . The U-Space ecosystem, described in Chapter 3 may also be considered as a vehicle coordination and control system within the Transport and Mobility critical system. Therefore, a specific critical infrastructure system for drones has not been identified.

As a summary, Table 4.2 above depicts the proposed smart territory' systems and a set of entities that are included/participate in each of the systems.

4.1.2.2 Critical infrastructures interdependencies

Additionally, as already indicated, the relationships and interdependencies among critical systems should also be considered in the conceptual smart territory model. Some of these dependencies have already been discussed in the previous description of the identified systems. In a concrete way, the operation of a territory (or of a subset of systems within) can be represented using influence diagrams where each system is denoted as a node, and relationships and dependences among systems are depicted using labelled arrows. The generation of these diagrams should be based on a systematic analysis on which dependencies are needed system by system. Identification of dependencies can be supported on expert knowledge of infrastructure operation or statistical analysis of real data.

As an example of this process, an expert-based analysis has been performed on the Electricity Grid dependencies resulting in Figure 4.2 . Within the influence diagram (which only considers first order dependencies of the electricity grid system), the outgoing arrows from each system indicate that the source system provides a service or supplies a product to the target system as indicated in the arrow tag. For instance, the outgoing arrows labeled as "Electricity Supply" indicate that most systems rely on the electricity grid to consume electricity. Conversely, the incoming arrows to the Electricity Grid system indicate that it relies on the ICT system for control and supervision, on the Gas and Oil distribution system to obtain fossil energy sources for energy production, on the Banking and Finance systems to negotiate and set the energy prices...

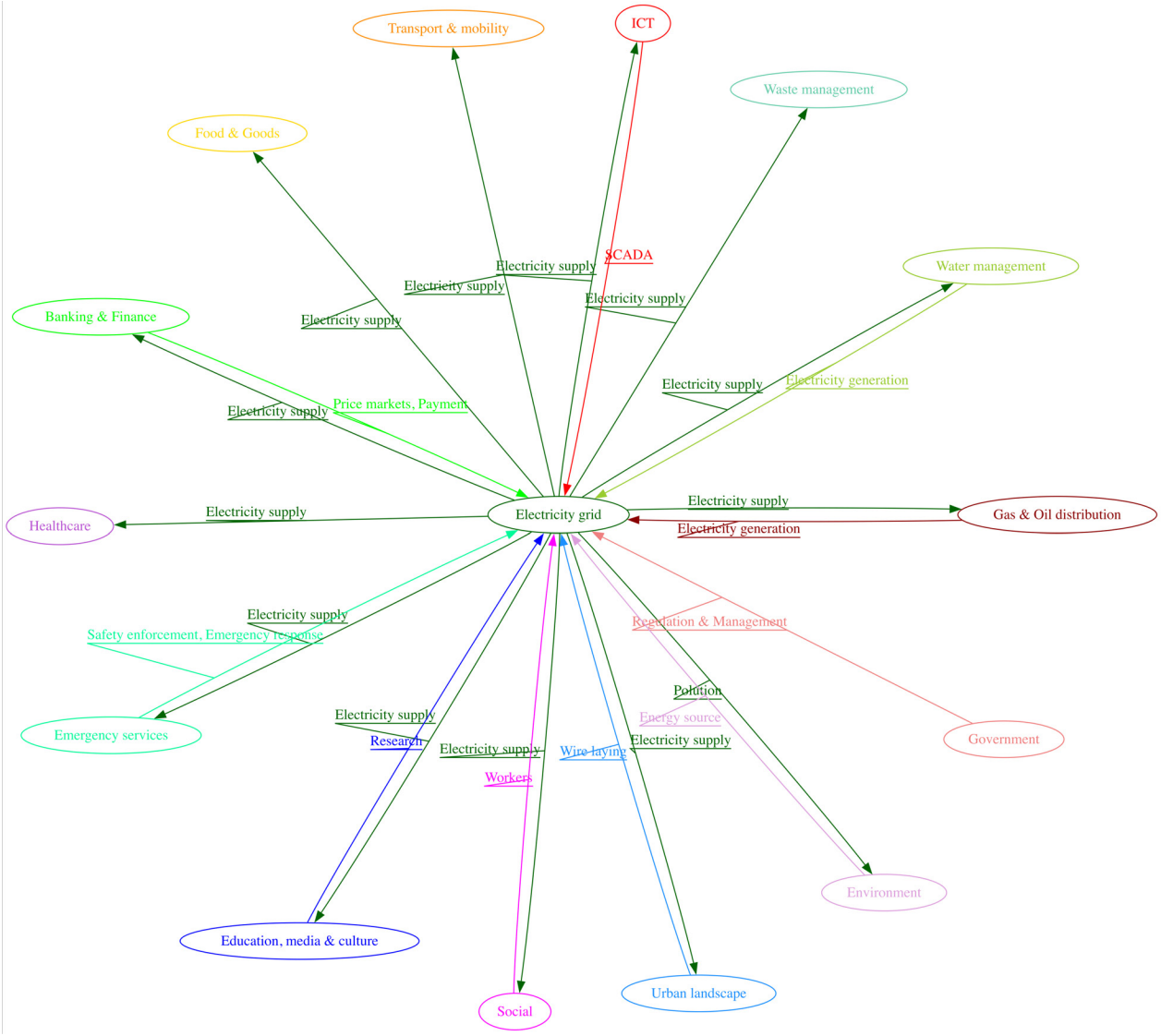


Figure 4.2: Example of influence diagram representing interdependencies of the electricity grid system.

The outgoing arrows from each system indicate that the source system provides a service or supplies a product to the target system as indicated in the arrow tag.

If the previous analysis is repeated for all city systems, it is possible to generate a comprehensive influence diagram by aggregating the different partial diagrams, as the one shown in Figure 4.3 . The purpose of such an influence diagram is twofold. On the one hand, it allows to clearly identify which relationships must be considered when modeling each system. Then, it also allows a decision-maker to analyze between which cascading effects (first-order, second order... interdependences between systems) may occur as a consequence of a risk. However, as a result of this analysis, it is also possible to conclude that, if no additional structure is set up to organize the dependencies homogeneously, modeling them is a challenging task.

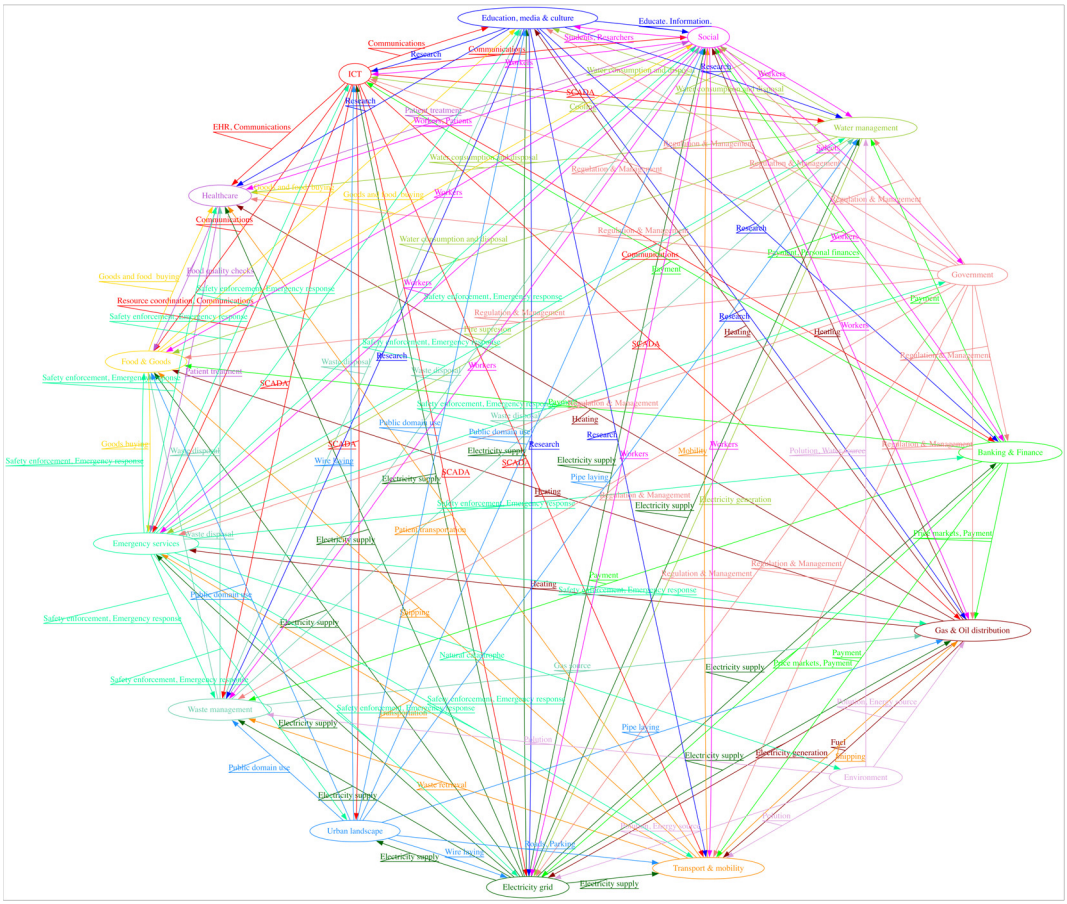


Figure 4.3: Comprehensive view of the interdependencies among all considered systems. The outgoing arrows from each system indicate that the source system provides a service or supplies a product to the target system as indicated in the arrow tag.

4.1.2.3 Measuring smart infrastructures operation

Finally, to understand and analyze the behavior of a smart territory, it is necessary to define a set of metrics that describe the level of performance of its different processes and entities. This quantitative operational assessment becomes even more important when, in the context of data-driven resilience enhancement, it is necessary to analyze the impact of risks or to compare mitigation strategies. However, the heterogeneity and complexity of smart territories makes it difficult to define a single set of explainable metrics that address the main drivers of city performance. In fact, and following the previous formal framework, these metrics can be defined at different levels of granularity:

- **Entity level**: very detailed, domain-specific metrics that refer to the attributes of each of the entities of a city. For example, the load level of a tank, the occupancy of a train, the electricity consumption of a house...

- **CI system level**. Ideally, it would be a common, comparable metric that would allow comparing the level of service of each of the CI systems. However, this is usually not possible, and domain-specific aggregated KPIs (Key Performance Indicator) for each of the CI systems are usually used. Some examples would be demand evolution, demand coverage or response time to incidents, particularized for each system.
- **City/territory level**. In practice, these metrics that provide an overall view of the city usually refer to figures that measure the aggregated economic or social impact of an event or measure. For example, changes in a city's Gross Domestic Product, loss of human lives, variation in the number of citizens... Another possible option is to use scoring system (i.e.: combinations with different weights) to group heterogeneous metrics obtaining a single KPI for the whole city that considers the impact of the different CI systems [Ins+12].

In the context of smart-territories, instrumentation of infrastructures is possible through IoT sensors (for instance, through the architectures presented in Chapter 2). Therefore, entity level metrics can be measured in real time and aggregated in other data fusion level to build the remaining metrics. In addition, for each of these levels, other filtering and aggregation levels can be applied concurrently, such as: time interval, spatial distribution, attributes of the entities/systems analyzed that give rise to different classifications.

The preference for one level of granularity or another depends on the intended purpose of the metric and the target audience. For example, if they are addressed to the operator of a specific CI system, the operator will require detailed metrics for each of the entities it is composed of. These metrics can be used to make tactical decisions that improve its operation, or to generate targeted alerts on specific elements of the infrastructure that can prompt quick reactive actions. In contrast, a municipal decision-maker (e.g., a politician) requires a global view of the city's performance with metrics aggregated at the city level or for each of the IC systems. With this holistic view and the comparison of different what-if scenarios; strategic, long-term decisions such as the implementation of preventive measures can be taken. Ideally, **a decision-support system, should be flexible, providing both levels of granularity to suit the needs of the target user**.

4.1.3 DSS based on hybrid simulation

In the previous section (Section 4.1.2), a conceptual model has been proposed by considering a smart territory comprehensively as a set of interconnected CI systems that are monitored and can be evaluated by a set of metrics. That abstract representation of a human settlement is the first step towards an actionable simulation model that can represent various scenarios and is useful in the context of data-driven decision making. In this sense, Section 4.1.3.1 describes a hybrid simulation framework that links ABM and network-based simulation to analyze each CI and their dependencies in the context of the cities' formal model. Then, Section 4.1.3.2 points how the proposed simulation framework enables risk-analysis and data-driven decision and describes some implementation details that allow model reusability and simulator federation.

4.1.3.1 Hybrid simulation model of smart territories critical infraestructures

Of the various approaches described for simulating CI in Section 4.1.1 , a hybrid framework based on agent-based modelling (ABM) [KB12] and network-based modelling is proposed in this section. ABM fits well with the paradigm of smart territories as CAS: each of the agents models one of the city functional entities (e.g.: a citizen, a hospital, a power plant...) and the overall behaviour of the city emerges from its adaptive, autonomous behaviour. In general, ABM agents can be heterogeneous and governed by simple rules that can consider stochastic components. As the city agents interact at different operational levels, different systems are involved (in practice, understood as an aggregation of agents cooperating to fulfil a common goal or provide a common service). Therefore, the smart territory can be modeled as a MAS in which a set of individual autonomous agents form the different systems (also known as organizations in MAS argot) identified in Section 4.1.2.1 :

$$\text{Agents} = \{a_1, a_2, ..., a_i, ..., a_N\} = \text{autonomous agents in the city model} \tag{4.1}$$

$$\text{Systems} = \{Healthcare,\ ElectricalGrid,\ WaterDistribution,\ ICT,\ ...,\ GenericSystem\} = \text{systems in the city model} \tag{4.2}$$

Modelling the interactions and interdependences between systems can be a challenging task, as demonstrated by the influence diagram in Figure 4.3 . In fact, agents might be part of multiple of the previously identified systems, interacting with other agents within those systems, as shown in the lower half of Figure 4.4 . For instance, a hospital (modeled as an agent) is part of the healthcare system (e.g.: exchanging patients with other hospitals) and that main function depends on the adequate interrelationships with other systems, which will be functional parts of the same hospital agent: the electrical grid (e.g.: consuming electricity from a distributor), water distribution networks, ICT network (it exchanges information with other hospitals over its own ICT infrastructure)... Each of the agents within the dependency systems, on its own, also depend on interactions with agents of the same systems and possibly other systems. Thus, **the proposed model must reflect actions that occur at two different levels: 1) between different systems (inter-system dependencies) and 2) actions across a single system (intra-system dependencies)**. Network modelling is here used to organize these relationships in a simple way. **A multiplexed-network approach is used to clearly identify and model each of these dependencies while following an agent-based model**. It consists in imposing a multilayer structure that limits agents' interactions in two different planes: the agent dimension and the system layer dimension. As depicted in the upper part of Figure 4.4 , agents can be considered as being composed of a set of subagents each of them operating in an individual city system while maintaining internal causal and operational relationships among them to model the whole agent behavior. In the proposed implementation, these sub-agents can be standardized to fulfil different roles within a system, allowing to build modular agents by aggregation of standard sub-agents. Individual subagents can be instantiated by providing different parameterization values to each modular agent.

$$\text{agent} \equiv \{subagent_{healthcare}, subagent_{electricalgrid}, subagent_{waternetwork}, subagent_{ICT}, ..., subagent_{genericsystem}\} \tag{4.3}$$

As shown in Figure 4.4 , the inter-system dependencies can be explained by the interactions among sub-agents across the agent dimension. Then, the behavior of a single, complete system (e.g.: the healthcare system) can be explained by the interactions of all the corresponding subagents (e.g.: all the hospitals subagents) across all full territory agents that form part of that system (i.e.: not all agents are required to participate in all systems). Therefore, interactions of sub-agents across the layer dimension explains the intra-system dependencies, each of the layers/networks corresponding of the identified city critical infrastructure systems.

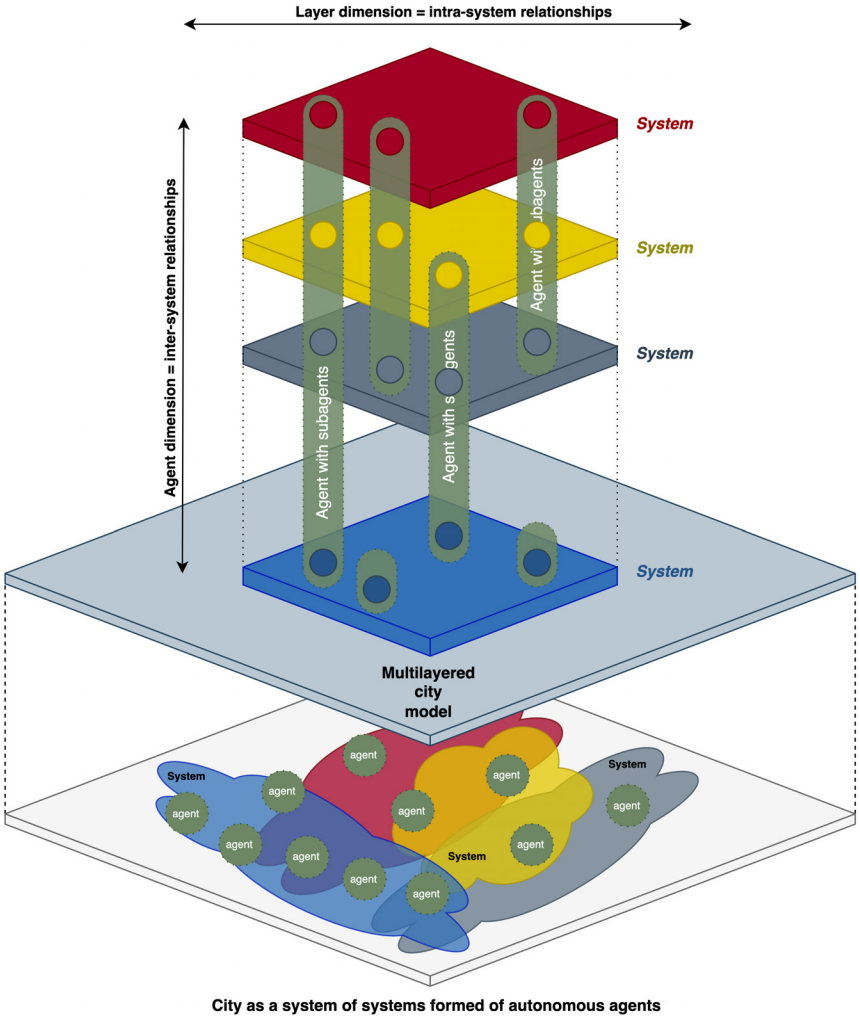


Figure 4.4: Agent and system dimensions within a smart territory
The city as a set of agents organized in systems (down view). Multi-layered abstraction to organized relationships between agents and systems (top view).

These interactions govern the evolution over time of each subagent internal state (Figure 4.5). Multiple tools can be used to define a subagent's state such as finite state machines, status flags or a set of continuous dynamic magnitudes. Interactions can be modelled as well-defined,

explainable rules which might consist of a series of conditions, mathematical expressions, or stochastic processes. Rules are expected to be parameterizable at an individual level so that different agent behaviors can be configured. Interaction rules can be decomposed in three different rulesets. First, subagent **Internal Rules (IR)** govern the internal operational trends of the subagent that can be explained in an isolated way (do not depend on other subagents). Then, **System Interaction Rules (SR)** represent relationships with other subagents at the same system level. Thus, these interactions explain the network effects within a system, considering how that system subagents affect each other (intra-system dependencies). Finally, an agent encapsulates multiple subagents within a single functional entity. At this level, **Agent Interaction Rules (AR)**, explain the relationships between the different subagents of a single agent (intersystem dependencies). That is, they contemplate how other CI systems affect the behavior of the subagent. To ease the understanding and modelling process in, intra-agent dependencies (which will be then made explicit in the AR of each subagent) can be described with an influence diagram mapping dependencies between subagents.

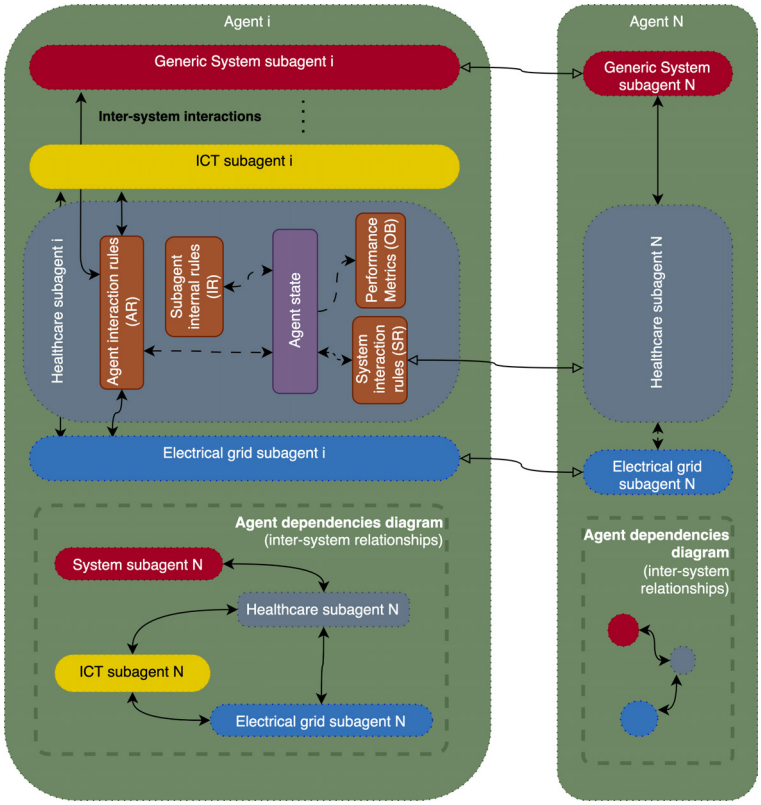


Figure 4.5: Agent as a set of sub-agents modeling inter-system dependencies

Continuing with the hospital example, the associated healthcare subagent state could consist in its bed occupancy, available healthcare providers or medical stockage. Its IR would model processes such as Emergency Room (ER) operation, wards operation, evolution of healthcare quality. Then, additional subagents would be required to model the hospital's water or electricity consumption would. The provision of those services would be contingent on the interactions modelled with the SR across the corresponding systems. These interactions can affect the healthcare service and vice versa, as modelled by AR. For instance, an increase

in patients can result in additional electricity demand; conversely, a blackout can decrease healthcare quality. Overall, the subagent state evolves over time because of the concurrent application of the three set of rules, each considering the state of a given group or interconnected subagents.

$$\begin{aligned}\text{subagent state}|_t = (AR \circ SR \circ IR)(&parameters\ , internal\ state|_{t-1},\\ &state\ of\ subagents\ within system|_{t-1}, state\ of\ subagents\ in\ agent|_{t-1})\end{aligned} \tag{4.4}$$

As already discussed, within the scope of a smart territory, its entities performance evolution can be monitored over time. For each sub-agent a subset of metrics can be extracted from its state. This extraction is modelled by an **observability function (OB)** that allows simulating entity-level metrics for each subagent.

$$subagent\ metrics|_t = OB(subagent\ state|_t) \tag{4.5}$$

Summing up, **a subagent is composed of a set of parameters, its state, the interaction rules, and its observability model**:

$$subagent = (parameterization, state, IR, SR, AR, OB) \tag{4.6}$$

Abstracting to the system level, each system in S will be composed of a set of subagents operating in that system (layer), as shown in Figure 4.6 . Relationships between subagents (intra-system dependencies) can be easily displayed as a network or dependency diagram where the nodes are the system subagents and edges represent functional or causal relationships between them. On its own, each relationship is modelled by the corresponding SR defined at subagent level:

$$Generic\ system \equiv \{a_1^{SYSTEM}, a_2^{SYSTEM}, a_3^{SYSTEM}, ..., a_N^{SYSTEM}\} \tag{4.7}$$

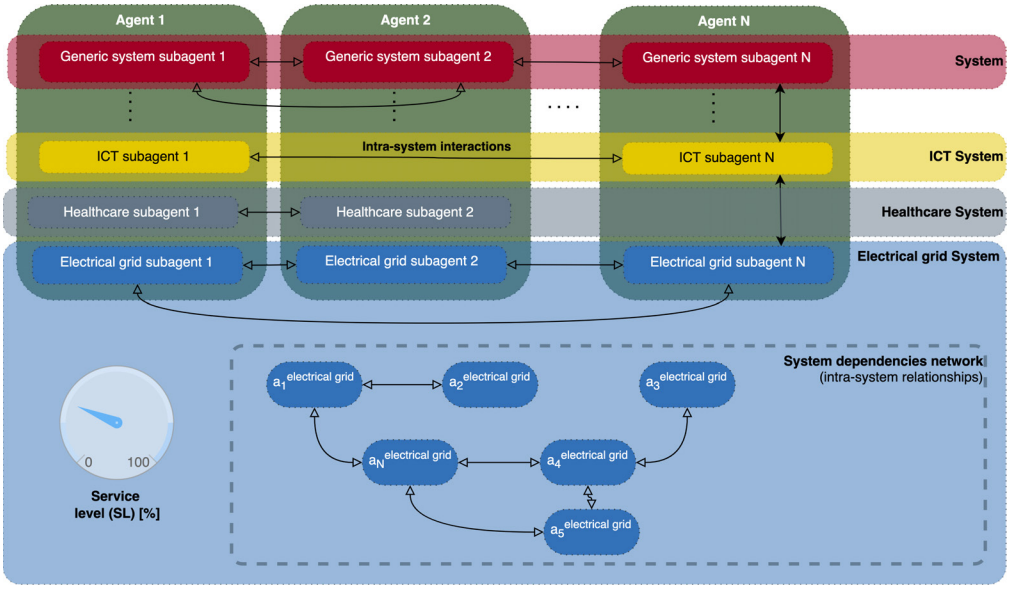


Figure 4.6: Proposed multi-layered agent-based model

Systems performance can also be quantified and monitored within the model. Domain-specific metrics can be obtained by aggregating the metrics of the subagents operating at that level. Specific functions, named **Aggregated Observability (AOB)**, can be used to model the construction of such metrics. Alternatively, a comparable, generic measurement between different systems can also be obtained by quantifying the degree of degradation suffered with respect to the nominal operation (i.e.: correct, without failures) of each system. For this purpose, a metric called **Service Level (SL)** is defined. It can be understood as a quantitative, aggregated, system-level metric that is derived from operational metrics of individual modelled agents. Although operational, domain-specific meaning might be lost, SL degradations can be still tracked back to individual subagents. If required, city-level metrics could be obtained by using a score function over the SL of all city systems. Therefore, for each city system under consideration, the following can be defined:

$$Metrics_{generic\ system}|_t = AOB_{generic\ system}(state\ of\ subagents\ in\ system|_t) \tag{4.8}$$

$$SL_{generic\ system}|_t \in [0,1] = SA_{generic\ system}(state\ of\ subagents\ in\ system|_t) \tag{4.9}$$

Where **SA (System Aggregation)** is a system-dependent, modeler-defined, monotonically increasing function that allows representing the state of an overall infrastructure system from the observable state of each of its subagents. It is expected that the service level will evolve over time with a SL=0 representing a complete degradation of services and SL=1 representing a full level of service. It is the modeller job to design the SA function so that the resulting Service Level accurately represents the operational status of the whole service.

4.1.3.2 Adaptation of the proposed hybrid model for decision-making

The goal of the proposed comprehensive simulation framework is to be able to assess the impact of hazardous events on the operation of city and to compare the effect of different risk-mitigation alternatives following a simulation-driven approach. In this sense, Figure 4.7 envisions a **risk analysis and decision methodology** that integrates the simulation framework within it. First, an initial stage allows defining the scope of the problem to be solved: identifying relevant risks to analyze and which city systems should be considered. Then, in Stage 2, a modeler would be charged of generating an actionable simulation model of the considered systems as an implementation of the comprehensive hybrid simulation framework. This simulation capability enables performing a data-driven risk-analysis in Stage 3 using metrics at different aggregation levels. Afterwards, in Stage 4, the risks are evaluated against acceptable thresholds using the obtained quantitative data. Possible mitigation alternatives are proposed and evaluated with the model to estimate the value of each one in terms of risk reduction. Finally, in Stage 5, the decision maker has a clear picture of the existing risks and a set of assessed alternatives to mitigate them. A comparison between them is possible using the numerical insights provided by the proposed simulation framework to reach a final risk-informed decision to enhance city resiliency.

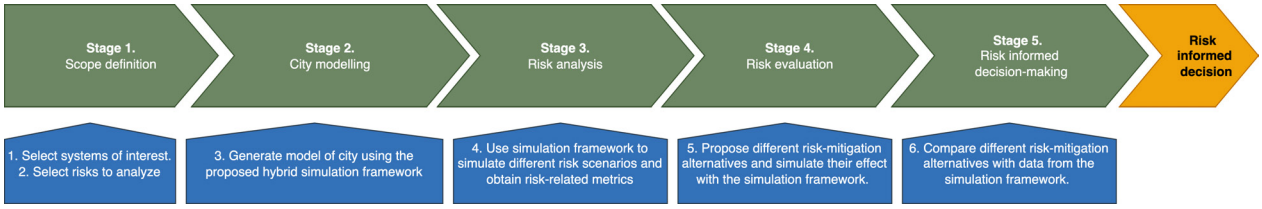


Figure 4.7: Typical steps in a risk-informed decision making process

To facilitate the implementation of this methodology, it is essential to implement the previous simulation framework considering the ease of use from the decision-maker point of view. Therefore, it is desired to reduce the modeling work required to address each decision problem. It is also required to promptly provide results for a broad time horizon so that multiple risk scenarios can be analyzed over the medium to long term. Finally, the simulation results must be easily interpretable by the decision-maker. This subsection discusses how these desiderata can be naturally obtained from the granular nature of Agent-Based Modeling.

Regarding the ease-of-modelling, considering **reusability within the implementation** allows to leverage previous modeling work by partially reusing it for other problems. To implement this reusable approach, a modular design based on multiple inheritance object-oriented programming (OOP) is used, as shown in Figure 4.8 . First, for each city system under analysis, a series of programming classes are defined and implemented. Each class corresponds to a different subagent within the system, individually implementing the behavior and interactions of an entity of the subsystem based on a state and a set of well-defined rules (the previously defined IR and SR rules along with the observability function). These subagents, within each subsystem, exchange events with each other that model the relevant intra-system dependencies. Then, inheritance is used to build new classes that aggregate the behavior of subagents within their source sub-system while adding an inter-system behavior

(AR function). These classes refer to the agents that represent complex entities within the city simultaneously operating in multiple sub-system. The OOP structure clearly encapsulates sub-agents' behaviors in individual classes, enabling to model each sub-agent at different detail levels and easily swapping different sub-agent models. In fact, it is possible to fully simulate each-subsystem independently. Then, complex systems are not modeled from scratch but by aggregation of the pre-existing modular subagents forming agents. This modular programmatic approach intuitively arises from the agent-based analysis and it enables code reusing. Moreover, each class is parameterizable so that differently configured instances of each subagent and agents can be instantiated in different simulation scenarios.

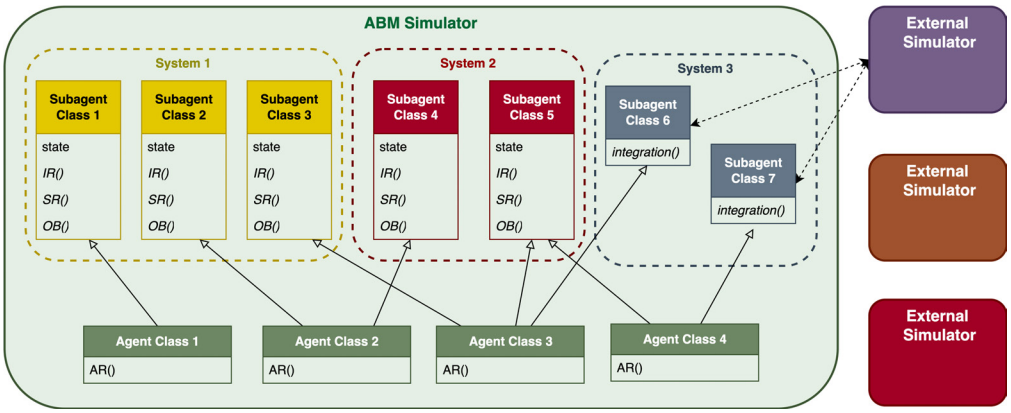


Figure 4.8: Modular class design for agents and sub-agents based on multiple inheritance

Another advantage of the proposed modular OOP approach is that it also allows **substituting the modeling of a subsystem by an external simulator**. The integration with external simulators lessens further the modelling and validation effort as well-tested, domain-specific simulators are broadly available. If the external simulator exposes a programmatic or networked API, which is usually available, the ABM simulator can integrate it. To do so, "meta"-subagents are created as the reflection of an external simulation entity within the ABM simulation. These agents oversee controlling and retrieving data from the external simulation. Therefore, they mirror external entities state so that it can be used within the internal simulation. Moreover, they also allow interactions with internal sub-agents that are abstracted from the integration of the external system. Thus, the proposed simulation framework infrastructure is ready to federate multiple simulators, i.e.: to conduct a distributed and coordinated simulation. Even if usually possible, the extent of the integration is dependent on the capabilities of the external simulator. Overall, this feature enables the integration of advanced, complete, and stable simulation capabilities for different city systems. Furthermore, returning to the idea of simulating entities external to the cities, simulation federation would also allow to simulate in parallel multiple cities interrelated to each other by means of the aforementioned boundary entities/agents.

Summing up, the proposed hybrid (i.e.: ABM-network modelling) framework eases the modelling process, as dependencies among systems are included as reasonably well-known, easily explainable, and implementable relationships between subagents. Those relationships can be explained as simple parameterizable rules and mathematical formulas that can be common and standardized across subagents representing a given function within the system.

Then, each agent can be properly calibrated to instantiate the actor it represents. Moreover, once a high-level modelling framework (i.e.: types of agents, considered relationships...) has been established, each agent (or subagent) within the simulation can be modelled with a different level of detail. In fact, the model can also be easily extended with the inclusion of new agents of the same or different nature or performing simulators federation.

Another of the desired features to enable risk-analysis is the ability to quickly simulate several scenarios with different risks and implementable decision alternatives. Therefore, the simulation framework uses **discrete simulation in fast time** [MJ18]. This type of timing does not introduce dependencies with real time, which allows to quickly generate results for different scenarios using a discretized in simulation time interval. Besides, the framework is configurable in the sense that each agent can be programmatically implemented taking advantage of code reusing between sub-agents or using integration of external simulators. Moreover, each agent depends on a set of parameters that may yield to different simulation scenarios. The **configuration capabilities** range from the agents to be simulated per system or the dependencies between them. Additionally, the simulation of contingencies or threats must be considered in the framework. A hazardous event, HE, can be defined as the occurrence at given time stamps of hazards or threat. Each of them could be parameterized as a variation of the nominal parameters of the affected subagents. Then, because of the described operation, the change of these parameters may have consequences in other subagents due to the interdependencies between them. These contingencies can be manually defined to occur at a certain time step or be simulated stochastically.

$$HE = (hazard_1|_{t=T}, threat_2|_{t=T}, threat_2|_{t=T+2}...) \tag{4.10}$$

Finally, to enable the data-driven approach, the **output of the simulation framework is provided through time series**: the observability function of each subagent and system is used to extract a set of time series with relevant metrics for each agent/system. In this sense, the granularity of the model also allows the generation of metrics that monitor the city for the different levels discussed in the previous section, which helps the decision-maker in understanding the underlying trends. Thus, the framework supports quantitative predictive capabilities to analyse risks. Additionally, it is also possible to compare with the output data different "what-if" alternative scenarios that might also include the existence of non-nominal conditions (hazardous events, contingencies).

These implementation characteristics (i.e.: reusability, simulator federation, fast-time simulation, simulation of hazardous event, granular output timeseries) adapt the simulation framework to its use as a decision support tool. For example, the absence of the simulator federation would make it necessary to model complex systems from scratch, increasing the effort required to implement each decision problem in the simulator. Similarly, the lack of simulation in fast time would make it impossible to analyse scenarios in the long term because the computation times would be unmanageable. Ultimately, as it will be shown in Section 4.1.5 , these features make it easier to carry out simulations that assess the impact of risks and to analyse different mitigation strategies implementing an optimization by simulation approach.

4.1.4 Instantiation of the DSS for healthcare resiliency enhancement

Once the hybrid simulation framework has been defined in Section 4.1.3 , its uselfuness as a DSS within a risk-oriented, simulation-driven decision process must be demonstrated. This subsection is going to **instantiate the model for operational and strategic decision-making in the healthcare domain**. The objective is two-fold:

- By instantiating the abstract simulation model, it is possible to demonstrate how it helps to articulate the interdependences between CIs and implement an actionable simulation model that will serve as part of a DSS.
- As discussed in the thesis's author publication [Car+25a], there is a gap in literature in DSS that cover healthcare CI interdependencies with ICT dependencies considering compounding threats. The instantiation will also cover this gap by considering the interdependencies between these two critical systems within smart territories.

As depicted in Figure 4.9 , the implemented simulator considers three of the smart territories' systems discussed in Section 4.1.2 : **ICT system, healthcare system and social system**. At high level, the implemented model allows to simulate patients flow in a healthcare system together with the corresponding outcomes depending on the attention quality. Within the social system, population groups are considered to model different patient demand sources, but mainly infective disease propagation will be studied. In parallel, a simplified ICT system is simulated together with its interdependencies with the healthcare system. In the coauthored publication [Car+24a], an extension of this model also considering the transportation and mobility system has been proposed, integrating a federated preexisting simulator, but it is not discussed here for brevity reasons.

To consider those systems, the agents described in Table 4.3 are implemented in the simulator. To do so, the OOP methodology described in Section 4.1.3 is used. In it, each agent is derived from a set of individual sub-agents, each participating in a single system. For each subagent, a state and a series of internal and system rules have been defined and implemented, as briefly indicated in Table 4.4 . A detailed explanation of each subagent is provided in the next sections. Then, the already described interdependencies between systems are considered at agent-level, coordinating the behavior of the different sub-agents.

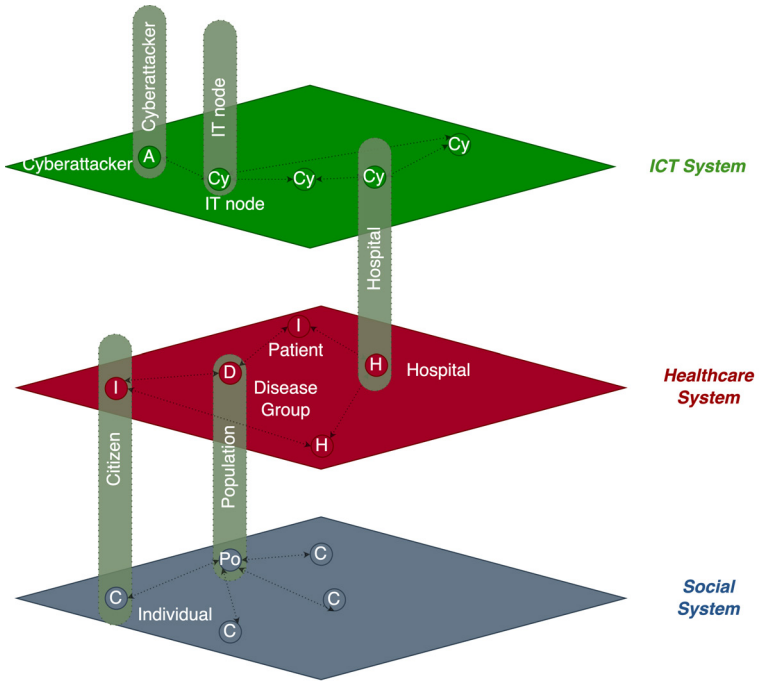


Figure 4.9: Systems, agents, and sub-agents considered in the proof-of-concept healthcare and ICT simulator

Agent	Sub-agents	Description / AR
Cyber-attacker	Cyber-attacker	An entity that can generate botnet, DDOS and ransomware attacks on cyber-infrastructures. Attacks are propagated through a network of cyber-infrastructures respecting its topology. No direct dependencies are considered with other systems.
Cyber infraestructure	IT node	An ICT service provider that does not provide direct service to a hospital. Cyber-infrastructures relate to each other forming a network of dependences. If a given infrastructure becomes unavailable due to an attack, this unavailability propagates to other dependent infrastructures.
Citizen	Individual	A citizen of the city. The citizen can get infected or contract a disease and require medical attention (patient sub agent).
	Patient	
Population	Individuals Group	An entity that groups related citizens (population subagent) that interact over time and in which an infective disease can be transmitted (disease group sub-agent) . It models hospital demand.
	Disease Group	
Hospital	Hospital	A hospital of the city that attends patients. Hospitals form a network (hospital sub-agents) referring patients between them to balance capacity. They depend on the correct operation of ICT infrastructure (cyber-infrastructure sub-agent) that can reduce the hospital capacity and care quality if unavailable.
	IT node	

Table 4.3: Agents considered in the implemented DSS simulation model

For each agent, the corresponding sub-agents are depicted with color coding indicating the system each sub-agent belongs to. The description columns discusses the relationship between the different subagents (AR rules) to implement the overall agent behavior

Sub-Agent	Description	Parameters	State	IR	SR	OB
Cyber-attacker	Representation of any entity carrying out a cyber-attack.	▪ Target ▪ Type of attack (botnet, ransomware, DDOS)	▪ Attack status	▪ Attack generation	▪ Attack transmission	▪ Number of attacked entities
IT node	Representation of an ICT service provider or physical server.	▪ Service Capacity ▪ Vulnerability level ▪ Recovery Speed ▪ Network Topology	▪ Availability status	▪ Nominal operation. ▪ Recovery logic	▪ Dependence on other cyber-infrastructures. ▪ Attack defense and re-transmission logic	▪ Availability status
Disease group	Healthcare demand generation function trough disease transmission modeling.	▪ SIR parameters ▪ Health state transition probabilities	▪ Disease progression	▪ SIR model. ▪ Base medical needs	▪ Patient generation	▪ Disease incidence
Patient	Representation of the health situation of any citizen.		▪ Infective status ▪ Health status	▪ Disease course progress.	▪ Seeking medical attention in hospital	▪ Health status ▪ Waiting times in hospital.
Hospital	Representation of a hospital.	▪ Healthcare capacity ▪ Healthcare quality ▪ Healthcare system topology	▪ Occupancy ▪ Attention quality	▪ Medical care process (recovery process of patients)	▪ Patient reception /discharge ▪ Patient referral to other hospitals	▪ Services occupancy
Individuals group	Group of related citizens	▪ Citizen list			▪ Citizens groups	
Individual	Representation of a citizen of the city.					

Table 4.4: Subagents considered in the implemented DSS simulation model
For each subagent, columns identify the required parameters, its state, the internal and system rules and the observable metrics.

4.1.4.1 Healthcare system model

As depicted in the red slice of Figure 4.9 , the Healthcare System (HS) is modelled as a network of hospitals and populations of potential patients. Three types of subagents are considered within the HS model:

- `Disease Group` subagents aggregate the healthcare needs of different citizens within the same population agent. Therefore, they simulate disease generation and propagation to model patient demand.
- `Patient` subagents model the health impact of each disease at an individual level. As the disease evolves through different stages following a stochastical model, patients may require medical treatment in different healthcare levels/services, whose availability and quality of service influences the patient outcome.
- `Hospital` subagents model healthcare providing facilities that attend incoming patients subject to capacity restrictions.

`Disease Group` subagent

Starting with the description of `Disease Group` agents, they dynamically generate new `Patient` agents following three internal processes (IR, following the methodology on the proposed framework): baseline healthcare demand generation, multiple casualty incidents, and infectious disease propagation. A combination of these three processes can be simultaneously used to generate realistic demand patterns and create `Patient` agents at system level (SR).

Initially, a reference healthcare demand is modeled, to represent the usual healthcare needs of a population (e.g.: chronic diseases, cardiovascular diseases, etc.). Patients are generated over time given a baseline incidence (number of affected patients per 100.000 inhabitants). Also, multiple casualty incidents can also be included as a sudden surge in healthcare demand.

Then, a Susceptible-Infectious-Recovered (SIR) [KMW97] model is used to simulate the spread of contagious diseases within a population and across populations. SIR models simulate infections and recoveries using a set of differential equations parameterized with the transmission rate (number of new infections per infected person and time step) and the recovery rate (number of recoveries per infected person and time step) of the disease. They usually assume a constant population size and that recovered individuals develop lasting immunity. Alternatively, in the coauthored publications [Alb+23b] and [Car+24a], disease propagation has been modeled by simulating the interactions of individual citizens through their daily activities.

`Patient` subagent

Once a citizen contracts a disease, it is proposed to statistically model its evolution at an individual level (`Patient` agent) using a Markov chain with different health states (defined by set H), as is common in literature [Suk+12]. Generated ill patients can traverse through 5 different illness states: very mild symptoms, mild symptoms, moderate symptoms, severe symptoms, and critical symptoms; before recovering or eventually dying from the disease, as shown below:

$$h \in \{healthy,\ very\ mild,\ mild,\ moderate,\ severe,\ critical,\ death\} \equiv H \tag{4.11}$$

From all the possible transitions within a Markov model of 7 states, transitions are restricted (as shown in Figure 4.10 and defined in set T) to those that imply recovering to the healthy state, staying in the same health state, worsening to the next health state or dying:

$$t \in \{stay,\ worsening,\ recovery,\ death\} \equiv T \tag{4.12}$$

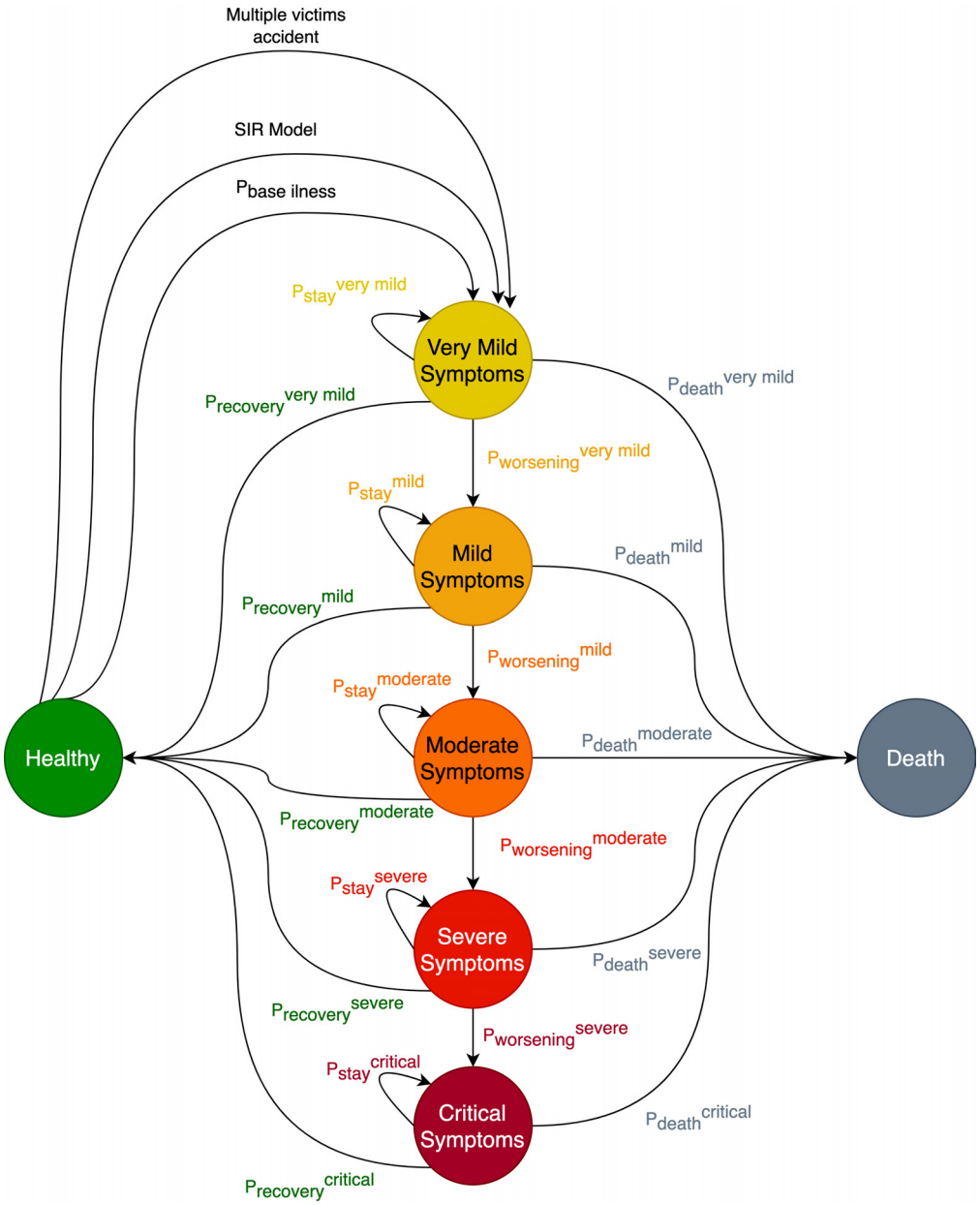


Figure 4.10: Disease evolution model in Patient subagents

In Markov models, transitions between states are governed by stationary probabilities, which

will be denoted $p(t|h)$, meaning the probability of performing transition t from state h. Given a state, the sum of all its transition probabilities must equal 1:

$$P(stay|h) + P(worsening|h) + P(recovery|h) + P(death|h) = 1 \tag{4.13}$$

Then, within the simulation loop, random experiments following the transition probabilities distributions of a given state are used to simulate changes of states.

As the proposed Markov model aims to imitate a disease progression, transition probabilities must be crafted to imitate it in terms of the length of stay at each state, worsening rates or death rates. Literature shows that transition probabilities can be obtained from aggregate administrative data present in Electronic Health Records, which would be readily available for HS decision-makers [Tho+20]. In addition, mHealth systems such as the one proposed in Section 2.2 could be used to gather this data. One of such readily available information is the average length of stay in each of the health states ($h \in H$) that we will denote as follows: T_h, and will be measured in number of simulation steps.

It seems logic to assume that the number of time steps required for a health state change follows a geometric distribution of mean T_h. This means that, as time passes, it is expected that the patient will either recovery or worsen, decreasing the probability of continuing in the same state. Given this assumption and due to the properties of the geometric distribution, the probability of changing of state at any time step is $1/T_h$. Then, the transition probability of staying in the same state can be computed as:

$$p(stay|h) = 1 - \frac{1}{T_h} \tag{4.14}$$

The rest of transition probabilities can be obtained from aggregated statistics describing the overall prognosis of a given disease under study. This means statistics indicating the proportion of patients that recover from a given state or that die from a disease level. Again, it is expected that this information is easily available for decision makers and allows an easy parameterization of the model without resorting to an excessive number of parameters (which is one of the problems of ABM identified in the literature). These statistics are denoted as:

$$Q^h_{recovery}, \quad Q^h_{worsening}, \quad Q^h_{death} \tag{4.15}$$

defining the proportion of patients who eventually recovering/worsening/dying in state h. From this parameters, the remaining transition probabilities can be computed as:

$$P(worsening|h) = \frac{Q^h_{worsening}}{T_h}, \quad P(recovery|h) = \frac{Q^h_{recovery}}{T_h} \quad P(death|h) = \frac{Q^h_{death}}{T_h} \tag{4.16}$$

considering that the overall probability of changing of state is $1/T_h$ and the aggregate proportions must be respected.

Using the previous transition probabilities, it is possible to simulate over time the disease evolution of a `Patient` subagent as part of its IR function. Additionally, diseases with different severities (i.e.: different recovery times, proportion of patients reaching critical stages, different death rates...) can be modelled by tuning the aggregated parameters of different independent Disease (denoted di) instances (which may be simulated concurrently). That is, the required parameters to model a disease are:

$$d_i = (T_h,\ Q^h_{recovery},\ Q^h_{worsening},\ Q^h_{death}) \forall h \in H \tag{4.17}$$

`Hospital` subagent

Another key aspect of the proposed disease evolution model is that health states may be linked with a required level of healthcare service that favors patients' recovery. In this sense, `Hospital` subagents are in charge of providing healthcare services to `Patient` subagents as part of their SR functionality. Four types of healthcare resources are envisioned in this model (as depicted in Figure 4.11 , and denoted by set C): assistance through mobile health services (as the one proposed in Section 2.2 , appropriate for mild patients), in-person consultation with a clinician (optimal for patients with moderate symptoms), general hospitalization (required by patients for severe symptoms), and critical care hospitalization (used for critical patients):

$$c = \{mHealth,\ inPerson,\ generalBed,\ ICU\} \equiv C \tag{4.18}$$

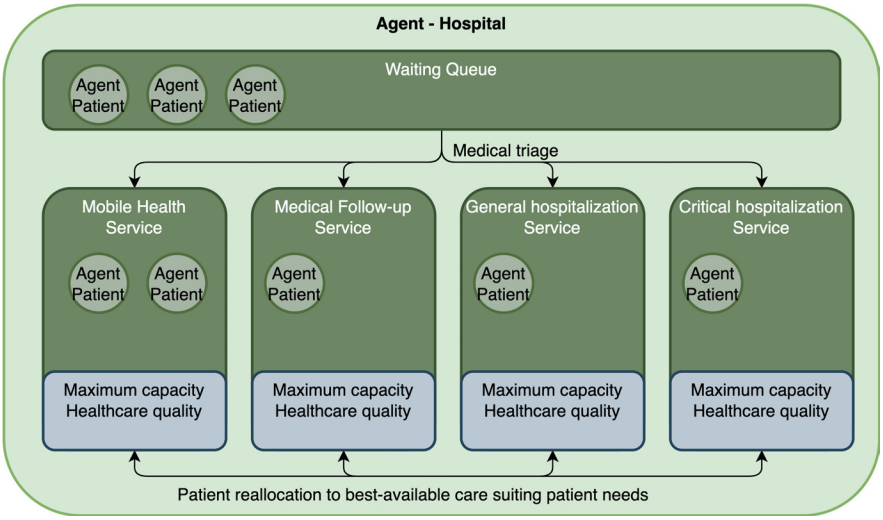


Figure 4.11: Hospital subagent model with different healthcare services

As the health state of a patient deteriorates, it can seek medical attention in a `Hospital` subagent. However, healthcare resources are limited (a maximum capacity is enforced at each care level) which may result in the patient receiving medical care at a lower service than expected or even having to wait to receive any medical care at all. At each simulation step, the allocation of medical resources to patients is reevaluated so that patients may be

promoted to higher level services as resources become available, or as needed due to potential health deterioration. Ultimately, the hospital agent ensures the best possible patient care with constrained resources using a First Come-First Served approach and providing the highest quality healthcare service available and compatible with the current illness state of the Patient.

Received healthcare services affect patients' outcome, particularly when a patient cannot access the required care. Thus, the previous transition probabilities governing disease evolution should be conditional to the received healthcare services. To this regard, the previous disease model must be extended to depend on the type of received medical care:

$$d_i = (T_{h,c},\ Q^{h,c}_{recovery},\ Q^{h,c}_{worsening},\ Q^{h,c}_{death})\ \forall\ h \in H\ ,\ c \in C \tag{4.19}$$

where h represents the health state and c the care level received. To do so, for each parameter, a matrix or table similar to Table 4.5 must be constructed to represent the double dependency (i.e.: current state and medical care). In it, the patient's prognosis will worsen as the care received moves away from the care required. Then, at each simulation step, the appropriate set of parameters is used to compute the transition probabilities, extending the double dependency (i.e.: health state and health level) into the simulation. As in the previous cases, this aggregate information can be obtained from health records.

Health state / Medical care	Very mild symptoms	Mild symptoms	Moderate symptoms	Severe symptoms	Critical symptoms
No follow up	$T_{very\ mild,\ no\ followup}$	$T_{mild,\ no\ followup}$	$T_{moderate,\ no\ followup}$	$T_{severe,\ no\ followup}$	$T_{critical,\ no\ followup}$
Mhealth	-	$T_{mild,\ mhealth}$	$T_{moderate,\ mhealth}$	$T_{severe,\ mhealth}$	$T_{critical,\ mhealth}$
In-person consultation	-	-	$T_{moderate,\ in-person}$	$T_{severe,\ in-person}$	$T_{critical,\ in-person}$
General hospitalization	-	-	-	$T_{severe,\ general}$	$T_{critical,\ general}$
Critical hospitalization	-	-	-	-	$T_{critical,\ icu}$

Table 4.5: Aggregated parameters modeling disease evolution

Parameters are dependent on the health state and medical care received. As an example, the average length state (T) in each state is depicted. Similar tables would be needed for the rest of aggregated parameters (Q)

Additionally, healthcare facilities Quality of Service (QoS) might not be constant. It can be dependent on hospital level of occupancy (i.e.: saturated services tend to perform worse) or other factors such as the dependence with the IC services. To model this, an attention quality level is introduced at each care service ($c \in C$). This parameter can be manually tuned or be dependent on some of the previous factors. As the attention quality declines, the prognosis of patients is expected to worsen. To introduce this dependency, it is necessary to establish a relationship between the attention quality and the transition probabilities that shape the course of the disease. In this sense, let us assume that the probabilities calculated so far correspond to those obtained for a quality of service equal to 1, i.e., the nominal behavior of each service. Similarly, let us assume that as the quality of care degrades, the transition probability is modified up to a limit (i.e.: non-nominal probability in Figure 4.12) that will be matched to the probabilities that would be obtained at the previous level of

care. That is, if a critical hospitalization service has an attention quality of 0, that would be equivalent to receiving the care of a general hospitalization bed. Then, at each simulation step, the transition probability is gradually changed (e.g.: following a quadratic law) between the nominal and non-nominal probabilities as modulated by the attention quality. This relationship is depicted in Figure 4.12 .

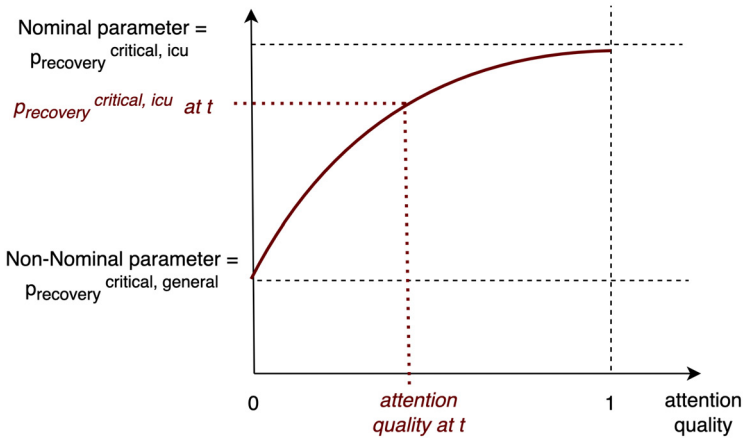


Figure 4.12: Attention quality effect on recovery probability

So far, the model has been detailed with a focus on disease progression. In turn, the relationship between subagents at healthcare level (i.e.: SR functionality) is defined as a probabilistic dependency matrix that governs how `Patient` from each `disease group` seek attention in the set of hospitals. Moreover, `Hospital` behave as a networked healthcare provider, meaning that patients can be transferred between hospitals once their maximum capacity has been reached.

The previous system-level model can be easily parameterized to represent different scenarios with an assorted number of agents, different diseases, or service capacities. Then, as an output, among other, the model provides information both at agent and system level including:

- healthcare service usage (available and occupied capacity per level),
- unattended patients' statistics (waiting times),
- treated patients' statistics (mean treatment time),
- incidence of different diseases or deaths in each population and deaths in each population group.

4.1.4.2 ICT system model

According with the high level description on the model in Figure 4.9 , an ICT system is simulated concurrently with the HS model to represent the information communication and processing capabilities needed for its operation. This system is modeled as a network of two types of autonomous agents (see green slice in the figure): cyber-attackers and IT nodes. This implementation could also take advantage of simulation federation and include an off-the-shelf communications simulator.

The first type of subagent are `cyber-attackers`, that generate attacks (with different parameterizable threat levels) against IT nodes. Different types of attacks can be modelled, i.e.:

- Botnet (it infects other entities to be used to send ransomware or DDOS attacks),
- Ransomware (the infected node becomes unavailable for a given time) and
- DDOS (short-term attack which might yield an infrastructure unavailable depending on its capacity).

Attacks can be targeted to specific IT nodes or be general. Similarly, it is possible to model the difficulty in detecting the cyber-attack and the average response time to recover from the degradation caused by the attack.

The second one are `IT nodes` themselves, that represent functional entities (e.g.: servers, servers or IT service providers...) in a network that may be the subject of the cyberattacks. Each IT node agent can be parameterized with a different service capacity (number of requests that can be handled by unit of time), vulnerability level against attacks (classification of the probability to be affected by an incoming cyberattack), and a different recovery capacity (time required to recover from a suffered cyberattack). This way, it is possible to stochastically model cyberinfrastructure resiliency against an attack. If the attack is finally successful, the IT node might become degraded for some time.

Events (statically and programmatically defined) are used to model functional dependencies between IT nodes and the propagation of the cyber-attack processes (SR functionality). These dependencies are parameterized using a graph that represents the network topology in which the proper functioning of some IT nodes depends on other agents (i.e.: other IT nodes).

To model that the health care system depends on the underlying IT services (`Hospital` agent AR functionality), each hospital subagent is coupled with an IT node subagent operating within the IT system. If that IT subagent becomes unavailable, the interdependency generates a direct effect in the associated healthcare system subagent, which indirectly impacts the HS as a whole. Depending on the type of attack and unavailability time, hospital services might become unavailable (especially those services reliant on the IT system such as the mHealth service) or service quality of service might be degraded (affecting patient outcome and increasing patient mean stay time in each service).

Finally, as an added value output for decision-makers, the simulator allows to analyze the number of received attacks in each IT node and its availability/quality of service over time.

——— * * * ———

This subsection has explored how the critical infrastructure modeling framework proposed in Section 4.1.3 is a valid tool to implement a simulation-based DSS. Particularly, the framework has been used to model both the healthcare system and ICT system together with their interdependencies. The model has been implemented using the Python library Mesa [KMC20], which maps the ABM paradigm into object-oriented programming. A file-based interface has been designed to configure simulation scenarios. It allows defining which agents

to simulate along with their parameters, dependencies, and network topology. Also, threat occurrence can be configured as an event. As an output, a file is provided with the resulting timeseries and Key Performance Indicators (KPIs). This file is easily processed in Python to provide an understandable, graphical representation of the simulation results. The simulator is computationally lightweight and parallelizable. Montecarlo experiments can also be run to provide statistically relevant results.

4.1.5 Validation use-case

To end the section, the usage of the previous HS and ICT simulation model as a DSS is to be exemplified. To this end, the methodology exposed in Section 4.1.3.2 is going to be followed in the subsequent sections through an example use-case.

4.1.5.1 Step 1. Scope definition

The first step in the decision-making methodology was to define the decision scenario identifying the systems of interest and which risks to analyze. In this sense, let us imagine that a regional-level healthcare provider wants to improve the resiliency of its operation against compounding-mixed threats. To do so, the healthcare provider models its infrastructure as a set of autonomous agents following the proposed architecture, as shown in Figure 4.13 . Two towns A and B (of 40,000 and 150,000 inhabitants, respectively) are served by two homonymous hospitals with on-site consultation services, general hospitalization, and critical care units (each with different maximum capacities). In particular, the decision maker wants to evaluate and improve the resilience of Hospital B, with an in-person attention capacity of 1000 people, 278 beds and 30 critical beds respectively. The hospital has no dedicated cyber defense resources, so it is very vulnerable to potential cyber-attacks. Additionally, in case of need, hospital A, which has a larger capacity, can support the needs of hospital B by accepting referred patients.

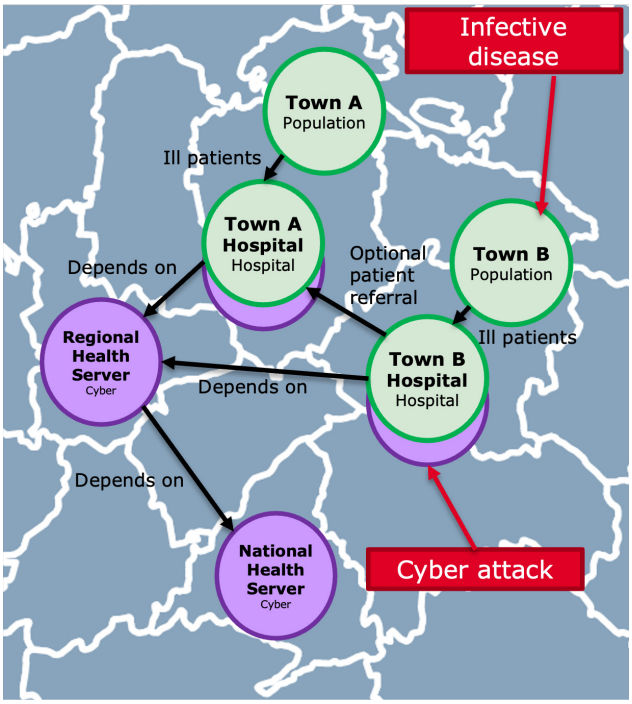


Figure 4.13: Validation scenario
The IT system operates at hierarchically dependent levels: national level, regional level, and hospital level. The health care system is composed of two hospitals in two different populations: Town A and Town B. Town A hospital may support Town B hospital if needed, accepting transferred patients.

The decision-maker wants to consider the coexistence of a cyber-threat against Hospital B Information and Telecommunications system and the spread of an infective disease within Population B. These compounding threats can occur with different intensity levels, with the impact on the infrastructure varying accordingly.

4.1.5.2 Step 2. Systems modeling

The next step would be to model the systems needed for the decision scenario using the proposed hybrid simulation framework and the help of the conceptual model of smart territories. This step has already been performed in Section 4.1.4 . However, an important open point still remains: **the correctness of the model should be evaluated to ensure the validity of its outcomes**. In this sense, validating ABM is a challenging endeavour due to the parameterization dimensionality and usual stochasticity of agent behaviour [FM14]. In fact, there does not exist a broadly accepted method to perform ABM validation. Authors of [HK20] propose a four step approach to validate and test an ABM:

1. **Cross-validation**. If the model has been generated as an extension of a well-known model or if it exits an already validated model for the same problem, the outputs of the ABM can be compared with the ones of the previously simulated model.
2. **Sensitivity analysis**. This step consists in assessing if the output of the model changes as expected to variations of the input parameters.
3. **Comparison to data**. If available, data extracted from the real system is a valuable

tool to validate the model. However, in many processes real data might not be available or be subjected to access restrictions.

4. **Model testing**. As a last step, the usefulness of the proposed model must be assessed, ensuring that it can provide new, valid insights for a decision-maker in a real scenario.

Although all steps should be ideally performed in a new model, authors argue that due to the complexity of ABM and possible limitations on comparable information, it might not be possible to carry out all steps. Regarding ABM validation steps 1 to 3, several technical methods can be used to carry out them. As discussed in [CKL24] and [Klü08], one can resort to **data-driven/statistical methods** to assess the obtained results. A proper statistical analysis of the model might require a significative number of runs covering an extensive combination of input parameters to provide. Alternatively, **face validation or visualization techniques** can be also used to either compare different models' outputs (i.e.: ABM validation step 1), verify that the model responds to changes in the input data (i.e.: ABM validation step 2) or compare the results with real data (i.e.: ABM validation step 3). Face validation consists in a human expert assessing that the simulated system correctly imitates the real system, and its results are coherent. This can be realized using graphical representations of the model output and agent's internal state over time.

Based on the previous methodology, the model validation will be articulated through the decision use case itself, as the present use case is a representative usage scenario of the simulation tool as a DSS. As an output of the use case execution, agents' behaviour over time and aggregate statistics will be obtained. With them, the following ABM model validation steps will be performed using the face validation technique:

- **ABM model validation step 1**. A similar model does not exist in the literature. Thus, it is not possible to directly perform cross-validation. However, the proposed model builds upon the SIR disease model. Thus, the obtained disease propagation curves must be consistent with the typical "bell-like" graph of SIR models.
- **ABM model validation step 2**. The execution of the use case involves running the simulation model with a matrix of different combinations of the configuration parameters (e.g., different threats, different contingency measures). This serves as a sensitivity analysis as it allows to check that varying the input data leads to different results, and that these results are as expected. For example, an increase in the infection rate should lead to a higher number of infections and consequently a higher number of hospitalizations.
- **ABM model validation step 3**. Unfortunately, hospital occupancy, wait times or cyber-attacks information is usually not openly available to the public. Thus, it is not possible to perform a comparison with real data or perform statistical validation.
- **ABM model validation step 4**. The use case will prove that the proposed models can generate useful insights to the decision maker who will be able to compare different contingency responses to enhance resiliency through a contingency matrix.

4.1.5.3 Step 3. Risk analysis

Continuing with the next step, a set of scenarios with different risks (i.e.: cyber attack and infective disease spread) is to be analyzed. Moreover, the variability and unpredictability of those risks should be considered. In this regard, the proposed simulation model facilitates to parameterize different risk scenarios and evaluate the impact on the hospital under analysis. In this way, a baseline scenario is configured consisting of the 8 possible combinations of the following parameters:

- A cyber-attack with high severity (i.e.: high recovery time, denoted `highAttack`) and a cyberattack with low severity (named `lowAttack`).
- An infectious disease with a high contagion rate (i.e.: fast spread over the population, referred as `highContagion`) and a disease with a lower contagion rate (denoted as `lowContagion`).
- The previous infectious disease having high pathogenicity (i.e.: it causes high severe disease with high mortality and requiring advanced healthcare, named `highSeverity`) and another having lower severity (referred as lowSeverity).

These risk scenarios (i.e.: a combination of `highAttack`/`lowAttack`, `highContagion`/ `lowContagion`, `highSeverity`/`lowSeverity`) can be easily parameterized within the proposed hybrid ABM model. The required parameters (already described in Section 4.1.3.2) are aggregated parameters that are usually accessible by decision-makers or healthcare managers. For instance, the pathogenicity level of a disease can be set by tuning the Q^h_{death} parameter at each health state and severity level. Similarly, the disease contagion level can be set tuning the transmission rate of the SIR model.

Then, a Montecarlo simulation with ten realizations for each combination has been performed to obtain valuable metrics to assess the impact of each of the compounding threats combination. Moreover, this parameter variation will serve as the basis for the sensitivity analysis of the model. Each realization of the simulated threat scenarios will simulate the HS and supporting IT system for 350 days.After running the scenarios, visualization techniques can be used to assess the impact on the HS of each risk scenario. Moreover, they will also serve to perform cross-validation and sensitivity analysis using a face validation technique. In this sense, Figure 4.14 , Figure 4.15 and Figure 4.16 show the average results (across the 10 Montecarlo runs) for a subset of the analyzed risk scenarios. For clarity, the figures show only a subset of the risk combinations.

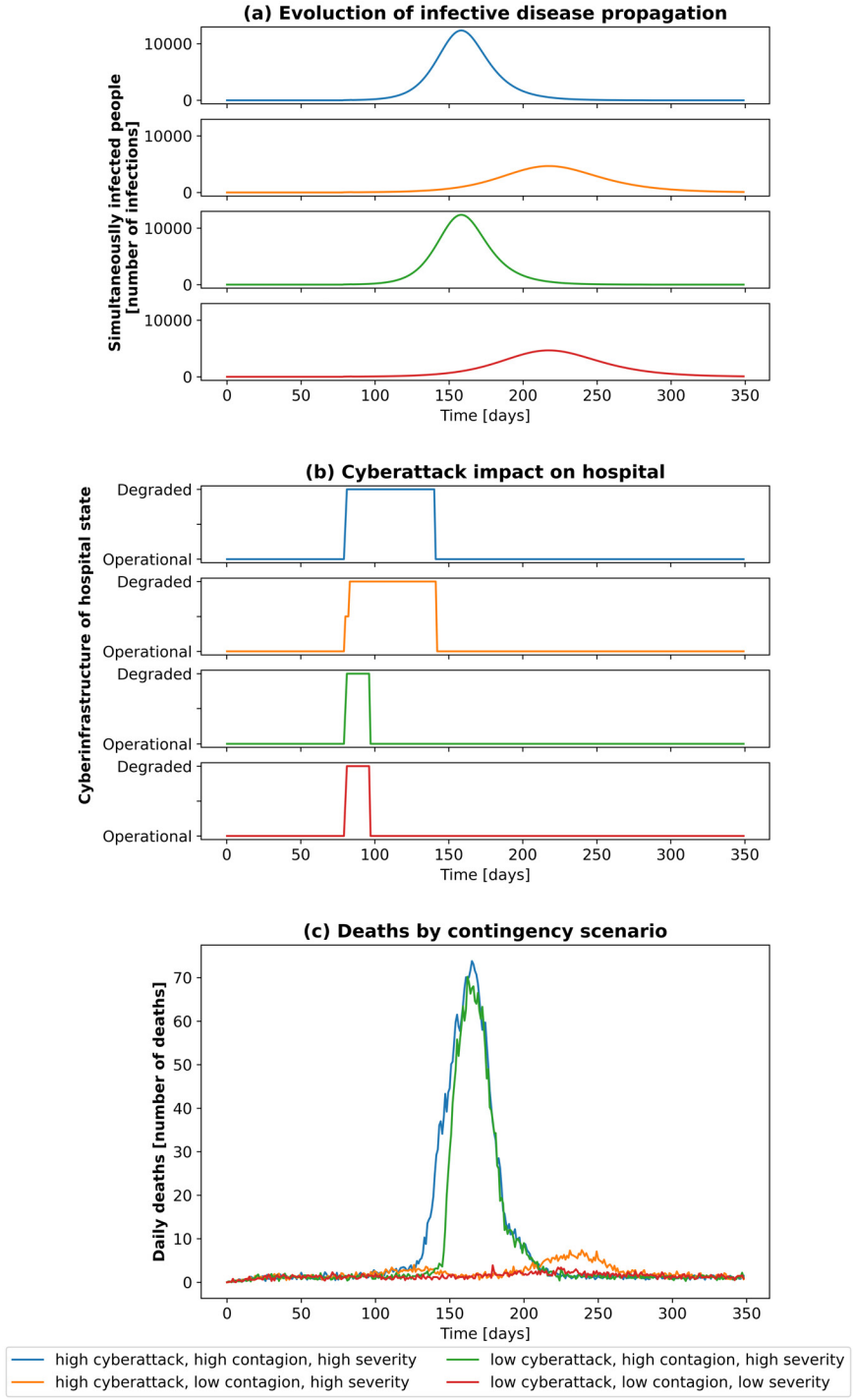


Figure 4.14: Validation scenario results at both agent and aggregate level
(a) Mean evolution of contagious disease, (b) cyberattack impact for different severity levels, and (c) mean impact in terms of daily deaths. In all subfigures a subset of risk scenarios is shown as indicated by the color legend.

Regarding the materialized threats, Figure 4.14 (a) shows the evolution of the number of people infected by the simulated contagious disease. As expected in a transmission model derived from a SIR model, the disease incidence follows a "bell" curve (ABM model validation

step 1). Also, it can be seen how the model correctly responds to changes in the configured parameters. In this sense, high levels of infectivity (`highContagion` scenarios) lead to a high peak of disease incidence (blue and green curves) or, on the contrary (`lowContagion` scenarios), to a flattened curve (orange and red curves). The cyberattack risk is displayed in subfigure b that shows how the occurrence of a cyberattack degrades the hospital's information services. Again, the different configured levels of cyberattack severity result in different service unavailability times. A low severity cyber-attack (`lowAttack` scenarios) can be resolved in a shorter time (green and red curves), whereas a high severity one (`highAttack` scenarios) requires a substantial recovery time (blue and orange curves). Therefore, this proves that the simulation model is adequately sensitive to changes in the input parameters (ABM model validation step 2).

The final impact of these threats on the population at a macro level is shown in Figure 4.14 (c) that depicts the average number of reported deaths. Further supporting the sensitivity analysis, it can be seen how the materialization of both threats can create a surge in deaths that is temporally aligned with the occurrence of those threats. This is clearly visible in the orange curve with a first surge (around t=100) corresponds with the cyberattack effect and a second surge (around t=200) corresponds with the disease infections impact.

At a micro/agent level, the impact of these threats on hospital infrastructure can also be analysed, as shown in Figure 4.15 which depicts the occupancy of the different health services of Hospital B (i.e.: in person consultation, general hospitalization and ICU hospitalization) across time. Again, the impact of the described threats depends on the experienced combination of parameters (risk scenarios, shown as different colour curves). A severe cyber-attack and a rapidly transmitted pathogenic pandemic (i.e.: blue curves in the figure) causes a synergistic negative effect that is observed as a single peak in occupancy across all services. On the other hand, a slower-transmission pandemic (i.e.: orange and red curves in the figure depicts), can decouple the negative effects of the two threats by delaying the contagions peak. In both cases, the cyber-attack causes a first drop in the quality of care, which results in longer care times. This leads to an increase in hospital occupancy at all care levels. On the other hand, the pandemic causes an increase in hospital demand that may end up collapsing the hospital, as shown by the plateaus in occupancy once the maximum capacity has been reached. Different disease parameterization (i.e.: different Q values) may result in different rates of use of different services.

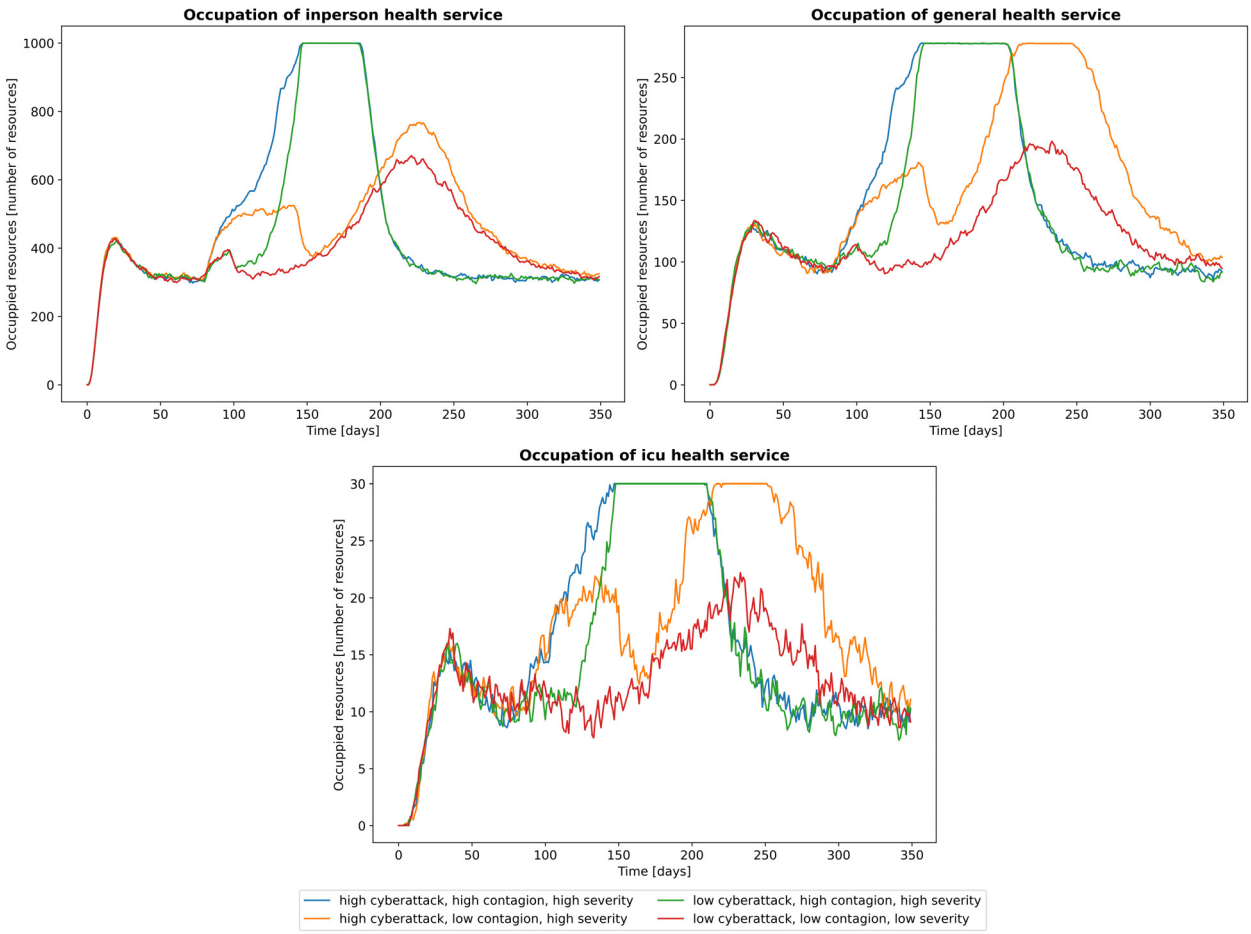


Figure 4.15: Hospital B health services occupancy for different threat levels

Once the collapse of a service has been reached, the collapse severity can be assessed by analysing the utilization rate (i.e.: ratio between the patient arrival rate and the mean occupancy rate) in Figure 4.16 . Utilization rates lower than 1 indicate that the service can meet the care demand. Then, after the 1-threshold is exceeded, the higher the utilization rate, the more degraded a health care service is, resulting in longer waiting times (patients are left unattended) and worse attention quality for attended patients. Degradation in hospital services ultimately impacts on the number of reported deaths (Figure 4.14 (c)) which is clearly superior in the most critical combination of threats (blue curve) versus the least threatening combination (red curve).

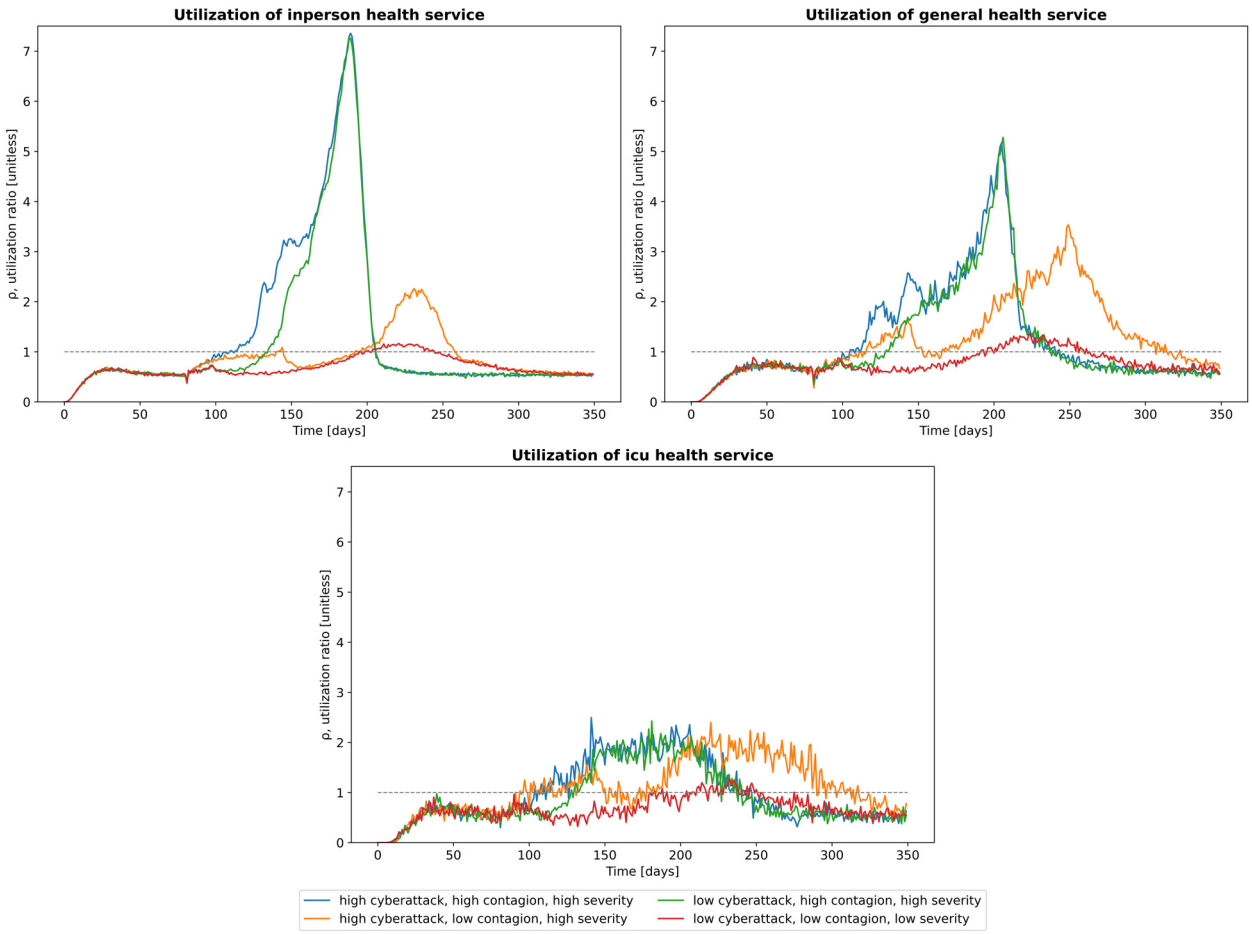


Figure 4.16: Utilization rate of Hospital B services for the different threat levels
Utilization rate metric measures the ration between arrival rate and occupancy rate

In the previous discussion, the model has been validated by cross-validation and sensitivity analysis. Moreover, **the simulated use-case shows how the proposed simulation framework can assess the impact of varied cyber and natural threats in a HS and its associated populations**, providing useful information to decision makers. A correct parameterization of the scenario allows to analyse the different impact of diseases with different transmission and medical care characteristics.

4.1.5.4 Steps 4 and 5. Risk evaluation and informed decision-making

The resulting number of deaths in some of the previous risk scenarios may not be deemed admissible. In that case, the next step of the methodology is to propose different risk-mitigation alternative. The decision maker must enhance the resilience of hospital B to reduce the experienced overload and the cost in terms of deaths and/or money. Consequently, after a detailed analysis of the impact results (with the above metrics or other available metrics such as waiting times per health care service, death rates for each service...), different alternative decisions/countermeasures might be proposed within the limited available resources. Specifically, in this example use case, the following contingency measures are considered:

- An increase of 50% (`lowBeds` alternative) or 100% (`highBeds` alternative) in the number of available resources in some healthcare services.
- A dedicated cyber-defense capacity that increases the hospital's recovery capacity and that it is expected to reduce the recovery time after the cyber-attack to 15 days (`lowSecurity` alternative) or 24 hours (`highSecurity` alternative).
- The inclusion of a remote health monitoring service (`mHealth` alternative) for home monitoring of patients.
- A networked operation of the healthcare system, referring exceeding patients of Hospital B to Hospital A (`referral` alternative).

The optimal decision alternative may not be the same for all scenarios. In this sense, the proposed simulation-based DSS can be used again to evaluate the obtained improvement with each of the potential decision alternatives. Thus, for each of the 8 risk scenarios, a set of contingency scenarios is run combining the threats with each of the proposed contingency measures. To analyse the results in an orderly manner, contingency matrices may be used. These matrices are double-entry tables that show (for each decision alternative and each possible risk scenario) a merit function to assess its suitability.

After running the simulations, Figure 4.17 show a contingency matrix using the number of deaths as figure of merit. Rows correspond to each of the risk scenarios detailed previously, whereas columns correspond to each of the contingency/response options. An additional column named "baseline" corresponds to the reference result when no measures are implemented. Also, a colour code across each row is used to easily compare alternatives. The color code translates alternatives with higher improvements relative to the baseline to green colors, and alternatives that worsen the figure of merit to red colors. Yellow colors indicate that no significant difference is found.

	baseline	highBeds	highSecurity	lowBeds	lowSecurity	mHealth	referral
highAttack_highContagious_highSeverity	3047.0	1730.0	2548.0	2344.0	2536.0	2092.0	1219.0
highAttack_highContagious_lowSeverity	1058.0	620.0	882.0	654.0	920.0	708.0	736.0
highAttack_lowContagious_highSeverity	785.0	607.0	664.0	612.0	702.0	666.0	696.0
highAttack_lowContagious_lowSeverity	552.0	558.0	497.0	550.0	508.0	568.0	541.0
lowAttack_highContagious_highSeverity	2461.0	1452.0	2528.0	1885.0	2505.0	1793.0	1119.0
lowAttack_highContagious_lowSeverity	906.0	566.0	854.0	606.0	893.0	620.0	644.0
lowAttack_lowContagious_highSeverity	691.0	564.0	698.0	582.0	708.0	654.0	621.0
lowAttack_lowContagious_lowSeverity	501.0	490.0	503.0	495.0	499.0	494.0	505.0

Figure 4.17: Contingency table with the number of accumulated deaths as merit function
The baseline column shows the number of deaths in the baseline scenario. In the other columns, a lower value indicates that the alternative succeeds in reducing the impact of the threat.

From the contingency table, a decision maker could conclude that in cases with low severity contingencies (e.g.: low attack severity, low contagion rate and low disease severity) the proposed alternatives hardly reduce the number of reported deaths. Even, results are worse in some cases but this could be due to noise and different realizations of the stochastic process. However, the number of deaths can be significantly reduced in the case of high severity contingencies. Especially, with patient referral alternatives or with a high increase in the number of beds. The mHealth alternative provides intermediate results that may be interesting since the implementation cost of an mHealth service may be lower than that of the other alternatives. On the other hand, cybersecurity strategies only seem relevant in the case

of low virulence, low contagion (i.e.: no compound threat) with a high severity cyberattack.

Overall, this contingency table show how **the simulation tool enables the decision maker to compare decision alternatives considering interdependences between healthcare and IT systems**. Also, these tables could be enhanced by combining several decision alternatives (which can be simulated with the proposed tool) to check for synergic effects. These tables can be understood in two ways. On the one hand, from a strategic point of view, they can allow the decision-maker to select the best investment alternative that globally maximizes the merit function by giving a probability to each of the contingency scenarios. On the other hand, from a tactical point of view, they constitute an action guide for the decision-maker who, in the event of encountering one of the described risk scenarios, can easily select the most appropriate strategy. In both cases, they prove that the simulation framework is a valuable tool for the decision maker.

4.1.6 Conclusion

To guarantee the safety of smart territories inhabitants and the normal operation of the urban space, the concept of resilience and risk management must be introduced into the planning practices. However, due to the complexity of infrastructures, risk analysis and assessment is challenging and often only partially carried out without considering the interdependencies among smart territories domains. Among the section's main contributions, in first place, a **key element is the comprehensive conceptual model of the human settlement**, which aims at overcoming the domain-specific approaches and explores the smart territory as a set of interrelated systems linked by a Social System deployed over the Urban Landscape. Contrary to other theoretical examples in literature, the contribution aims to provide a general view of the city covering all dependencies.

As a possible solution to better manage uncertainty in cities, the section has proposed the use of a comprehensive, hybrid simulation framework as a decision support system to guide risk analysis with quantitative data. In literature, hybrid modelling methods emerge as superior choices for urban city modelling compared to simpler approaches. In this sense, the **proposed practical design of a hybrid simulation framework takes advantage of the benefits of ABM and network-based modelling**. Agent-based modelling (ABM) offers a method for investigating the emergent behaviours within a city, while network-based modelling enables the seamless representation of interdependencies among various entities. As a result, the city is conceived as a collection of independent agents, each comprised of subagents and incorporating stochastic elements. This framework facilitates the creation of functional simulation models that offer insights into city trends. Additionally, as differentiating contribution, it organizes entity dependencies in a manner conducive to easy computer implementation, enables simulator federation and discrete simulation in accelerated time. A key aspect is that the **framework mirrors the smart-territory paradigm by providing multi-level simulated metrics on city operation**, thereby facilitating decision-making-oriented visualization.

To support the applicability of the proposed simulation framework, **it has been applied for a subset of systems (i.e.: ICT system and healthcare system) used to deploy a demonstration scenario**. The implementation exemplifies how the tailored framework

can work as a data-driven DSS tool that enables risk-informed decision making, by assessing different risk scenarios and decision alternatives. Although the final implementation of the DSS and simulation used for validation has been limited to two systems due to the time-consuming nature of the modelling process, the described smart territory concept and simulation framework are general thus they can be easily extended.

Future work could involve incorporating more critical infrastructure systems into the model and improving the model validation stage through real data analysis. Regarding additional systems, the USPASIM platform described in Section 3.2 could be integrated in the proposed framework through simulators federation. This way, it could be possible to simulate drone operations and analyze their impact on other subsystems.

Another interesting development would be the integration of the proposed simulation model within a **digital twin** of healthcare and ICT infrastructures. Such a system would incorporate mHealth applications (as the ones discussed in Section 2.2) and WSNs to monitor in real time the status of patients, hospitals, IT nodes... This information would be analyzed in real time and could be used for rapid decision making (data-based DSS component). In addition, data would also serve as a starting point to parameterize simulation scenarios based on the existing real situation. Then, the simulation models proposed on this section would be used to analyze different what-if scenarios. Such a system would integrate elements within the three technological pillar of smart-territories, as depicted in Figure 4.18 .

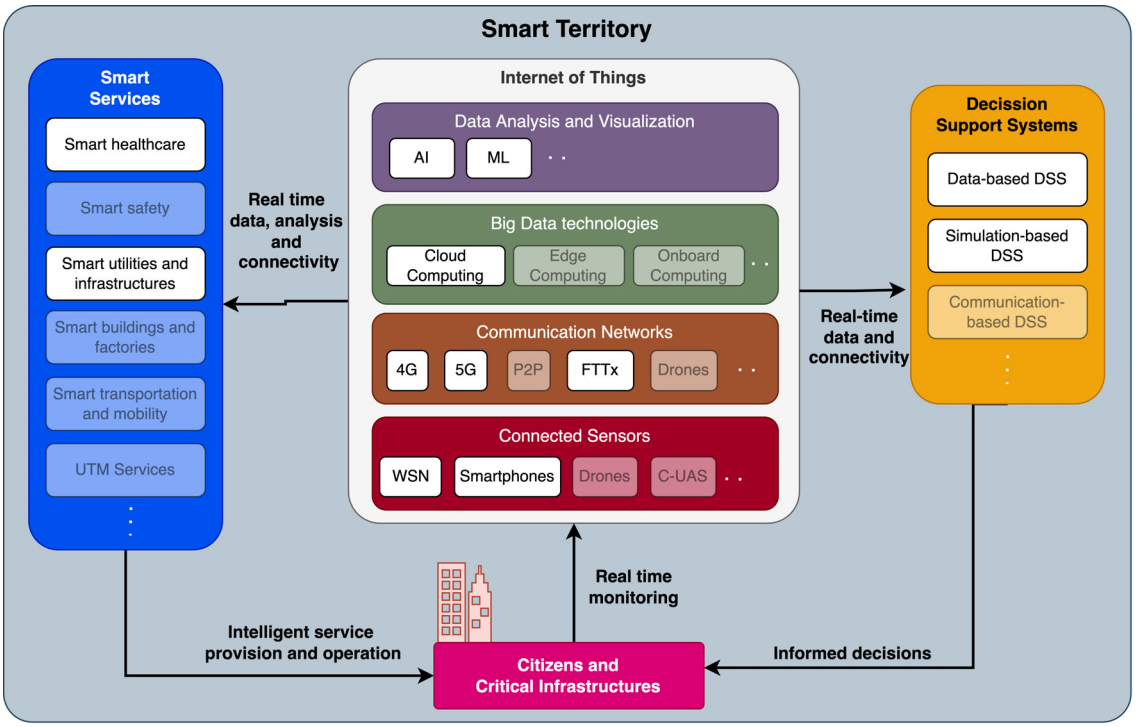


Figure 4.18: Technological elements in a digital-twin for smart healthcare and infrastructures
Relevant technological elements are highlighted over the full architecture

Finally, additional lines of research also include the exploration of artificial intelligence (AI) for outcomes' explainability, enabling a deeper understanding of simulation results for decision

making. How to activate a semi-automated workflow that is triggered by anomaly detection and focuses on providing prescriptive analysis may also be an interesting line of work.

4.2 Decision making at regulatory level: leveraging USPASIM platform for decision making

The U-Space ecosystem, presented in Section 3.1 , is a digital ecosystem that interconnects multiple entities to enable safe and efficient drone operations. As already discussed there, technological advances must come in hand with an enabling regulatory framework. However, U-Space complexity makes it difficult to assess regulation impact on efficiency or safety. To this end, and as depicted in Figure 4.19 , this section aims to showcase how the USPASIM platform (proposed in Section 3.2) can be used as a DSS to inform regulatory decisions at U-Space level.

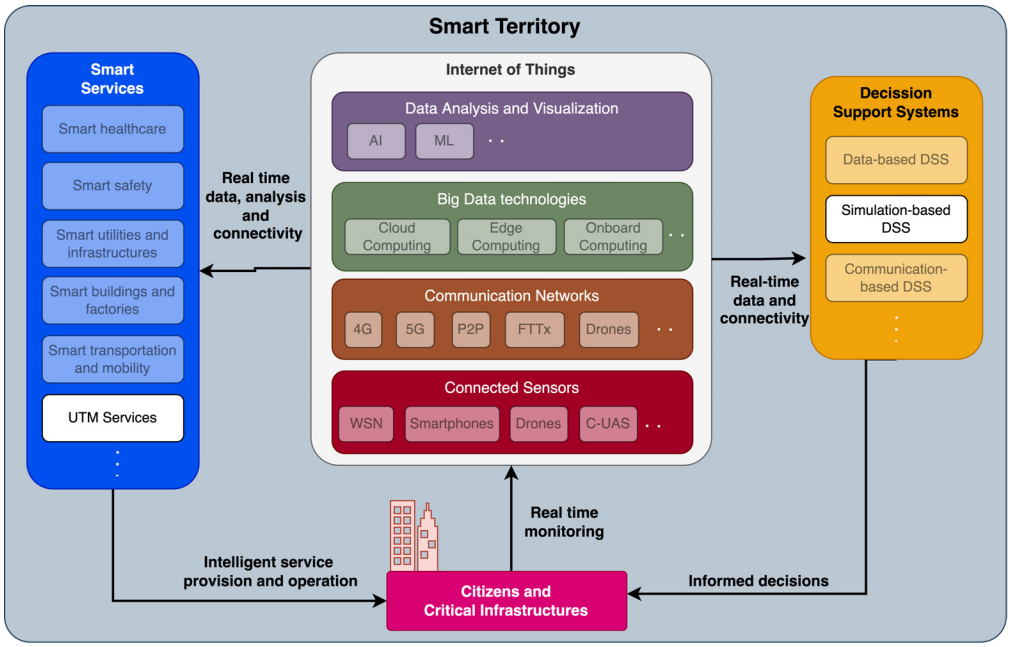


Figure 4.19: Technological elements in the proposed decision-support system
Relevant technological elements are highlighted over the full architecture

In this sense, Section 4.2.1 describes a decision problem faced by regulators: u-plan representation affects airspace usage efficiency, and an inefficient usage might hinder high-density scenarios. To solve this problem, different u-plan representation alternatives have been proposed in literature, for which further insights are needed in order to guide a regulatory decision process. Those decision alternatives (i.e.: existing representation models) are presented in Section 4.2.2 , including: single 4D volume format, 4D volume per segment model, point-based approach and linestring-based u-plan.

To analyze each alternative, the USPASIM strategic phase simulation capabilities can be used to execute comparative experiments, as it will be done in Section 4.2.3 . Then, in Section 4.2.4 , the analysis of those results provides quantitative metrics to compare each alternative in terms of airspace usage efficiency and on the computational requirements for strategic conflict detection algorithms. Therefore, it is demonstrated how the USPASIM framework can inform decision-makers by identifying the implications of each model. In this sense, in Section 4.2.5 , it will be concluded that, in order to enable a prospective high-density

scenario, regulators may need to adopt a new u-plan format or to introduce 4D volume size limitations at the expense of higher computational requirements.

The discussion and results of this section are based on this thesis's author contributions to the following publication:

Supporting publications

- **[CMB25]** - David Carramiñana, José M. Molina, and Juan Besada. "Evaluation of U-Plan Representation Impact on U-Space Capacity and Computational Needs". In: *25th Integrated Communications, Navigation and Surveillance Conference*. Brussels, Belgium, Apr. 2025, where the author has defined the decision-making methodology, reviewed decision alternatives, executed simulation experiments, and analyzed the results.

4.2.1 Description of the regulatory decision problem

As previously discussed, UTM systems, and particularly the U-Space ecosystem, rely on the prior declaration of flight operations (usually known as flight plan or u-plan) so that a priori separation between operations can be enforced. Indeed, Commission Implementing Regulation 2021/664 [Eur21] mandates that drone operators must submit a flight authorization request (i.e.: a u-plan) before each flight. Among other information, the U-plan must define the expected 4D trajectory of the drone (i.e.: 2D route with height and time information). Usually (as detailed in GM1 to article 10(2) [EAS22a]), UAVs trajectories are modelled as a set of 4D volumes, each consisting in a 2D polygon together with vertical limits and a duration.

While there is an intrinsic incentive to limit the size of 4D volumes (the less airspace they use, the less likely they are to conflict and the more likely they are to be authorized) European regulation does not specify the refinement level (i.e.: the maximum volume size) that operators must apply on u-plan definition. Consequently, to perform a linear trajectory between two points at a constant height, an operator may decide to reserve all the airspace around that line for the entire duration of the operation. Although this approach is simple for the operator and facilitates conflict detection, it makes an inefficient use of airspace. Alternatively, an operator may discretize that 4D volume into a set of smaller 4D volumes (both in size and duration) that represent an envelope of the aircraft's position as it moves along its trajectory. In this case, airspace usage is more efficient, but conflict detection complexity is expected to grow quadratically with the number of 4D volumes (as each 4D volume must be check against every other volume).

Therefore, it can be foreseen that, for a same operation, the selection of granularity used for its representation has an impact on airspace usage efficiency and on the computational complexity of the deconfliction process. In this regard, authors of [Muñ+23] and [Gon+25] have already explored the efficiency issue in a limited scenario where drones are guided through corridors. Results show that the authorization rate in a scenario can range from 40% if a single 4D volume is defined for each operation to 75% with multiple 4D volumes. Therefore, depending on the u-plan refinement level, capacity can be greatly affected.

In the future, as the number of drones increases and airspace becomes a contested resource, this inefficiency may artificially generate airspace congestion. As an alternative, existing literature has already proposed other u-plan representation models. Particularly, authors in [Bes+20] describe a line string-based representation which will be further explored in the following section. Similarly, in [JD22], a point-based representation that creates protection bubbles around the aircraft trajectory is explored. In this sense, **quantitative analysis is needed to evaluate the efficiency of each model and to assess whether they have an impact on computational needs**.

4.2.2 Decision alternatives: review on unmanned aerial operation representation models

In order to enable high-density scenarios, different decision alternatives (i.e.: alternative u-plan representation models) will be identified in this subsection. To start, the simplest way to ensure an efficient airspace usage may be to limit the size of declared 4D volumes. However, this may not be possible for some types of operations. Indeed, some operators may need to declare a wide operation area that confines many possible trajectories as the exact trajectory of the drone is not known in advance. This is usually the case of Search and Rescue or emergency operations (as discussed in Section 3.2.3) and can be well represented using a **single 4D volume**. Also, swarm operations (such as the ones enabled by the architecture in Section 2.1) could be modeled as a signe 4D volume, although their integration in the U-Space is still an open topic. This representation approach has been depicted in yellow in Figure 4.20 and is further described in Section 4.2.2.1 .

However, there are a larger set of operations (e.g.: infrastructure inspection flights, precision agriculture, goods delivery, as described in Section 3.2.3) that undergo a meticulous flight planning process. As a result, a set of waypoints is usually defined in advance to describe linear flight patterns or segments to be followed by the aircraft. In this case, if only one operational volume is defined, the available fine description of the operation would be disregarded thus reserving more airspace than necessary. For these cases, other u-plan representations with **multiple 4D volumes** (Section 4.2.2.2 and Section 4.2.2.3) or **based on a linestring model** (Section 4.2.2.4) may be beneficial and will be discussed in this section.

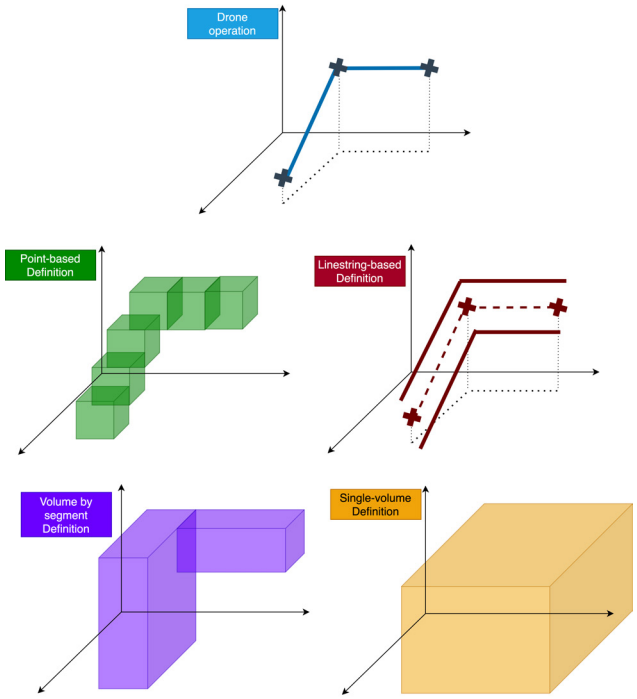


Figure 4.20: Different alternative u-plan representation models studied in this paper. A same drone operation (in blue) is represented using the different formats.

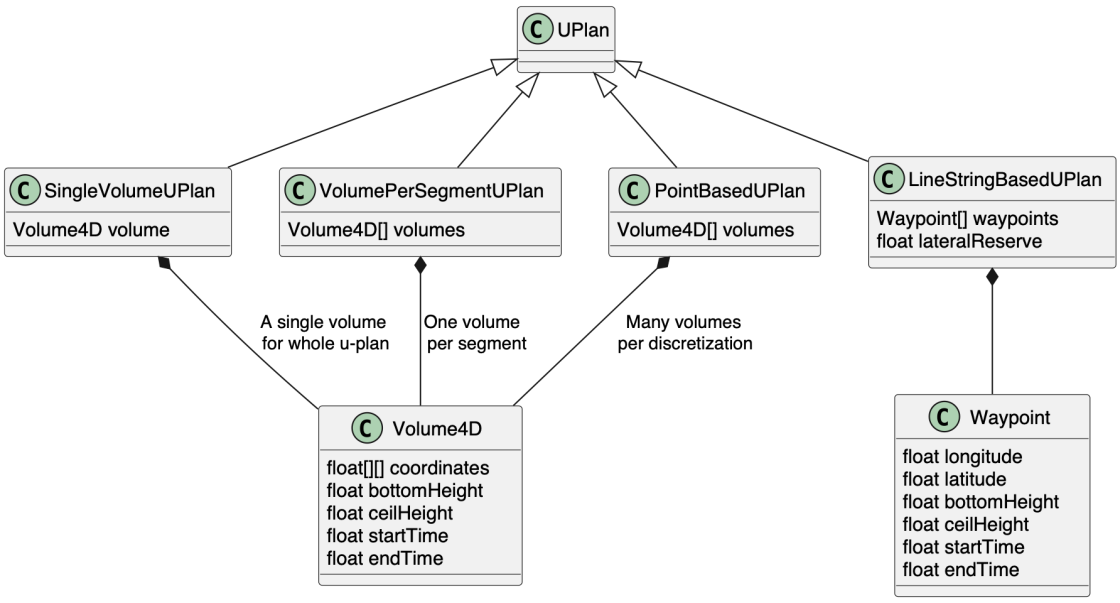


Figure 4.21: UML class diagram indicating the parameters used in each of the U-Plan alternative representations under analysis.

4.2.2.1 Single 4D volume definition

The first type of u-plan representation to be analyzed is the one that consists in declaring a single 4D volume that circumscribes all the utilized airspace. This approach is conformant with current regulation and is also described in [AST22] (named as area-based operational intent). The parameters required to characterize a u-plan following this representation are depicted in Figure 4.21 , and entails the characterization of a single 4D volume including:

- A 2D representation of the reserved area. That is: a polygonal shape on the surface of the earth which can be specified through a list of coordinates indicating the vertices of such shape.
- Height limits. A bottom height and ceil height must be defined to represent the vertical excursion of the flight. These limits are applied uniformly through all the 4D volume.
- Time limits. Finally, the duration of the operation must be considered by defining a start time (from which the allocation is active) and an end time (until which the 4D volume may be occupied).

In general, the polygonal shape can be arbitrary to reflect any drone position throughout the operation while also including tolerances (e.g.: flight technical error, navigation system error, etc.) to confine possible deviations. Both heights and time limits must also consider the whole operation, meaning that they must encompass any possible height excursion and the overall flight duration.

Given two u-plans described under this representation, conflict detection would consist in checking whether there is simultaneously horizontal, vertical and time intersection between both 4D volumes. Horizontal intersection can be checked with polygon intersection algorithms such as the one described in [9], while vertical and time intersection just require interval checking.

4.2.2.2 Volume per segment definition

If a linear operation is to be represented, it is possible to apply the approach studied in [Muñ+23] (named as polyline format), which is also described in [AST22] (named as trajectory-based intent). This representation, which we will name 4D volume per segment, consists in defining a 4D volume for each of the segments within the linear trajectory, where segments mean a linear path that connects two waypoints. Thus, a u-plan would consist in a list of consecutive 4D volumes, as depicted in purple in Figure 4.20 .

In this case, the horizontal representation consists in generating a rectangular buffer around each represented segment (i.e.: line) considering a longitudinal and transversal tolerance. Vertical limits shall go from the altitude of the lowest waypoint to the altitude of the highest one. As for the time limits, the minimum time shall consider when the drone will start flying that segment and the duration of the segment. In this way, it can be observed how the height and time limits no longer refer to the entire flight but only to the segment, so that portions of airspace that are not really utilized can be freed up. Also, it should be noted that in order to ensure flight plan envelope continuity, it is necessary to guarantee that an intersection exists between the 4D volumes of two consecutive segments.

Finally, strategic conflict detection could be performed by iteratively executing the procedure defined in Subsection A for each of the 4D volumes of each considered u-plan. Thus, a slight increase in computational load should be expected.

4.2.2.3 Point-based definition

In [SES21] project, a very fine representation model for linear trajectories is proposed (which will be named point-based definition). This model, which is similarly proposed in [JD22], consists in specifying the aircraft position over time in a large number of points throughout its trajectory. For each of the points, a voxel is defined around it to model the uncertainty about the drone position for the appropriate time margins. In essence, a voxel can be assimilated to a 4D volume in which uncertainty has been incorporated into the horizontal, vertical and time definitions. Thus, this flight plan format also consists of the definition of a set (which will usually be more numerous than in the previous case) of 4D volumes.

For the determination of this u-plan format, a trajectory estimation engine is needed to obtain an almost continuous evolution of the aircraft position over time. Then, this information can be finely sampled/discretized (either in distance or time) so that a 4D volume is defined for each of the sampled points. In this case, an appropriate choice of sampling frequency is of vital importance as it determines the representation refinement level, and therefore the model's efficiency in airspace usage.

This choice is not trivial since, as already introduced, there is a trade-off between the number of volumes and the computational load involved in processing them. This is due to conflict detection assessing intersections (by applying the already described algorithm) for each 4D volume of the flight plan under consideration against each 4D volume of each conflicting u-plan. Therefore, the number of checks have a $O(n^2)$ complexity.

4.2.2.4 Linestring-based definition

Finally, the representation model proposed in [Bes+20] seeks the advantages of a detailed flight definition but limiting the computational load required in the detection of strategic conflicts. In this case, which will be referred as linestring-based definition, authors propose to directly represent the linear trajectories at u-plan level. Thus, the u-plan consists of a series of waypoints to be visited (i.e.: a linestring, as reflected in Figure 4.20), defining for each of them their position, and an altitude and time margin. These margins model flight uncertainties such as different flight speeds (i.e.: minimum and maximum times the waypoint will be visited) or flight errors (i.e.: bottom and ceil heights). As for horizontal excursions, a lateral deviation parameter is used to consider transversal deviations over the expected linear trajectory.

The main advantage of this format is that given the waypoint information it is possible to infer drone cinematic information (i.e.: position, height and timing) at any point along the trajectory assuming a rectilinear motion at constant speed. Indeed, for any point within a given segment (i.e.: pair of waypoints), it is possible to also obtain a position, height and time interval by linearly interpolating between the values of the waypoints using distance as an index. Similarly, this process could be carried out considering time as the interpolation

variable.

The previous idea makes the data format very compact while faithfully representing the trajectory followed by the aircraft. However, it requires a different conflict detection algorithm that involves checking for conflicts between all pairs of flight plan segments. Given a pair of segments, the algorithm:

1. It checks for horizontal intersection by building a rectangular buffer polygon around each linear segment with the lateral reserve parameter. Then, intersection between both buffers is checked. If no intersection is detected, the following stage is not performed.
2. It projects the horizontal conflicting area into each segment under analysis so that a possibly conflicting subsegment is identified in each u-plan. Height and time intersection is to be checked between those subsegments only. To do so:
3. It computes the heights margins at the subsegments under consideration by performing a linear interpolation from the waypoints' height values.
4. It computes the time margins at the subsegments under consideration by performing a linear interpolation from the waypoints' time values.
5. It compares heights and time margins of both subsegments to check for intersection. If both are found, a conflict is present.

Overall, this algorithm is more complex than its volume-based counterpart. However, it is not expected that it will represent a significant increase on the computational load, as the number of entities to be checked will be lower than, for instance, in the point-based representation.

4.2.3 Simulation-based decision-making support: USPASIM setup

In the previous section, four different alternative u-plan representation models have been identified. In order to take an informed regulatory decision, it is needed to evaluate their impact on airspace occupation efficiency and computational cost. This will be measured through the following metrics:

- **Authorization rate** (i.e.: number of authorized flights over number of submitted flights). This metric can be used as a subrogate of airspace usage efficiency.
- **Conflict detection time per checked u-plan** that measures the time computational cost of each representation alternative. Normalization with the number of flights tested allows to standardize the metric regardless of the scenario size.

To quantify these metrics, the U-Space simulator described in Section 3.2.2 can be used as part of a model-based DSS. To do so, the USPASIM framework has been adapted to work with each of the flight plan models. Also, the reference implementation of U-Space services (mainly, the Strategic Conflict Detection service) included in the simulator have been adapted to perform conflict detection for both 4D volumes and line strings.

Then, each metric has been measured for each decision alternative on a common scenario. The comparison scenario consists of a set of 6400 random linear trajectories that are generated to emulate free-flight drone operations. This way, the scenario features a wide variety of

operations both in and out of strategic conflict. Traffic generation has been performed within an `Operator` agent using the linear traffic pattern (as described in Section 3.2.3). This way, it is possible to generate flights limited in a rectangular geographical extent (in this case of 1200 km^2). Each random flight (only multirrotors are considered) consists of several waypoints for a total flight length between 5 km and 10 km. Different altitudes are also considered with heights changing at each waypoint between 0 and 120m AGL. As for time distribution, flights take-off time has been set uniformly throughout a 24 h period. In this sense, Figure 4.22 depicts an example of planned flight distribution within the scenario in a given time snapshot.

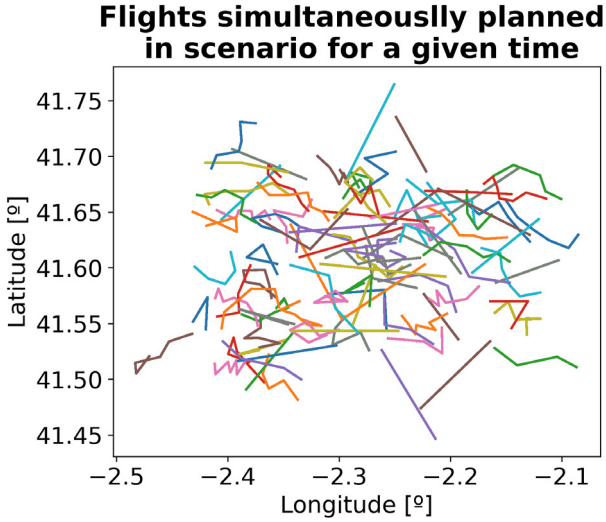


Figure 4.22: Geographical distribution and shapes of flights concurrently flown in the test scenario in a given time instant.

After scenario traffic has been generated, each operation has been represented using the aforementioned u-plan data models (i.e.: single-volume, volume per segment, point-based, linestring-based definitions). To do so, a trajectory prediction model is first used to obtain a detailed definition of the flight. Then, a purpose-built sampling algorithm is able to discretize the trajectory over travelled distance and extract relevant information for each of the u-plans models. Particularly, for the point-based model, three alternative representations are studied with the trajectory being is sampled every: 250 m, 500 m or 1000 m. This will allow to draw conclusions about the relationship between 4D volume size and both efficiency and computational load. Finally, uncertainty margins have been harmonized (± 20 m in height, 20 m in the horizontal plane and ±15% in speed deviation) across all representations.

Once the input information has been generated, the strategic authorization process (as described in Section 3.1.2) is simulated for each representation model. Thus, 6 independent tests are performed (i.e.: u-plan representations and different sampling levels are not mixed across tests). In each test, `Operator` agents gradually submit flights for authorization. At submission, within the reference USSP services, strategic conflicts with previously approved operations are checked by looking for 4D intersections with the previously described algorithms. Consequently, non-conflicting u-plans are marked as authorized. Operation replanning to

avoid conflicts is not considered for simplicity.

The fast time simulation capability allows to run different realizations of the validation scenario to obtain statistically representative results (Montecarlo simulation). Therefore, the comparison scenario has been repeated 10 times (i.e.: runs) with 10 different sets of random flights. As an output, it is possible to provide an estimate of the authorization rate and conflict detection time metrics to guide the decision process. The results will be discussed in the next section.

4.2.4 Comparative results on airspace occupation and computational efficiency to guide the decision process

4.2.4.1 Qualitative results analysis

The experimental setup allows to graphically compare how much airspace is reserved for each of the proposed u-plan models. The objective is to demonstrate if different representations yield to different airspace allocations in order to inform the regulatory decision process. In this sense, Figure 4.23 to Figure 4.25 depict a selected operation as an example on how it has been converted to the different representation formats. The intended operation trajectory is depicted in each figure in maroon color (as stars for the 2D view and as dashed line in the height and time plots).

Figure 4.23 shows the horizontal representation of each of the alternative u-plan formats as constructed by the purpose-built transformation algorithm. It can be seen how the algorithm is capable of building 4D volumes around the linear segments and how the number of volumes differs on the type of representation. Also, the overall horizontal dimensions are aligned across representations and circumscribe the intended operation with additional safety margins.

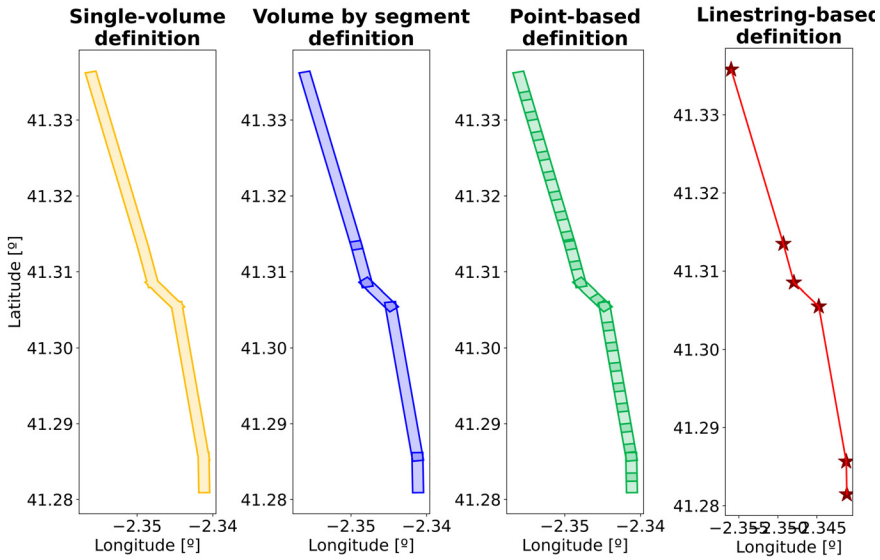


Figure 4.23: Horizontal (i.e.: 2D view) of a same operation represented with the four considered u-plan formats.

Horizontal safety margins have not been represented in the linestring-based representation to avoid confusion with the single-volume definition. Nevertheless, they have been considered in the simulation.

Different airspace bookings depending on the type of representation is best observed in Figure 4.24 and Figure 4.25 which show respectively the height and time limits allocated for the operation as a function of the distance to take-off point. Indeed, in the case of a single 4D volume (in yellow), a range of heights from 0 to 40 m is continuously occupied, whereas in the finer formats, heights that are actually not part of the operation can be freed up. Similarly, in the time case, the 4D volume is reserved for the whole duration of the operation which is not needed to safely perform it. Alternatively, the representations that consider the linear nature of operations allow to only reserve airspace portions as they will be occupied by the drone, releasing them after the drone has passed through them. Also, as we move to finer representations with lower discretization levels, the reserved airspace matches the actual trajectory more closely. For instance, the first segment (between 0 a 0.4 relative distances) representation in the volume-per-segment u-plan (in blue) fully occupies the 0-50 range. In turn, heights close to ground are gradually freed in the point-based (in green) and linestring-based definitions (in red). The same effect on airspace usage efficiency can be seen in the time plot. In this case, the reserved airspace increases over time due to the accumulation of uncertainty introduced in the aircraft speed.

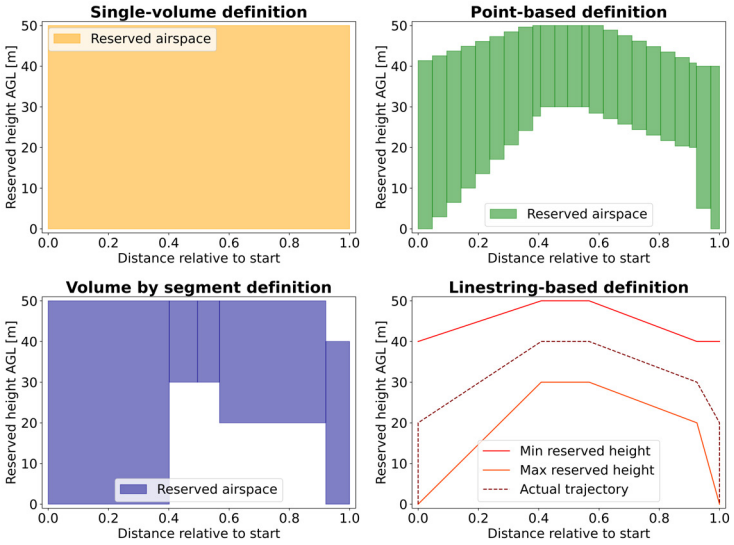


Figure 4.24: Comparison of height reservations depending on u-plan representation
Vertical profile (i.e.: height over travelled distance relative to total length) of a same operation represented with the four considered u-plan formats.

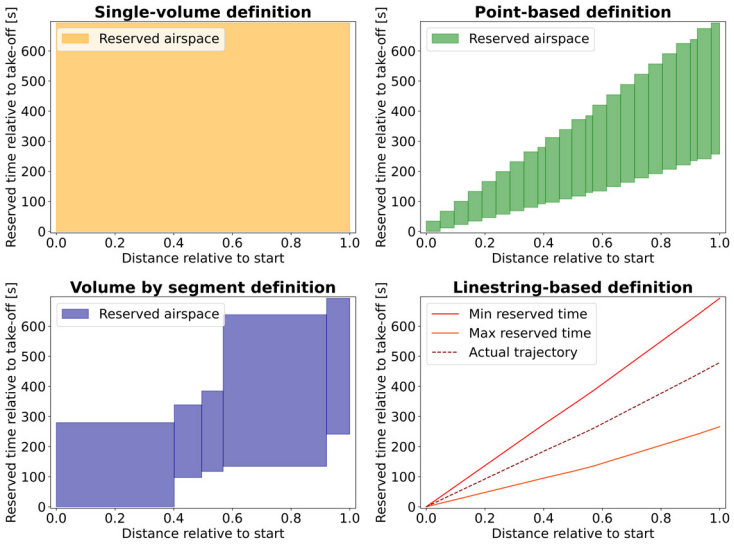


Figure 4.25: Comparison of time reservations depending on u-plan representation
Time profile (i.e.: reserved times relative to take-off time, over travelled distance relative to total length) of a same operation represented with the four considered u-plan formats.

From the point of view of the authorization process, a more efficient representation of the operations should result in a higher authorization rate. When the representations are coarse, "spurious" conflicts (due to excessive airspace allocation) are detected. However, these conflicts do not represent a safety problem as they happen outside of the actual operation envelope. This situation is exemplified in Figure 4.26 that show conflicts detected between a pair of u-plans using two different representations (single 4D volume vs point-based). In the top figure, a coarse representation is used, and a conflict is clearly detected (in red) as both operations reserve the whole low-level airspace. Thus, only one u-plan could be authorized. Instead, with a finer representation (in the bottom), it can be seen how, at that refinement level, both operations cross at different altitudes and no conflict exists. As a result, both operations could be authorized. In reality, both cases could have been authorized since, as shown by the actual drone trajectories depicted as coarse colored lines, in neither case operations collide. While this is a simple example for the sake of clarity, this situation can occur simultaneously in both time and altitude and with several pairs of flight plans.

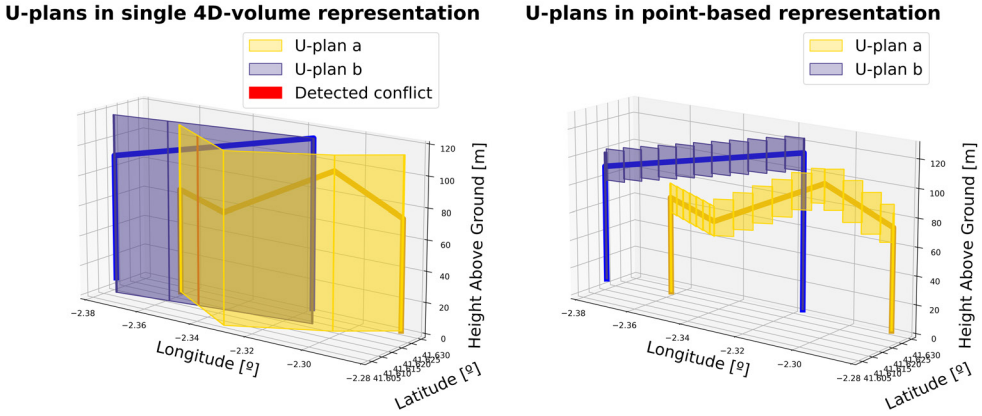


Figure 4.26: Comparison of conflict detection results depending on the type of u-plan representation. Actual trajectories are depicted as coarse lines.

4.2.4.2 Occupation efficiency analysis

The preceding analysis has shown anecdotal evidence of the improvements on airspace occupation efficiency depending on the type of u-plan representation. From the decision-making point of view, quantitative results are needed to inform a decision. In this sense, Figure 4.27 shows a boxplot depicting the mean authorization rate distribution across the 10 different runs for each type of analyzed u-plan format. As expected, those u-plans formats that utilize a coarser representation of an operation (e.g.: single polygon and volume by segment representations) achieve worse authorization rates than those with a finer representation (e.g.: linestring-based and point-based). In addition, the various discretization levels studied for the point-based alternative show that as the number of points increases, the authorization rates improve. As for the comparison between the two most efficient representations, the linestring-based approach performs better although similar results could be expected in with the point-based alternative provided that the discretization frequency is increased even more.

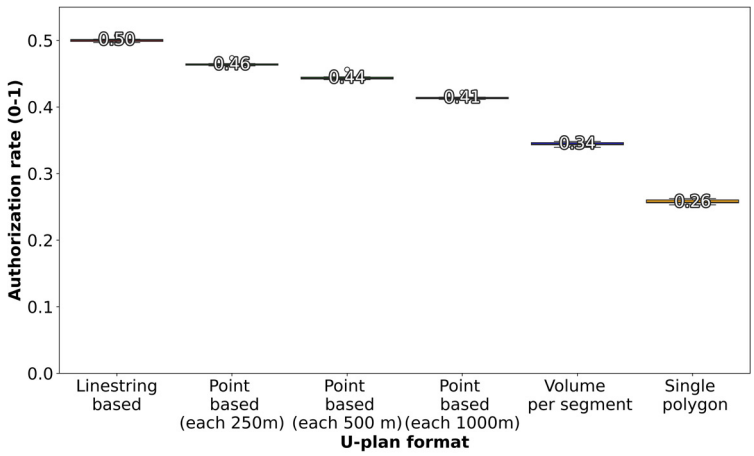


Figure 4.27: Analysis on u-plan occupation efficiency.
Boxplot comparing airspace occupation efficiency through the u-plan authorization rate for each u-plan format. Label depicts median value.

4.2.4.3 Computational efficiency analysis

However, increasing the discretization frequency is expected to have an impact on the computational load that is studied in this subsection. To this end, Figure 4.28 shows a boxplot depicting the statistical distribution of the time needed to check conflicts between a pair of flights. Indeed, results demonstrate how u-plan as the discretization distance decreases required processing times greatly increase. In this sense, the point-based alternative yield processing times that can be an order of magnitude higher than the others due to the larger number of spatio-temporal entities that must be compared. On the contrary, the linestring-based approach achieve processing times that are comparable to those of the single-volume representation. All tests have been run in a i9-13900K CPU with 32GB of RAM.

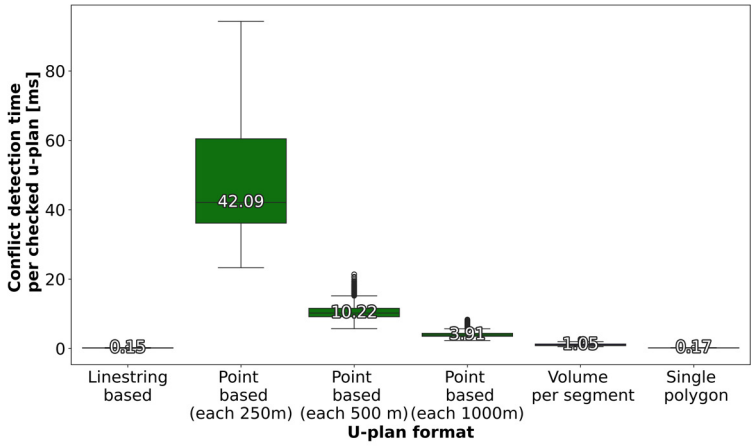


Figure 4.28: Analysis on u-plan computational costs.
Boxplot comparing time computational cost of detecting conflicts in each u-plan format through the conflict detection time per checked u-plan.

4.2.5 Discussion and conclusion: simulation-based u-plan representation insights for decision-makers

Results in previous section prove that that for 4D volume u-plan representations a strong inverse dependency exists between the size of the 4D volume (i.e.: how closely reserved airspace matches the actual operation) and airspace occupation efficiency. Also, it has been quantified how as the number of considered 4D volume increases so does the computational load in the strategic conflict detection process. As an alternative, the linestring-based alternative achieves the highest authorization rates without increasing the computational costs.

Besides the efficiency improvements, trajectory-aware u-plans also result in increased safety since it is possible to improve conformance monitoring. This way, a finer operation definition allows to detect eventual inflight deviations earlier so that the pilot can be promptly alerted and directed to solve them. Tactical conflict detection services can also use this information to avoid false alarms. For instance, a linear extrapolation of the in-flight trajectory of a drone can trigger a collision alert, but with a detailed u-plan it may be possible to infer first that the aircraft is changing its path conforming to the declared operation. Finally, detailed knowledge of the trajectory also allows to improve the techniques for automatic alternative flights generation in case of conflicts (strategic conflict resolution). This way, modifications to the route or heights could then be made to avoid the different flight plans. On the negative side, some operators may be reluctant to share the details of their track record for privacy or competence reasons. However, this point could be mitigated by considering the USSPs as neutral entities that do not share sensitive information with the operators they are providing a service to.

Overall, this section demonstrates how a simulation model such as USPASIM can serve as a key element of a DSS. Indeed, through the U-Space ecosystem simulation model, decision-makers can gather quantitative and qualitative insights on each of their decision alternatives

——— *** ———

Chapter summary

Informed decision-making through Decision-Support Systems (DSS) is one of the features of smart territories. Particularly, model-based DSS provide prediction and optimization capabilities through statistical models and simulation frameworks. Specially, they allow to implement a what-if analysis that quantitatively compares the outcomes of different decision alternatives.

In this sense, Section 4.1 has explored how a simulation model can be used to enhance the resiliency of Critical Infrastructures (CI) in smart-territories. The section addresses two literature gaps: a lack of a simulation framework that provides an comprehensive view on CIs interdependencies, and a lack of tools to jointly analyze the infrastructures of the healthcare and ICT domains. To this end, the section has proposed a hybrid simulation framework that uses agent-based simulation to consider intersystem dependencies and a network-based approach to model intra-system dependencies. This framework has been built on top of a novel conceptual model of smart territories' critical infrastructures as a system of interconnected and measurable systems. Finally, the model is instantiated in a healthcare and ICT decision use-case to demonstrate that it can provide multi-level quantitative metrics to guide both operational and strategic decisions.

Finally, Section 4.2 has discussed how regulatory-level decisions can also be informed by simulation tools. Particularly, the USPASIM simulation framework has been used to compare alternative u-plan representation models in terms of airspace usage efficiency and computational costs. Montecarlo simulations allow to conclude that linestring-based u-plan alternatives achieve superior authorization rates without increasing conflict detection costs.

Chapter 5

Conclusion

In the last decades, technological advances have gradually integrated in the operation of modern human settlements until becoming one of the main driving forces of economic, societal and environmental prosperity. This trend is reflected in the concept of smart territories, which are those that use technological solutions to improve their infrastructures and services efficiency and resilience, and the quality of life of citizens. In this sense, the main technological pillars of smart territories are: monitoring through instrumentation and intelligence using architectures that leverage the IoT, provision of advanced smart services to citizens, and informed decision-making based on data. In this context, and as depicted in Figure 5.1 , this thesis has proposed novel systems, applications and models across the whole smart territories technological stack. Moreover, it has done so while considering drones as an enabler of both IoT systems and smart services provision.

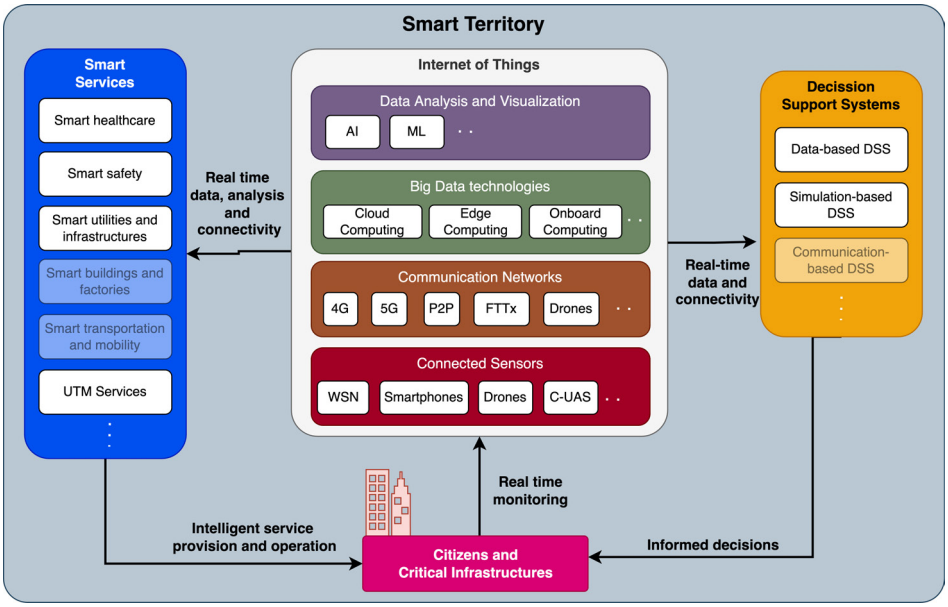


Figure 5.1: Smart territory technological elements considered across the thesis
Technological elements considered in the thesis are highlighted over the full architecture

To start, IoT architectures have been discussed from a data fusion and resource management perspective. Particularly, two applications have been discussed. On one hand, the usage of drones as distributed sensors for wide-area surveillance has been explored. In this case, autonomous flight control and data processing allows to scale the number of drones and the geographical reach of the platform. To this end, the proposed **micro-service architecture for the operation of a swarm of drones** allows to automatically command its flight while retaining explainable supervision of the operation. Also, an indoor testbed platform with digital-twin capabilities has been developed to test the architecture and enable safe and regulation-compliant validation of drone-flight algorithms and communication technologies.

On the other hand, the usage of smartphones as a pervasive data gathering and communication platform for healthcare purposes has been discussed. mHealth applications are a cost-effective way to both geographically and temporally extend the reach of healthcare. However, the lack of standardized comprehensive architectures hiders the development of such applications. In this sense, the proposed **mHealth reference architecture** allows to implement a wide set of common mHealth systems requirements and to integrate health-related AI processing modules. To prove so, the architecture has been instantiated for a SARS-CoV-2 patient monitoring use-case . Overall, both technological applications (i.e.: drone swarms and mHealth) showcase how IoT architectures allow for effective remote monitoring in smart territories.

Besides serving as a monitoring platform, drones can also be used to support communication networks in remote areas, perform infrastructures inspections, or deliver novel aerial mobility and delivery services. Yet, for this to happen, a middleware platform is needed to coordinate unmanned aerial operations in a safe and efficient manner. To this end, the U-Space is the European UTM system: a set of smart services that interconnects operators, authorities and drones to authorized and monitor flight execution. Among the challenges to deploy an operational U-Space ecosystem is the cost and complexity of live demonstrations and validations with real drones. As a solution, the **USPASIM simulation framework** has been presented. The proposal surpasses existing literature by incorporating realistic traffic generation capabilities, a model to consider human pilot behavior and reactions, collaborative surveillance systems (i.e.: ADS-B, FLARM, Remote ID) and non-collaborative (i.e.: radar) surveillance systems.

Another challenge to boost a drone market is the lack of tools to guarantee fair airspace access and prevent congestion in high-demand areas. In this regard, **a new U-Space strategic authorization framework** inspired on European regulations (but surpassing them) has been proposed. The framework is distributable, flexible and allows operators to state their preferences. It differs in time the authorization decision and is articulated through a set of queues of different priority. Access to these queues is controlled through the use of a set of tokens that are assigned to the operators with different fairness criteria and that must be consumed with each submitted flight following a cost that depends on geography and space. Results show how the framework enables the implementation of different notions of fairness that promote different operators' behaviors, as well as to mitigate geographical congestion.

The use of technology in smart territories is not limited to the intelligent provision of services (e.g: through the U-Space) but also extends to planning and decision-making processes. In this sense, IoT data gathering and analysis platforms such as those described above are useful for

informing decision-makers. In addition, simulation tools allow for prospective and comparative scenario analysis. One of the difficulties decision-making in smart territories is considering infrastructures interdependences. This causes a lack of tools that provide a holistic view of them with metrics that can be understood by decision-makers. To alleviate this situation, a conceptual model of the smart territory as a system of systems is proposed. The model can be implemented within a **hybrid ABM and network-based simulation framework that is oriented as a decision-support tool**. An instantiation of the framework for healthcare and ICT use cases demonstrates its uselfuness in both strategic and tactical decision-making.

Finally, the integration of drones into smart territories also requires regulatory decisions that can be informed by data. For this purpose, the USPASIM platform has been used to evaluate **whether different air operations representation models (u-plan) have an impact on airspace efficiency and computational requirements**. Thus, a new usage of the USPASIM platform as a DSS emerges. It has been demonstrated that the platform provides useful metrics to compare the different u-plan alternatives and orient regulators decisions.

Overall, **this thesis has tried to provide technological advances across the whole technological stack of smart territories**: it has developed novel monitoring architectures and applications, provided solutions for the U-Space as an smart-service, and improved existing data and simulation-based decision-support systems capabilities.

5.1 Thesis contributions

A clear identification of the thesis contributions has been depicted in Table 5.1 together with their supporting publications.

Topic	Existing challenges and literature gaps	Thesis Contributions	Supporting Publications
Monitoring technologies for smart territories	• Autonomous drone swarm control with AI may hinder explainable human supervision. • Lack of platforms to perform safe and legal development and validations of drone systems.	• Multi-level swarm control and data fusion architecture for drone-based surveillance operations incorporating AI and human supervision. • Indoor testbed based on digital-twin technologies for safe drone flight and algorithm deployment.	**[Arr+23]** , [Car+24c] , **[San+24]**, [Car+25b], **[Rey+25]**
	• Health monitoring systems based on smartphone applications (mHealth) are built with ad-hoc, costly, non-interoperable solutions. • Existing architectures lack standardized structure and health-related components.	• Microservices based architecture leveraging a health-centric data fusion model to implement a comprehensive set of mHealth requirements with integrated AI modules.	**[Gon+24]**
Smart services enabled by drones: the U-Space ecosystem Smart services	• Slow U-Space deployment due to cost and complexity of validation tests. • Lack of comprehensive simulation framework of the U-Space ecosystem covering human intervention and surveillance sensors.	• USPASIM: U-SPAce ecosystem SIMulation framework with realistic traffic generation, human intervention models, cooperative and non-cooperative surveillance models.	**[Bes+22a], [Bes+22b],** [Rap+23a]
	• European regulation-mandated authorization approach creates fairness issues in airspace access and does not consider congestion management. • Current fairness and congestion management proposals do not alight with European requirements, cannot be distributable / scalable, are not flexible	• Alternative authorization workflow based on a deferred authorization decision with multiple prioritization queues. • A token-based gate-keeping scheme to implement fairness and congestion-management policies while operators state their preferences.	[Car+24b]
Informed decision-making in smart territories	• Resiliency enhancement in critical infrastructures require simulation-based decision support systems. • Existing models do not cover interdependences comprehensively, nor offer useful metrics for decision-making. • Information and Communication systems impact on Healthcare systems have not been previously considered in DSSs.	• Holistic conceptual model of smart territories as a system of measurable systems. • Hybrid simulation framework with multi-level metrics as a decision-support system. • Simulation model for healthcare strategic and operational decision-making in presence of cyberattacks and pandemics.	[Car+23], [Alb+23a] , **[Car+24a], [Car+25a]**
	• Lack of data to inform regulatory decisions on aerial operation representations models (u-plans) to enforce an efficient airspace usage.	• USPASIM usage as a decision-support system for the U-Space ecosystem. • Metrics measuring efficiency and computational cost of different u-plan models	[CMB25]

Table 5.1: Summary of thesis contributions
Publications in bold are references articles, whereas non-highlighted references refer to conference contributions.

5.2 Limitations and future work

Limitations of the thesis's proposals and possible open lines of research have been already identified throughout the previous chapters. Yet, some points are common across all solutions. An improvement point is related to performing **full-scale operational validation tests** instead of relying on the use of simulations/testbed environments. For instance, the deployment of a drone swarm in a real operational environment is yet to be explored. In this sense, communications, security and regulatory issues are expected and should be addressed before a comercial implementation is viable. Similarly, in the case of the proposed mHealth architecture, its deployment for a large number of users should be validated in the future, together with an implementation of EHR interoperability. In the case of the proposed U-Space authorization framework, tests with real operators in coordination with regulators are needed to ensure that the proposal meets their expectations. Also, the comprehensive simulation framework for resiliency-enhancing planning should incorporate more critical infrastructures to become a full-scale DSS. To this end, the integration of the USPASIM simulator could be undertaken to consider urban aerial mobility interdependencies with other infrastructures.

Although architectures and systems that integrate AI modules have been proposed, the development of those models is limited. In this sense, an open point is the **development of AI models that can be integrated into the proposals of the thesis**. This includes AI-assisted drone fleet control algorithms, for which fully distributed approaches could be of interest. Likewise, AI models could be helpful to implement more complex token allocation policies in the context of the proposed U-Space authorization framework. Such models could predict dynamic patterns in operators' behavior (e.g.: flights concentrated in one area, on certain days of the week...) and consider them to ensure fair airspace access. AI techniques can also help to improve the proposed DSS system by integrating results/metrics explainability techniques (e.g.: tracing a fault causality chain through the interdependencies network) or even semi-automatic decision-making features (e.g.: generation of decision alternatives).

There is growing concern about the **regulatory, data protection and ethical issues** of technological solutions, and particularly in AI-based systems. To this end, the European High-Level Expert Group (HLEG) has delivered a set of key ethical requirements that are relevant for the usage of AI-based tools [Hig19]. These guidelines include requirements such as: human oversight, data privacy, transparency and explainability, technical robustness and safety, non-discrimination and fairness, accountability, and societal well-being. Although some of these requirements are implicit in some of the proposed solutions (e.g.: data privacy in mHealth applications, fairness in the authorization framework for U-Space), a complete analysis of the solutions from this perspective is an open point. This ethical analysis could identify new areas of future improvement.

Forward-looking, to enable a vision of smart-territories where drone operations are pervasive, the gap between research and commercial deployments still needs to be bridged. Today's proposals on drones and U-Space need to improve on robustness, reliability and adopt a end-user perspective. This way, it will be possible to develop economically viable business cases that will thrive on tomorrow's smart territories resulting in a more sustainable society.

References

[AAR21] AARONIA AG. *AARTOS X7*. 2021.

[Abd+21] Mohamed Abdelkader, Samet Güler, Hassan Jaleel, and Jeff S. Shamma. "Aerial Swarms: Recent Applications and Challenges". In: *Current Robotics Reports* 2.3 (Sept. 2021), pp. 309–320. DOI: 10.1007/s43154-021-00063-4.

[Abd+23] Safa E. Abdelsamad et al. "Vision-Based Support for the Detection and Recognition of Drones with Small Radar Cross Sections". In: *Electronics* 12.10 (Jan. 2023), p. 2235. DOI: 10.3390/electronics12102235.

[Abr77] Norman Abramson. "THE ALOHA SYSTEM: Another Alternative for Computer Communications". In: *Fall Joint Computer Conference* 37 (Jan. 1977), pp. 281–285. DOI: 10.1145/1478462.1478502.

[Ael+24] Peyman Aela, Hung-Lin Chi, Ali Fares, Tarek Zayed, and Minkoo Kim. "UAV-based Studies in Railway Infrastructure Monitoring". In: *Automation in Construction* 167 (Nov. 2024), p. 105714. DOI: 10.1016/j.autcon.2024.105714.

[Aer23] Aerobits. *idME (remoteID)*. 2023.

[AK20] Shubhani Aggarwal and Neeraj Kumar. "Path Planning Techniques for Unmanned Aerial Vehicles: A Review, Solutions, and Challenges". In: *Computer Communications* 149 (Jan. 2020), pp. 270–299. DOI: 10.1016/j.comcom.2019.10.014.

[Aha+20] Mohd Abdul Ahad, Sara Paiva, Gautami Tripathi, and Noushaba Feroz. "Enabling Technologies and Sustainable Smart Cities". In: *Sustainable Cities and Society* 61 (Oct. 2020), p. 102301. DOI: 10.1016/j.scs.2020.102301.

[Ai+20] Bo Ai, Andreas F. Molisch, Markus Rupp, and Zhang-Dui Zhong. "5G Key Technologies for Smart Railways". In: *Proceedings of the IEEE* 108.6 (June 2020), pp. 856–893. DOI: 10.1109/JPROC.2020.2988595.

[Air22] Airbus. *Airbus UTM's USim Brings Advanced Simulation Technology to Researchers Exploring Future Traffic Management Operations*. https://acubed.airbus.com/blog/airbus-utm/airbus-ut-ms-u-sim-brings-advanced-simulation-technology-to-researchers-exploring-future-traffic-management-operations/. (accessed 2025-01-13). 2022.

[Air25] Airelectronics. *X8 Flying Wing*. https://www.aeroexpo.online/prod/airelectronics/product-185532-28782.html. (accessed 2025-06-21). 2025.

[Alb+23a] César Alberte, David Carramiñana, Ana M. Bernardos, and Juan A. Besada. "Managing Pandemics Through Agent-Based Simulation: A Case Study Based on COVID-19". In: *18th International Conference on Soft Computing Models*

in Industrial and Environmental Applications (SOCO 2023). Cham: Springer Nature Switzerland, 2023, pp. 185–195. DOI: 10.1007/978-3-031-42529-5_18.

[Alb+23b] César Alberte, David Carramiñana, Ana M. Bernardos, and Juan A. Besada. "Managing Pandemics Through Agent-Based Simulation: A Case Study Based on COVID-19". In: *18th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2023)*. Ed. by Pablo García Bringas et al. Lecture Notes in Networks and Systems. Cham: Springer Nature Switzerland, 2023, pp. 185–195. DOI: 10.1007/978-3-031-42529-5_18.

[Ali+23] Shrouk A. Ali, Shaimaa Ahmed Elsaid, Abdelhamied A. Ateya, Mohammed ElAffendi, and Ahmed A. Abd El-Latif. "Enabling Technologies for Next-Generation Smart Cities: A Comprehensive Review and Research Directions". In: *Future Internet* 15.12 (Dec. 2023), p. 398. DOI: 10.3390/fi15120398.

[Ama23] Amazon. *Amazon Is Launching Ultra-Fast Drone Deliveries in Italy, the UK, and a Third Location in the U.S.* https://www.aboutamazon.com/news/operations/amazon-prime-air-drone-delivery-updates. (accessed 2025-04-16). Oct. 2023.

[Anw+23] Anwen Tobalske, Natalie Chen, Brian Lovinguth, and Nick Mageras. *Proprietary IQVIA Digital Health Data Assets and Expertise for Decision Making*. Tech. rep. 2023.

[Aou+23] Driss Aouladhadj, Ettien Kpre, Virginie Deniau, Aymane Kharchouf, Christophe Gransart, and Christophe Gaquière. "Drone Detection and Tracking Using RF Identification Signals". In: *Sensors* 23.17 (Jan. 2023), p. 7650. DOI: 10.3390/s23177650.

[Ard13] Ardupilot Team. *MissionPlanner: Mission Planner Ground Control Station for ArduPilot*. 2013.

[Arr+23] Raúl Arranz, David Carramiñana, Gonzalo de Miguel, Juan A. Besada, and Ana M. Bernardos. "Application of Deep Reinforcement Learning to UAV Swarming for Ground Surveillance". In: *Sensors* 23.21 (Jan. 2023), p. 8766. DOI: 10.3390/s23218766.

[ART20] ART. *Drone Detection Radar - ART Drone Sentinel - Anti Drone Radar System*. https://www.advancedradartechnologies.com/products/art-drone-sentinel-2/. (accessed 2021-11-05). 2020.

[ASD21a] ASD-STAN. *Aerospace Series-Unmanned Aircraft Systems — Part 002: Direct Remote Identification*. 2021.

[ASD21b] ASD-STAN D05/WG08. *Direct Remote ID. Introduction to the European Uas Digital Remote Id Technical Standard*. 2021.

[Ash+16] Reem Ashour et al. "Site Inspection Drone: A Solution for Inspecting and Regulating Construction Sites". In: *2016 IEEE 59th International Midwest Symposium on Circuits and Systems (MWSCAS)*. Oct. 2016, pp. 1–4. DOI: 10.1109/MWSCAS.2016.7870116.

[Ass22] Associated Press. *Heavy Russian Barrage on Ukraine, No Water for Much of Kyiv*. https://apnews.com/article/russia-ukraine-kyiv-black-sea-europe-kharkiv-870091b0922a131aefb2692bb88f5c6d. (accessed 2023-07-13). Oct. 2022.

[AST19] ASTM. *F3411-19. Standard Specification for Remote ID and Tracking*. 2019. DOI: 10.1520/F3411-19.

[AST22] ASTM. *F3548 Standard Specification for UAS Traffic Management (UTM) UAS Service Supplier (USS) Interoperability*. Mar. 2022.

[AGC22] Jean-Philippe Aurambout, Konstantinos Gkoumas, and Biagio Ciuffo. "A Drone Hop from the Local Shop? Where Could Drone Delivery as a Service Happen in Europe and the USA, and How Many People Could Benefit from It?" In: *Transportation Research Interdisciplinary Perspectives* 16 (Dec. 2022), p. 100708. DOI: 10.1016/j.trip.2022.100708.

[AOJ22] Kebira Azbeg, Ouail Ouchetto, and Said Jai Andaloussi. "BlockMedCare: A Healthcare System Based on IoT, Blockchain and IPFS for Data Management Security". In: *Egyptian Informatics Journal* (July 2022). DOI: 10.1016/J.EIJ.2022.02.004.

[Băd+11] Costin Bădică, Zoran Budimac, Hans-Dieter Burkhard, and Mirjana Ivanovic. "Software Agents: Languages, Tools, Platforms". In: *Computer Science and Information Systems* 8.2 (2011), pp. 255–298. DOI: 10.2298/CSIS110214013B.

[BLG16] Mathieu Bagot, Pascale Launay, and Frédéric Guidec. "A Flexible Architecture for Mobile Health Monitoring". In: *2016 IEEE 18th International Conference on e-Health Networking, Applications and Services, Healthcom* (Nov. 2016), pp. 1–6. DOI: 10.1109/HEALTHCOM.2016.7749442.

[Ban+15] Oresti Banos et al. "Design, Implementation and Validation of a Novel Open Framework for Agile Development of Mobile Health Applications". In: *BioMedical Engineering Online* 14.2 (2015), pp. 1–20. DOI: 10.1186/1475-925X-14-S2-S6.

[BDH09] Yaakov Bar-Shalom, Fred Daum, and Jim Huang. "The Probabilistic Data Association Filter". In: *IEEE Control Systems Magazine* 29.6 (Dec. 2009), pp. 82–100. DOI: 10.1109/MCS.2009.934469.

[BWT11] Yaakov Bar-Shalom, Peter K. Willet, and Xian Tian. *Tracking and Data Fusion: A Handbook of Algorithms*. Yaakov Bar-Shalom, 2011.

[Bar+22] Palak Barapatre, Yash Ingolikar, Prajakta Desai, Pooja Jajoo, and Prasheel Thakre. "A Secured Architecture for IoT-based Healthcare System". In: *3C Empresa: Investigación y pensamiento crítico* 11.2 (2022), pp. 222–230. DOI: 10.17993/3cemp.2022.110250.222-230.

[Bar+15] G. Bartoli, R. Fantacci, F. Gei, D. Marabissi, and L. Micciullo. "A Novel Emergency Management Platform for Smart Public Safety". In: *International Journal of Communication Systems* 28.5 (2015), pp. 928–943. DOI: 10.1002/dac.2716.

[Bar04] David K. Barton. *Radar System Analysis and Modeling*. Artech House, Oct. 2004.

[BA15] Sara Beheshtifar and Abbas Alimoahmmadi. "A Multiobjective Optimization Approach for Location-Allocation of Clinics". In: *International Transactions in Operational Research* 22.2 (2015), pp. 313–328. DOI: 10.1111/itor.12088.

[Ber+20] Emily Zechman Berglund et al. "Smart Infrastructure: A Vision for the Role of the Civil Engineering Profession in Smart Cities". In: *Journal of Infrastructure Systems* 26.2 (June 2020), p. 03120001. DOI: 10.1061/(ASCE)IS.1943-555X.0000549.

[Ber+12] R. D. Berndt, M. C. Takenga, S. Kuehn, P. Preik, G. Sommer, and S. Berndt. "SaaS-platform for Mobile Health Applications". In: *International Multi-Conference on Systems, Signals and Devices* (2012), pp. 1–4. DOI: 10.1109/SSD.2012.6198120.

[Ber+19] Sofian Berrouiguet, María Luisa Barrigón, Jorge Lopez Castroman, Philippe Courtet, Antonio Artés-Rodríguez, and Enrique Baca-García. "Combining Mobile-Health (mHealth) and Artificial Intelligence (AI) Methods to Avoid Suicide Attempts: The Smartcrises Study Protocol". In: *BMC Psychiatry* 19.1 (2019), pp. 1–9. DOI: 10.1186/s12888-019-2260-y.

[BFT11] Dimitris Bertsimas, Vivek F. Farias, and Nikolaos Trichakis. "The Price of Fairness". In: *Operations Research* 59.1 (Feb. 2011), pp. 17–31. DOI: 10.1287/opre.1100.0865.

[Bes+20] Juan Besada, Ivan Campaña, Luca Bergesio, Ana Bernardos, and Gonzalo de Miguel. "Drone Flight Planning for Safe Urban Operations". In: *Personal and Ubiquitous Computing* (Jan. 2020). DOI: 10.1007/s00779-019-01353-7.

[Bes+22a] Juan A. Besada, Ivan Campaña, David Carramiñana, Luca Bergesio, and Gonzalo de Miguel. "Review and Simulation of Counter-UAS Sensors for Unmanned Traffic Management". In: *Sensors* 22.1 (Jan. 2022), p. 189. DOI: 10.3390/s22010189.

[Bes+22b] Juan A. Besada, David Carramiñana, Luca Bergesio, Ivan Campaña, and Ana M. Bernardos. "Modelling and Simulation of Collaborative Surveillance for Unmanned Traffic Management". In: *Sensors* 22.4 (Jan. 2022), p. 1498. DOI: 10.3390/s22041498.

[Bes+18] Juan A. Besada et al. "Drone Mission Definition and Implementation for Automated Infrastructure Inspection Using Airborne Sensors". In: *Sensors* 18.4 (Apr. 2018), p. 1170. DOI: 10.3390/s18041170.

[Bia17] Andrzej Bialas. "Computer Support for Risk Management in Critical Infrastructures". In: *Advances in Network Systems*. Ed. by Maciej Grzenda, Ali Ismail Awad, Janusz Furtak, and Jarosław Legierski. Advances in Intelligent Systems and Computing. Cham: Springer International Publishing, 2017, pp. 197–216. DOI: 10.1007/978-3-319-44354-6_12.

[BT20] Hamoud S. Bin-Obaid and Theodore B. Trafalis. "Fairness in Resource Allocation: Foundation and Applications". In: *Network Algorithms, Data Mining, and Applications*. Ed. by Ilya Bychkov, Valery A. Kalyagin, Panos M. Pardalos, and Oleg Prokopyev. Cham: Springer International Publishing, 2020, pp. 3–18. DOI: 10.1007/978-3-030-37157-9_1.

[Bis+18] Igor Bisio, Chiara Garibotto, Fabio Lavagetto, Andrea Sciarrone, and Sandro Zappatore. "Unauthorized Amateur UAV Detection Based on WiFi Statistical Fingerprint Analysis". In: *IEEE Communications Magazine* 56.4 (Apr. 2018), pp. 106–111. DOI: 10.1109/MCOM.2018.1700340.

[Bje+21] Daniel Bjelica, Artur Bjelica, Marijana Despotović-Zrakić, Božidar Radenković, Dušan Barać, and Marko Đogatović. "Designing an IT Ecosystem for Pregnancy Care Management Based on Pervasive Technologies". In: *Healthcare* 9.1 (Dec. 2021). DOI: 10.3390/HEALTHCARE9010012.

[Bo 22] Italo Buleje Bo Wen Vince S. Siu. "Health Guardian Platform: A Technology Stack to Accelerate Discovery in Digital Health Research". In: *IEEE Interna-*

tional Conference on Digital Health (ICDH) (2022), pp. 40–46. DOI: 10.1109/ICDH55609.2022.00015.

[BS09] Christopher L Bowman and Alan N. Steinberg. "Systems Engineering Approach for Implementing Data Fusion Systems". In: *Handbook of Multisensor Data Fusion*. 2nd ed. 2009.

[Bur+11] Michelle Nicole Burns et al. "Harnessing Context Sensing to Develop a Mobile Intervention for Depression". In: *Journal of medical Internet research* 13.3 (Aug. 2011), e1838. DOI: 10.2196/JMIR.1838.

[Bus+05] Brian Bush, L. Dauelsberg, Rene Leclaire, D. Powell, and S. DeLand. *Critical Infrastructure Protection Decision Support System (CIP/DSS) Project Overview*. Jan. 2005.

[But24] Philip Butterworth-Hayes. *Is Europe Ready for U-space? Yes – But...* Nov. 2024.

[Cab+24] Daniel Caballero-Martin, Jose Manuel Lopez-Guede, Julian Estevez, and Manuel Graña. "Artificial Intelligence Applied to Drone Control: A State of the Art". In: *Drones* 8.7 (July 2024), p. 296. DOI: 10.3390/drones8070296.

[CT12] Margarita Cabrera and Francesc Tarrés. *Multiplexación por división en frecuencias ortogonales (OFDM)*. 2012.

[Cam15] Cambridge Centre For Risk Studies. *World Cities Risk 2015-2025*. Sept. 2015.

[Cam20] Iván Campaña Ramos. "Contributions to Drone Fleet Mission Planning and Monitoring". PhD thesis. E.T.S.I. Telecomunicación (UPM), 2020.

[Cam+24] Hamish A. Campbell et al. "Emerging Research Topics in Drone Healthcare Delivery". In: *Drones* 8.6 (June 2024), p. 258. DOI: 10.3390/drones8060258.

[Cam+15] Daniel Camps-Mur, Eduard Garcia-Villegas, Elena Lopez-Aguilera, Paulo Loureiro, Paul Lambert, and Ali Raissinia. "Enabling Always on Service Discovery: Wifi Neighbor Awareness Networking". In: *IEEE Wireless Communications* 22.2 (Apr. 2015), pp. 118–125. DOI: 10.1109/MWC.2015.7096294.

[CZL22] Junwei Cao, Guihua Zhang, and Dong Liu. "The Impact of Using mHealth Apps on Improving Public Health Satisfaction during the COVID-19 Pandemic: A Digital Content Value Chain Perspective". In: *Healthcare, MDPI* 10.3 (2022), p. 479. DOI: 10.3390/HEALTHCARE10030479.

[Car+21a] Stuart W. Card, Adam Wiethuechter, Robert Moskowitz, and Anei Gurtov. *Drone Remote Identification Protocol (DRIP) Requirements*. Internet Draft draft-ietf-drip-reqs-18. Internet Engineering Task Force, Sept. 2021.

[Car+23] David Carramiñana, Ana M. Bernardos, Juan A. Besada, and José R. Casar. "Enhancing Health Care Infrastructure Resiliency Through an Agent-Based Simulation Methodology". In: *2023 International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*. Sept. 2023, pp. 1–4. DOI: 10.1109/ICT-DM58371.2023.10286957.

[Car+24a] David Carramiñana, Ana M. Bernardos, Juan A. Besada, and José R. Casar. "Towards Resilient Cities: A Hybrid Simulation Framework for Risk Mitigation through Data-Driven Decision Making". In: *Simulation Modelling Practice and Theory* 133 (May 2024), p. 102924. DOI: 10.1016/j.simpat.2024.102924.

[Car+25a] David Carramiñana, Ana M. Bernardos, Juan A. Besada, and José R. Casar. "Enhancing Healthcare Infrastructure Resilience through Agent-Based Simulation

Methods". In: *Computer Communications* 234 (Mar. 2025), p. 108070. DOI: 10.1016/j.comcom.2025.108070.

[Car+24b] David Carramiñana, Juan A. Besada, José M. Molina, and Ana M. Bernardos. "Scalable U-Space Authorization Workflow Enabling Fairness and Capacity Management". In: *SESAR Innovation Days 2024*. Rome, Italy, 2024.

[Car+21b] David Carramiñana, Iván Campaña, Luca Bergesio, Ana M. Bernardos, and Juan A. Besada. "Sensors and Communication Simulation for Unmanned Traffic Management". In: *Sensors* 21.3 (Jan. 2021), p. 927. DOI: 10.3390/s21030927.

[Car+21c] David Carramiñana, Iván Campaña, Luca Bergesio, Ana M. Bernardos, and Juan A. Besada. "Sensors and Communication Simulation for Unmanned Traffic Management". In: *Sensors* 21.3 (Jan. 2021), p. 927. DOI: 10.3390/s21030927.

[CMB25] David Carramiñana, José M. Molina, and Juan Besada. "Evaluation of U-Plan Representation Impact on U-Space Capacity and Computational Needs". In: *25th Integrated Communications, Navigation and Surveillance Conference*. Brussels, Belgium, Apr. 2025.

[Car+25b] David Carramiñana, Lucas Rey, Raúl Arranz, Juan Besada, and Ana Bernardos. "An AI-assisted Multi-Drone Architecture for Surveillance Operations". In: *International Conference on on Drones and Unmanned Systems*. Granada, Spain, 2025.

[Car+24c] David Carramiñana et al. "A Digital Twin Concept to Prototype Multi-Drone Based Applications". In: *14th International Conference on the Internet of Things*. Oulu, Finland, 2024.

[Cec+19] Nathália Pinto Cechetti, Ericles Andrei Bellei, Daiana Biduski, João Pedro Mazuco Rodriguez, Mateus Klein Roman, and Ana Carolina Bertoletti De Marchi. "Developing and Implementing a Gamification Method to Improve User Engagement: A Case Study with an m-Health Application for Hypertension Monitoring". In: *Telematics and Informatics* (2019). DOI: 10.1016/j.tele.2019.04.007.

[Cha+18] Xianyu Chang, Chaoqun Yang, Junfeng Wu, Xiufang Shi, and Zhiguo Shi. "A Surveillance System for Drone Localization and Tracking Using Acoustic Arrays". In: *2018 IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM)*. July 2018, pp. 573–577. DOI: 10.1109/SAM.2018.8448409.

[Chi14] Virgil Chichernea. "The Use of Decision Support Systems (Dss) in Smart City Planning and Management". In: *Journal of Information Systems & Operations Management* (2014), pp. 1–14.

[Chi+20] Christopher Chin, Karthik Gopalakrishnan, Hamsa Balakrishnan, Maxim Egorov, and Antony Evans. "Tradeoffs between Efficiency and Fairness in Unmanned Aircraft Systems Traffic Management". In: (2020).

[Chi+23a] Christopher Chin, Karthik Gopalakrishnan, Hamsa Balakrishnan, Maxim Egorov, and Antony Evans. "Protocol-Based Congestion Management for Advanced Air Mobility". In: *Journal of Air Transportation* 31.1 (Jan. 2023), pp. 35–44. DOI: 10.2514/1.D0298.

[Chi+21a] Christopher Chin, Karthik Gopalakrishnan, Maxim Egorov, Antony Evans, and Hamsa Balakrishnan. "Efficiency and Fairness in Unmanned Air Traffic Flow

Management". In: *IEEE Transactions on Intelligent Transportation Systems* 22.9 (Sept. 2021), pp. 5939–5951. DOI: 10.1109/TITS.2020.3048356.

[Chi+21b] Christopher Chin, Karthik Gopalakrishnan, Maxim Egorov, Antony Evans, and Hamsa Balakrishnan. "Efficiency and Fairness in Unmanned Air Traffic Flow Management". In: *IEEE Transactions on Intelligent Transportation Systems* 22.9 (Sept. 2021), pp. 5939–5951. DOI: 10.1109/TITS.2020.3048356.

[Chi+23b] Christopher Chin, Victor Qin, Karthik Gopalakrishnan, and Hamsa Balakrishnan. "Traffic Management Protocols for Advanced Air Mobility". In: *Frontiers in Aerospace Engineering* 2 (May 2023). DOI: 10.3389/fpace.2023.1176969.

[Chi+23c] Christopher Chin, Victor Qin, Karthik Gopalakrishnan, and Hamsa Balakrishnan. "Traffic Management Protocols for Advanced Air Mobility". In: *Frontiers in Aerospace Engineering* 2 (May 2023). DOI: 10.3389/fpace.2023.1176969.

[Chi+21c] Sheng Min Chiu, Yi Chung Chen, Jou Wei Lin, Cheng Hsien Tsai, and Chiang Lee. "Interactive Mobile App for Self-Supervised Health Management". In: *3rd IEEE Eurasia Conference on Biomedical Engineering, Healthcare and Sustainability, ECBIOS 2021* (2021). DOI: 10.1109/ECBIOS51820.2021.9510524.

[CC19] Su Yeon Choi and Dowan Cha. "Unmanned Aerial Vehicles Using Machine Learning for Autonomous Flight; State-of-the-Art". In: *Advanced Robotics* 33 (Mar. 2019), pp. 1–13. DOI: 10.1080/01691864.2019.1586760.

[Cho+21] Tilendra Choudhary, L. N. Sharma, M. K. Bhuyan, and Kangkana Bora. "Identification of Human Breathing-States Using Cardiac-Vibrational Signal for m-Health Applications". In: *IEEE Sensors Journal* 21.3 (Feb. 2021), pp. 3463–3470. DOI: 10.1109/JSEN.2020.3025384.

[Cla+25] Cecilia Claramunt, Marina Jiménez, Pablo Gascó, and David Phy. "ACUTE Project - Facts and Findings on UAS Traffic in European Cities". In: *Integrated Communication Navigation and Surveillance Conference*. Brussels, Belgium, Apr. 2025.

[CSF21] Adam P. Cohen, Susan A. Shaheen, and Emily M. Farrar. "Urban Air Mobility: History, Ecosystem, Market Potential, and Challenges". In: *IEEE Transactions on Intelligent Transportation Systems* 22.9 (Sept. 2021), pp. 6074–6087. DOI: 10.1109/TITS.2021.3082767.

[CKL24] Andrew Collins, Matthew Koehler, and Christopher Lynch. "Methods That Support the Validation of Agent-Based Models: An Overview and Discussion". In: *Journal of Artificial Societies and Social Simulation* 27.1 (2024), p. 11.

[CS20] CORUS project and SESAR Joint Undertaking. *Concept of Operations for European UTM Systems. 3rd Edition*. Sept. 2020.

[COR23] CORUS-XUAM. *Concept of Operations for European UTM Systems. 4th Edition*. v4, July 2023.

[DAC22] DACUS Project. *Final Optimized Drone DCB*. Aug. 2022.

[Dam13] Renata Dameri. "Searching for Smart City Definition: A Comprehensive Proposal". In: *International Journal of Computers & Technology* 11 (Oct. 2013), p. 2544. DOI: 10.24297/ijct.v11i5.1142.

[Dan+19] Yangpeng Dan, Jianxin Yi, Xianrong Wan, Yunhua Rao, and Benjing Wang. "LTE-based Passive Radar for Drone Detection and Its Experimental Results".

In: *The Journal of Engineering* 2019.20 (2019), pp. 6910–6913. DOI: 10.1049/joe.2019.0583.

[Dan+20] Shuping Dang, Osama Amin, Basem Shihada, and Mohamed-Slim Alouini. "What Should 6G Be?" In: *Nature Electronics* 3.1 (Jan. 2020), pp. 20–29. DOI: 10.1038/s41928-019-0355-6.

[Ded21] Dedrone. *Dedrone First to Offer Both United States and European Union Drone Remote ID Capability*. https://www.dedrone.com/press/dedrone-first-to-offer-both-united-states-and-european-union-drone-remote-id-capability. (accessed 2021-11-10). Feb. 2021.

[Ded23] Dedrone. *Counter-Drone Solution - Dedrone*. https://www.dedrone.com/products/counter-drone-solution. (accessed 2021-11-05). 2023.

[dVG09] Isabel del Pozo de Poza, Miguel Vilaplana Ruiz, and Colin Goodchild. "Assessing Fairness and Equity in Trajectory-Based Operations". In: *9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO)*. Aviation Technology, Integration, and Operations (ATIO) Conferences. American Institute of Aeronautics and Astronautics, Sept. 2009. DOI: 10.2514/6.2009-6973.

[del+20] Ana Belén del Río-Lanza, Ana Suárez-Vázquez, Leticia Suárez-Álvarez, and Víctor Iglesias-Argüelles. "Mobile Health (Mhealth): Facilitators and Barriers of the Intention of Use in Patients with Chronic Illnesses". In: *Journal of Communication in Healthcare* 13.2 (Apr. 2020), pp. 138–146. DOI: 10.1080/17538068.2020.1777513.

[dev21] ONNX Runtime developers. *ONNX Runtime*. 2021.

[Dia+22] Heiman Dianat, Suzanne Wilkinson, Peter Williams, and Hamed Khatibi. "Choosing a Holistic Urban Resilience Assessment Tool". In: *International Journal of Disaster Risk Reduction* 71 (Mar. 2022), p. 102789. DOI: 10.1016/j.ijdrr.2022.102789.

[DR04] Julius V. DiFranco and William L. Rubin. *Radar Detection*. Institution of Engineering and Technology, Jan. 2004. DOI: 10.1049/SBRA028E.

[DK09] Susanne Dirks and Mary Keeling. *A Vision of Smarter Cities: How Cities Can Lead the Way into a Prosperous and Sustainable Future*. Dec. 2009.

[DJI17] DJI. *Aeroscope*. https://www.dji.com/es/aeroscope. (accessed 2025-05-05). 2017.

[DJI19] DJI. *DJI Assistant 2*. 2019.

[DJI25] DJI. *DJI Matrice 400*. https://enterprise.dji.com/es/photo. (accessed 2025-06-21). 2025.

[Do+21] Hai T. Do et al. "Energy-Efficient Unmanned Aerial Vehicle (UAV) Surveillance Utilizing Artificial Intelligence (AI)". In: *Wireless Communications and Mobile Computing* 2021.1 (2021), p. 8615367. DOI: 10.1155/2021/8615367.

[Dor20] Vidal Dordal. *An Introduction to Computer Networks*. 2nd ed. 2020.

[Dor+24] Daniela Doroftei, Geert De Cubber, Salvatore Lo Bue, and Hans De Smet. "Quantitative Assessment of Drone Pilot Performance". In: *Drones* 8.9 (Sept. 2024), p. 482. DOI: 10.3390/drones8090482.

[DPS14] Monica Dragoicea, Monica Patrascu, and George Alexandru Serea. "Real Time Agent Based Simulation for Smart City Emergency Protocols". In: *2014 18th*

International Conference on System Theory, Control and Computing (ICSTCC). Oct. 2014, pp. 187–192. DOI: 10/gm64nk.

[Dro19a] Dronecode Project. *QGroundControl*. https://qgroundcontrol.com/. (accessed 2025-05-11). 2019.

[Dro19b] DroneShield. *DroneSentry*. https://www.droneshield.com/dronesentry. (accessed 2021-10-25). 2019.

[Dro20] Droniq. *HOD4track*. https://droniq.de/en/products/hod4track-deutschland. (accessed 2021-11-02). 2020.

[Dro21] Droniq. *GBSAS Stationary*. https://droniq.de/en/products/gbsas-stationar-deutschland. (accessed 2021-11-02). 2021.

[Du+18] Yao Du, Kezhi Wang, Kun Yang, and Guopeng Zhang. "Energy-Efficient Resource Allocation in UAV Based MEC System for IoT Devices". In: *2018 IEEE Global Communications Conference (GLOBECOM)*. Dec. 2018, pp. 1–6. DOI: 10.1109/GLOCOM.2018.8647789.

[DM17] Matthew Duffield and Tim W. McLain. "A Well Clear Recommendation for Small UAS in High-Density, ADS-B-Enabled Airspace". In: *AIAA Information Systems-AIAA Infotech @ Aerospace*. American Institute of Aeronautics and Astronautics, 2017. DOI: 10.2514/6.2017-0908.

[EAS16] EASA. *Warsaw Declaration: «Drones as a Leverage for Jobs and New Business Opportunities»*. Nov. 2016.

[EAS22a] EASA. *ED Decision 2022/022/R - AMC and GM to Implementing Regulation (EU) 2021/664 — Issue 1*. Dec. 2022.

[EAS22b] EASA. *Technical Specification for ADS-L Transmissions Using SRD-860 Frquency Band*. Dec. 2022.

[EAS25] EASA. *Drones & Air Mobility Basics Explained | EASA*. https://www.easa.europa.eu/en/domains/drones-air-mobility/drones-air-mobility-landscape/basics-explained. (accessed 2025-04-16). 2025.

[Eby95] M. S. Eby. "A Self-Organizational Approach for Resolving Air Traffic Conflicts". In: Sept. 1995.

[Els+21] Mohamed Elsayed, Mohamed Reda, Ahmed S. Mashaly, and Ahmed S. Amein. "Review on Real-Time Drone Detection Based on Visual Band Electro-Optical (EO) Sensor". In: *2021 Tenth International Conference on Intelligent Computing and Information Systems (ICICIS)*. Dec. 2021, pp. 57–65. DOI: 10.1109/ICICIS52592.2021.9694151.

[Elv+21] Luís B. Elvas et al. "Disaster Management in Smart Cities". In: *Smart Cities* 4.2 (2021). DOI: 10.3390/smartcities4020042.

[EL17] Peyman Emamian and Juan Li. "SimpleHealth - A Mobile Cloud Platform to Support Lightweight Mobile Health Applications for Low-End Cellphones". In: *2017 IEEE 19th International Conference on e-Health Networking, Applications and Services, Healthcom* (Dec. 2017), pp. 1–6. DOI: 10.1109/HEALTHCOM.2017.8210759.

[Ema+23] Mathieu Emanuelli, Christian Passarge, Hartmann Estruch, and Matt Stewart. *Digital Transformation for Consumer Health: A Proven Approach to Enhance Consumer Health Customer Engagement*. Tech. rep. 2023.

[ENA21] ENAIRE. *ENAIRE invierte más de 1,3 millones de euros en su plataforma de gestión automatizada y digital de drones (U-Space)*. https://www.enaire.es/es_ES/2021_08_26/ndp_adjudicacion_u-space. (accessed 2025-01-10). 2021.

[Esc24] Damià Fuentes Escoté. *DJITelloPy*. Dec. 2024.

[EUR09a] EUROCAE. *RTCA DO-260 - Minimum Operational Performance Standards for 1090 MHz Extended Squitter Automatic Dependent Surveillance – Broadcast (ADS-B) and Traffic Information Services – Broadcast (TIS-B) | Engineering360*. 2009.

[EUR09b] EUROCONTROL. *ESCAPE*. https://www.eurocontrol.int/simulator/escape. (accessed 2025-05-05). 2009.

[EUR22] EUROCONTROL. *U-SPACE Services Implementation Monitoring Report*. Oct. 2022.

[Eur19] European Commission. *Commission Delegated Regulation (EU) 2019/945 of 12 March 2019 on Unmanned Aircraft Systems and on Third-Country Operators of Unmanned Aircraft Systems*. June 2019.

[Eur21] European Commission. *Commission Implementing Regulation (EU) 2021/664 of 22 April 2021 on a Regulatory Framework for the U-space (Text with EEA Relevance)*. Apr. 2021.

[Eur22a] European Commission. *A Drone Strategy 2.0 for a Smart and Sustainable Unmanned Aircraft Eco-System in Europe*. Nov. 2022.

[Eur22b] European Directive. *Directive (EU) 2022/2557 of the European Parliament and of the Council of 14 December 2022 on the Resilience of Critical Entities and Repealing Council Directive 2008/114/EC*. 2022.

[EEM20] Antony D. Evans, Maxim Egorov, and Steven Munn. "Fairness in Decentralized Strategic Deconfliction in UTM". In: *AIAA Scitech 2020 Forum*. Orlando, FL: American Institute of Aeronautics and Astronautics, Jan. 2020. DOI: 10.2514/6.2020-2203.

[Fan+12] Xi Fang, Satyajayant Misra, Guoliang Xue, and Dejun Yang. "Smart Grid — The New and Improved Power Grid: A Survey". In: *IEEE Communications Surveys & Tutorials* 14.4 (2012), pp. 944–980. DOI: 10.1109/SURV.2011.101911.00087.

[Faz+22] Matia Fazio et al. "LinguAPP: An m-Health Application for Teledentistry Diagnostics". In: *International Journal of Environmental Research and Public Health* (2022). DOI: 10.3390/ijerph19020822.

[Fed20] Federal Aviation Administration. *Unmanned Aircraft Systems (UAS) Traffic Management (UTM) Concept of Operations V2.0*. Mar. 2020.

[Fed21a] Federal Office for Information Security. *What Are Critical Infrastructures?* https://www.bsi.bund.de/EN/Themen/Regulierte-Wirtschaft/Kritische-Infrastrukturen/Allgemeine-Infos-zu-KRITIS/allgemeine-infos-zu-kritis.html?nn=911334. (accessed 2025-05-05). 2021.

[Fed21b] Federal Register. *Remote Identification of Unmanned Aircraft*. Jan. 2021.

[FIW24] FIWARE. *FIWARE FOR SMART CITIES AND TERRITORIES: A DIGITAL TRANSFORMATION JOURNEY*. 2024.

[FLA21] FLARM. *Atom UAV*. https://www.flarm.com/en/drones/atom-uav/. (accessed 2021-10-26). 2021.

[FLA18] FLARM Technology. *FLARM UAS ELECTRONIC ID*. https://flarm.com/technology/eid/. (accessed 2020-09-27). 2018.

[FLI22] FLIR. *Radar de panel de vigilancia de drones en aire y en tierra del Ranger R8SS-3D | Teledyne FLIR*. https://www.flir.es/products/ranger-r8ss-3d/. (accessed 2021-11-05). 2022.

[Fol+23] Frans Folkvord, Nadine Bol, Giacomo Stazi, Lutz Peschke, and Francisco Lupiáñez-Villanueva. "Preferences in the Willingness to Download an mHealth App: Discrete Choice Experimental Study in Spain, Germany, and the Netherlands". In: *JMIR Formative Research* 7 (Dec. 2023), e48335. DOI: 10.2196/48335.

[Fre+22] Rodolphe Fremond et al. "Demonstrating Advanced U-space Services for Urban Air Mobility in a Co-Simulation Environment". In: *12th SESAR Innovation Days*. Budapest, Dec. 2022.

[FM14] Marcia R. Friesen and Robert D. McLeod. "A Survey of Agent-Based Modeling of Hospital Environments". In: *IEEE Access* 2 (2014), pp. 227–233. DOI: 10.1109/ACCESS.2014.2313957.

[Fro+19] Guillermo Frontera, Ivan Campaña, Ana M. Bernardos, and Juan A. Besada. "Formal Intent-Based Trajectory Description Languages for Quadrotor Aircraft". In: *IEEE Transactions on Aerospace and Electronic Systems* 55.6 (Dec. 2019), pp. 3330–3346. DOI: 10.1109/TAES.2019.2907396.

[Fro+21] Guillermo Frontera, Jesús Cuadrado, Emiliano Bartolomé, and Carlos Querejeta. "SkyWay Simulator: An Integrated ATM/UTM Simulator for Autonomous Operations". In: *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*. Oct. 2021, pp. 1–9. DOI: 10.1109/DASC52595.2021.9594294.

[GKK23] Panagiota Galetsi, Korina Katsaliaki, and Sameer Kumar. "Exploring Benefits and Ethical Challenges in the Rise of mHealth (Mobile Healthcare) Technology for the Common Good: An Analysis of Mobile Applications for Health Specialists". In: *Technovation* 121 (2023), p. 102598. DOI: 10.1016/J.TECHNOVATION.2022.102598.

[GK88] P.P. Gandhi and S.A. Kassam. "Analysis of CFAR Processors in Nonhomogeneous Background". In: *IEEE Transactions on Aerospace and Electronic Systems* 24.4 (July 1988), pp. 427–445. DOI: 10.1109/7.7185.

[Gar+20] Victor M Garcia et al. "Management of Real-Time Data for a Smart Flooding Alert System". In: *2020 IEEE International Smart Cities Conference (ISC2)*. Sept. 2020, pp. 1–8. DOI: 10.1109/ISC251055.2020.9239092.

[Gar22] P. K. Garg. "Characterisation of Fixed-Wing Versus Multirotors UAVs/Drones". In: *Journal of Geomatics* 16.2 (Oct. 2022), pp. 152–159. DOI: 10.58825/jog.2022.16.2.44.

[GNW22] Paolo Gerli, Julio Navio Marco, and Jason Whalley. "What Makes a Smart Village Smart? A Review of the Literature". In: *Transforming Government: People, Process and Policy* 16.3 (Jan. 2022), pp. 292–304. DOI: 10.1108/TG-07-2021-0126.

[Gha+18] Mohammad Ghamari et al. "Detailed Examination of a Packet Collision Model for Bluetooth Low Energy Advertising Mode". In: *IEEE Access* 6 (2018), pp. 46066–46073. DOI: 10.1109/ACCESS.2018.2866323.

[Gha+16] Mohammed Ghazal, Samr Ali, Marah Al Halabi, Nada Ali, and Yasmina Al Khalil. "Smart Mobile-Based Emergency Management and Notification System". In: *2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*. Aug. 2016, pp. 282–287. DOI: 10.1109/W-FiCloud.2016.64.

[GPN15] J. Ramon Gil-Garcia, Theresa Pardo, and Taewoo Nam. "What Makes a City Smart? Identifying Core Components and Proposing an Integrative and Comprehensive Conceptualization". In: *Information Polity* 20 (July 2015), pp. 61–87. DOI: 10.3233/IP-150354.

[God03] David R. Godschalk. "Urban Hazard Mitigation: Creating Resilient Cities". In: *Natural Hazards Review* 4.3 (Aug. 2003), pp. 136–143. DOI: 10.1061/(ASCE)1527-6988(2003)4:3(136).

[Goh+24] Charlene Enhui Goh et al. "Development of a Dental Diet-Tracking Mobile App for Improved Caries-Related Dietary Behaviours: Key Features and Pilot Evaluation of Quality". In: *Digital Health* 10 (2024). DOI: 10.1177/20552076241228433/ASSET/IMAGES/LARGE/10.1177_20552076241228433-FIG2.JPEG.

[Gon+25] Juan José Ramos González, Zhiqiang Liu, Jose Luis Muñoz-Gamarra, Enric Pastor, and Cristina Barrado. "Impact Assessment of the Strategic Planning Performance in Shared U-space Volumes". In: *European Journal of Transport and Infrastructure Research* 25.1 (Jan. 2025), pp. 67–85. DOI: 10.59490/ejtir.2025.25.1.7472.

[Gon+24] Ana González Bermúdez, David Carramiñana, Ana M. Bernardos, Luca Bergesio, and Juan A. Besada. "A Fusion Architecture to Deliver Multipurpose Mobile Health Services". In: *Computers in Biology and Medicine* 173 (May 2024), p. 108344. DOI: 10.1016/j.compbiomed.2024.108344.

[GZ23] Xiaohui Gu and Guoan Zhang. "A Survey on UAV-assisted Wireless Communications: Recent Advances and Future Trends". In: *Computer Communications* 208 (Aug. 2023), pp. 44–78. DOI: 10.1016/j.comcom.2023.05.013.

[Gup+20] Aditya Dinesh Gupta, Prerna Pandey, Andrés Feijóo, Zaher Mundher Yaseen, and Neeraj Dhanraj Bokde. "Smart Water Technology for Efficient Water Resource Management: A Review". In: *Energies* 13.23 (Jan. 2020), p. 6268. DOI: 10.3390/en13236268.

[Gut+17] Michael Guterres, Stanley Jones, Greg Orrell, and Robert Strain. "ADS-B Surveillance System Performance With Small UAS at Low Altitudes". In: *AIAA Information Systems-AIAA Infotech @ Aerospace*. Grapevine, Texas: American Institute of Aeronautics and Astronautics, Jan. 2017. DOI: 10/gm645j.

[Hab+18] Hadi Habibzadeh, Tolga Soyata, Burak Kantarci, Azzedine Boukerche, and Cem Kaptan. "Sensing, Communication and Security Planes: A New Challenge for a Smart City System Design". In: *Computer Networks* 144 (Oct. 2018), pp. 163–200. DOI: 10.1016/j.comnet.2018.08.001.

[UN-18] UN-Habitat. *Guide to the City Resilience Profiling Tool*. 2018.

[Hah91] E.L. Hahne. "Round-Robin Scheduling for Max-Min Fairness in Data Networks". In: *IEEE Journal on Selected Areas in Communications* 9.7 (Sept. 1991), pp. 1024–1039. DOI: 10.1109/49.103550.

[Hai+16] Abdelfatteh Haidine et al. “The Role of Communication Technologies in Building Future Smart Cities”. In: *Smart Cities Technologies*. IntechOpen, Dec. 2016. DOI: 10.5772/64732.

[Hal00] Robert E Hall. “The Vision of A Smart City”. In: *Proceedings of 2nd International Life Extension Technology Workshop*. Paris, France, 2000.

[Har02] Hiroshi Harada. *Simulation and Software Radio for Mobile Communications*. The Artech House Universal Personal Communications Series. Boston: Artech House, 2002.

[Har+10] C. Harrison et al. “Foundations for Smarter Cities”. In: *IBM Journal of Research and Development* 54.4 (July 2010), pp. 1–16. DOI: 10.1147/JRD.2010.2048257.

[Has+16] Ibrahim Abaker Targio Hashem et al. “The Role of Big Data in Smart City”. In: *International Journal of Information Management* 36.5 (Oct. 2016), pp. 748–758. DOI: 10.1016/j.ijinfomgt.2016.05.002.

[HEN20] HENDSOLDT. *SPEXER 2000 3D Radar*. https://dam.hensoldt.net/m/59f155d472802b78/original/SPEXER-2000-3D-Radar-Datasheet-English.pdf. (accessed 2025-05-11). 2020.

[Hen+23] S. Hengy, E. Bavu, A. Matwyschuk, P. Naz, and B Fischer. “DEEPLOMATICS Project : Multimodal Localization and Identification of Small UAVs”. In: *Distributed Multi-Spectral and Multistatics Sensing STO-MP-SET-312*. 2023.

[Hen+18] Aicha Idriss Hentati, Lobna Krichen, Mohamed Fourati, and Lamia Chaari Fourati. “Simulation Tools, Environments and Frameworks for UAV Systems Performance Analysis”. In: *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*. June 2018, pp. 1495–1500. DOI: 10.1109/IWCMC.2018.8450505.

[Her+17] Constantinos Heracleous, Panayiotis Kolios, Christos G. Panayiotou, Georgios Ellinas, and Marios M. Polycarpou. “Hybrid Systems Modeling for Critical Infrastructures Interdependency Analysis”. In: *Reliability Engineering & System Safety* 165 (Sept. 2017), pp. 89–101. DOI: 10.1016/j.ress.2017.03.028.

[HGH20] HGH Infrared. *SPYNEL IR Camera: A Proven Solution for Drone Swarms/Small UAVs Detection & Tracking*. 2020.

[Hig19] High-Level Expert Group on AI. *Ethics Guidelines for Trustworthy AI | Shaping Europe’s Digital Future*. https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai. (accessed 2024-08-01). Apr. 2019.

[HE16] Jacco Hoekstra and Joost Ellerbroek. “BlueSky ATC Simulator Project: An Open Data and Open Source Approach”. In: June 2016.

[Hom+16] Jeffrey Homola, Thomas Prevot, Joey Mercer, Nancy Bienert, and Conrad Gabriel. “UAS Traffic Management (UTM) Simulation Capabilities and Laboratory Environment”. In: *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*. Sept. 2016, pp. 1–7. DOI: 10.1109/DASC.2016.7778078.

[Hsu+24] Jia-Hao Hsu, Chung-Hsien Wu, Esther Ching-Lan Lin, and Po-See Chen. “Mood-Sensing: A Smartphone App for Digital Phenotyping and Assessment of Bipolar Disorder”. In: *Psychiatry Research* 334 (2024), p. 115790. DOI: 10.1016/J.PSYCHRES.2024.115790.

[HS18] James Humann and Eric Spero. "Modeling and Simulation of Multi-UAV, Multi-Operator Surveillance Systems". In: *2018 Annual IEEE International Systems Conference (SysCon)*. Apr. 2018, pp. 1–8. DOI: 10.1109/SYSCON.2018.8369546.

[HK20] Elizabeth Hunter and John Kelleher. "A Framework for Validating and Testing Agent-based Models: A Case Study from Infectious Diseases Modelling." In: *Conference papers* (Jan. 2020). DOI: 10.21427/2xjb-cq79.

[Ins+12] Institute for the Protection and Security of the Citizen (Joint Research Centre), Roberto Filippini, Giorgios Giannopoulos, and Muriel Schimmer. *Risk Assessment Methodologies for Critical Infrastructure Protection.Part I, a State of the Art*. LU: Publications Office of the European Union, 2012.

[Iri+17] Sarah J Iribarren, Kenrick Cato, Louise Falzon, and Patricia W. Stone. "What Is the Economic Evidence for mHealth? A Systematic Review of Economic Evaluations of mHealth Solutions". In: *PloS one* 12.2 (2017), e0170581. DOI: 10.1371/journal.pone.0170581.

[ITU19] ITU. *P.525 : Cálculo de La Atenuación En El Espacio Libre*. 2019.

[ITU21] ITU. *P.372 : Ruido Radioeléctrico*. 2021.

[JUP25] Christian Janke, Maarten Ujut de Haag, and Eugene Pik. "Development of Global Regulations for Uncrewed Aircraft Systems - Europe and Beyond". In: *Integrated Communication Navigation and Surveillance Conference*. Brussels, Belgium, Apr. 2025.

[JAR24] JARUS. *JARUS Guidelines on Specifc Operations Risk ASsessment*. 2024.

[Jav+22] Mohd Javaid, Abid Haleem, Ravi Pratap Singh, Rajiv Suman, and Shanay Rab. "Significance of Machine Learning in Healthcare: Features, Pillars and Applications". In: *International Journal of Intelligent Networks* 3 (2022), pp. 58–73. DOI: 10.1016/J.IJIN.2022.05.002.

[Jen01] Nicholas R. Jennings. "An Agent-Based Approach for Building Complex Software Systems". In: *Communications of the ACM* 44.4 (Apr. 2001), pp. 35–41. DOI: 10.1145/367211.367250.

[Jep25] Jeppesen. *Total Airspace and Airport Modeler*. https://ww2.jeppesen.com/airspace-solutions/total-airspace-and-airport-modeler/. (accessed 2025-05-05). 2025.

[Jia+19] Mengda Jia, Ali Komeily, Yueren Wang, and Ravi S. Srinivasan. "Adopting Internet of Things for the Development of Smart Buildings: A Review of Enabling Technologies and Applications". In: *Automation in Construction* 101 (May 2019), pp. 111–126. DOI: 10.1016/j.autcon.2019.01.023.

[JKK18] Ohyun Jo, Yong-Kyu Kim, and Juyeop Kim. "Internet of Things for Smart Railway: Feasibility and Applications". In: *IEEE Internet of Things Journal* 5.2 (Apr. 2018), pp. 482–490. DOI: 10.1109/JIOT.2017.2749401.

[Joh19] Elizabeth E. Joh. "Policing the Smart City". In: *International Journal of Law in Context* 15.2 (June 2019), pp. 177–182. DOI: 10.1017/S1744552319000107.

[JD22] Antoine Joulia and Thomas Dubot. "A Simulation Framework for U-space Tactical Conflict Detection and Resolution Services". In: *AIAA AVIATION 2022 Forum*. Chicago, United States: American Institute of Aeronautics and Astronautics, June 2022. DOI: 10.2514/6.2022-3543.

[Jov+21] Snezhana Jovanoska et al. “Passive Sensor Processing and Data Fusion for Drone Detection”. In: *NATO STO Meeting Proceedings: MSG-SET-183 Specialists' Meeting on Drone Detectability: Modelling the Relevant Signature* (July 2021), p. 16.

[Jul+20] Arthur Juliani et al. *Unity: A General Platform for Intelligent Agents*. May 2020. DOI: 10.48550/arXiv.1809.02627. arXiv: 1809.02627 [cs].

[Kab16] Yasin Kabalci. “A Survey on Smart Metering and Smart Grid Communication”. In: *Renewable and Sustainable Energy Reviews* 57 (May 2016), pp. 302–318. DOI: 10.1016/j.rser.2015.12.114.

[Kam+20] Daouda Kamissoko et al. “Resilience of Interconnected Infrastructures and Systems: The RESIIST Project”. In: *Applied Technologies*. Ed. by Miguel Botto-Tobar, Marcelo Zambrano Vizuete, Pablo Torres-Carrión, Sergio Montes León, Guillermo Pizarro Vásquez, and Benjamin Durakovic. Communications in Computer and Information Science. Cham: Springer International Publishing, 2020, pp. 257–270. DOI: 10.1007/978-3-030-42517-3_20.

[KMC20] Jackie Kazil, David Masad, and Andrew Crooks. “Utilizing Python for Agent-Based Modeling: The Mesa Framework”. In: *Social, Cultural, and Behavioral Modeling*. Ed. by Robert Thomson, Halil Bisgin, Christopher Dancy, Ayaz Hyder, and Muhammad Hussain. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2020, pp. 308–317. DOI: 10.1007/978-3-030-61255-9_30.

[Ken+19] Blen M. Keneni et al. “Evolving Rule-Based Explainable Artificial Intelligence for Unmanned Aerial Vehicles”. In: *IEEE Access* 7 (2019), pp. 17001–17016. DOI: 10.1109/ACCESS.2019.2893141.

[KMW97] William Ogilvy Kermack, A. G. McKendrick, and Gilbert Thomas Walker. “A Contribution to the Mathematical Theory of Epidemics”. In: *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* 115.772 (Jan. 1997), pp. 700–721. DOI: 10.1098/rspa.1927.0118.

[KA20] Mohammad Ayoub Khan and Fahad Algarni. “A Healthcare Monitoring System for the Diagnosis of Heart Disease in the IoMT Cloud Environment Using MSSO-ANFIS”. In: *IEEE access : practical innovations, open solutions* 8 (2020), pp. 122259–122269. DOI: 10.1109/ACCESS.2020.3006424.

[Kha+22] Muhammad Asif Khan, Hamid Menouar, Aisha Eldeeb, Adnan Abu-Dayya, and Flora D. Salim. “On the Detection of Unauthorized Drones—Techniques and Future Perspectives: A Review”. In: *IEEE Sensors Journal* 22.12 (June 2022), pp. 11439–11455. DOI: 10.1109/JSEN.2022.3171293.

[Kim+17] Juhyun Kim, Cheonbok Park, Jinwoo Ahn, Youlim Ko, Junghyun Park, and John C. Gallagher. “Real-Time UAV Sound Detection and Analysis System”. In: *2017 IEEE Sensors Applications Symposium (SAS)*. Mar. 2017, pp. 1–5. DOI: 10.1109/SAS.2017.7894058.

[Kio+21] Athanasios Kiourtis, Argyro Mavrogiorgou, Chrysostomos Symvoulidis, Charalampos Tsigkounis, and Dimosthenis Kyriazis. “Indexing of Cloud Stored Electronic Health Records for Consented Third Party Accessing”. In: *IEEE* (2021). DOI: 10.23919/FRUCT50888.2021.9347651.

[Kle+22] Spyridon Kleftakis, Argyro Mavrogiorgou, Nikolaos Zafeiropoulos, Konstantinos Mavrogiorgos, Athanasios Kiourtis, and Dimosthenis Kyriazis. "A Comparative Study of Monolithic and Microservices Architectures in Machine Learning Scenarios". In: *2022 IEEE International Conference on Computing, ICOCO 2022* (2022), pp. 352–357. DOI: 10.1109/ICOCO56118.2022.10031648.

[KT75] L. Kleinrock and F. Tobagi. "Packet Switching in Radio Channels: Part I - Carrier Sense Multiple-Access Modes and Their Throughput-Delay Characteristics". In: *IEEE Transactions on Communications* 23.12 (Dec. 1975), pp. 1400–1416. DOI: 10.1109/TCOM.1975.1092768.

[Klü08] Franziska Klügl. "A Validation Methodology for Agent-Based Simulations". In: *Proceedings of the 2008 ACM Symposium on Applied Computing*. SAC '08. New York, NY, USA: Association for Computing Machinery, Mar. 2008, pp. 39–43. DOI: 10.1145/1363686.1363696.

[KB12] Franziska Klügl and Ana L. C. Bazzan. "Agent-Based Modeling and Simulation". In: *AI Magazine* 33.3 (Sept. 2012), pp. 29–29. DOI: 10.1609/aimag.v33i3.2425.

[Koc18] L I Kochetkova. "Pipeline Monitoring with Unmanned Aerial Vehicles". In: *Journal of Physics: Conference Series* 1015.4 (May 2018), p. 042021. DOI: 10.1088/1742-6596/1015/4/042021.

[Kon+15] Qingkai Kong, Young-Woo Kwon, Louis Schreierz, Steven Allen, Richard Allen, and Jennifer Strauss. "Smartphone-Based Networks for Earthquake Detection". In: *2015 15th International Conference on Innovations for Community Services (I4CS)*. July 2015, pp. 1–8. DOI: 10.1109/I4CS.2015.7294490.

[KNT13] G.M Koretsky, J.F. Nicoll, and M.S. Taylor. *A Tutorial on Electro-Optical/Infrared (EO/IR) Theory and Systems*. 2013.

[KS21] Wojciech Kozłowski and Kacper Suwar. "Smart City : Definitions, Dimensions, and Initiatives". In: *European Research Studies Journal* XXIV (2021). DOI: 10.35808/ersj/2442.

[Kri21] Moez Krichen. "Anomalies Detection Through Smartphone Sensors: A Review". In: *IEEE Sensors Journal* 21.6 (Mar. 2021), pp. 7207–7217. DOI: 10.1109/JSEN.2021.3051931.

[Lal+21] Yandja Lalle, Mohamed Fourati, Lamia Chaari Fourati, and João Paulo Barraca. "Communication Technologies for Smart Water Grid Applications: Overview, Opportunities, and Research Directions". In: *Computer Networks* 190 (May 2021), p. 107940. DOI: 10.1016/j.comnet.2021.107940.

[Lan+17] Haylee Lane, Mitchell Sarkies, Jennifer Martin, and Terry Haines. "Equity in Healthcare Resource Allocation Decision Making: A Systematic Review". In: *Social Science & Medicine* 175 (Feb. 2017), pp. 11–27. DOI: 10.1016/j.socscimed.2016.12.012.

[LA21] Michael Lee and Travis Atkison. "VANET Applications: Past, Present, and Future". In: *Vehicular Communications* 28 (Apr. 2021), p. 100310. DOI: 10.1016/j.vehcom.2020.100310.

[LMP24] Christopher J. Leet, Robert Morris, and Priyank Pradeep. "Safe, Efficient, and Fair UTM Airspace Management". In: *43rd AIAA/Digital Avionics Systems Conference (DASC)*. San Diego, CA, 2024.

[Lei+21] Lei Lei, Gaoqing Shen, Lijuan Zhang, and Zhilin Li. "Toward Intelligent Cooperation of UAV Swarms: When Machine Learning Meets Digital Twin". In: *IEEE Network* 35.1 (Jan. 2021), pp. 386–392. DOI: 10.1109/MNET.011.2000388.

[LS22] Nathaniel M. Levine and Billie F. Spencer. "Post-Earthquake Building Evaluation Using UAVs: A BIM-Based Digital Twin Framework". In: *Sensors* 22.3 (Jan. 2022), p. 873. DOI: 10.3390/s22030873.

[Liu+25] Ruiqi Liu, Leyi Zhang, Ruyue Yu-Ngok Li, and Marco Di Renzo. "The ITU Vision and Framework for 6G: Scenarios, Capabilities, and Enablers". In: *IEEE Vehicular Technology Magazine* 20.2 (June 2025), pp. 114–122. DOI: 10.1109/MVT.2025.3532887.

[Liu+23a] Yanchao Liu, Zhenyu Zhou, Waseem Naqvi, and Jun Chen. "Strategic Deconfliction of Unmanned Aircraft Based on Hexagonal Tessellation and Integer Programming". In: *Journal of Guidance, Control, and Dynamics* 46.12 (2023), pp. 2362–2372. DOI: 10.2514/1.G007459.

[Liu+21] Yanming Liu, Kai Liu, Jinglin Han, Lipeng Zhu, Zhenyu Xiao, and Xiang-Gen Xia. "Resource Allocation and 3-D Placement for UAV-Enabled Energy-Efficient IoT Communications". In: *IEEE Internet of Things Journal* 8.3 (Feb. 2021), pp. 1322–1333. DOI: 10.1109/JIOT.2020.3003717.

[Liu+23b] Yu Liu, Huaiyan Jiang, Yumei Qi, and Jinsheng Yang. "M-Health of Nutrition: Improving Nutrition Services with Smartphone and Machine Learning". In: *Mobile Information Systems* 2023 (2023). DOI: 10.1155/2023/3979020.

[Liu+20] Yuxuan Liu, Hu Liu, Yongliang Tian, and Cong Sun. "Reinforcement Learning Based Two-Level Control Framework of UAV Swarm for Cooperative Persistent Surveillance in an Unknown Urban Area". In: *Aerospace Science and Technology* 98 (Mar. 2020), p. 105671. DOI: 10.1016/j.ast.2019.105671.

[Lob+23] Elton H. Lobo et al. "Design and Development of a Smartphone App for Hypertension Management: An Intervention Mapping Approach". In: *Frontiers in Public Health* 11 (2023). DOI: 10.3389/FPUBH.2023.1092755.

[LP21] Michal Lom and Ondrej Pribyl. "Smart City Model Based on Systems Theory". In: *International Journal of Information Management* 56 (Feb. 2021), p. 102092. DOI: 10/gkct9p.

[LL17] Scott M Lundberg and Su-In Lee. "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017.

[LMC12] José Rafael Luque, María José Morón, and Eduardo Casilari. "Analytical and Empirical Evaluation of the Impact of Gaussian Noise on the Modulations Employed by Bluetooth Enhanced Data Rates". In: *EURASIP Journal on Wireless Communications and Networking* 2012.1 (Dec. 2012), p. 94. DOI: 10.1186/1687-1499-2012-94.

[LZZ18] Jiangbin Lyu, Yong Zeng, and Rui Zhang. "UAV-Aided Offloading for Cellular Hotspot". In: *IEEE Transactions on Wireless Communications* 17.6 (June 2018), pp. 3988–4001. DOI: 10.1109/TWC.2018.2818734.

[Ma+19] Meiyi Ma, Sarah M. Preum, Mohsin Y. Ahmed, William Tärneberg, Abdeltawab Hendawi, and John A. Stankovic. "Data Sets, Modeling, and Decision Making

in Smart Cities: A Survey". In: *ACM Trans. Cyber-Phys. Syst.* 4.2 (Nov. 2019), 14:1–14:28. DOI: 10.1145/3355283.

[MBJ19] Aakif Mairaj, Asif I. Baba, and Ahmad Y. Javaid. "Application Specific Drone Simulators: Recent Advances and Challenges". In: *Simulation Modelling Practice and Theory* 94 (July 2019), pp. 100–117. DOI: 10.1016/j.simpat.2019.01.004.

[Mal19] Mateusz Malanowski. *Signal Processing for Passive Bistatic Radar*. Artech House Radar Library. Norwood: Artech house, 2019.

[MJ18] Guido Manfredi and Yannick Jestin. "An Introduction to Fast Time Simulations for RPAS Collision Avoidance System Evaluation". In: *8th International Conference for Research in Air Transportation*. Barcelona, Spain, 2018.

[Mar+17] Emerson Alberto Marconato et al. *AVENS - A Novel Flying Ad Hoc Network Simulator with Automatic Code Generation for Unmanned Aircraft System*. Jan. 2017. HDL: 10125/41924.

[McG+13] John W. McGillicuddy et al. "Mobile Health Medication Adherence and Blood Pressure Control in Renal Transplant Recipients: A Proof-of-Concept Randomized Controlled Trial". In: *JMIR Research Protocols* 2.2 (2013). DOI: 10.2196/resprot.2633.

[MNS16] Sara Meerow, Joshua P. Newell, and Melissa Stults. "Defining Urban Resilience: A Review". In: *Landscape and Urban Planning* 147 (Mar. 2016), pp. 38–49. DOI: 10.1016/j.landurbplan.2015.11.011.

[Meh+15] Garret Mehl et al. "Harnessing mHealth in Low-Resource Settings to Overcome Health System Constraints and Achieve Universal Access to Health". In: *Behavioral Healthcare and Technology: Using Science-Based Innovations to Transform Practice*. 2015, p. 239.

[Mei09] Lorenz Meier. *MAVlink Protocol*. 2009.

[MBB21] Rico Merkert, Matthew J. Beck, and James Bushell. "Will It Fly? Adoption of the Road Pricing Framework to Manage Drone Use of Airspace". In: *Transportation Research Part A: Policy and Practice* 150 (Aug. 2021), pp. 156–170. DOI: 10.1016/j.tra.2021.06.001.

[MG15] Arvind Merwaday and Ismail Guvenc. "UAV Assisted Heterogeneous Networks for Public Safety Communications". In: *2015 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*. Mar. 2015, pp. 329–334. DOI: 10.1109/WCNCW.2015.7122576.

[Mil+19] Jose A. Millan-Romera, José Joaquín Acevedo, Ángel R. Castaño, Hector Perez-Leon, Carlos Capitán, and Aníbal Ollero. "A UTM Simulator Based on ROS and Gazebo". In: *2019 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED UAS)*. Nov. 2019, pp. 132–141. DOI: 10.1109/REDUAS47371.2019.8999705.

[MKA21] Lambros Mitropoulos, Annie Kortsari, and Georgia Ayfantopoulou. "A Systematic Literature Review of Ride-Sharing Platforms, User Factors and Barriers". In: *European Transport Research Review* 13.1 (Dec. 2021), p. 61. DOI: 10.1186/s12544-021-00522-1.

[Moh+20] Nader Mohamed, Jameela Al-Jaroodi, Imad Jawhar, Ahmed Idries, and Farhan Mohammed. "Unmanned Aerial Vehicles Applications in Future Smart Cities".

In: *Technological Forecasting and Social Change* 153 (Apr. 2020), p. 119293. DOI: 10.1016/j.techfore.2018.05.004.

[Moh+23] Kamel K. Mohammed, Eman I. Abd El-Latif, Noha Emad El-Sayad, Ashraf Darwish, and Aboul Ella Hassanien. "Radio Frequency Fingerprint-Based Drone Identification and Classification Using Mel Spectrograms and Pre-Trained YAM-Net Neural". In: *Internet of Things* 23 (Oct. 2023), p. 100879. DOI: 10.1016/j.iot.2023.100879.

[MCK16] Saraju P. Mohanty, Uma Choppali, and Elias Kougianos. "Everything You Wanted to Know about Smart Cities: The Internet of Things Is the Backbone". In: *IEEE Consumer Electronics Magazine* 5.3 (July 2016), pp. 60–70. DOI: 10.1109/MCE.2016.2556879.

[MD17] Andras Molnar and Zsolt Domozi. "Tracking Production Volumes of Open-Pit Mines with Photogrammetry". In: *2017 European Modelling Symposium (EMS)*. Manchester: IEEE, Nov. 2017, pp. 100–106. DOI: 10.1109/EMS.2017.28.

[Mor+21] Marta Mora-Traverso et al. "An M-Health Telerehabilitation and Health Education Program on Physical Performance in Patients with Hip Fracture and Their Family Caregivers: Study Protocol for the ActiveHip+ Randomized Controlled Trial". In: *Research in Nursery & Health* (2021). DOI: 10.1002/nur.22218.

[MS18] Dan Mototolea and Cornelis Stolk. "Detection and Localization of Small Drones Using Commercial Off-the-Shelf FPGA Based Software Defined Radio Systems". In: *2018 International Conference on Communications (COMM)*. June 2018, pp. 465–470. DOI: 10.1109/ICComm.2018.8484827.

[Al-+19] Amjed Al-Mousa, Belal H Sababha, Nailah Al-Madi, Amro Barghouthi, and Remah Younisse. "UTSim: A Framework and Simulator for UAV Air Traffic Integration, Control, and Communication". In: *International Journal of Advanced Robotic Systems* 16.5 (Sept. 2019), p. 1729881419870937. DOI: 10.1177/1729881419870937.

[MLH21] Mohammad Mozaffari, Xingqin Lin, and Stephen Hayes. "Toward 6G with Connected Sky: UAVs and Beyond". In: *IEEE Communications Magazine* 59.12 (Dec. 2021), pp. 74–80. DOI: 10.1109/MCOM.005.2100142.

[MHR20] Sharon Mozgai, Arno Hartholt, and Albert Rizzo. "The Passive Sensing Agent: A Multimodal Adaptive mHealth Application". In: *2020 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops 2020* (2020), pp. 1–3. DOI: 10.1109/PERCOMWORKSHOPS48775.2020.9156177.

[Muj+21] Omkar Mujumdar, Haluk Celebi, Ismail Guvenc, Mihail Sichitiu, Sunghyun Hwang, and Kyu-Min Kang. "Use of LoRa for UAV Remote ID with Multi-User Interference and Different Spreading Factors". In: *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*. Apr. 2021, pp. 1–7. DOI: 10/gm642q.

[Mun57] James Munkres. "Algorithms for the Assignment and Transportation Problems". In: *Journal of the Society for Industrial and Applied Mathematics* 5.1 (Mar. 1957), pp. 32–38. DOI: 10.1137/0105003.

[Muñ+23] Jose Luis Muñoz Gamarra, Juan José Ramos González, Zhiqiang Liu, and Agustí Sobejano Ventayol. "Impact of USSPs performance in shared U-space volumes". In: *SESAR Innovation Days*. SESAR Joint Undertaking, 2023.

[MMR18] Thomas Münzberg, Tim Müller, and Wolfgang Raskob. "A Future-Oriented Agent-Based Simulation to Improve Urban Critical Infrastructure Resilience". In: *Urban Disaster Resilience and Security: Addressing Risks in Societies*. Ed. by Alexander Fekete and Frank Fiedrich. The Urban Book Series. Cham: Springer International Publishing, 2018, pp. 261–284. DOI: 10.1007/978-3-319-68606-6_16.

[Mur+16] Deepak Murugan, Akanksha Garg, Tasneem Ahmed, and Dharmendra Singh. "Fusion of Drone and Satellite Data for Precision Agriculture Monitoring". In: *2016 11th International Conference on Industrial and Information Systems (ICIIS)*. Dec. 2016, pp. 910–914. DOI: 10.1109/ICIINFS.2016.8263068.

[MS21] Menaka Muthuppalaniappan LLB and Kerrie Stevenson. "Healthcare Cyber-Attacks and the COVID-19 Pandemic: An Urgent Threat to Global Health". In: *International Journal for Quality in Health Care* 33.1 (Jan. 2021), mzaa117. DOI: 10.1093/intqhc/mzaa117.

[NO20] Max Nathan and Henry Overman. "Will Coronavirus Cause a Big City Exodus?" In: *Environment and Planning B: Urban Analytics and City Science* 47.9 (Nov. 2020), pp. 1537–1542. DOI: 10.1177/2399808320971910.

[Nat23] National Protective Security Authority. *Critical National Infrastructure*. 2023.

[NRG20] Julio Navío-Marco, Beatriz Rodrigo-Moya, and Paolo Gerli. "The Rising Importance of the "Smart Territory" Concept: Definition and Implications". In: *Land Use Policy* 99 (Dec. 2020), p. 105003. DOI: 10.1016/j.landusepol.2020.105003.

[Ola+23] Damilola Oladimeji, Khushi Gupta, Nuri Alperen Kose, Kubra Gundogan, Linqiang Ge, and Fan Liang. "Smart Transportation: An Overview of Technologies and Applications". In: *Sensors* 23.8 (Jan. 2023), p. 3880. DOI: 10.3390/s23083880.

[Oos+21] Atsushi Oosedo, Hiroaki Hattori, Ippei Yasui, and Kenya Harada. "Unmanned Aircraft System Traffic Management (UTM) Simulation of Drone Delivery Models in 2030 Japan". In: *Journal of Robotics and Mechatronics* 33.2 (Apr. 2021), pp. 348–362. DOI: 10.20965/jrm.2021.p0348.

[ope25] opendroneid. *Opendroneid/Receiver-Android*. Open Drone ID. Apr. 2025.

[Opt25] OptiTrack. *Motion Capture Systems*. http://optitrack.com/index.html. (accessed 2025-05-04). 2025.

[Ott+17] Sadeeb Ottenburger, Miimu Airaksinen, Isabel Pinto-Seppä, and Wolfgang Raskob. "Enhancing Urban Resilience via a Real-Time Decision Support System for Smart Cities". In: *2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC)*. June 2017, pp. 836–844. DOI: 10.1109/ICE.2017.8279970.

[OMA20] Fatma Outay, Hanan Abdullah Mengash, and Muhammad Adnan. "Applications of Unmanned Aerial Vehicle (UAV) in Road Safety, Traffic and Highway Infrastructure Management: Recent Advances and Challenges". In: *Transportation Research Part A: Policy and Practice* 141 (Nov. 2020), pp. 116–129. DOI: 10.1016/j.tra.2020.09.018.

[Ouy14] Min Ouyang. "Review on Modeling and Simulation of Interdependent Critical Infrastructure Systems". In: *Reliability Engineering & System Safety* 121 (Jan. 2014), pp. 43–60. DOI: 10.1016/j.ress.2013.06.040.

[Par+17] Jihun Park, Dae Hoe Kim, Young Sook Shin, and Sang-ho Lee. "A Comparison of Convolutional Object Detectors for Real-Time Drone Tracking Using a PTZ Camera". In: *2017 17th International Conference on Control, Automation and Systems (ICCAS)*. Oct. 2017, pp. 696–699. DOI: 10.23919/ICCAS.2017.8204318.

[Par21] Parrot. *Parrot FreeFlight 6.7 Application Update Integrates New Flight Features and Direct Remote Identification Support*. https://www.parrot.com/en/newsroom/parrot-freeflight-6.7.4-update. (accessed 2021-11-10). June 2021.

[Pen+12] Wei Peng, Yubai Li, Huan Li, and Bolong Wen. "A Novel High-sensitivity ADS-B Receiver Based on RF Direct Logarithmic Detecting". In: *2012 International Conference on Computer Application and System Modeling*. Atlantis Press, Aug. 2012, pp. 735–737. DOI: 10.2991/iccasm.2012.186.

[Per+22] José Joaquín Peralta Abadía, Christian Walther, Ammar Osman, and Kay Smarsly. "A Systematic Survey of Internet of Things Frameworks for Smart City Applications". In: *Sustainable Cities and Society* 83 (Aug. 2022), p. 103949. DOI: 10.1016/j.scs.2022.103949.

[Per+13] Daniel Perez, Ivan Maza, Fernando Caballero, David Scarlatti, Enrique Casado, and Anibal Ollero. "A Ground Control Station for a Multi-UAV Surveillance System". In: *Journal of Intelligent & Robotic Systems* 69.1 (Jan. 2013), pp. 119–130. DOI: 10.1007/s10846-012-9759-5.

[Pie+21] Paola Pierleoni et al. "Real-Time Smart Monitoring System for Atrial Fibrillation Pathology". In: *Journal of Ambient Intelligence and Humanized Computing* (2021). DOI: 10.1007/s12652-019-01602-w.

[Rad21] RadioBro Corporation. *UAS - Remote ID - Counter UAS*. https://www.radiobro.com/uas/. (accessed 2021-11-10). 2021.

[RPH19] Soundarya Ramesh, Thomas Pathier, and Jun Han. "SoundUAV: Towards Delivery Drone Authentication via Acoustic Noise Fingerprinting". In: *Proceedings of the 5th Workshop on Micro Aerial Vehicle Networks, Systems, and Applications*. DroNet'19. New York, NY, USA: Association for Computing Machinery, June 2019, pp. 27–32. DOI: 10.1145/3325421.3329768.

[Rap+23a] Daniel Raposo, David Carramiñana, Juan Besada, and Ana Bernardos. "A Realistic UAS Traffic Generation Tool to Evaluate and Optimize U-Space Airspace Capacity". In: *18th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2023)*. Cham: Springer Nature Switzerland, 2023, pp. 24–33. DOI: 10.1007/978-3-031-42536-3_3.

[Rap+23b] Daniel Raposo, David Carramiñana, Juan Besada, and Ana Bernardos. "A Realistic UAS Traffic Generation Tool to Evaluate and Optimize U-Space Airspace Capacity". In: *18th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2023)*. Lecture Notes in Networks and Systems. Cham: Springer Nature Switzerland, 2023, pp. 24–33. DOI: 10.1007/978-3-031-42536-3_3.

[RW25] Wolfgang Raskob and Stefan Wandler. “Agent-Based Modelling to Identify Possible Measures in Case of Critical Infrastructure Disruption”. In: *Proceedings of the ISCRAM 2015 Conference*. Kristiansand, Norway, May 2025, p. 6.

[RH20] Marvin Rausand and Stein Haugen. *Risk Assessment: Theory, Methods, and Applications*. Second edition. Wiley Series in Statistics in Practice. Hoboken, NJ: Wiley, 2020.

[Raz+18] Sohail Razzaq, Costas Xydeas, Michael E. Everett, Anzar Mahmood, and Thamer Alquthami. “Three-Dimensional UAV Routing With Deconfliction”. In: *IEEE Access* 6 (2018), pp. 21536–21551. DOI: 10.1109/ACCESS.2018.2824558.

[Rey+25] Lucas Rey et al. “A Performance Analysis of You Only Look Once Models for Deployment on Constrained Computational Edge Devices in Drone Applications”. In: *Electronics* 14.3 (Jan. 2025), p. 638. DOI: 10.3390/electronics14030638.

[Rez+23] Seyed MHS Rezvani, Maria João Falcão, Dragan Komljenovic, and Nuno Marques de Almeida. “A Systematic Literature Review on Urban Resilience Enabled with Asset and Disaster Risk Management Approaches and GIS-Based Decision Support Tools”. In: *Applied Sciences* 13.4 (Jan. 2023), p. 2223. DOI: 10.3390/app13042223.

[RSG16] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “"Why Should I Trust You?": Explaining the Predictions of Any Classifier”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’16. New York, NY, USA: Association for Computing Machinery, Aug. 2016, pp. 1135–1144. DOI: 10.1145/2939672.2939778.

[REH20] Marta Ribeiro, Joost Ellerbroek, and Jacco Hoekstra. “Review of Conflict Resolution Methods for Manned and Unmanned Aviation”. In: *Aerospace* 7.6 (June 2020), p. 79. DOI: 10.3390/aerospace7060079.

[RMC15] Víctor Rodríguez-Fernández, Héctor D. Menéndez, and David Camacho. “Design and Development of a Lightweight Multi-UAV Simulator”. In: *2015 IEEE 2nd International Conference on Cybernetics (CYBCONF)*. June 2015, pp. 255–260. DOI: 10.1109/CYBConf.2015.7175942.

[Ros+20] V. Rosato et al. “The European Infrastructure Simulation and Analysis Centre (EISAC) Initiative and Its Technological Assets”. In: *2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO)*. Sept. 2020, pp. 1848–1851. DOI: 10.23919/MIPRO48935.2020.9245340.

[Ryz18] Ryze Robotics. *Tello*. https://www.ryzerobotics.com/es/tello. (accessed 2025-05-04). 2018.

[Sac+20] Peter Sachs, Anthony Evans, Maxim Egorov, Robert Hoffman, and Bert Hackney. *Evaluating Fairness in UTM Architecture and Operations*. Feb. 2020.

[Sak+23] Naoki Sakane et al. “The Effect of a mHealth App (KENPO-app) for Specific Health Guidance on Weight Changes in Adults With Obesity and Hypertension: Pilot Randomized Controlled Trial”. In: *JMIR mHealth and uHealth* 11 (2023), e43236. DOI: 10.2196/43236.

[San+24] Rubén San-Segundo, Lucía Angulo, Manuel Gil-Martín, David Carramiñana, and Ana M. Bernardos. “Hybrid Artificial Intelligence Strategies for Drone Navigation”. In: *AI* 5.4 (Dec. 2024), pp. 2104–2126. DOI: 10.3390/ai5040103.

[San+18] Pedro M. Santos et al. "PortoLivingLab: An IoT-Based Sensing Platform for Smart Cities". In: *IEEE Internet of Things Journal* 5.2 (Apr. 2018), pp. 523–532. DOI: 10.1109/JIOT.2018.2791522.

[El-+19] Shaker El-Sappagh, Farman Ali, Abdeltawab Hendawi, Jun Hyeog Jang, and Kyung Sup Kwak. "A Mobile Health Monitoring-and-Treatment System Based on Integration of the SSN Sensor Ontology and the HL7 FHIR Standard". In: *BMC Medical Informatics and Decision Making* (May 2019). DOI: 10.1186/S12911-019-0806-Z/FIGURES/11.

[Sar+21] Fatemeh Sarhaddi et al. "Long-Term Iot-Based Maternal Monitoring: System Design and Evaluation". In: *Sensors* (2021). DOI: 10.3390/s21072281.

[Sch21] Brendan Schulman. *FAA Remote ID: What It Means for You and Your DJI Drone*. https://viewpoints.dji.com/blog/faa-remote-id-what-it-means-for-you-and-your-dji-drone. (accessed 2021-11-10). 2021.

[SN21] Pavel Sedivy and Ondrej Nemec. "Drone RCS Statistical Behaviour". In: *Proceedings of the MSG-SET-183 Specialists' Meeting on "Drone Detectability: Modelling the Relevant Signature", North Atlantic Treaty Organization (NATO), Held Virtually (via WebEx)*. 2021, pp. 4–1.

[SG19] Deepti Sehrawat and Nasib Singh Gill. "Smart Sensors: Analysis of Different Types of IoT Sensors". In: *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*. Apr. 2019, pp. 523–528. DOI: 10.1109/ICOEI.2019.8862778.

[SV22] Azamat Seidakhmetov and Omid Fatahi Valilai. "Drone Based Delivery System: Restrictions and Limitations". In: *Changing Tides: The New Role of Resilience and Sustainability in Logistics and Supply Chain Management – Innovative Approaches for the Shift to a New Era. Proceedings of the Hamburg International Conference of Logistics (HICL), Vol. 33*. Berlin: epubli GmbH, 2022, pp. 351–373. DOI: 10.15480/882.4711.

[Sei+24] Ulzhalgas Seidaliyeva, Lyazzat Ilipbayeva, Kyrmyzy Taissariyeva, Nurzhigit Smailov, and Eric T. Matson. "Advances and Challenges in Drone Detection and Classification Techniques: A State-of-the-Art Review". In: *Sensors* 24.1 (Jan. 2024), p. 125. DOI: 10.3390/s24010125.

[Sen17] Ulysses Sengupta. "Complexity Theory: The Urban Is a Complex Adaptive System". In: Sept. 2017, pp. 249–265. DOI: 10.4324/9781315576282-21.

[SES16] SESAR Joint Undertaking. *European Drones Outlook Study*. Nov. 2016.

[SES17] SESAR Joint Undertaking. *U-Space Blueprint*. 2017.

[SES18] SESAR Joint Undertaking. *European ATM Master Plan Drone Roadmap: Roadmap for the Safe Integration of Drones into All Classes of Airspace*. 2018.

[SES19] SESAR Joint Undertaking. *Initial View on Principles for the U-space Architecture*. July 2019.

[SES20] SESAR Joint Undertaking. *SUPPORTING SAFE AND SECURE DRONE OPERATIONS IN EUROPE: A Preliminary Summary of SESAR U-space Research and Innovation Results (2017–2019)*. 2020.

[SES22] SESAR Joint Undertaking. *Demonstrating the Every Day Benefits of U-space*. 2022.

[SES25] SESAR Joint Undertaking. *European ATM Master Plan: 2025 Edition*. 2025.

[SES21] SESAR Project. *PJ34-W3 AURA "ATM U-SPACE INTERFACE"*. 2021.

[ST16] Roberto Setola and Marianthi Theocharidou. "Modelling Dependencies between Critical Infrastructures". In: *Managing the Complexity of Critical Infrastructures*. Springer, Cham, 2016, pp. 19–41.

[SFD22] Sven Seuken, Paul Friedrich, and Ludwig Dierks. "Market Design for Drone Traffic Management". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 36.11 (June 2022), pp. 12294–12300. DOI: 10.1609/aaai.v36i11.21493.

[SE21] Ahmed M. Shahat Osman and Ahmed Elragal. "Smart Cities and Big Data Analytics: A Data-Driven Decision-Making Use Case". In: *Smart Cities* 4.1 (Mar. 2021), pp. 286–313. DOI: 10.3390/smartcities4010018.

[Sha+19] Hazim Shakhatreh et al. "Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges". In: *IEEE Access* 7 (2019), pp. 48572–48634. DOI: 10.1109/ACCESS.2019.2909530.

[SSG23] Charul Sharma, Sanjay Kumar Sharma, and Dharmendra Gill. "Reassessing Smart City Components: An Overview of the Dynamic Nature of Smart City Concept". In: *IOP Conference Series: Earth and Environmental Science* 1186.1 (May 2023), p. 012017. DOI: 10.1088/1755-1315/1186/1/012017.

[SHI+14] Huaizhou SHI, R. Venkatesha Prasad, Ertan Onur, and I.G.M.M. Niemegeers. "Fairness in Wireless Networks:Issues, Measures and Challenges". In: *IEEE Communications Surveys & Tutorials* 16.1 (2014), pp. 5–24. DOI: 10.1109/SURV.2013.050113.00015.

[Shi+24] Nuo Shi et al. "Feasibility of a Mobile Health App-Based Self-Management Program for Chinese Patients with Breast Cancer Receiving Chemotherapy: A Randomized Controlled Pilot Study". In: *DIGITAL HEALTH* 10 (2024). DOI: 10.1177/20552076241231560.

[SBK21] Rakesh Shrestha, Rojeena Bajracharya, and Shiho Kim. "6G Enabled Unmanned Aerial Vehicle Traffic Management: A Perspective". In: *IEEE Access* 9 (2021), pp. 91119–91136. DOI: 10.1109/ACCESS.2021.3092039.

[SSH21] Niven Junliang Sie, Sutthiphong Srigrarom, and Sunan Huang. "Field Test Validations of Vision-based Multi-camera Multi-drone Tracking and 3D Localizing with Concurrent Camera Pose Estimation". In: *2021 6th International Conference on Control and Robotics Engineering (ICCRE)*. Apr. 2021, pp. 139–144. DOI: 10.1109/ICCRE51898.2021.9435654.

[SKH18] Bhagya Nathali Silva, Murad Khan, and Kijun Han. "Towards Sustainable Smart Cities: A Review of Trends, Architectures, Components, and Open Challenges in Smart Cities". In: *Sustainable Cities and Society* 38 (Apr. 2018), pp. 697–713. DOI: 10.1016/j.scs.2018.01.053.

[Sin+21] Maulshree Singh, Evert Fuenmayor, Eoin P. Hinchy, Yuansong Qiao, Niall Murray, and Declan Devine. "Digital Twin: Origin to Future". In: *Applied System Innovation* 4.2 (June 2021), p. 36. DOI: 10.3390/asi4020036.

[Sko18] Brent Skorup. *Auctioning Airspace*. SSRN Scholarly Paper. Rochester, NY, Nov. 2018. DOI: 10.2139/ssrn.3290842.

[Sob+18] Alvaro Sobrinho, Leandro Dias Da Silva, Angelo Perkusich, Maria Eliete Pinheiro, and Paulo Cunha. "Design and Evaluation of a Mobile Application to Assist the Self-Monitoring of the Chronic Kidney Disease in Developing Countries".

In: *BMC Medical Informatics and Decision Making* 18.1 (2018), pp. 1–14. DOI: 10.1186/s12911-018-0587-9.

[SAD23] Mohsen Soori, Behrooz Arezoo, and Roza Dastres. "Internet of Things for Smart Factories in Industry 4.0, a Review". In: *Internet of Things and Cyber-Physical Systems* 3 (Jan. 2023), pp. 192–204. DOI: 10.1016/j.iotcps.2023.04.006.

[SK19] Stavros Souravlas and Stefanos Katsavounis. "Scheduling Fair Resource Allocation Policies for Cloud Computing through Flow Control". In: *Electronics* 8.11 (Nov. 2019), p. 1348. DOI: 10.3390/electronics8111348.

[Spa11] Spain Law. *Ley 8/2011, de 28 de abril, por la que se establecen medidas para la protección de las infraestructuras críticas.* 2011.

[Sri+21] Sutthiphong Srigrarom, Niven Jun Liang Sie, Huimin Cheng, Kim Hoe Chew, Mengda Lee, and Photchara Ratsamee. "Multi-Camera Multi-drone Detection, Tracking and Localization with Trajectory-based Re-identification". In: *2021 Second International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP)*. Jan. 2021, pp. 1–6. DOI: 10.1109/ICA-SYMP50206.2021.9358454.

[SSC13] Brandon Stark, Brennan Stevenson, and YangQuan Chen. "ADS-B for Small Unmanned Aerial Systems: Case Study and Regulatory Practices". In: *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*. May 2013, pp. 152–159. DOI: 10/gm645t.

[SB09] Alan N. Steinberg and Christopher L. Bowman. "Revisions to the JDL Data Fusion Model". In: *Handbook of Multisensor Data Fusion*. CRC Press, 2009.

[SOR15] Jonathan D. Stevenson, Siu O'Young, and Luc Rolland. "Estimated Levels of Safety for Small Unmanned Aerial Vehicles and Risk Mitigation Strategies". In: *Journal of Unmanned Vehicle Systems* 3.4 (Dec. 2015), pp. 205–221. DOI: 10.1139/juvs-2014-0016.

[Str20] Christian Struwe. *How to Make Your DJI Drone Compliant with the French Remote ID Requirement*. https://enterprise-insights.dji.com/blog/how-to-make-your-dji-drone-compliant-with-french-remote-id. (accessed 2021-11-10). 2020.

[Suk+12] Rafid Sukkar, Elyse Katz, Yanwei Zhang, David Raunig, and Bradley T. Wyman. "Disease Progression Modeling Using Hidden Markov Models". In: *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. Aug. 2012, pp. 2845–2848. DOI: 10.1109/EMBC.2012.6346556.

[SRZ19] Jianing Sun, Katarzyna Radecka, and Zeljko Zilic. "FoodTracker: A Real-time Food Detection Mobile Application by Deep Convolutional Neural Networks". In: *The 16th International Conference on Machine Vision Applications* (2019).

[Swi16] Swiss Foundation for Mine Action (FSD). *Case Study No.11: Simulation – Drones for Search and Rescue in Emergency Response | Capacity4dev*. 2016.

[Sye+19] Liyakathunisa Syed, Saima Jabeen, Manimala S., and Abdullah Alsaeedi. "Smart Healthcare Framework for Ambient Assisted Living Using IoMT and Big Data Analytics Techniques". In: *Future Generation Computer Systems* (Dec. 2019). DOI: 10.1016/J.FUTURE.2019.06.004.

[Tal+21] Marieh Talebkhah, Aduwati Sali, Mohsen Marjani, Meisam Gordan, Shaiful Jahari Hashim, and Fakhrul Zaman Rokhani. "IoT and Big Data Applications

in Smart Cities: Recent Advances, Challenges, and Critical Issues". In: *IEEE Access* 9 (2021), pp. 55465–55484. DOI: 10.1109/ACCESS.2021.3070905.

[TDL23] Jun Tang, Haibin Duan, and Songyang Lao. "Swarm Intelligence Algorithms for Multiple Unmanned Aerial Vehicles Collaboration: A Comprehensive Review". In: *Artificial Intelligence Review* 56.5 (May 2023), pp. 4295–4327. DOI: 10.1007/s10462-022-10281-7.

[TX23] Yiwen Tang and Yan Xu. "Incorporating Optimization in Strategic Conflict Resolution for UAS Traffic Management". In: *IEEE Transactions on Intelligent Transportation Systems* (2023), pp. 1–13. DOI: 10.1109/TITS.2023.3290682.

[TXI22] Yiwen Tang, Yan Xu, and Gokhan Inalhan. "An Integrated Approach for On-Demand Dynamic Capacity Management Service in U-Space". In: *IEEE Transactions on Aerospace and Electronic Systems* 58.5 (Oct. 2022), pp. 4180–4195. DOI: 10.1109/TAES.2022.3159317.

[Tan+25] Zhibo Tang, He Ma, Youmin Qu, and Xingpeng Mao. "UAV Detection with Passive Radar: Algorithms, Applications, and Challenges". In: *Drones* 9.1 (Jan. 2025), p. 76. DOI: 10.3390/drones9010076.

[Tha21] Thales. *ScaleFlyt Remote ID: Identification & Tracking for Safe Drone Operations*. https://www.thalesgroup.com/en/markets/aerospace/drone-solutions/scaleflyt-remote-id-identification-tracking-safe-drone-operations. (accessed 2021-11-04). 2021.

[Tho+20] Praveen Thokala, Peter Dodd, Hassan Baalbaki, Alan Brennan, Simon Dixon, and Kinga Lowrie. "Developing Markov Models From Real-World Data: A Case Study of Heart Failure Modeling Using Administrative Data". In: *Value in Health* 23.6 (June 2020), pp. 743–750. DOI: 10.1016/j.jval.2020.02.012.

[Tia+19] Shuo Tian, Wenbo Yang, Jehane Michael Le Grange, Peng Wang, Wei Huang, and Zhewei Ye. "Smart Healthcare: Making Medical Care More Intelligent". In: *Global Health Journal* 3.3 (Sept. 2019), pp. 62–65. DOI: 10.1016/j.glohj.2019.07.001.

[Tur15a] Turf.js. *randomLineString*. 2015.

[Tur15b] Turf.js. *randomPolygon*. 2015.

[US 13] U.S. department of Homeland Security. *National Infrastructure Protection Plan 2013*. 2013.

[uAv20] uAvionix. *ping20Si*. https://uavionix.com/products/ping20si/. (accessed 2021-10-22). 2020.

[Ugw+24] Paulinus O. Ugwoke, Abubakar Abdulsalam, Chukwuemeka C. Ejiofor, Paulinus O. Ugwoke, Abubakar Abdulsalam, and Chukwuemeka C. Ejiofor. "Internet of Things: The Key Enabler of Smart City Development". In: *Smart Cities - Foundations and Perspectives*. IntechOpen, July 2024. DOI: 10.5772/intechopen.114997.

[Ull+20] Zaib Ullah, Fadi Al-Turjman, Leonardo Mostarda, and Roberto Gagliardi. "Applications of Artificial Intelligence and Machine Learning in Smart Cities". In: *Computer Communications* 154 (Mar. 2020), pp. 313–323. DOI: 10.1016/j.comcom.2020.02.069.

[Un+21] Ka Chun Un et al. “Observational Study on Wearable Biosensors and Machine Learning-Based Remote Monitoring of COVID-19 Patients”. In: *Scientific Reports* (2021). DOI: 10.1038/s41598-021-82771-7.

[US 20] US News & World Report. *Don’t Stand So Close: Singapore Trials Automated Drones to Check*. www.usnews.com/news/technology/articles/2020-08-06/dont-stand-so-close-singapore-trials-automated-drones-to-check. (accessed 2025-04-15). 2020.

[Vil22] Antonio Villarreal. *Causes of UAV Loss*. https://www.dtuav.com/post/causes-of-uav-loss. (accessed 2025-06-22). Mar. 2022.

[Vil+15] Vladimir Villarreal, Ramon Hervas, Jesus Fontecha, and Jose Bravo. “Mobile Monitoring Framework to Design Parameterized and Personalized M-Health Applications According to the Patient’s Diseases”. In: *Journal of Medical Systems* (Oct. 2015). DOI: 10.1007/s10916-015-0324-1.

[Vir+22] Chioma Virginia Anikwe et al. “Mobile and Wearable Sensors for Data-Driven Health Monitoring System: State-of-the-art and Future Prospect”. In: *Expert Systems with Applications* (Sept. 2022). DOI: 10.1016/J.ESWA.2022.117362.

[WLS21] Jian Wang, Yongxin Liu, and Houbing Song. “Counter-Unmanned Aircraft System(s) (C-UAS): State of the Art, Challenges, and Future Trends”. In: *IEEE Aerospace and Electronic Systems Magazine* 36.3 (Mar. 2021), pp. 4–29. DOI: 10.1109/MAES.2020.3015537.

[WDZ19] Zhe Wang, Lingjie Duan, and Rui Zhang. “Adaptive Deployment for UAV-Aided Communication Networks”. In: *IEEE Transactions on Wireless Communications* 18.9 (Sept. 2019), pp. 4531–4543. DOI: 10.1109/TWC.2019.2926279.

[Wel+22] Emily M. Wells, Mariel Boden, Ilana Tseytlin, and Igor Linkov. “Modeling Critical Infrastructure Resilience under Compounding Threats: A Systematic Literature Review”. In: *Progress in Disaster Science* 15 (Oct. 2022), p. 100244. DOI: 10.1016/j.pdisas.2022.100244.

[Wen+24] Zhang Wen, Junjie Zhao, Yan Xu, and Antonios Tsourdos. “A Co-simulation Digital Twin with SUMO and AirSim for Testing Lane-based UTM System Concept”. In: *2024 IEEE Aerospace Conference*. Mar. 2024, pp. 1–11. DOI: 10.1109/AERO58975.2024.10521156.

[Win25] Wing. *Wing Drone Delivery. The Future of Delivery Is Here*. https://wing.com/. (accessed 2025-04-16). 2025.

[Won+20] Chun Ka Wong et al. “Artificial Intelligence Mobile Health Platform for Early Detection of COVID-19 in Quarantine Subjects Using a Wearable Biosensor: Protocol for a Randomised Controlled Trial”. In: *BMJ Open* (2020). DOI: 10.1136/bmjopen-2020-038555.

[Woo09] Michael Wooldridge. *An Introduction to Multiagent Systems*. Wiley, 2009.

[Wor16] World Health Organization. *Global Diffusion of eHealth: Making Universal Health Coverage Achievable. Report of the Third Global Survey on eHealth*. 2016.

[Wro+23] Zofia Wrona et al. “Overview of Software Agent Platforms Available in 2023”. In: *Information* 14.6 (June 2023), p. 348. DOI: 10.3390/info14060348.

[Xu+17] Boyi Xu, Lida Xu, Hongming Cai, Lihong Jiang, Yang Luo, and Yizhi Gu. “The Design of an M-Health Monitoring System Based on a Cloud Computing

Platform". In: *Enterprise Information Systems* 11.1 (Jan. 2017), pp. 17–36. DOI: 10.1080/17517575.2015.1053416.

[XZZ18] Jie Xu, Yong Zeng, and Rui Zhang. "UAV-Enabled Wireless Power Transfer: Trajectory Design and Energy Optimization". In: *IEEE Transactions on Wireless Communications* 17.8 (Aug. 2018), pp. 5092–5106. DOI: 10.1109/TWC.2018.2838134.

[Xue+18] Min Xue, Joseph Rios, Joseph Silva, Zhifan Zhu, and Abraham K. Ishihara. "Fe^3: An Evaluation Tool for Low-Altitude Air Traffic Operations". In: *2018 Aviation Technology, Integration, and Operations Conference*. American Institute of Aeronautics and Astronautics, 2018. DOI: 10.2514/6.2018-3848.

[Yan+22] Chen Yang et al. "Using 5G in Smart Cities: A Systematic Mapping Study". In: *Intelligent Systems with Applications* 14 (May 2022), p. 200065. DOI: 10.1016/j.iswa.2022.200065.

[Yan+21] Fawei Yang, Feng Xu, Francesco Fioranelli, Julien Le Kernec, Shaoqiang Chang, and Teng Long. "Practical Investigation of a MIMO Radar System Capabilities for Small Drones Detection". In: *IET Radar, Sonar & Navigation* 15.7 (2021), pp. 760–774. DOI: 10.1049/rsn2.12082.

[Yaq+17] Ibrar Yaqoob, Ibrahim Abaker Targio Hashem, Yasir Mehmood, Abdullah Gani, Salimah Mokhtar, and Sghaier Guizani. "Enabling Communication Technologies for Smart Cities". In: *IEEE Communications Magazine* 55.1 (Jan. 2017), pp. 112–120. DOI: 10.1109/MCOM.2017.1600232CM.

[Yus+20] C Yuste et al. "IMUP: APLICACIÓN DE NUEVAS TECNOLOGÍAS EN EL SEGUIMIENTO DOMICILIARIO DE PACIENTES NEFROLÓGICOS COVID-19". In: *Congreso Nacional de La Sociedad Española de Nefrología*. Virtual, 2020.

[Zha+20] Hao Zhang, Bin Xin, Li-hua Dou, Jie Chen, and Kaoru Hirota. "A Review of Cooperative Path Planning of an Unmanned Aerial Vehicle Group". In: *Frontiers of Information Technology & Electronic Engineering* 21.12 (Dec. 2020), pp. 1671–1694. DOI: 10.1631/FITEE.2000228.

[ZP11] Pengcheng Zhang and Srinivas Peeta. "A Generalized Modeling Framework to Analyze Interdependencies among Infrastructure Systems". In: *Transportation Research Part B: Methodological* 45.3 (Mar. 2011), pp. 553–579. DOI: 10/bz6d2b.

[Zha+25] Yunke Zhang et al. "MetaCity: Data-driven Sustainable Development of Complex Cities". In: *The Innovation* 6.2 (Feb. 2025), p. 100775. DOI: 10.1016/j.xinn.2024.100775.

[Zha+19] Ziyi Zhao et al. "A Simulation Framework For Fast Design Space Exploration Of Unmanned Air System Traffic Management Policies". In: *2019 Integrated Communications, Navigation and Surveillance Conference (ICNS)*. Apr. 2019, pp. 1–10. DOI: 10.1109/ICNSURV.2019.8735348.

[ZB20] Safiah Zulkifli and Alessio Balleri. "Design and Development of K-Band FMCW Radar for Nano-Drone Detection". In: *2020 IEEE Radar Conference (RadarConf20)*. Sept. 2020, pp. 1–5. DOI: 10.1109/RadarConf2043947.2020.9266538.