COMP4DRONES

DELIVERABLE

D5.1 “Architecture for Communications and Security — First Release”

Project Title	**COMP4DRONES**
Grant Agreement number	826610
Call and topic identifier	H2020-ECSEL-2018
Funding Scheme	Research & Innovation Action (RIA)
Project duration	36 Months [1 October 2019 – 30 September 2022]
Coordinator	Mr. Rodrigo Castiñeira (INDRA)
Website	www.**COMP4DRONES**.eu

ECSEL Joint Undertaking
Electronic Components and Systems for European Leadership

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826610

Document fiche			
Authors:	№	Short name	Partner name
	1	INDRA	INDRA SISTEMAS SA
	2	AIT	AIT AUSTRIAN INSTITUTE OF TECHNOLOGY GMBH
	3	FB	FORSCHUNG BURGENLAND GMBH
	11	SCALIAN	EUROGICIEL
	18	TEKNE	TEKNE SRL
	22	ANYWI	ANYWI TECHNOLOGY BV
	24	TNL	THALES NEDERLAND BV
	28	ACORDE	ACORDE TECHNOLOGIES SA
	31	IKERLAN	IKERLAN S. COOP
	32	CEA	COMMISSARIAT A L’ENERGIE ATOMIQUE ET AUX ENERGIES ALTERNATIVES
	34	AI	AITEK SPA
	44	ROT	RO TECHNOLOGY SRL
	45	MODIS	MODIS CONSULTING SRL
	46	IFAT	INFINEON TECHNOLOGIES AUSTRIA AG
Internal reviewers:	Pavel Zemčík [BUT], Andries Stam [ALM]		
Work Package:	WP5		
Task:	T5.1		
Nature:	R		
Dissemination:	PU		

Document History			
Version	Date	Contributor(s)	Description
V0.1	2020-05-28	TEKNE	D5.1 template
V0.2	2020-05-29	TEKNE (Editor) ACORDE CEA IKERLAN MODIS ROT TNL	First contributions New C4D document template
V0.21	2020-05-16	TEKNE (Editor) ACORDE AI AIT CEA FB IFAT MODIS ROT SCALIAN TEKNE TNL	Second contributions
V0.3	2020-05-17	INDRA IKERLAN	Integrated intermediate version Third contributions
V0.31	2020-05-18	ANYWI	
V0.32	2020-07-03	MODIS ANYWI	
V0.34	2020.07.06	AIT	Fourth contributions

		INDRA MODIS ROT ACORDE CEA IKERLAN TNL	KPI added
V0.35	2020-07-07	AITEK TEKNE	KPI added
V0.36	2020-07-10	IFAT MODIS SCALIAN IKERLAN INDRA CEA	Fourth contributions KPI added/refined
V0.40	2020-07-20	SCALIAN ACORDE TNL AIT	Addressing review comments
V0.42	2020-07-22	ROT IKERLAN ACORDE	Addressing review comments
V0.43	2020-07-23	ACORDE AIT IFAT CEA MODIS ANYWI IKERLAN AI FB	Addressing review comments
V1.0	2020-07-29	TEKNE (Editor)	Candidate release

Keywords:	UAV, drone, communications, communications security, communications architecture, communications components
Abstract (few lines):	This document describes the components of the **COMP4DRONES** communication framework, their functionalities, architecture, and interfaces. It reports the results on month 10 of the project and traces the roadmap for the prosecution.

DISCLAIMER

This document does not represent the opinion of the European Community, and the European Community is not responsible for any use that might be made of its content. This document may contain material, which is the copyright of certain **COMP4DRONES** consortium parties, and may not be reproduced or copied without permission. All **COMP4DRONES** consortium parties have agreed to full publication of this document. The commercial use of any information contained in this document may require a license from the proprietor of that information.

Neither the **COMP4DRONES** consortium as a whole, nor a certain party of the **COMP4DRONES** consortium warrant that the information contained in this document is capable of use, nor that use of the information is free from risk, and does not accept any liability for loss or damage suffered by any person using this information.

ACKNOWLEDGEMENT

This document is a deliverable of **COMP4DRONES** project. This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement Nº 826610

Table of Contents

Table of Figures

Table of Tables

Definitions, Acronyms and Abbreviations

Acronym	Title
API	Application Programming Interface
AUC	Area Under Curve
BVLOS	Beyond Visual Line of Sight
C4D	**COMP4DRONES** (short name of the project)
C4I	Command, Control, Communications, Computers, and Intelligence
CMPD	Drone Mission and Data Processing Center
COTS	Commercial Off-The-Shelf
D-SLAM	Decoupled Simultaneous Localization and Mapping
DDS	Data Distribution Service
DoS	Denial of Service
DTN	Disruption Tolerant Networking
DTW	Dynamic-Time-Warping
FPV	First Person View
FTP	File Transfer Protocol
FPR	False Positive Rate
GCS	Ground Control Station
GDPR	General Data Protection Regulation
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GS	Ground Station
HD	High definition (video)
HIDS	Host-based Intrusion Detection System
HIL	Hardware-In-the-Loop
HW	Hardware
HMAC	Keyed-Hashing for Message Authentication
I2C	Inter-Integrated Circuit (a wide-spread hardware interface type)
IDS	Intrusion Detection System
IoT	Internet of Thing
IPS	Indoor Positioning System
k-NN	k-Nearest Neighbours
KB	Knowledge Base
KET	Key Enabling Technology
KPI	Key Performance Indicator
LoRa	Long Range (LoRa is a spread spectrum modulation technique for LPWAN)
LPWAN	Low Power Wide Area Network
MANET	Mobile Ad-hoc NETwork
ML	Machine Learning
MPTCP	MultiPath TCP
MSE	Mean Squared Error
NB-IoT	Narrowband Internet of Things (NB-IoT is a cellular wireless communication standard for LPWAN)
OSSEC	Open Source HIDS SECurity
PVT	Position Velocity Time
QUIC	Quick UDP Internet Connection
QoS	Quality of Service
RF	Radio Frequency
RMSE	Root Mean Squared Error

ROC	Receiver Operating Characteristic (Curve)
SCAP	Security Content Automation Protocol
SCF	Store-Carry-Forward
SLAM	Simultaneous Localization and Mapping
SPI	Serial Peripheral Interface (a wide-spread hardware interface type)
SCTP	Stream Control Transmission Protocol
SVM	Support Vector Machines
SW	Software
SW-API	Software Application Programming Interface
TDOA	Time Difference Of Arrival
TLS	Transport Layer Security
TRL	Technology Readiness Level
TWR	Two Way Ranging
UAS	Unmanned Aircraft System (UAVs operate as part of an Unmanned Aircraft System which also includes a remote-pilot/ground-station, a link for control and management, and other necessary components.)
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
UTM	UAS Traffic Management
UWB	Ultra-Wideband
VANET	Vehicular Ad-hoc NETwork
VPN	Virtual Private Network

Executive Summary

Drones/UAVs can perform air operations that manned aircrafts struggle with, and their use brings significant economic savings and environmental benefits whilst reducing the risk to human life. However, current drone embedded architectures are organized in loosely coupled monolithic boards, which are composed of processor, memory and communication resources (e.g. flight control, planning and vision boards) and do not support the continuous development of drone applications such as the ever-increasing demand on autonomy. The project **COMP4DRONES** aims to define a set of technological frameworks made of replaceable components according to the functionality or capability required for a mission making the drone adaptable and able to evolve.

This report addresses one of the **COMP4DRONES** framework technological brick: trusted communication. This framework is defined **to ensure robust and efficient communications** in the system (ground station(s) to drone(s)) **even in presence of malicious attackers**. To do so, this document details the set of objectives/functionalities required to achieve "trusted communication" and which define the perimeter of the framework.

From this list of objectives/functionalities, a set of components is defined to specify the role and the key performance indicators that each of these components should fulfil in the framework.

Finally, a number of components' solutions are presented as instantiation of how each component could be implemented.

1 Introduction

1.1 Document overview

This document describes the components for trusted communication that will be developed in work package WP5 “Trusted communication”, their functionalities, architecture and interfaces.

Drawing on standard UAS components and on the deliverable D2.1 “Framework Specification” [4], the document first outlines an architecture context where to locate components for communication and security and their interaction paths (Section 1.2 “Architecture”).

Then, the components are described in the Chapter 2 “U-Space components portfolio” and in the Chapter 3 “System functions components portfolio”. The following elements are defined:

- the component synopsis: requirements, KETs (Key Enabling Technologies), standards, improvement over the state of the art, tasks of the work package WP5 in which the component is developed, use case in which it is experimented;
- the component architecture and technologies;
- the component roadmap for the rest of the project (years 2 and 3);
- how the components will enter in the project outcome evaluation (that is the KPI — Key Performance Indicators);
- the planned TRL (Technology Readiness Level) of the component.

The remaining sections of this chapter, 1.3 “WP5 objectives and Key Enabling Technologies” and 1.4 “From the use cases to the project outcome evaluation”, relate the above listed elements to important aspects of WP5 and of **COMP4DRONES**.

1.2 Architecture

An architecture for communications and security for drone systems does not exist in isolation, but rather is an integral part of the complete UAS. It is therefore prudent that any reference architecture is based on standard UAS architectures. In order to facilitate the intended cooperation between C4D partners, it must also be sufficiently specific, especially with regard to identifying components. At the same time, in an R&D project, overly stringent interaction models between components, set too early in the project may hamper innovation.

With these restrictions in mind, we define a loose architectural canvas (Figure 1.1) in this report D5.1, to be refined and validated by the report D5.2 that will be the final version of the document (to be released in the month 30 of the project).

The diagram in Figure 1.1 is the blueprint that allows us locate D5.1 components and their logical interaction paths, at precisely the required specificity for this project stage.

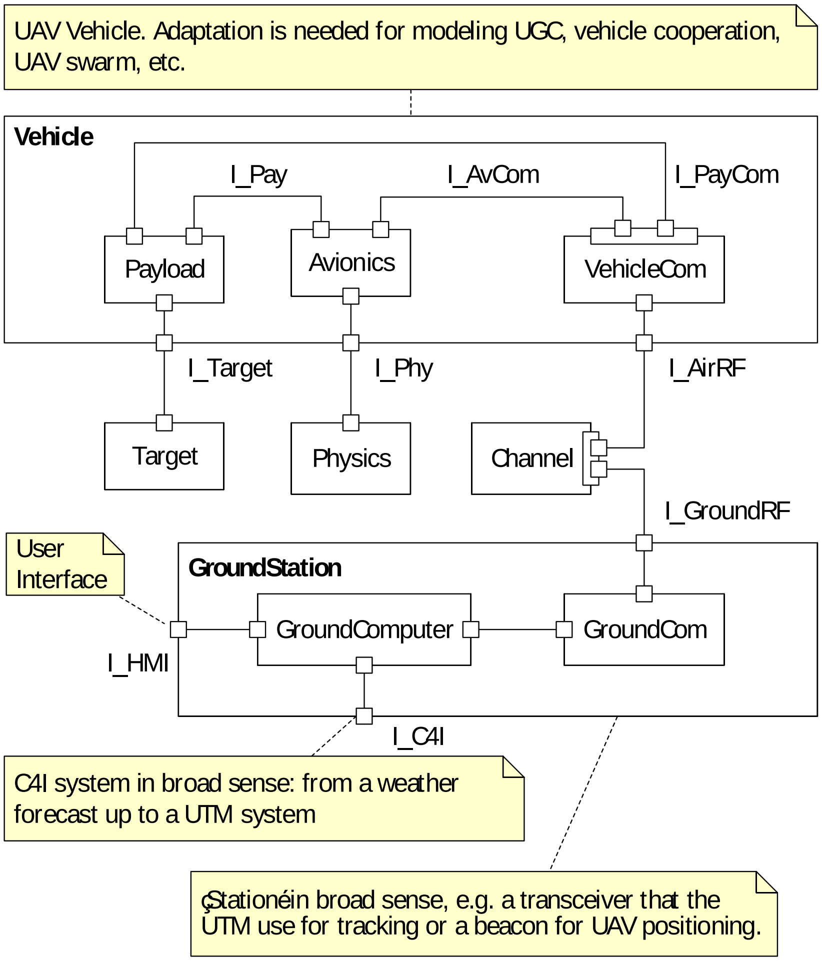

UAV Vehicle. Adaptation is needed for modeling UGC, vehicle cooperation, UAV swarm, etc.
Vehicle
I_Pay
I_AvCom
I_PayCom
Payload
Avionics
VehicleCom
I_Target
I_Phy
I_AirRF
Target
Physics
Channel
I_GroundRF
User Interface
GroundStation
GroundComputer
GroundCom
I_HMI
I_C4I
C4I system in broad sense: from a weather forecast up to a UTM system
çStationéin broad sense, e.g. a transceiver that the UTM use for tracking or a beacon for UAV positioning.

Figure 1.1: D5.1 Architectural context for component interaction paths

A brief description of the blocks in Figure 1.1 follows.

- **Vehicle** — The unmanned vehicle. In **COMP4DRONES** there are activities and use cases (UC3-D1, UC5-D1) that consider Unmanned Ground Vehicles (UGV) other than Unmanned Air Vehicles (UAV). In the context of the work package WP5 “Trusted Communications”, it is not necessary to model the UAV and UGV propulsion systems, the airframe of the former, the body and chassis of the latter.
- **Payload** — UAV sensors and associated recording devices, actuators, onboard systems, goods to be delivered, etc. carried onboard which are used to accomplish a specified mission.
- **I_Pay** — In simple up to mid-complex vehicle the Payload block is controlled, and the mission products are retrieved through the I_Pay interface and the Avionics block, so there may be no I_PayCom interface.
- **Target** — The object of a particular action, to be sensed or acted upon.
- **I_Target** — Generic interface between the payloads of various types and the corresponding targets.
- **Avionics** — The *flight control functions* of the block *Avionic* (another common term is *Autopilot*) are guidance (to follow the given flight profile reacting to disturbances, e.g. wind, and obstacles), navigation (to localize the UAV and detect obstacles), and control (to execute guidance commands and stabilize the UAV).
Avionics includes the sensors (e.g. the GPS) the flight control needs. It is not necessary to model the actuators in the context of the work package WP5.

Mission control functions are required for high levels of automation, such as to recognize the target and act upon it, and they may also be allocated to sub-blocks of the Payload block.

- **Physics** — The block Physics models the environment in which the UAV moves and acts.
- **I_Phy** — Generic interface between the block Physic and the sensors of the block Avionics.
- **GroundStation** — The control of the UAV is achieved through the GroundStation and the communication system (GroundCom and VehicleCom). The GroundStation incorporates the functionality to generate, load and execute the UAV mission and to disseminate useable information data products to various systems at higher hierarchical level through the I_C4I interface. The GroundStation terminal can be located in any platform, (e.g. a van, a vessel, even another air platform).
- **GroundComputer** — The computing elements in the GroundStation.
- **I_HMI** — Human Machine Interface through which the operator interacts with and uses the functions of the GroundStation.
- **I_C4I** — Interface to C4I systems. The term C4I (Command, Control, Communications, Computers, and Intelligence) is used in broad sense. With respect to the UAS (Unmanned Aircraft System, that is UAV and ground station) in Figure 1.1, a C4I system is external, can exercise authority and direction being at higher hierarchical level (e.g. a UAS Traffic Management), can provide communication and computing resources, can provide information (e.g. a wheater forecast system).
- **VehicleCom** and **GroundCom** — The radiocommunication link between the blocks GroundStation and Vehicle. These two blocks include the transceivers and the antennas whose radiofrequency ports are modelled with the interfaces I_AirRF and I_GroundRF. The Figure 1.1 shows a common link for UAV command and control, payload command and control, and mission product retrieval. These functions may be accomplished on separate, independent links.
- **I_GroundRF** and **I_AirRF** — Model elements of the air radiofrequency ports of the communication antennas.
- **Channel** — The propagation medium of radiofrequency waves.

1.3 WP5 objectives and Key Enabling Technologies

WP5 objectives — This deliverable D5.1 is fundamental for the work package WP5, of which it is the first one. Also partners that don’t participate in the task T5.1 contributed with the descriptions of the components they will develop and of the functionalities these must have to reach the objectives. The specific objectives of the task T5.1 are:

- Efficient communication middleware.
- Path management for multi-path communication.
- Relaying/Routing.
- Compression of sensor data (video, images etc.)
- Specific issues of multicopters vs. fixed-wing drones.

The general objectives of other WP5 tasks are:

- T5.2 “Robust Multi-Radio Communications” — Objective: permanent availability of efficient communication medium.
- T5.3 “Security Management” — Objective: security management.
- T5.4 “Reactive Security” — Objective: Reactive security.

The Table 1.1 gives the relation between the objectives of WP5 tasks and the WP5 components.

Table 1.1: Relation between the objectives of WP5 tasks and the WP5 components

Objectives of WP5 tasks										
T5.1 objective: Efficient communication middleware	T5.1 objective: Path management for multi-path communication	T5.1 objective: Relaying/Routing	T5.1 objective: Compression of sensor data (video, images etc.)	T5.1 objective: Specific issues of multicopters vs. fixed-wing drones	T5.2 objective: Permanent availability of efficient communication medium	T5.3 objective: Security management	T5.4 objective: Reactive security	Description summary and identifier of the component(s)		
	✓							Multipath communication		
								WP5-12-ANYWI	WP5-13-ANYWI	WP5-18-CEA
		✓			✓			Message switching when there is no direct link or path		
								WP5-20-TNL		
✓								Network to support agent-based coordination of a UAV fleet		
								WP5-03-SCALIAN		
✓								Generic Autonomous Management Framework for interaction between control mechanism and system specific management components		
								WP5-17-FB		
					✓	✓	✓	LPWAN for identification, tracking, and management/warning messages in emergency situations		
								WP5-05-TEK		
			✓					Video coding and compression		
								WP5-06-AI		
				✓				Suite for the MANTIS fixed-wing UAV		
								WP5-IND-1 (UAV↔GCS HD video for payload and control commands)		
								WP5-IND-2 (payload ↔ fuselage ↔ avionics wiring ports for control and video)		
								WP5-IND-3 (Communication between WP5-IND-1 and the user interface)		
								WP5-IND-4 (User interface for WP5-IND-4)		
								WP5-IND-5 (UAV ↔ UTM communications)		
							✓	Anti-jamming and anti-spoofing features in geo-referencing system		
								WP5-11-ACO		
					✓			Robust and enriched communication for the Indoor Positioning System		
								WP5-19-ACO		
						✓		Hardware for crypto and security-relevant operations		
								WP5-14-IFAT		
						✓		Software crypto		
								WP5-08-ROT	WP5-16-AIT (Primitives and protocols partly based on WP5-14-IFAT)	
							✓	Security by analysis		
								WP5-01-CEA	WP5-02-IKER	WP5-07-MODIS

Key Enabling Technologies — The components’ descriptions given in the following are organized in two groups: *U-Space* (Chapter 2) and *System Functions* (Chapter 3). This division follows the **COMP4DRONES** deliverable D2.1 “Framework Specification” [4] that starts from the definition of the U-Space as described in the *SESAR Joint Undertaking* in the document “European ATM Master Plan − Roadmap for the safe integration of drones into all classes of airspace” [6]:

> *U-Space is a set of new services relying on a high level of digitalization and automation of functions and specific procedures designed to support safe, efficient and secure access to airspace for large numbers of drones. As such, U-Space is an enabling framework designed to facilitate any kind of routine mission, in all classes of airspace and all types of environment, even the most congested, while addressing an appropriate interface with manned aviation and air traffic control.*

Then, in addition to the services “to support safe, efficient and secure access to airspace for large numbers of drones”, [4] identifies the other functionalities and the supporting frameworks needed to have a system of drones full functioning in all missions, refers to them as KET (Key Enabling Technologies), and classifies them in four sets:

a) Drone Capabilities for U-Space;
b) System Functions;
c) Payload Technologies;
d) Tools.

The description of a component specifies the KETs that the component addresses. Being about communications, the work package WP5 deals with components whose functionalities are in the U-Space (Table 1.2) and System Function (Table 1.3) sets. Moreover, some components offer support for payload systems (e.g. cameras).

Table 1.2: U-Space services (from [4], Figure 61, page 94)

U-Space services	
E-Identification	Communication, Navigation and Surveillance
Geofencing	Tracking
Security	Emergency Recovery
Telemetry	Detect and Avoid
Command and control	Vehicle to Vehicle communication
Operations management	Vehicle to Infrastructure communication (V2I)

Table 1.3: System Functions (from [4], Figure 62, page 114)

System Functions	
Intelligent Mission Management	Intelligent Data Handling
Intelligent Outer Loop Control	Swarm formation and cooperation
Take-off and Landing	UAV and UGV
Planning and Scheduling	Network Centric Communications
Contingency Management	Over the Horizon Communications
Deconfliction	Propellant Storage & Feed
Indoor Positioning	Regenerative Fuel Cells
Geofencing	Rechargeable Batteries
Georeferencing	Consumable Fuel Cell
Simultaneous Localization and Mapping	Internal Combustion
Intelligent Vehicle System Monitoring	High Power Density Propulsion

1.4 From the use cases to the project outcome evaluation

Use cases — The requirements of the WP5 components, as well as those of the other work packages, derive both from the industrial and research interests of the CD4 partners and from the use cases of the project. For this reason, at M4 there was an internal preliminary version of the report D1.1 “Specification of Industrial Use Cases” [3] whose official release is at the same time of this document (M10).

Other than for requirements, use cases are important because they contribute to the *verification*—did we build the system right? Moreover, they are essential for the *validation*—did we build the right system? For these two aspects, *the description of a component specifies the use case* in which the component is tested in the field to complete the verification and the validation, *and it specifies the requirements too*. For easy reference, the uses cases defined in [3] are listed in the Table 1.4.

Table 1.4: Use cases and demonstrators of COMP4DRONES (from [3])

Use case		Demonstrator	
ID	Field/Name	ID	Application/Name
UC1	Transport	D1	Road transport: traffic management & monitoring, incident detection
		D2	Port infrastructure: supervision, maritime drone applications
		D3	Monitoring railway: construction works, infrastructure maintenance activities
UC2	Construction	D1	Digitalization of civil infrastructure construction
		D2	Monitoring underground infrastructure construction process
UC3	Logistics	D1	Deployment of an autonomous communication system in hard-to-access areas thanks to a highly automated multi-vehicles system
		D2	Logistics in 5G urban environment: clinical sample delivery in hospital campus
UC4	Surveillance & Inspection	D1	Inspection of offshore turbines structure with hyperspectral technology carried by drones
		D2	Fleet of multi robot navigating and mapping in an unknown environment
UC5	Agriculture	D1	Crop monitoring
		D2	Wine production

ID = Identifier

Each use case addresses a wide *field of applications*, according to which it is named, and includes different *technical demonstrators*. Each demonstrator addresses a specific application according to which it is named.

Project objectives and outcome evaluation — There is another aspect for which use cases are important in a research and development project: the experimentation in the use cases is essential for the *project outcome evaluation*—did we achieve the objectives? **COMP4DRONES** has five project-wide *Objectives* (O1 … O5) that, together with their corresponding *Success Criteria* (SC1.1 … SC5.1) are listed in the Table 1.5.

Success Criteria are the *project wide key indicators*: they will be used to evaluate whether the objectives will have been achieved at the end of the project. Being indicators, they are made of numbers, and the **COMP4DRONES** Technical Proposal suggests the contexts where the measures that produce these numbers can be taken: the *Measurable Outcomes* (MO1.1 … MO5.2) of the Table 1.5.

Table 1.5: CD4 objectives (O), success criteria (SC), and measurable outcomes (MO)

O1			To ease the integration and customization of embedded drone systems.
	SC1.1		Demonstrate a potential gain for design efficiency of embedded drone platforms by reducing their integration, customization and maintenance efforts by 35%.
	SC1.2		Demonstrate a potential reuse of pre-qualified drone application components, leading to an effort reduction of 35% for system-level assurance activities.
		MO1.1	A reference architecture of a modular embedded drone platform, as a virtual entity that embodies a common set of APIs to access the hardware and network resources, flexible scheduling, reconfiguration and resource protection mechanisms.
		MO1.2	A method for incremental assurance and predictability that facilitates the integration and customization of drone modules with minimum impact on other modules and the overall timing and safety determinism.
		MO1.3	A repository of generic hardware-independent application components that will be easily parameterized and pluggable on embedded platforms implementing the proposed reference architecture.
O2			To enable drones to Integrate intelligent perception for safe control loops.
	SC2.1		Demonstrate a potential raise of technology innovation led by increasing autonomy by 40% of dull, dirty, dangerous and difficult tasks with an acceptable level of safety.
	SC2.2		Demonstrate a potential raise of artificial intelligence technology used in drones which will lead to an increased level of security of about 35% via automated health-monitoring and data-analytics systems.
		MO2.1	A reference architecture of a modular embedded drone platform, as a virtual entity that embodies a common set of APIs to access the hardware and network resources, flexible scheduling, reconfiguration and resource protection mechanisms.
		MO2.2	Intelligent real-time data analytics algorithms supporting self-adaptation mechanisms for fault-tolerant missions.
		MO2.3	Runtime safety monitoring algorithms to enable dynamic safe-operational modes and safe control transfer to human operators, whenever it is needed.
O3			To ensure the deployment of trusted communications
	SC3.1		Demonstrate a potential raise of technology trustworthiness led by 30% reduction of cybersecurity risks of drone-to-drone and drone-to-ground communications.
		MO3.1	A lightweight communication framework supporting MIMO (multiple input multiple output) wireless communication and middleware.
		MO3.2	Robust-multi-radio communications.
		MO3.3	Mechanisms for secure communications, including reactive security (e.g. anomaly-based intrusion detection).
O4			To minimize the design and verification efforts for complex drone applications (design and verification tools).
	SC4.1		Demonstrate a potential gain for design and assurance efficiency of embedded drone platforms by reducing specification, verification & validation and implementation efforts by 30%.
		MO4.1	Drone system modelling and code generation tools.
		MO4.2	Drone system validation and verification tools.
		MO4.3	Drone system analysis and optimization tools.

Table 1.5: CD4 objectives (O), success criteria (SC), and measurable outcomes (MO)

O5	To ensure sustainable impact and creation of an industry-driven community.		
	SC5.1	Demonstrate a potential sustainable impact in drone industry by increasing the harmonization, scalability and market growth of drone technologies by 30%.	
		MO5.1	Ecosystem of **COMP4DRONES** software components, tools, methods, supported by an infrastructure of repository, change management and ticketing system.
		MO5.2	Community for maintenance, evolution and industrialization of the ecosystem, supported by governance board, rules, policies and quality models.

Being project wide, Success Criteria pertain to the global result and so, following [7], can be taken as *key result indicators* «to reflect the fact that many measures are a summation of more than one team’s input. […] The common characteristics of these measures is that they are the result of many actions carried out by many teams over a period of time, hence the use of the term *result*, and they are good summary measures, hence the term *key*. »

The «many measures [that] are a summation of more than one team’s input» are the KPIs/metrics, which are related to the use cases experimentation of components, subsystems, systems and services.

The description of a component specifies the KPIs/metrics thereof (the Table 1.6. gives an example).

Table 1.6: Template of the KPIs/metrics

№	KPI	#	Metric	MO
1	Availability	1	MTBF	MO3.x
1	Availability	2	MTTR	MO3.y
2	Speed	1	Time to travel from A to B	MO3.z

№ = Component wide KPI number   KPI = KPI description
# = KPI wide metric number   Metric = Metric description
MO = Measurable Outcome (see Table 1.5) that the metric refers to and refines

The communications components in this document address mainly the objective O3 “To ensure the deployment of trusted communications”, but are also related with the objective O1 “To ease the integration and customization of embedded drone systems” in terms of their reusability and of the resources they require (computational, energy, weight, dimensions).

2 U-Space components portfolio

This chapter describes the components that address mainly Key Enabling Technologies of the "Drone Capabilities for U-Space" set (see "Key Enabling Technologies" at page 15).

2.1 Communications for HD video, telemetry, and commands and controls (WP5-IND-1)

The Table 2.1 provides the synopsis of the component WP5-IND-1.

Table 2.1: Component WP5-IND-1

Identifier: WP5-IND-1	**Partner:** INDRA	**Expected TRL:** TRL 6
Name: Avionics – Communications / Radio Links		
License: Proprietary	**Owner:** INDRA	**Contact:** airala@indra.es dlamas@indra.es
Description: This component of the avionics enables the reception, management and forward of HD (High Definition) video from the three types of payloads to the operator's ground station, the telemetry of the UAV to the GCS, and control commands from the GCS to the UAV.		
Satisfied Requirement: IDENTIFIER — DEFINITION • DEM1-P&C-1 — VLOS requirements & regulation requirements, according to "Real Decreto" 1036/17, signed on December 17th 2017". • DEM1-SEC-2 — The drone shall fly following the flight plan created and authorised, as well as the contingency route to recover the RPA in the event of loss link. • DEM1-SEC-3 — The communications between Mantis-GCS systems must be resilience against cyber-attacks. • DEM1-INT-1 — The bandwidth of the air and ground radio links must be sufficient to send the video in HD. • DEM1-SEC-4 — The drone must incorporate a secure communication module for communication and control.		
Key Enabling Technology: KET — CATEGORY • KET: Vehicle to Infrastructure communication (V2I) — CATEGORY: U-Space Capabilities • KET: Network Centric Communications Systems — CATEGORY: System Functions		
Improvement: (HW) Analysing the avionics subsystems involved and designing the wiring route as well as the physical execution of that route on all aircraft available in the project. (SW) Settings, parameterization, configuration and tests that allow to efficiently increase the bandwidth needed to send HD video and telemetry between the air and ground radio links.		
Contributor: Indra	**Task:** T5.1	
• **Use Case:** • UC1 — Transport • **Demonstrator:** • D1 — Road transport: traffic management & monitoring, incident detection		

2.1.1 Architecture context and interfaces

The Figure 2.1 depicts the architecture context of the component WP5-IND-1.

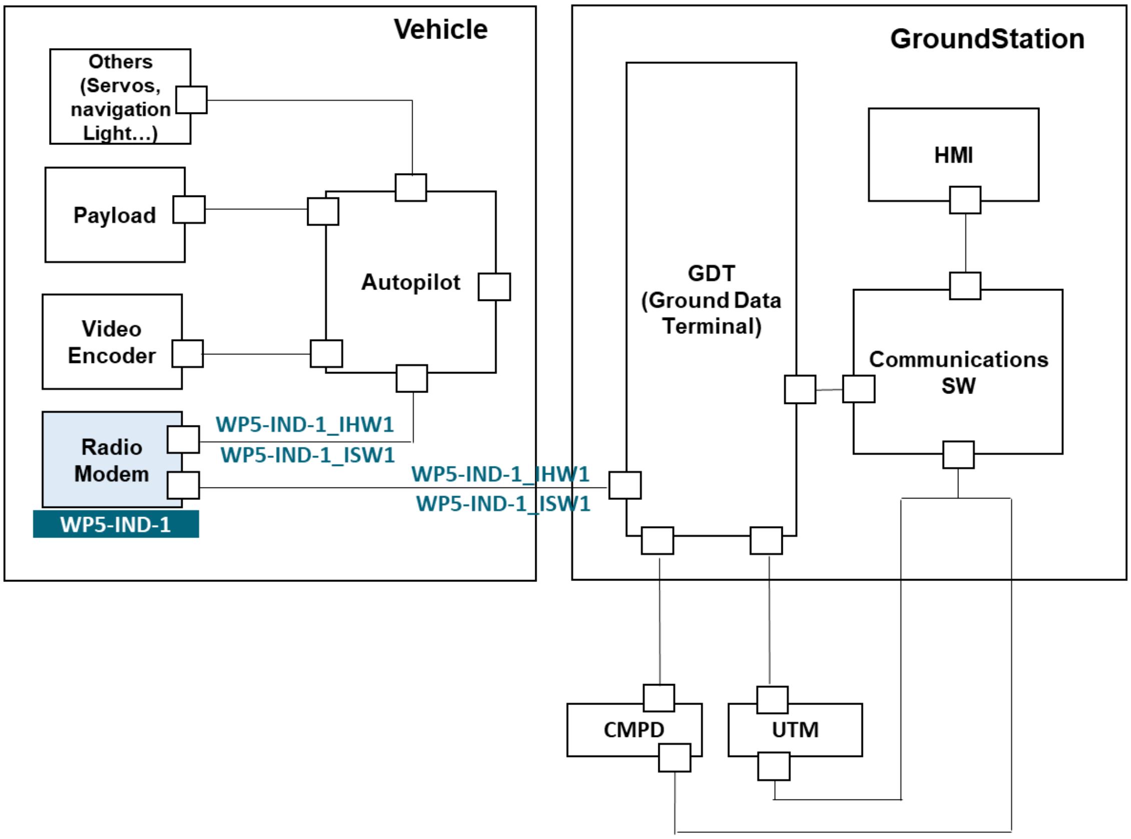


Figure 2.1: Architecture context of the component WP5-IND-1

Any data exchange between the ground station and the aircraft and vice versa, is centralized in the autopilot in which data is transmitted through a single data link. The hardware and software interfaces used are described below:

- WP5-IND-1_IHW1: Internal hardware communication interface between the autopilot and the radio modem boarded of the aircraft (ADT). It is an ethernet interface through which both the data and video sent from the aircraft to the ground station are managed, as well as the Command and Control commands sent from the ground station to the aircraft
- WP5-IND-1_ISW1: Internal communication software interface between the autopilot and the radio modem of the aircraft (ADT). It is a communications interface with TCP/IP protocol through which both the data and video sent from the aircraft to the ground station are managed, as well as the Command and Control commands sent from the ground station to the aircraft
- WP5-IND-1_IHW2: Internal hardware communication interface between the radio modem of the aircraft (ADT) and the one located on the ground station (GDT). Communication is via "Wireless" in the frequency band of 2.304 - 2.39 GHz
- WP5-IND-1_ISW2: Internal communication software interface between the radio modem of the aircraft (ADT) and the one located at the ground station (GDT). The software communications protocol is owned by the manufacturer of the radio modem. It provides the bandwidth and range needed for complex data intensive applications. It is a digital data link that uses Maximal Ratio Combining (MRC), Maximal likelihood (ML) decoding and Low-Density Parity Check (LDPC) to achieve robust RF performance.

2.1.2 Internals and technologies

The software architecture of the MANTIS system is structured in layers—see also the **COMP4DRONES** report D3.1 “Specification of Integrated and Modular Architecture for Drones” [3].

The different software components in avionics of the aircraft are classified, depending on their level of abstraction, into two layers (Figure 2.2).

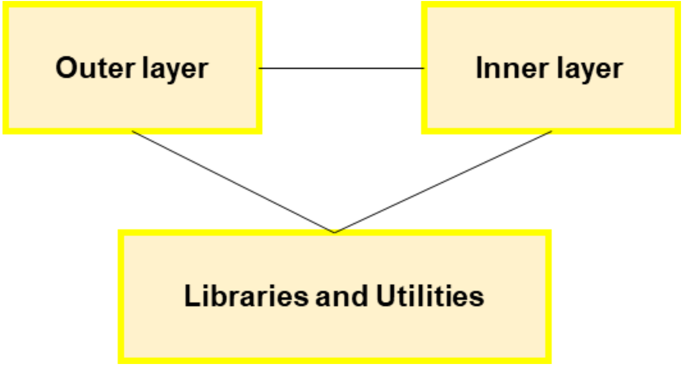


Figure 2.2: The two-layers software architecture of the MANTIS system

Within each layer there are certain basic software packages depending on their functionality (Figure 2.3).

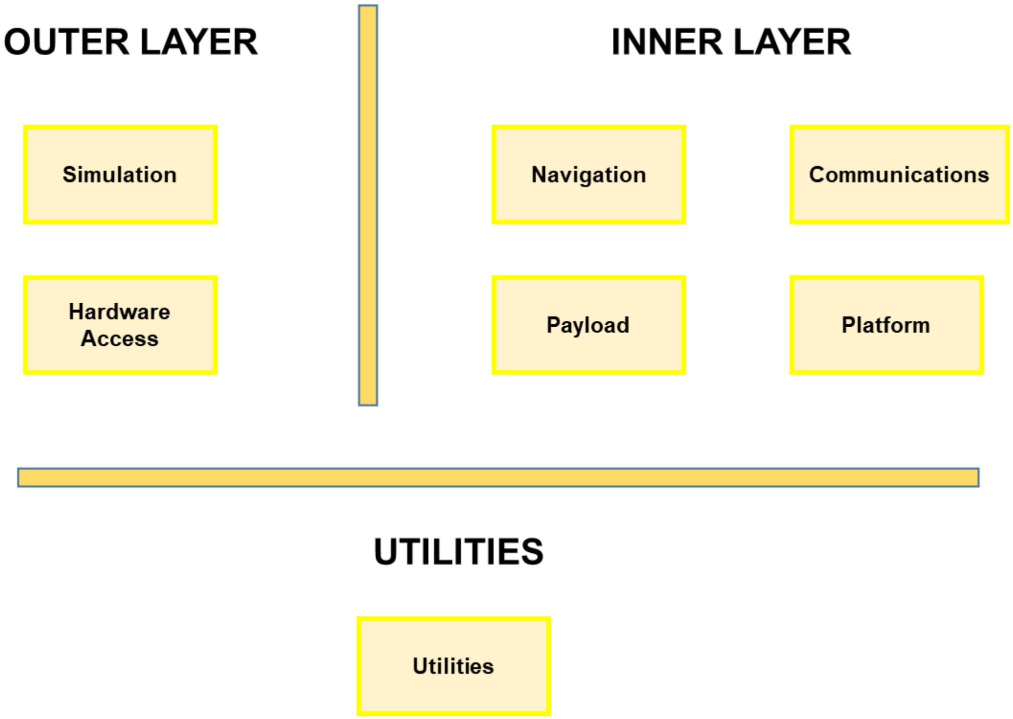


Figure 2.3: Functionalities of software packages

Each of the basic software packages has separate software units, which will be used for the development of the different components.

Specifically, the WP5-IND-1 component should make use of the external layer's "Hardware Access" SW and "Communications" packets in the inner layer:

- Hardware Access SW Package: From this package, the WP5-IND-1 component must use the following SW drives:

- Access commands from the ground station via the radio link (ethernet port control)
- Access to video from payment load (LVDS port control), sending to the video encoder and then to the radio link (ethernet port control)
- Access to navigation equipment (IMU, altimeter and pitot) for sending telemetry data to the ground control station.

- "Communications" SW Package: This package contains SW drives that implement high-level communications messaging with different devices that exchange information with the autopilot. Each component performs its task independently. The SW units to be used to implement the WP5-IND-1 component's task will be those related to the radio link, payment load, video encoder and aircraft navigation equipment.

The video-related portion of the payment payload will be implemented in the WP5-IND-3 component

2.1.3 Component roadmap

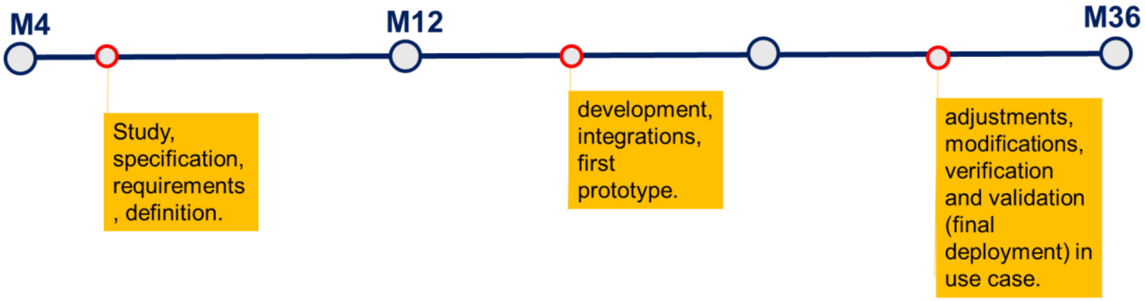

M4
M12
M36
Study, specification, requirements , definition.
development, integrations, first prototype.
adjustments, modifications, verification and validation (final deployment) in use case.

Figure 2.4: Roadmap of the component WP5-IND-1

2.1.3.1 Results at M10

The work carried out during the first period of the project (M4-M10) focused on the definition of the requirements associated to the specific UC1 Demonstrator 1 involving the MANTIS drone, study and analysis of the avionics subsystems involved, specification of the internals and interfaces and definition of the component.

2.1.3.2 Plans for the year 2 and the year 3

In year 2 and year 3, the work planned will focus on the physical and internal design of the component and the physical execution, performing all the settings, parameterization, configuration and tests that allow to efficiently increase the bandwidth needed to send HD video and telemetry between the air and ground radio links.

- Year 2: development, integrations, and deployment of first prototype.
- Year 3: adjustments, modifications, verification and validation (final deployment) in the Use Case 1 Demonstrator 1.

We plan to measure component characteristics, and compute metrics and KPIs according to the Table 2.2, and so to contribute to evaluate the achievement of the project objective O1.

Table 2.2: KPI and metric of the component WP5-IND-1

№	KPI	#	Metric	SC
1	Component reuse	1	The number of basic software packages and reused software units of the architecture defined will be measured	SC1.2

№ = Component wide KPI number
KPI = KPI description
# = KPI wide metric number
Metric = Metric description
SC = Success Criteria (see Table 1.5) to which the metric contributes

2.2 Payload/fuselage/avionics wiring ports (WP5-IND-2)

The Table 2.3 provides the synopsis of the component WP5-IND-2.

Table 2.3: Component WP5-IND-2

Identifier: WP5-IND-2	**Partner:** Indra	**Expected TRL:** TRL6
Name: Communications - Ports		
License: Proprietary	**Owner:** Indra	**Contact:** airala@indra.es dlamas@indra.es
Description: Configuration, Control and Video, Communications and Wiring Ports between payload and fuselage, and between fuselage and avionics.		
Satisfied Requirement: IDENTIFIER — DEFINITION • DEM1-FNC-3 — OEM Camera shall provide HD video (min 720p). • DEM1-FNC-5 — The drone shall provide HD video in real time to the Mission Center over 4G.		
Key Enabling Technology: KET — CATEGORY • KET: Vehicle to Infrastructure communication (V2I) — CATEGORY: U-Space Capabilities • KET: Network Centric Communications Systems — CATEGORY: System Functions		
Improvement: (HW) analysis of design requirements, routing, manufacturing, assembly and tests necessary to adapt the connectors that allow the connection between the different payment loads and the fuselage, both for the sending of video and for the control of the different subsystems of the payment loads. The solution must be valid for all 3 types of payment charges (CP Single Visible HD, CP Single Infrared HD and CP DUAL HD).		
Contributor: Indra	**Task:** T5.1	
Use Case: • UC1 — Transport **Demonstrator:** • D1 — Road transport: traffic management & monitoring, incident detection		

2.2.1 Architecture context and interfaces

The Figure 2.5 depicts the architecture context of the component WP5-IND-2.

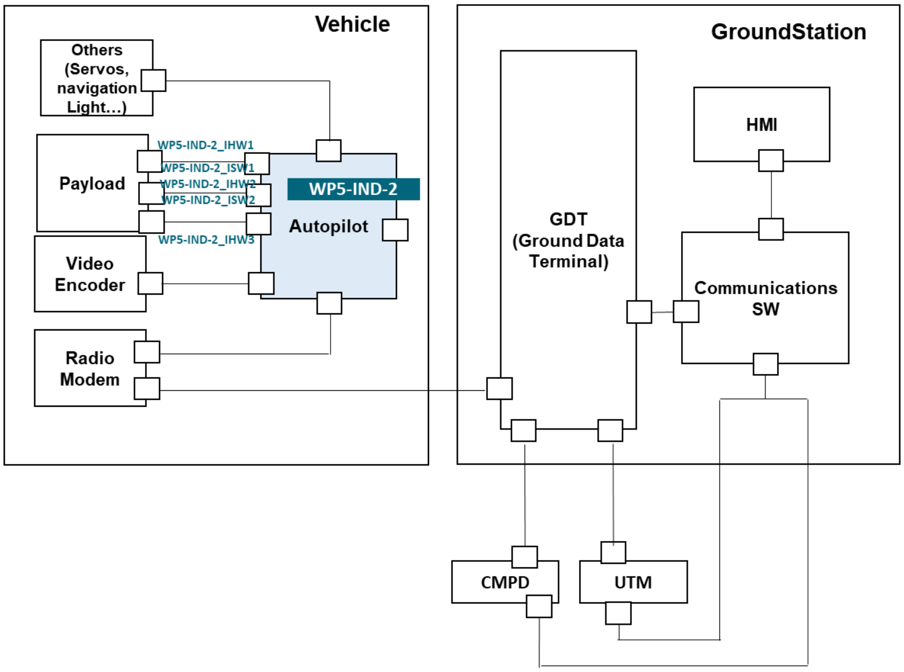


Figure 2.5: Architecture context of the component WP5-IND-2

The various payloads to be developed are physically connected to the aircraft via hardware interfaces (connection pins) available in the fuselage. These hardware interfaces enable software communication between the different payload equipment and the autopilot. The hardware and software interfaces used are described below:

- WP5-IND-2_IHW1: Internal hardware payload power interface. There will be different positive inputs depending on the voltage level required for each of the payment charges (between 5-9 volts)
- WP5-IND-2_IHW2: Internal communication hardware interface between payloads and avionics autopilot. This interface will allow the control of the payload, as well as the reading of data. This interface will be designed to control serial communications ports.
- WP5-IND-2_IHW3: Internal hardware interface of the video to be sent from the payloads to the avionics autopilot. This interface will be designed for the use of high-resolution Y/Pb/Pr 4:2:2 (LVDS) digital communications.
- WP5-IND-2_ISW1: Internal communication software interface between the autopilot and payload allowed by the Control, as well as the reading of data. There will be different software communication protocols depending on the protocol used by each of the optics. Ideally, the goal is that they are at most 2 different protocols, one for visible optics (single and dual) and one for infrared optics (single and dual) even if the optical models used are not the same.
- WP5-IND-2_ISW2: Internal video software interface to be sent from payloads to avionics autopilot. It will be necessary to implement the communication protocol high-level software to manage the video through the DIGITAL interface LVDS.

2.2.2 Internals and technologies

Hardware modifications and software developments will be required on the system.

- Hardware Modifications: The evolution of analog-to-digital video involves expanding the number of pins that the fuselage interconnects with the payload since through those pins the video is sent, needing 2 pins (analog video) to between 6 and 20 pins depending on the communications port. In addition, it will be needed to adapt the avionics to receive the new digital communications through the port (LVDS).
- Software Developments: It will be necessary to develop new software units within the software packages "Communications" and "Hardware Access" of the system software architecture. More specifically, the implementation of a new hardware access SW unit for LVDS ports will be required, as well as the SW unit that treats the messaging of that port at a high level within the communications packet.

In addition, this component will need to implement communications with the different cameras for control and configuration. To this end, the SW unit for accessing serial communications hardware will be used and, as far as possible, reuse the SW communications units with the old analog optics in case of compatibility of communications protocols.

2.2.3 Component roadmap

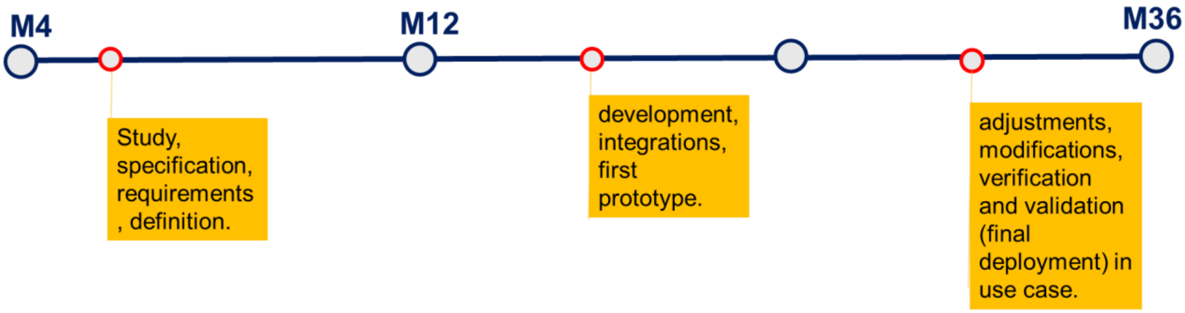

M4
M12
M36
Study, specification, requirements , definition.
development, integrations, first prototype.
adjustments, modifications, verification and validation (final deployment) in use case.

Figure 2.6: Roadmap of the component WP5-IND-2

2.2.3.1 Results at M10

The work carried out during the first period of the project (M4-M10) focused on the definition of the requirements associated to the specific UC1 Demonstrator 1 involving the MANTIS drone, study and analysis of design, routing, manufacturing, assembly and definition of the component.

2.2.3.2 Plans for the year 2 and the year 3

In year 2 and year 3, the work planned will focus on the tests necessary to adapt the connectors that allow the connection between the different payment loads and the fuselage, both for the sending of video and for the control of the different subsystems of the payment loads.

- Year 2: development, integrations, and deployment of first prototype
- Year 3: adjustments, modifications, verification and validation (final deployment) in the Use Case 1 Demonstrator 1.

We plan to measure component characteristics, and compute metrics and KPIs according to the Table 2.4, and so to contribute to evaluate the achievement of the project objective O1.

Table 2.4: KPI and metric of the component WP5-IND-2

№	KPI	#	Metric	SC
1	Component reuse	1	It will be measured on the basis of software units required for accessing the HW and the reuse of communications elements of the architecture.	SC1.2

№ = Component wide KPI number KPI = KPI description
# = KPI wide metric number Metric = Metric description
SC = Success Criteria (see Table 1.5) to which the metric contributes

2.3 Ground control station frontend/backend communication component (WP5-IND-3)

The Table 2.5 provides the synopsis of the component WP5-IND-3.

Table 2.5: Component WP5-IND-3

Identifier: WP5-IND-3	**Partner:** Indra	**Expected TRL:** TRL6
Name: Communications - GCS - Autopilot		
License: Proprietary	**Owner:** Indra	**Contact:** airala@indra.es dlamas@indra.es
Description: Communication between the frontend and the backend of the ground control station.		
Satisfied Requirement: IDENTIFIER — DEFINITION • DEM1-SEC-2 — The drone shall fly following the flight plan created and authorized, as well as the contingency route to recover the RPA in the event of loss link. • DEM1-FNC-6 — The drone must autonomously navigate with high position accuracy during landing. • DEM1-FNC-8 — The drone must communicate with the GCS and inform about its landing position.		
Key Enabling Technology: KET — CATEGORY • KET: Vehicle to Infrastructure communication (V2I) — CATEGORY: U-Space Capabilities • KET: Network Centric Communications Systems — CATEGORY: System Functions		
Improvement: (SW) Analysis, design, development and test of tests that enable communication between the frontend and the backend of the control station application so that the actions requested by the user, are sent to the autopilot.		
Contributor: Indra	**Task:** T5.2	
Use Case: • UC1 — Transport **Demonstrator:** • D1 — Road transport: traffic management & monitoring, incident detection		

2.3.1 Architecture context and interfaces

The Figure 2.7 depicts the architecture context of the component WP5-IND-3.

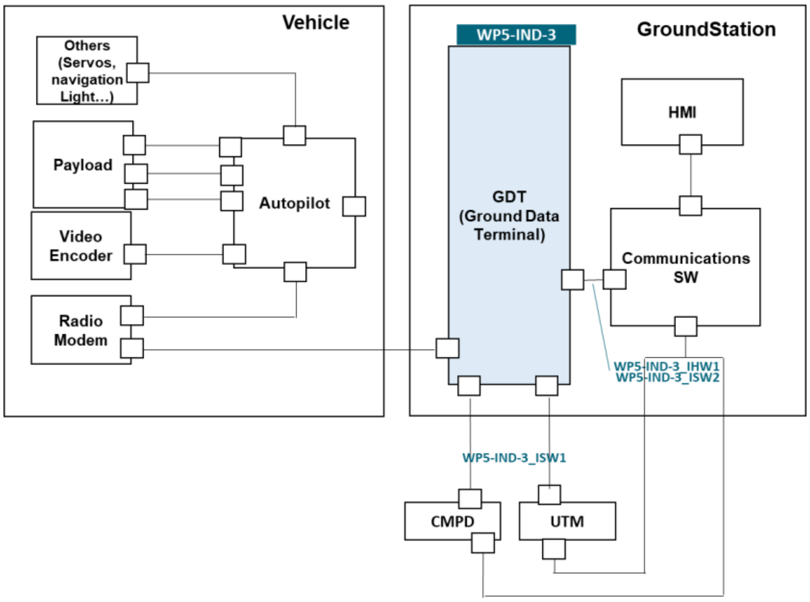


Figure 2.7: Architecture context of the component WP5-IND-3

The communications software (backend) and the Machine Man Interface (frontend) physically reside on the same GCS PC so it does not require hardware interfaces for its intercom, only software interfaces. Hardware interface will be required for communication with the autopilot.

- WP5-IND-3_ISW1: Internal communication software interface that enables the exchange of new data messaging from external systems (CMPD and UTM), between the frontend and the GCS backend. This is a communications interface with UDP protocol.
- WP5-IND-3_IHW1: Internal communication hardware interface between the GCS and the ground station radio modem (GDT). It is an Ethernet interface through which both the data and video received from the aircraft to the ground station are managed, as well as the Command and Control commands sent from the ground station to the aircraft.
- WP5-IND-4_ISW3: Internal communication software interface between the GCS and the ground station radio modem (GDT). It is a communications interface with TCP/IP protocol through which both the data and video sent from the aircraft to the ground station are managed, as well as the Command and Control commands sent from the ground station to the aircraft.

For communication with the autopilot, the interfaces defined in the WP5-IND-.1 component would be used.

2.3.2 Internals and technologies

It will be necessary to implement a new messaging and internal logic control between the frontend and the GCS backend that allows the aircraft operator to interact from the HMI with the CMPD and UTM systems and in turn, communicate with the autopilot and command actions.

More specifically, it will be necessary to implement the following messaging:

- Related to UTM system:
  - Internal messaging between frontend and backend to send flight plan authorization request from the GCS to the UTM system.
  - Internal messaging between frontend and backend to send telemetry from the MANTIS aircraft to the UTM system.
  - Internal messaging between frontend and backend to notify the UTM system that the MANTIS aircraft has started the flight.
  - Internal messaging between frontend and backend to notify the UTM system that the MANTIS aircraft has completed the flight.
- Related to CMPD system:
  - Internal messaging between frontend and backend to receive request for contingent mission from the CMPD system.
  - Internal messaging between frontend and backend to send confirmation of contingent mission flight plan authorization to the CMPD system.
  - Internal messaging between frontend and backend to notify the CMPD system that the MANTIS aircraft is in a position to undertake the mission.
  - Internal messaging between frontend and backend to receive mission start order from the CMPD system.
  - Internal messaging between frontend and backend to send telemetry and video of the MANTIS aircraft to the CMPD system.
  - Internal messaging between frontend and backend to receive an end-of-mission order from the CMPD system.

2.3.3 Component roadmap

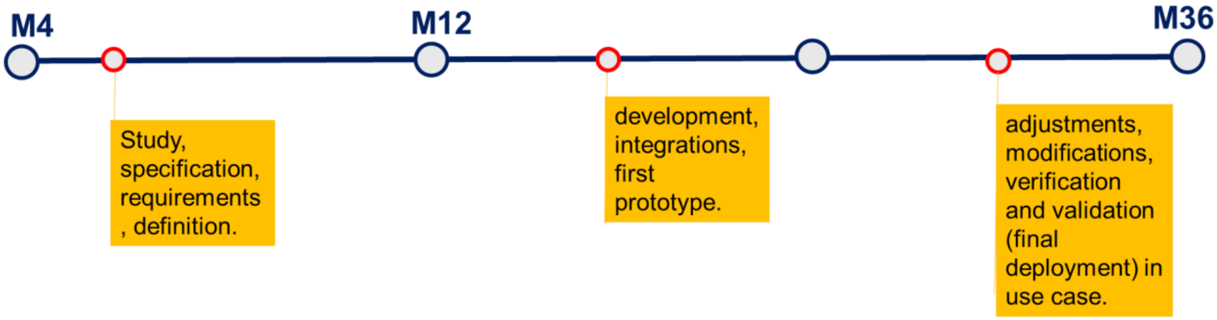

M4
M12
M36
Study, specification, requirements , definition.
development, integrations, first prototype.
adjustments, modifications, verification and validation (final deployment) in use case.

Figure 2.8: Roadmap of the component WP5-IND-3

2.3.3.1 Results at M10

The work carried out during the first period of the project (M4-M10) focused on the definition of the requirements associated to the specific UC1 Demonstrator 1 involving the MANTIS drone, study and analysis of the avionics subsystems involved, specification of the internals and interfaces and definition of the component.

2.3.3.2 Plans for the year 2 and the year 3

In year 2 and year 3, the work planned will focus on the design, development and test of tests that enable communication between the frontend and the backend of the control station application so that the actions requested by the user, are sent to the autopilot.

- Year 2: development, integrations, and deployment of first prototype.
- Year 3: adjustments, modifications, verification and validation (final deployment) in Use Case 1 Demonstrator 1.

We plan to measure component characteristics, and compute metrics and KPIs according to the Table 2.6, and so to contribute to evaluate the achievement of the project objective O3.

Table 2.6: KPI and metric of the component WP5-IND-3

№	KPI	#	Metric	O
1	Reducing cybersecurity risks	1	Radiofrequency communication drone ↔ GCS includes: ● Interference error detection: 32-bit of CRC, ARQ ● 128-bit AES encryption ● Different types of modulations that allow an "Excellent strong signal interference rejection Characteristics"	O3

№ = Component wide KPI number KPI = KPI description
# = KPI wide metric number Metric = Metric description
O = Project Objective (see Table 1.5) whose achievement the metric contributes to evaluate

2.4 Ground control station user interface and communication with CMPD (WP5-IND-4)

The Table 2.5 provides the synopsis of the component WP5-IND-4.

Table 2.7: Component WP5-IND-4

Identifier: WP5-IND-4	**Partner:** Indra	**Expected TRL:** TRL6
Name: Communications GCS-CMPD		
License: Proprietary	**Owner:** Indra	**Contact:** airala@indra.es dlamas@indra.es
Description: Human-machine interface of the ground control station that enable communication between the GCS (Ground Control Station) and the CMPD (Drone Mission and Data Processing Center).		
Satisfied Requirement: IDENTIFIER — DEFINITION • DEM1-FNC-5 — The drone shall provide HD video in real time to the Mission Center over 4G.		
Key Enabling Technology: KET — CATEGORY • KET: Operations management — CATEGORY: U-Space Capabilities		
Improvement: (SW) analysis, design, development and testing of the human-machine interface of the ground control station that enable communication between the ground control station and the CMPD system so that the operator interacts in communications with the CMPD.		
Contributor: Indra	**Task:** T5.2	
Use Case: • UC1 — Transport **Demonstrator:** • D1 — Road transport: traffic management & monitoring, incident detection		

2.4.1 Architecture context and interfaces

The Figure 2.9 depicts the architecture context of the component WP5-IND-4.

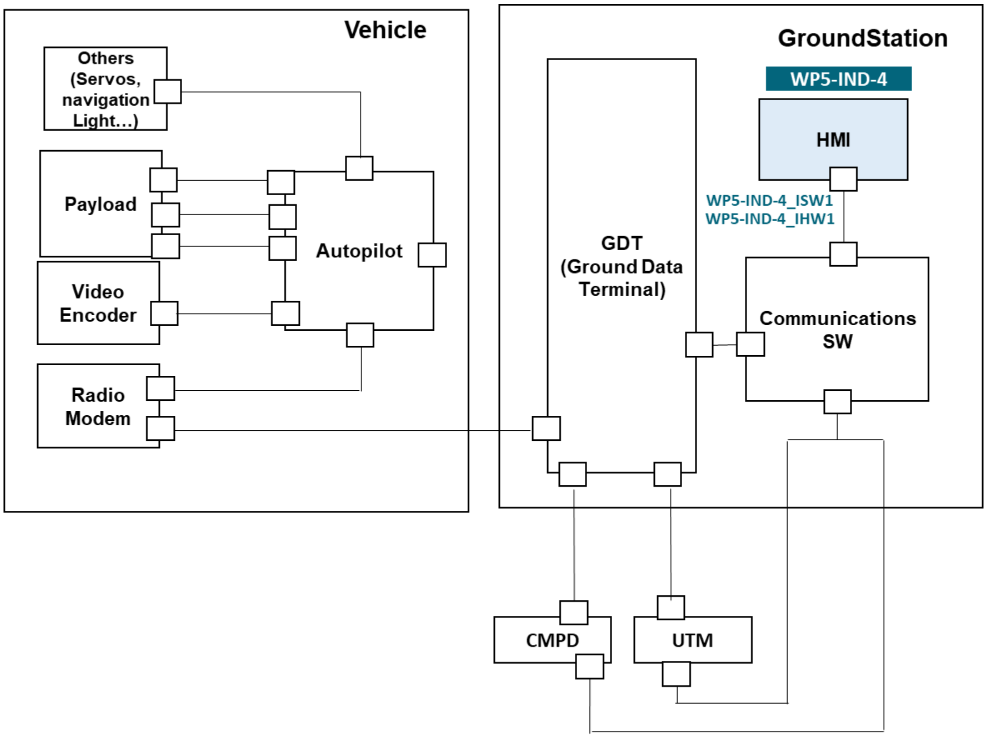


Figure 2.9: Architecture context of the component WP5-IND-4

- WP5-IND-4_IHW1: external HW interface for communication between the CMPD system and the GCS HMI. This is an ethernet communications interface.
- WP5-IND-4_ISW1: External software interface for communication between the CMPD system and the GCS HMI. The communication protocol is TCP/IP with HTTP technology.

2.4.2 Internals and technologies

In relation to the HMI, it will be necessary to:

- Modify information screens that allow the operator to show the operator the information provided by the CMPD to make the appropriate decisions during the execution of the flight plan.
- Associate additional events, new options available or existing ones in the HMI, to trigger the start of the communication flow with the CMPD.

It will be necessary to implement a new messaging and control logic that allows the aircraft operator to interact from the HMI with the CMPD system.

More specifically, it will be necessary to implement the following messaging:

- Messaging to receive request for contingent mission from the CMPD system. This messaging includes receiving mission data, as well as managing the response flow, handling errors and retries agreed upon in the analysis phase.
- Messaging to send confirmation of contingent mission flight plan authorization to the CMPD system. This messaging includes sending flight plan data in the format expected by CMPD, as well as managing the response flow, handling errors and retries agreed in the analysis phase.

- Messaging to notify the CMPD system that the MANTIS aircraft is in a position to undertake the mission. This messaging includes sending notification data in the format expected by CMPD, as well as managing the response flow, handling errors and retries agreed upon in the analysis phase.
- Messaging to receive mission start order from the CMPD system. This messaging includes receiving order data, as well as managing the response flow, handling errors and retries agreed upon in the analysis phase.
- Messaging to send telemetry and video from the MANTIS aircraft to the CMPD system. This messaging includes sending telemetry data in the format expected by CMPD, as well as managing the response flow, handling errors and retries agreed upon in the analysis phase.
- Messaging to receive end-of-mission order from the CMPD system. This messaging includes receiving order data, as well as managing the response flow, handling errors and retries agreed upon in the analysis phase.

2.4.3 Component roadmap

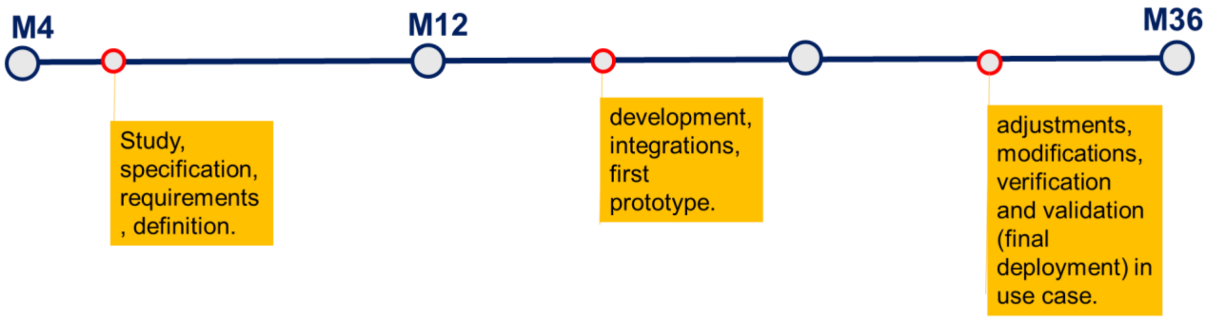

M4
M12
M36
Study, specification, requirements , definition.
development, integrations, first prototype.
adjustments, modifications, verification and validation (final deployment) in use case.

Figure 2.10: Roadmap of the component WP5-IND-4

2.4.3.1 Results at M10

The work carried out during the first period of the project (M4-M10) focused on the definition of the requirements associated to the specific UC1 Demonstrator 1 involving the MANTIS drone, study and analysis of the avionics subsystems involved, specification of the internals and interfaces and definition of the component.

2.4.3.2 Plans for the year 2 and the year 3

In year 2 and year 3, the work planned will focus on design, development and testing of the human-machine interface of the ground control station that enable communication between the ground control station and the CMPD system so that the operator interacts in communications with the CMPD.

- Year 2: development, integrations, and deployment of first prototype.
- Year 3: adjustments, modifications, verification and validation (final deployment) in the Use Case 1 Demonstrator 1.

Table 2.8: KPI and metric of the component WP5-IND-4

№	KPI	#	Metric	O
1	Reducing cybersecurity risks	1	Secure HTTPS protocol over 4G network will be used for communication between GCS and CMPD system	O3

№ = Component wide KPI number KPI = KPI description
# = KPI wide metric number Metric = Metric description
O = Project Objective (see Table 1.5) whose achievement the metric contributes to evaluate

2.5 UAS-UTM communications (WP5-IND-5)

The Table 2.9 provides the synopsis of the component WP5-IND-5.

Table 2.9: Component WP5-IND-5

Identifier: WP5-IND-5	**Partner:** Indra	**Expected TRL:** TRL6
Name: Communications - UAV - GCS - CMPD - UTM		
License: Proprietary	**Owner:** Indra	**Contact:** airala@indra.es dlamas@indra.es
Description: End-to-end communication between the MANTIS aircraft and the UTM platform.		
Satisfied Requirement: IDENTIFIER — DEFINITION • DEM1-FNC-2 — The drone operator shall create a flight plan based on the incident communicated by HORUS. (Transport Control Center, see the Use Case 1 in the C4D report D1.1 [3]) • DEM1-SEC-2 — The drone shall fly following the flight plan created and authorized, as well as the contingency route to recover the RPA in the event of loss link. • DEM1-SEC-3 — The communications between Mantis-GCS systems must be resilience against cyber-attacks. • DEM1-FNC-8 — The drone must communicate with the GCS and inform about its landing position.		
Key Enabling Technology: KET — CATEGORY • KET: Operations management — CATEGORY: U-Space Capabilities • KET: Network Centric Communications Systems — CATEGORY: System Functions		
Improvement: (SW) analysis, design, development and testing for end-to-end communication between the MANTIS aircraft and the UTM platform. Other features include those related to aircraft registration on the UTM platform, courier exchanged between the aircraft and UTM.		
Contributor: Indra	**Task:** T5.2	
Use Case: • UC1 — Transport **Demonstrator:** • D1 — Road transport: traffic management & monitoring, incident detection		

2.5.1 Architecture context and interfaces

The Figure 2.11 depicts the architecture context of the component WP5-IND-5.

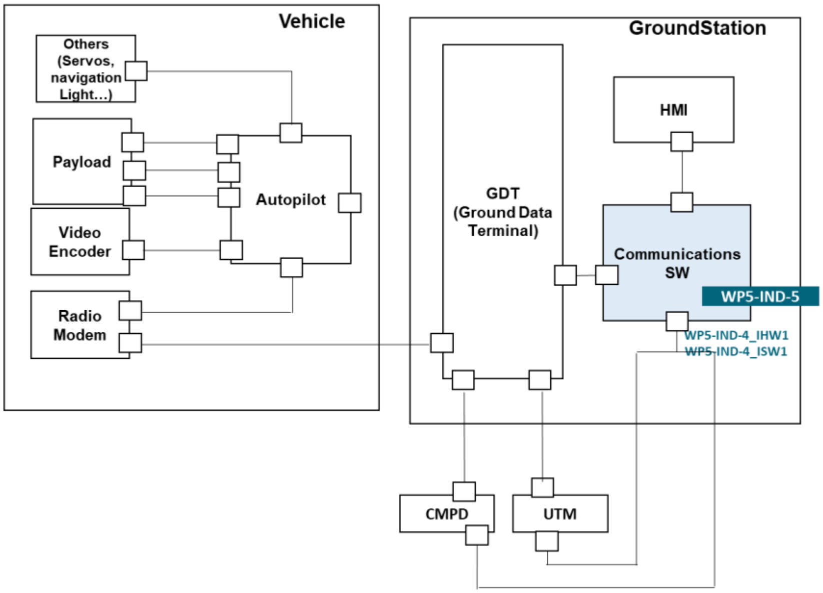


Figure 2.11: Architecture context of the component WP5-IND-5

- WP5-IND-5_IHW1: external Hardware interface for communication between the UTM system and the GCS HMI. This is an ethernet communications interface.
- WP5-IND-5_ISW1: External software interface for communication between the CMPD system and the GCS HMI. The communication protocol is TCP/IP with REST technology.

2.5.2 Internals and technologies

In relation to the HMI, it will be necessary to:

- Modify information screens that allow the operator to show the information provided by the UTM system so that he/she makes the appropriate decisions during the execution of the flight plan.
- Associate additional events, new options available or existing ones in the HMI, to trigger the start of the communication flow with the UTM system.

It will be necessary to implement a new messaging and control logic that allows the aircraft operator to interact from the HMI with the UTM system.

More specifically, it will be necessary to implement the following messaging:

- Messaging to send flight plan authorization request from the GCS to the UTM system. This messaging includes sending flight plan data in the format expected by UTM, as well as managing the response flow, handling errors and retries agreed in the analysis phase.
- Messaging to send telemetry data from the MANTIS aircraft to the UTM system. This messaging includes sending telemetry data in the format expected by UTM, as well as managing the response flow, handling errors and retries agreed in the analysis phase.
- Messaging to notify the UTM system that the MANTIS aircraft has started the flight. This messaging includes sending notification data in the format expected by UTM, as well as managing the response flow, handling errors and retries agreed upon in the analysis phase.
- Messaging to notify the UTM system that the MANTIS aircraft has completed the flight. This messaging includes sending notification data in the format expected by UTM, as well as managing the response flow, handling errors and retries agreed upon in the analysis phase.

Point-to-point communications UAV - GCS - CMPD - UTM will be validated using the components described above.

2.5.3 Component roadmap

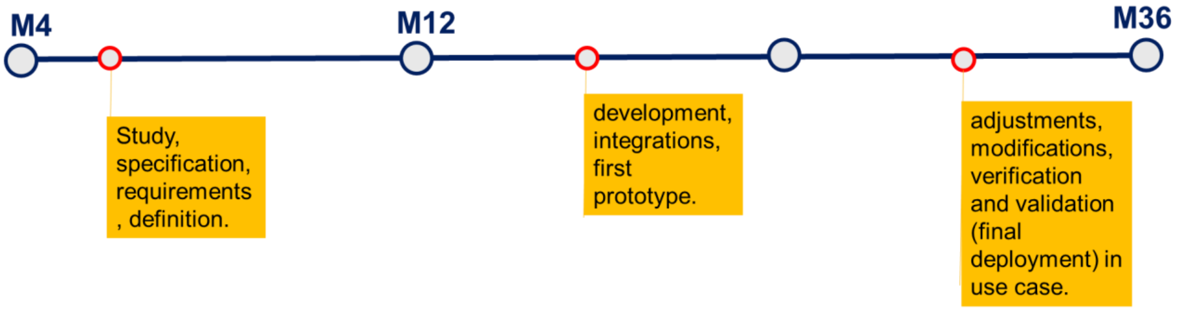

M4
M12
M36
Study, specification, requirements , definition.
development, integrations, first prototype.
adjustments, modifications, verification and validation (final deployment) in use case.

Figure 2.12: Roadmap of the component WP5-IND-5

2.5.3.1 Results at M10

The work carried out during the first period of the project (M4-M10) focused on the definition of the requirements associated to the specific UC1 Demonstrator 1 involving the MANTIS drone, study and analysis of the avionics subsystems involved, specification of the internals and interfaces and definition of the component.

2.5.3.2 Plans for the year 2 and the year 3

In year 2 and year 3, the work planned will focus on the design, development and testing for end-to-end communication between the MANTIS aircraft and the UTM platform. Other features include those related to aircraft registration on the UTM platform, courier exchanged between the aircraft and UTM.

- Year 2: development, integrations, and deployment of first prototype.
- Year 3: adjustments, modifications, verification and validation (final deployment) in the Use Case 1 Demonstrator 1.

We plan to measure component characteristics, and compute metrics and KPIs according to the Table 2.10, and so to contribute to evaluate the achievement of the project objective O3.

We plan to measure component characteristics, and compute metrics and KPIs according to the Table 2.10, and so contribute to evaluate the achievement of the project objective O3.

Table 2.10: KPI and metric of the component WP5-IND-5

№	KPI	#	Metric	O
1	Reducing cybersecurity risks	1	Secure HTTPS protocol over 4G network will be used for communication between GCS and UTM system	O3

№ = Component wide KPI number KPI = KPI description
# = KPI wide metric number Metric = Metric description
O = Project Objective (see Table 1.5) whose achievement the metric contributes to evaluate

2.6 LPWAN for identification, tracking, and emergency messages (WP5-05-TEK)

The Table 2.11 provides the synopsis of the component WP5-05-TEK.

Table 2.11: Component WP5-05-TEK

Identifier: WP5-05-TEK	**Partner:** TEK	**Expected TRL:** TRL4/TRL5
Name: Integrated long-range communication for UAV identification and monitoring		
License: Not Applicable	**Owner:** TEK	**Contact**: https://en.tekne.it/308/contact-us.html
Description: The WP5-05-TEK provides an integrated long-range communication link by which the unmanned vehicles can be identified and monitored.		
Satisfied Requirement: IDENTIFIER — DEFINITION • UC5-DEM10-FNC-002 — Vehicle identification. The system should provide a long-range communication link by which the unmanned vehicles can be identified.		
Key Enabling Technology: KET — CATEGORY • KET: E-Identification — CATEGORY: U-Space Capabilities • KET: Intelligent Vehicle System Monitoring — CATEGORY: System Functions		
Improvement: Low resource (power, weight, cost) UAV identification and tracking system, UAV integrated in open source IoT system.		
Contributor: TEK	**Task:** T5.1, T5.2, T5.4	
Use Case: • UC5 — Agriculture **Demonstrator:** • D1 — Crop monitoring		

2.6.1 Architecture context and interfaces

The Figure 2.13 depicts the architecture context of the component WP5-05-TEK.

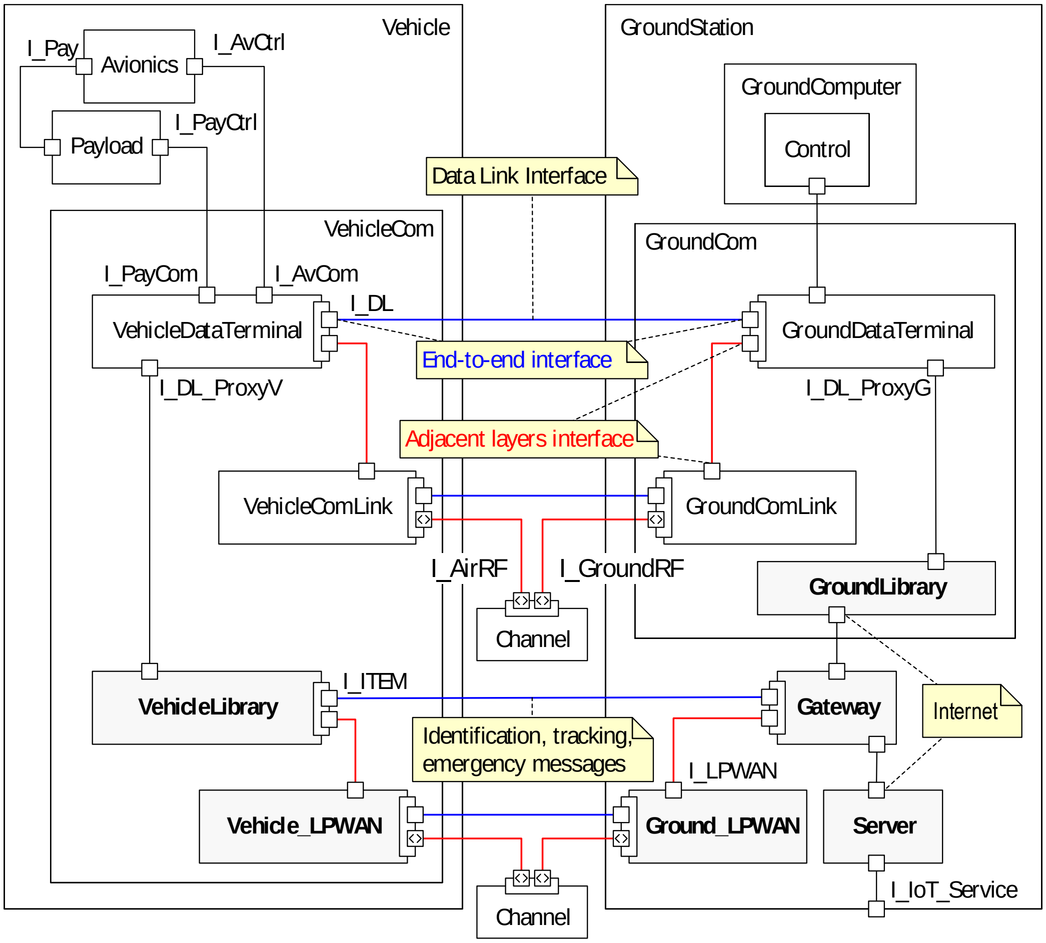


Figure 2.13: Architecture context of the component WP5-05-TEK

With respect to the architectural context blueprint of Figure 1.1, the Figure 2.13 adds the following details that apply to a generic UAS (Unmanned Aircraft System):

- **End-to-end** and **adjacent layers interfaces** — Following [18], the communication interfaces are modelled on two levels of abstraction:
    - as end-to-end protocol entities, model elements are connected horizontally through *end-to-end interfaces*;
    - as entities on different communication stack levels, they are connected vertically through *adjacent layers interfaces*.
- **Control** — The block *Control* is the element of the GroundStation block that loads and executes the mission and retrieves the data products[1]. It controls the unmanned vehicle through the blocks *VehicleCom* and *GroundCom* that are the communication elements. *Avionics* and *Payload* are the controllable blocks of the vehicle.

[1] The Figure 2.13 does not shows the blocks that provide the other required functionalities: to generate the mission and to exchange information with the C4I systems.

- **Data Link Interface (I_DL)** — The *Data Link Interface* enables the GroundStation block to generate and transmit, as well as to receive and understand messages for controls and status of the UAV (Avionics and Payload). The messages are according to a given protocol (such as MAVLink [19] and STANAG 4586 [20]) that is vehicle independent and has adaptability[2] characteristics (e.g. to add custom messages).

- **VehicleDataTerminal** — The *VehicleDataTerminal* block translates the Data Link Interface messages to/from the format that the blocks Avionic and Paylod require.

- **I_DL_ProxyV** — The *I_DL_ProxyV* (Vehicle Data Link Proxy) interface is a replica of the Data Link Interface that the VehicleDataTerminal element offers on a port to which additional on-board blocks can connect.

- **GroundDataTerminal** — The *GroundDataTerminal* block translates the Data Link Interface messages to/from the format that the block Control requires.

- **I_DL_ProxyG** — The *I_DL_ProxyG* (Ground Data Link Proxy) interface is a replica of the Data Link Interface that the VehicleDataTerminal element offers on a port to which additional ground blocks can connect. This working hypothesis will be verified during the project prosecution and, in the case, the WP5-05-TEK component will be adapted.

- **VehicleComLink** and **GroundComLink** — The communication stack layers below the application level are modelled with the blocks *VehicleComLink* and *GroundComLink*. These two blocks include the transceivers and the antennas whose radiofrequency ports are modelled with the interfaces I_AirRF and I_GroundRF.

2.6.2 Internals and technologies

The WP5-05-TEK component provides an LPWAN (Low Power Wide Area Network) based solution by which:

- the unmanned vehicles can be identified by the Ground Station;
- the unmanned vehicles can be tracked by the Ground Station;
- the unmanned vehicles and the Ground Station can exchange management and warning messages in emergency situations;
- compatibly with the bandwidth availability, the unmanned vehicles can transmit the mission products to the Ground Station.

Functional blocks — The WP5-05-TEK component is made up of the blocks that in the Figure 2.13 have a light gray background and that are detailed in the following.

- **I_ITEM** — The *I_ITEM* interface enables the GroundStation block to receive and understand as well as to generate and transmit messages for UAV identification and tracking, for management and warning in emergency situations, and for the mission products compatibly with the bandwidth availability. The protocol will be defined during the project prosecution.

- **VehicleLibrary** — The *VehicleLibrary* block translates the I_ITEM interface messages to/from the format that the I_DL_ProxyV interface requires. Moreover, VehicleLibrary includes the application level that manages the identification and tracking functions: retrieving UAV position through the I_DL_ProxyV interface, managing the UAV identifier, answering/sending solicited/unsolicited message to the Gateway block.

- **GroundLibrary** — The *GroundLibrary* block translates the message format that the I_DL_ProxyG interface requires to/from the format accepted by the block Gateway.

[22] *Adaptability* is the “Degree to which a product or system can effectively and efficiently be adapted for different or evolving hardware, software or other operational or usage environments.” [21]

- **Gateway** — The *Gateway* block uses the messages on the I_ITEM interfaces to communicate with the UAV in order to provide the following functionalities: identification, tracking, management and warning message in emergency situations, mission products retrieval. At lower level, it configures the LPWAN (the blocks Vehicle_LPWAN and Ground_LPWAN) through the interface I_LPWAN. At higher level, it offers the services to access these functionalities to the block GroundLibrary. with which it has an internet connection.

- **I_LPWAN** — The interface *LPWAN* between the blocks Gateway and Ground_LPWAN, which are on adjacent layers, is used by the former to receive and transmit messages and to issue LPWAN configuration messages.

- **Vehicle_LPWAN** and **Ground_LPWAN** — The *Vehicle_LPWAN* and *Ground_LPWAN* blocks provide the communication link to the

- **I_IoT_Service** — On the *I_IoT_Service* interface, the Sever block (see the next bullet) offers as Web Services the mission products, and the UAV identification, position, and messaging.

- **Server** — The *Server* block demonstrates the interoperability[3] and the reusability[4] of WP5-05-TEK, namely:
  - The Server block is an IoT (Internet of Thing) system that integrates the unmanned vehicles.
  - On the interface I_IoT_Service, the Server block offers as a Web Service the mission products that it receives and stores.
  - On the interface I_IoT_Service, the Server block offers as a Web Service the identification and the position of, as well as the messaging to/from the UAVs; the payloads of the Web Service are according to a UTM (UAS Traffic Management) standard that it implements in part.

Constraints — The constraints that apply to the WP5-05-TEK component are the following:

- The WP5-05-TEK component will be based on LPWAN (Low Power Wide Area Network) technologies, such as NB-IoT (Narrowband Internet of Things) or LoRa (Long Range).

- Messages will be encrypted.

- Identification and tracking messages will be solicited and unsolicited and will contain at least the vehicle identifier and the vehicle GPS based position.

- The WP5-05-TEK component will be verified under DoS (Denial of Service) jamming attacks.

2.6.3 Component roadmap

2.6.3.1 Results at M10

Within M10 we carried out the following activities: simulation with OMNeT++; experimentation with COTS transceivers and gateways; experimentation with open source IoT middleware and server; first experimentation with UAV equipped with LoRa transceiver.

2.6.3.2 Plans for the year 2 and the year 3

Month 22: WP5-05-TEK critical functionalities demonstrated, verified in laboratory/simulated environment.

Month 30: WP5-05-TEK key functionalities demonstrated, integrated and verified in controlled environment (TRL4).

Month 36: WP5-05-TEK validated in the use case (target TRL: between TRL4 and TRL5).

[3] *Interoperability* is the “Degree to which two or more systems, products or components can exchange information and use the information that has been exchanged.” [21]

[4] *Reusability* is the "Degree to which an asset can be used in more than one system, or in building other assets." [21]

We plan to measure component characteristics, and compute metrics and KPIs according to the Table 2.12, and so contribute to evaluate the achievement of the project objective O3.

Table 2.12: KPI and metric of the component WP5-05-TEK

№	KPI	#	Metric	MO
1	Data link availability	1	Communication, identification, and tracking range.	MO3.2 (multi-radio) MO3.3 (security)
1	Data link availability	2	Capability to work in crowded radio bands, number of mobile nodes.	MO3.2 (multi-radio) MO3.3 (security)
1	Data link availability	3	Update data rate for identification and tracking.	MO3.3 (security)

№ = Component wide KPI number
KPI = KPI description
# = KPI wide metric number
Metric = Metric description
MO = Measurable Outcome (see Table 1.5) that the metric refers to and refines

2.7 Path manager and scheduler (WP5-12-ANYWI)

The Table 2.13 provides the synopsis of the component WP5-12-ANYWI.

Table 2.13; Component WP5-12-ANYWI

Identifier: WP5-12-ANYWI	**Partner:** ANYWI	**Expected TRL:** TRL5
Name: Path manager and scheduler		
License: Proprietary	**Owner:** AnyWi	**Contact:** AnyWi
Description: Path manager to monitor connection availability and quality of the different base communication channels for drone use.		
Satisfied Requirement: IDENTIFIER — DEFINITION • UC4-INT-03 — Rover acting like a radio hub, ground station. • UC4-INT-03 — Communication independence from the environment. (Self-carried radio infrastructure). • UC4-SEC-06 — Typical communication range.		
Key Enabling Technology: KET — CATEGORY • KET: Communication, Navigation and Surveillance — CATEGORY: U-Space Capabilities • KET: Network Centric Communications Systems — CATEGORY: System Functions		
Improvement: Develop a path manager to monitor connection availability and quality of the different base communication channels. These channels can be BT-LE, 802.11 and other channels for local communication. Progress over SotA: Compared to e.g. standard MPTCP, this path manager will use information from lower layers of the communication stack to improve discovery of useful channels.		
Contributor: AnyWi	**Task:** T5.1	
Use Case: • UC4 — Surveillance & Inspection **Demonstrator:** • D2 — Fleet of multi robot navigating and mapping in an unknown environment		

2.7.1 Architecture context and interfaces

This WP5-12-ANYWI component implements two main functionalities included in the overall architectural canvas in Figure 1.1, namely the *VehicleCom* and *GroundCom* components. For the purpose of the implementation of this component, further internal sub-components must be implemented, as outlined in Figure 2.14 and described in the following.

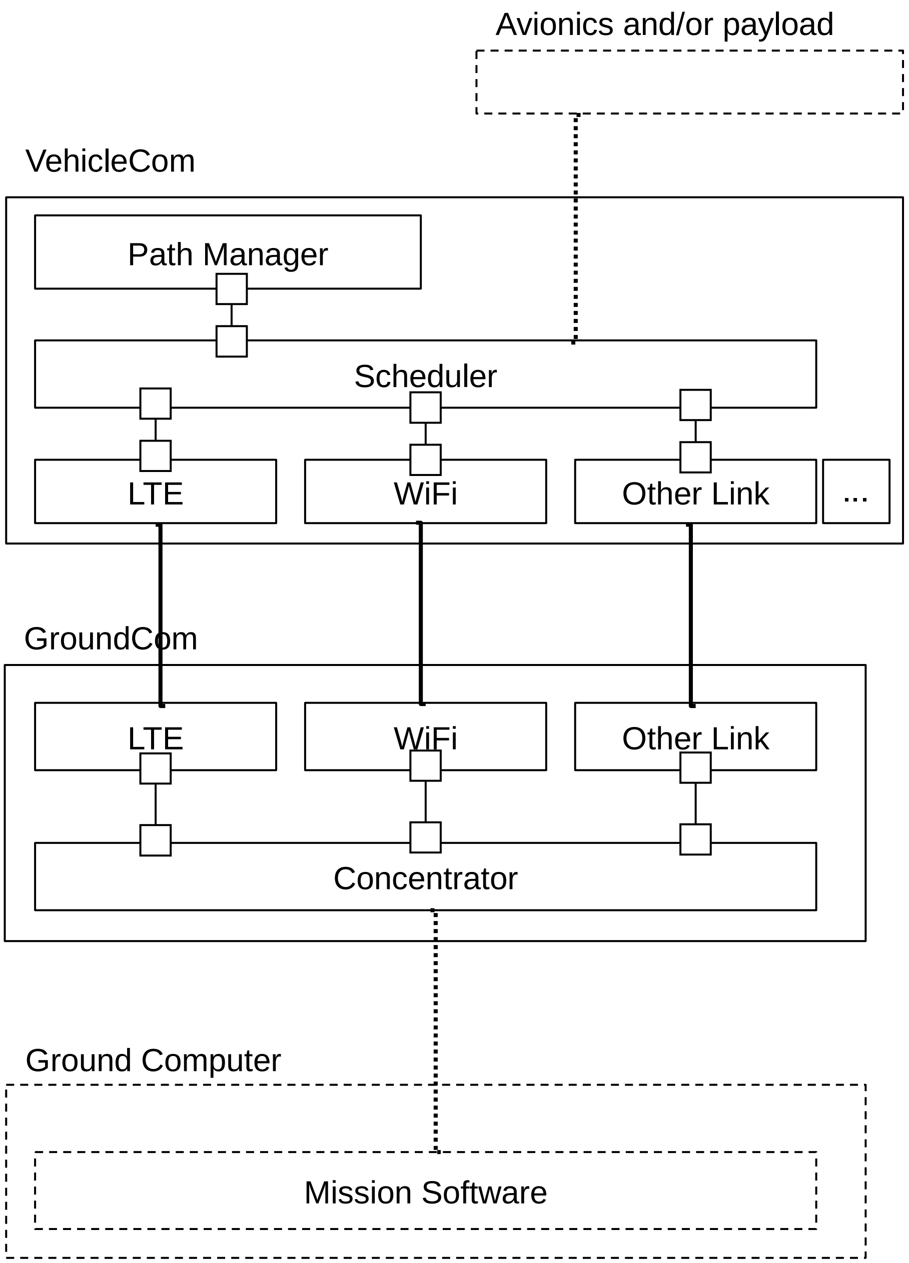


Figure 2.14: Architecture context of the component WP5-12-ANYWI

2.7.2 Internals and technologies

The ***internals***, in the form of components and sub-components of Figure 2.14, can be summarised as follows:

VehicleCom. In this implementation, the block VehicleCom organises the single links of communication from drone to ground as a bundling/merging of two or more individual links, that may be of the same type (e.g., two LTE/5G connections) or of a heterogeneous nature, such as one LTE and one Wi-Fi connection.

Path manager. This sub-component has the main objective to identify and monitor available links on a constant basis, to maintain an updated status of the availability of the communication link for all phases of flight, detecting and classifying available communications links with respect to reliability. It provides ongoing link metadata to the scheduler component, facilitating optimal link utilization when multiple links are available, and reliable handover between links. The path manager also handles establishing the links, which normally requires registration and authentication on the network that carries the link.

Scheduler. The scheduler has the task of deciding on a packet-by-packet basis which of the available links to use according to the decided policy (lowest latency, lowest jitter, lowest cost, etc.). In the initial version, an estimate of lowest latency is planned for the decision function.

LTE/Wi-Fi/Other comms modules. The architecture is designed to incorporate commercially available communication modules for mobile (LTE/5G) and Wi-Fi communication, and to interact with the APIs provided by these modules. As such the modules themselves and their internal firmware are out of scope for this component.

GroundCom. This module consists of the counterparties of the on-board path manager and scheduler: the single block Concentrator.

Concentrator. The Concentrator works as the ground-based counterparty of the scheduler and to some degree also of the path manager. The traffic received from the VehicleCom module and sent over different links is merged back into a single flow of packets and transmitted onwards to the endpoint, for instance the ground computer or a backend on the general internet. For traffic in the other direction, including packets for synchronisation and acknowledgement as in the TCP protocol, the concentrator employs a scheduler as well, which uses the same software as the one on the vehicle.

The ***technologies to be developed*** will extend on what is currently available in existing multipath capable protocols such as MPTCP (MultiPath TCP), SCTP (Stream Control Transmission Protocol) and the upcoming QUIC (Quick UDP Internet Connection) core. These systems leave path management largely outside of their specification scope. Additionally, they do not lend themselves well to carrying arbitrary application (mission software) data streams, as they impose very specific reliability models on the application.

In the scheduler and concentrator components, AnyWi will provide a simple alternative multipath protocol, loosely modelled after the existing standards, but geared more towards providing a virtual link interface. Such an interface will allow both carrying arbitrary (IP-based) data as streams or datagrams, and also keep the application layer agnostic to hardware link switchovers.

Path management will take into account trends in link reliability statistics, such that either multiple reliable paths can be used concurrently, or to enable transitioning the virtual link from one hardware link to another.

The implementation will be on Linux. Reference hardware is to be determined, though for use in drones, ARM-based devices are likely preferable. LTE and Wi-Fi modules are also to be determined.

2.7.3 Component roadmap

2.7.3.1 Results at M10

First design of a base framework for path manager and scheduler components.

2.7.3.2 Plans for the year 2 and the year 3

M12-M24: Develop functionality of path manager to consume link state metainformation from link drivers and firmware via a dedicated API (see WP5-13-ANYWI). Path manager will perform simple analysis of link state metainformation, resulting in link reliability and priority data relayed to the scheduler component. The scheduler component will be developed to multiplex application traffic across active links accordingly.

M25-M36: Develop path manager functionality to perform predictive analysis of link state metainformation, to yield optimized scheduling of application data by the scheduler component. Verify and validate path management capabilities.

It is planned to measure component characteristics, and compute metrics and KPIs according to the Table 2.14, and so contribute to the evaluation of the project objective O3.

Table 2.14: KPI and metric of the component WP5-12-ANYWI

№	KPI	#	Metric	MO
1	Robustness	1	Number of packets lost from drone to ground before switch to alternative link	MO3.2
1	Robustness	2	Number of packets lost from ground to drone before switch to alternative link	MO3.2
1	Robustness	2	Time from availability of link on operating system-level until availability in path manager	MO3.2

№ = Component wide KPI number KPI = KPI description
# = KPI wide metric number Metric = Metric description
MO = Measurable Outcome (see Table 1.5) that the metric refers to and refines

2.8 Link state Application Programming Interface (WP5-13-ANYWI)

The Table 2.15 provides the synopsis of the component WP5-13-ANYWI. This component accompanies WP5-12-ANYWI. It will be used to improve the functionality of path manager and scheduler by making available metainformation about the link status, such as information form lower layers that can help making better routing decisions at the transport layer. Examples of this are radio signal quality information, internal information about packet latencies, etc. The functionality of the API is of more general interest outside of the multipath links developed in WP5-13-ANYWI, which is the reason this is defined as a separate component.

Table 2.15: Component WP5-13-ANYWI

Identifier: WP5-13-ANYWI	Partner: ANYWI	Expected TRL: TRL6
Name: Link state API		
License: Commercial	**Owner:** AnyWi	**Contact:** AnyWi
Description: API to supply communication link state 6 metainformation to path manager.		
Satisfied Requirement: IDENTIFIER — DEFINITION • UC4-DEM2-SEC-002 — Communication: Robust against signal interruptions. This component serves in partial fulfilment of the requirement in that it provides information for other components to be more robust against signal interruptions.		
Key Enabling Technology: KET — CATEGORY • KET: Communication, Navigation and Surveillance — CATEGORY: U-Space Capabilities • KET: Network Centric Communications Systems — CATEGORY: System Functions		
Improvement: Develop an API to communicate connection quality and availability as well as available unused resources (such as unused links) to support on-board applications, with prioritized communication. Progress over SotA: standards, such as MPTCP, do not provide link quality information to other applications.		
Contributor: AnyWi	**Task:** T5.1, T5.2	
Use Case: • UC4 — Surveillance & Inspection **Demonstrator:** • D2 — Fleet of multi robot navigating and mapping in an unknown environment		

2.8.1 Architecture context and interfaces

The link state API is an interface internal to the component that provides the bridge between link drivers and the Path Manager (see Figure 2.14).

The Link State API is an internal sub-component intended to work with the various subcomponents of WP5-12-ANYWI, especially the path manager. It is intended to provide relevant information from lower layers of the OSI stack, notably the physical layers, about, e.g., radio signal quality to allow the path manager to estimate the quality of a link and hence to include it in the pool of available links or not. In the context of the overall architectural framework of Figure 1.1, it acts as an extension of the VehicleCom component. This is illustrated in the figure below.

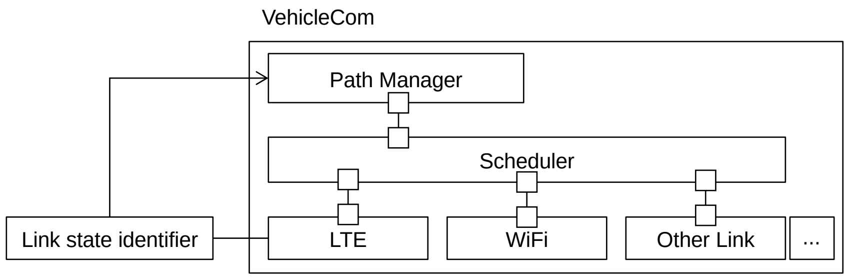


Figure 2.15: Architecture context of the link state identifier of component WP5-13-ANYWI

The Link state identifier is a subcomponent that queries the LTE/5G and Wi-Fi modules (and other modules, such as BTLE, as appropriate) for further information about the connection quality, organises it into a suitable data format and makes it available to an API at the level of the path manager. For the path manager in WP5-13-ANYWI, this level is Linux user space, meaning that the API will be available to normal user space applications in Linux.

2.8.2 Internals and technologies

The link state API is to transfer information from the state of the multi-radio communication system to the lightweight communication system and the multi-path network communication.

The focus of the link state API is on loss-free general applicability. It provides link independent state and reliability metrics to consuming applications, with path manager from WP5-12-ANYWI being the primary target.

That is, any link driver and firmware should be able to report the entirety of its relevant link state metainformation in such a fashion that consumers stay agnostic to internal link functionality, yet do not miss information vital to their purpose.

Commercially available LTE/5G modules, such as those provided by Sierra Wireless, have firmware interfaces that make the relevant information available, but only via the module drivers and the data formats depend on the type of connection (3G/LTE/5G) being active at the moment of the query. A layer on top of this is required to make the data more easily available to the path manager software, and to align the output across different access technologies.

Similar information, although typically less rich, is available from Wi-Fi modules (often restricted to RSSI values). This information depends on the available drivers, and also in this case is alignment needed across different vendors of Wi-Fi link technology.

For the implementation of the of the link state identifier, suitable open source frameworks will be explored to provide a starting ground. The initial candidate is the ModemManager package[5], which contains a layer to query the firmware, to provide the necessary raw data and handle concurrency issues with the access to the module firmware. A similar simple package will be sought for Wi-Fi modules, or the drivers will be queried directly.

2.8.3 Component roadmap

2.8.3.1 Results at M10

At M10, the first high level design of signal quality identifier has been developed.

2.8.3.2 Plans for the year 2 and the year 3

M12-M24: Research and develop best practice link state metainformation generically applicable to a range of example hardware. Develop link state sampling algorithms to provide appropriate data to the API.

M25-M36: Finalize API design and provide implementation of link state samplers for a range of example hardware pertinent to the use case. Verify and validate link state API.

It is planned to measure component characteristics, and compute metrics and KPIs according to the Table 2.16, and so contribute to evaluate the achievement of the project objective O3.

[5] https://www.freedesktop.org/wiki/Software/ModemManager/

Table 2.16: KPI and metric of the component WP5-13-ANYWI

№	KPI	#	Metric	MO
1	Availability of link status information from radio layer	1	Number of relevant data points for signal quality extracted from the radio layer of Mobile and WiFi link technologies	MO3.2

№ = Component wide KPI number KPI = KPI description
# = KPI wide metric number Metric = Metric description
MO = Measurable Outcome (see Table 1.5) that the metric refers to and refines

2.9 Hardware security component (WP5-14-IFAT)

The Table 2.17 provides the synopsis of the component WP5-14-IFAT.

Table 2.17: Component WP5-14-IFAT

Identifier: WP5-14-IFAT	**Partner:** IFAT	**Expected TRL:** TRL4/TRL5
Name: Hardware security component		
License: Proprietary	**Owner:** IFAT	**Contact:** IFAT
Description: The hardware component shall be a separate chip, which for security-reasons is physically separated from the main application microcontroller of the drone. Therefore, the hardware component provides a commonly used I2C or SPI interface to connected to the main microcontroller. Furthermore, the corresponding SW-API shall provide a set of corresponding C-based libraries to be called by the main application microcontroller and will be primarily used to support security-relevant operations.		
Satisfied Requirement: IDENTIFIER — DEFINITION • UC5-DEM9-FNC-xx — This hardware component provides security-measures to support the main application microcontroller of the drone with functionality such as platform integrity check options, cryptographically secured drone identification and drone and/or control unit authentication. • UC5-DEM9-FNC-xx — The corresponding firmware- and software-components provide APIs for use in the modular drone architecture framework, primarily supporting security-relevant tasks in the security management, such as support functions for establishing trusted communication.		
Key Enabling Technology: KET — CATEGORY • KET: Security — CATEGORY: U-Space Capabilities • KET: Network Centric Communications Systems — CATEGORY: System Functions		
Improvement: State-of-the-art components, such as TPMs are previously historically focused on PC-operating systems (such as Windows), and not yet optimized for the use in embedded systems. Therefore, the novel approach is to provide APIs which can be configured and used in a more flexible way, e.g. with reduced functionality specifically adapted for the integration into the targeted embedded system.		
Contributor: IFAT	**Task:** T5.3	
Use Case: • UC5 — Agriculture **Demonstrator:** • D2 — Wine production		

2.9.1 Architecture context and interfaces

The Figure 2.16 depicts the architecture context of the component WP5-14-IFAT.

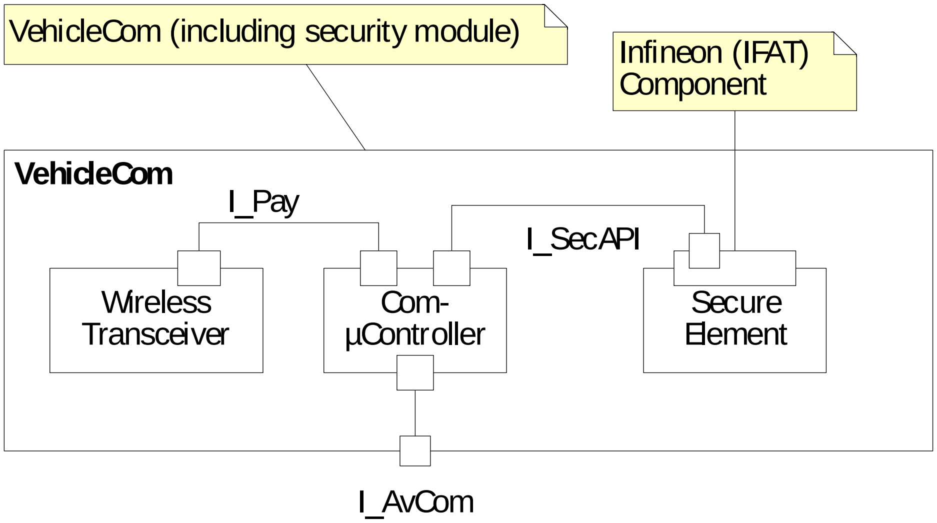


Figure 2.16: Architecture context of the component WP5-14-IFAT

As depicted in the Figure 2.16, the IFAT component mainly comprises: (1) the “Secure Element” (hardware security component) and (2) corresponding software/driver-parts which are added and executed either on one “Main Drone microcontroller”, or on a separated “communication microcontroller” of the “VehicleCom” component (“Com-µController” part)—however, this distinction is depending on the overall drone architecture (not defined by IFAT).

The corresponding primary interface is the “I_SecAPI”, which will be defined and later refined by IFAT in the course of WP5 partly in cooperation with WP3.

2.9.2 Internals and technologies

The “Secure Element” is a dedicated chip where security-critical operations are performed in a secured and temper-resistant environment, and therefore this chip is physically isolated from the general-purpose microcontroller—since this part is not defined by IFAT, it will either be the “Main Drone microcontroller” or a separate “Com-µController”, from IFAT’s perspective this counterpart will be generically denoted as “Host Controller” (this is a part defined and developed by other C4D partners).

The underlying technology of the “I_SecAPI” hardware-interface will either be SPI or I2C protocol (will be further detailed by IFAT in the next project steps).

2.9.3 Component roadmap

2.9.3.1 Results at M10

In the first WP5 phase between M04-M10 IFAT has worked on the selection of suitable “Secure Element” hardware-platform depending on the required API-functions and is currently further refined (including first inputs of the UC5 D2 demonstrator with respect to security). The intermediate result the most promising hardware candidate and option currently investigated is using the Infineon “OPTIGA Trust X” as the hardware platform as basis.

Furthermore, based on the draft reference architecture of the security-functionality IFAT has defined within WP3, in the course of WP5 the required API functions such as cryptographic primitives are currently defined. As depicted in Figure 2.17, some basic cryptographic primitives have been already defined, which are necessary hardware-security assisted TLS handshake. For such functions, within the course of WP5, IFAT will develop low-level API driver and software functions. Remark: These underlying

APIs and functionalities developed in WP5 will then partly be used as basis for developing modular/generic software-components within WP3.

Due to the advantages of the planned hardware-security-assisted TLS enhancement (and the planned modularity concept), it is seen as essential contribution by IFAT for to improve the security of drone identification, authentication, and data-privacy for drone-to-drone and drone-to-infrastructure communication.

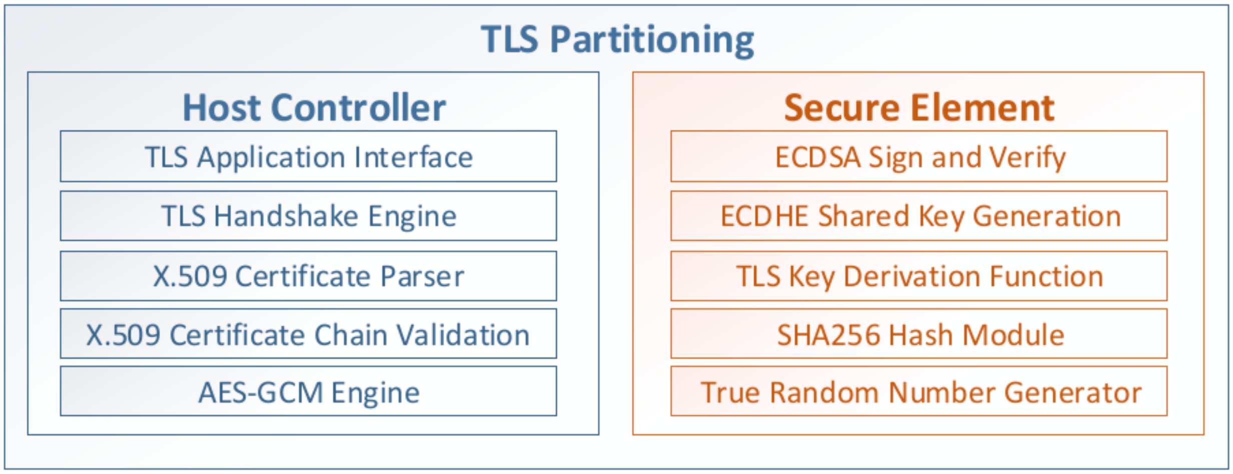


Figure 2.17: First selection of cryptographic primitives of the “Secure Element” (for TLS1.2 handshake)

2.9.3.2 Plans for the year 2 and the year 3

In the year 2 and 3 of the project, in the course of WP5, IFAT will focus on firmware and software development of the security-relevant API functions. In the 2nd year the focus will be the basis of lower-level API functions required as basis for the TLS use-case, while in the 3rd year IFAT will focus on higher-level functions and potential further security-relevant use cases.

Correspondingly, in the course of the following two associated main deliverables IFAT will report its results in further details in year 2 and year 3:

- Month 22: report D5.5 “APIs for Trusted Communication – first version” (Lead: IFAT).
- Month 30: report D5.6 “APIs for Trusted Communication – final version” (Lead: IFAT).

As previously defined the targeted TRL of this IFAT component is between TRL4-5. Furthermore, it is planned to contribute and to be partly tested in view of UC5 Demonstrator 2.

We plan to measure component characteristics, and compute metrics and KPIs according to the Table 2.18, and so contribute to evaluate the achievement of the project objective O3.

Table 2.18: KPI and metric of the component WP5-14-IFAT

№	KPI	#	Metric	MO
1	Improve security by adding hardware-based security component. Both metrics in comparison to state-of-the-art drone systems	1	Number of added hardware-security functions for helping to establish a secured communication channel for drone-to-X.	O3
		2	Number of credentials stored in a protected temper resistant memory (to protect against physical cloning attacks when a drone is captured).	O3

№ = Component wide KPI number KPI = KPI description
# = KPI wide metric number Metric = Metric description
O = Project Objective (see Table 1.5) whose achievement the metric contributes to evaluate

3 System functions components portfolio

This chapter describes the components that address mainly Key Enabling Technologies of the "System Function" set (see "Key Enabling Technologies" at page 15).

3.1 Security management toolchain (WP5-02-IKER)

The component WP5-02-IKER developed by IKERLAN is a Security Management Toolchain for the monitoring and control of the drone. There are several attacks, such as, eavesdropping, hacking, or identity spoofing, which compromise the security of communication links. WP5-02-IKER component provides different monitoring, update, and visualization features in order to detect anomalous behaviour and vulnerabilities within the drone. For that purpose, in this component, most security-relevant variables (such as software versions, abnormal execution and communication patterns, protocol and certificates) will be tightly measured and monitored.

The Table 3.1 provides the synopsis of the component WP5-02-IKER.

Table 3.1: Component WP5-02-IKER

Identifier: WP5-02-IKER	**Partner:** IKER	**Expected TRL:** TRL5
Name: Security Management Toolchain for Drone Monitoring and Control		
License: Open source	**Owner:** IKERLAN	**Contact:** mbarcelo@ikerlan.es
Expected Outcome: Security Management Toolchain for Drone Security Monitoring and Control. The tools must be able to detect anomalous behaviour and detect vulnerabilities within the drone.		
Description: This component provides a mechanism and a visualization interface to ensure that the drone is free of known vulnerabilities. This works as follows: the drone periodically sends information to a remote node, which processes it, extracts conclusions and shows them in a comprehensive manner.		
Satisfied Requirement: IDENTIFIER — DEFINITION • UC2-SEC-02 — Secure communications: The drone communication to send security related event data shall be secure. • UC2-SEC-03 — Collect history data: The drone shall collect and store a log of history data for future analysis. • UC2-SEC-04 — Vulnerability detection: The system shall give information about the drone status for the vulnerability detection. • UC2-SEC-05 — Software actualization: The system shall be able to update security libraries, protocols, certificates, or software versions.		
Standard: • ISO 27001 – Information Security Management • GDPR – General Data Protection Regulation		
Key Enabling Technology: KET — CATEGORY • Security — U-Space Capabilities • Communication, Navigation and Surveillance — U-Space Capabilities • Intelligent Vehicle System Monitoring — System Functions • Network Centric Communications Systems — System Functions • Over the Horizon Communications — System Functions		
Improvement: New vulnerabilities appear every day. In order to detect them in drone systems and perform the necessary actions as soon as possible, this component will periodically check for them and notify the administrator in case any vulnerability is detected. This will work not only against known vulnerabilities but with anomalous behaviour also.		
Contributor: IKERLAN	**Task:** T5.3	
Use Case:		

- UC2 — Construction

Demonstrator:

- D1 — Digitalization of civil infrastructure construction
- D2 — Monitoring underground infrastructure construction process

3.1.1 Architecture context and interfaces

The Figure 3.1 depicts the architecture context of the component WP5-02-IKER, includes the different modules that IKERLAN will develop, and shows the interactions and interfaces of the framework. The IKERLAN modules are highlighted in blue.

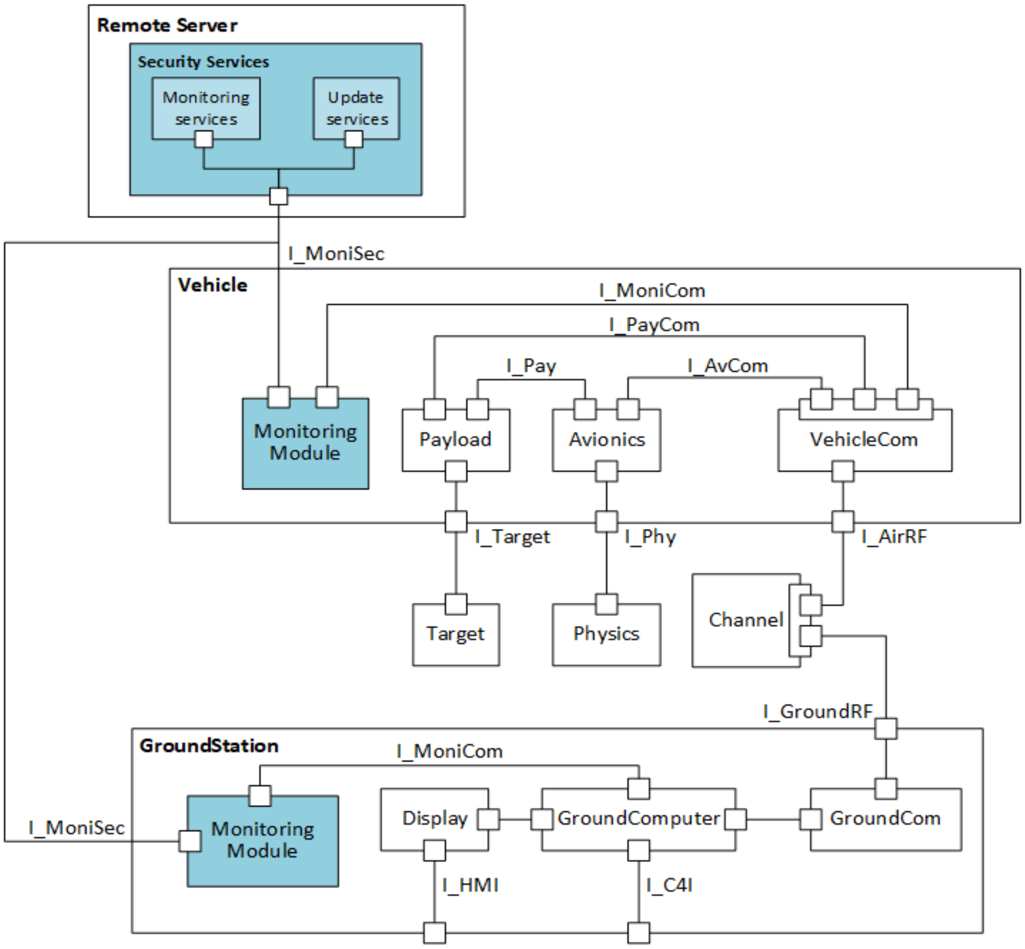


Figure 3.1: Architecture context of the component WP5-02-IKER

The modules can be classified into two different groups:

- **Monitoring Module**: This module is in charge of the data collection and information delivery to the security services. The communication uses the OSSEC message protocol and is sent compressed over a cyphered channel. This module also applies the required software updates. This module is included in the drone as well as in the Ground Station.
- **Security Services**: These services are located outside the drone infrastructure in a remote server. They process the received data and detect any vulnerability or anomalous behaviour.
  - **Monitoring services**: The monitor services are connected to the security modules through their OSSEC message protocol.

- **Update services**: The update services [8] [9] check if the security requirements regarding software components are up to date or need to be updated. If needed the updates are exchanged with the protocols used for each one of them.

3.1.2 Internals and technologies

For the next months, the development on the toolchain will evolve and adapt to overcome the issues we may face during the project. As it is not easy to foresee the future problems, we have identified some of the technologies we are going to use, extracted from our research on the monitoring matter.

Some of the technologies that we will use to construct our toolchain are the following:

- Suricata [10], to monitor the traffic and detect threats on it.
- SCAP [11], that is a framework of specifications to evaluate the security level of the devices.
- Wazuh [12], that is a tool to monitor the hosts in order to detect anomalous behaviour.
- Elasticsearch [13], to store the alerts and have easy access to the information retrieved.
- Hive [14], that is used to manage the alerts and include all the documentation.
- Docker [15], to limit the size and functionalized as well as to ease the update system.
- Swarm/Kubernetes [16] [17], that will help us to deploy easily in any environment.

3.1.3 Component roadmap

3.1.3.1 Results at M10

At M10 we have done the following:

- Define the requirements for the demonstrators involving the specific drones used in the construction use case (UC2).
- Study and evaluate different software update methods to decide which of them is more suitable for each of the drone of this use case.
- Define the security aspects that will be monitored within the API.
- Develop the preliminary version of the Security Management Toolchain.

3.1.3.2 Plans for the year 2 and the year 3

For next years, we foresee to have completed the following:

- Development of the Security Management Toolchain, including the monitoring and control capabilities of the firmware integrity and system configuration.

  The work regarding the implementation of the Toolchain within the different demonstrators and the in-lab tests, will be specified in the next phase.
- Year 2:
  - Testing for UC2 drone controller capabilities.
  - Build tools focusing on the compatibility with the UC2.
  - Analyse the data to verify it is between the expected parameters.
- Year 3:
  - Test toolchain functionalities individually to check the behaviour.
  - Implement the complete toolchain to verify compatibility.
  - Test the toolchain on controlled scenario simulating a real environment.

We plan to measure component characteristics, and compute metrics and KPIs according to the Table 3.2, and so contribute to evaluate the achievement of the project objective O3.

Table 3.2: KPI and metric of the component WP5-02-IKER

№	KPI	#	Metric	MO
1	Latency	1	Latency in the network (milliseconds)	MO3.3
2	Speed	1	Attack Detection time (seconds)	MO3.3
2	Speed	2	Update detection time (minutes)	MO3.3
3	Compatibility	1	Compatible standards	MO3.3

№ = Component wide KPI number KPI = KPI description
# = KPI wide metric number Metric = Metric description
MO = Measurable Outcome (see Table 1.5) that the metric refers to and refines

3.2 Safe fleet communications (WP5-03-SCALIAN)

The Table 3.3 provides the synopsis of the component WP5-03-SCALIAN.

Table 3.3: Component WP5-03-SCALIAN

Identifier: WP5-03-SCALIAN	**Partner:** SCALIAN	**Expected TRL:** TRL6
Name: Safe fleet communication		
License: Proprietary	**Owner:** SCALIAN	**Contact:** SCALIAN
Expected Outcome: Communication routing module (Software)		
Description: SCALIAN has worked on developing a generic architecture to allow fleet of UAVs or miscellaneous agents to perform a variety of missions. This architecture is composed of several components and a Knowledge Base whose role is to store information on the mission status and UAVs status. In a system like this, the communication system is very important to provide all the information of each agent to each other. This module is the abstraction of the communication to ensure that the agents KB are correctly synchronized.		
Satisfied Requirement: IDENTIFIER — DEFINITION • UC3-INT-003 — The system must provide a redundant, robust and secure communication so all the agents can have access to the shared knowledge base. • UC3-OPR-004 — The communication devices and their infrastructure shall be deployable in less than a day.		
Key Enabling Technology: KET — CATEGORY • KET: Security — CATEGORY: U-Space Capabilities • KET: Vehicle to Vehicle communication — CATEGORY: System Functions • KET: Vehicle to Infrastructure communication (V2I) — CATEGORY: System Functions • KET: Swarm formation and cooperation — CATEGORY: System Functions • KET: Network Centric Communications Systems — CATEGORY: System Functions		
Improvement: • Be compatible with a fleet of 25 to 50 agents. It should support the data communication, with notion of message priority for critical communication. • Permit to operate on “large” areas. • Offer on-demand video-stream from UAV without using a dedicated streaming device. • Be quickly deployable on the field and with an autonomous configuration of the network		
Contributor: SCALIAN	**Task:** T5.1	
Use Case: • UC3 — Logistics **Demonstrator:** • Deployment of an autonomous communication system in hard-to-access areas thanks to a highly automated multi-vehicles system (Metis project)		

3.2.1 Architecture context and interfaces

SCALIAN has worked on developing a generic architecture to allow fleet of UAVs or miscellaneous agents to perform a variety of missions. The genericity allows to use heterogeneous UAVs/agents and to allocate them different missions. For instance, a fleet of UAVs has performed logistic operations by dropping sensors over a huge area collaboratively, or a fleet of 4 UAVs have swipe scanned areas to build an orthophoto map.

Inside the architecture there are several components (from low level like control, up to high-level like task planning) and a Knowledge Base (KB). It is the component responsible for maintaining a shared mission status inside the system. Agents connect to it and are then able to get and update the mission status, but also share their own status.

The system supports several types of agents: various kind of heterogeneous UAVs, GCS and “Command and Control” agents. The UAVs use the mission status to plan their mission and report their status. The GCS and the “Command and Control” use the Knowledge Base to display the mission progress to the operators and allow them to send orders to the UAVs (e.g. return to base, emergency stop).

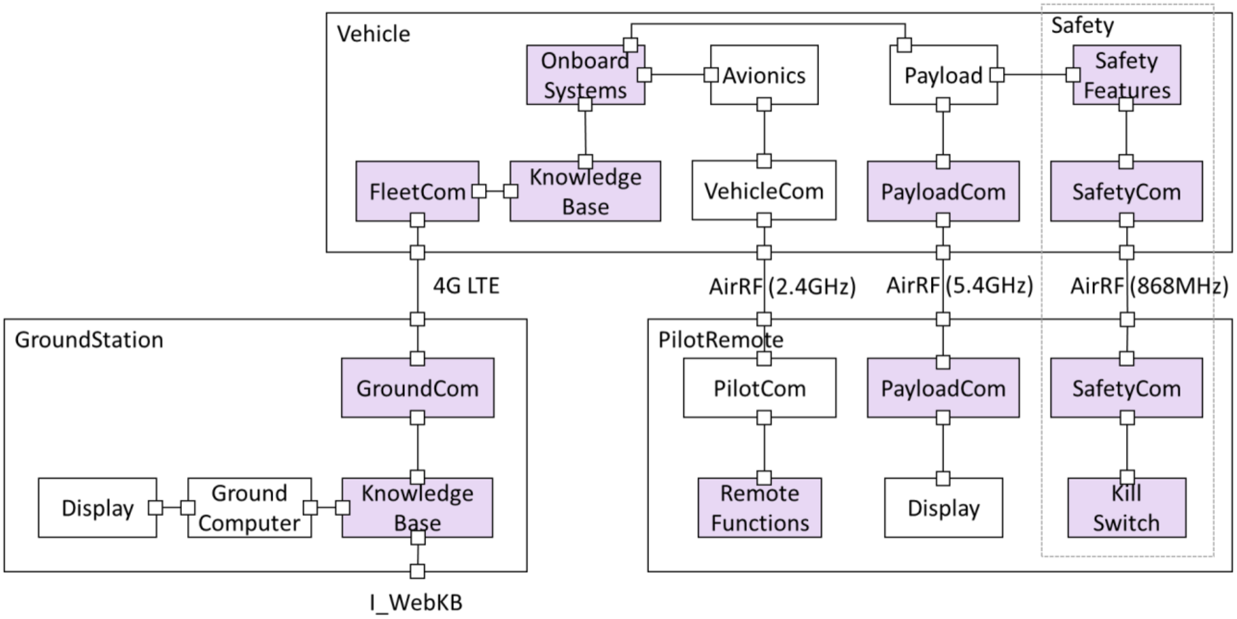


Figure 3.2: The UAVs use the data inside the KB to compute plans and decisions

In Figure 3.2 we present the components of our architecture according to the architecture blueprint presented in Figure 1.1. The components we add are highlighted in light purple. The different UAVs are autonomous and use the Knowledge Base to plan their next tasks. They rely on the FleetCom component to share their status and progress allowing the rest of the fleet to take them into account.

A pilot remote controller is represented as an independent block. The pilot is in the system only for safety and regulatory reasons. The UAVs never receives commands from their remote, unless the chain of command asks the pilot to remove their designated UAV from the fleet and to take control over it. This is used only when the system is compromised at a very high level, usually most problems are autonomously handled by the system. The pilot receives the payload camera feed so the operations can be validated. In the long term this human validation will be removed but aviation authorities forbid fully autonomous dropping operations for now.

Finally, due to the size of the UAVs and the criticality of the infrastructures in the operation areas we devised a safety Kill Switch component: it shuts down the motors and triggers a parachute. Again, a chain of command must order the pilot to take this action.

In a system like this, the communication system is very important to provide all the information of each agent to each other. It ensures the synchronization of the Knowledge Base of each agent. For future use, an interface to the Knowledge Base offers an access to its data from the web.

On the hardware part of the communication system, SCALIAN has decided to use a 4G private network. On the software part, a middleware named DDS (Data Distribution Service) is used to ensure a certain Quality of Service. In addition of DDS, a File Transfer system is using an FTP-based communication to share big files with the fleet and the GCS (e.g. digital elevation model or camera picture).

3.2.2 Internals and technologies

When we deploy a fleet of UAVs, we use a private 4G network as the physical communication layer. It is totally independent and does not rely on a third-party network provider. The network architecture is centralized on the main 4G antenna.

Each agent of the fleet has a 4G modem and is configured to automatically connect to the fleet network.

Most of time, the GCS computer is connected to the 4G antenna through an Ethernet cable, in order to save 4G bandwidth and to reduce the latency.

In our software controller named “EZ_Chains”, the communication is managed by a dedicated module using the DDS network middleware. This middleware simplifies complex network programming. It implements a publish–subscribe pattern for sending and receiving data, events, and commands among the nodes. DDS allows the user to specify Quality of Service (QoS) parameters to configure discovery and behaviour mechanisms up-front.

In our communication software DDS is mostly used for:

- Reliability mechanism (a message can’t get lost).
- Autonomous discovery of publishers and subscribers, to make the different participants of the fleet meet themselves.
- When joining the fleet network, the communication system automatically rebuilds the mission status and history in the Knowledge Base.
- After a communication disrupt, all the missed messages are retrieved in order to keep the Knowledge Base synchronized.

Based on these technologies, the communication system is responsible for the synchronization of the Knowledge Base between all the agents. They are two types of communication:

- periodical communication for data needed continuously (position, heartbeat);
- transaction-based communication.

The transaction system is a centralized system (i.e. client-server architecture) that allows any kind of agent in the fleet (including the GCS) to ask for the validation of a transaction query and to broadcast its effect to all the agents.

A transaction is a generic concept for any kind of event message or modification to do in the knowledge base. For example, the GCS can send a transaction query to create a new *geocage* or to order a drone to return to home. A drone can send a transaction query to declare and book a flight path.

3.2.3 Component roadmap

3.2.3.1 Results at M10

Thanks to the private 4G network and our software communication module, the system is quickly deployable in the field (the operational team needs to deploy only the 4G antenna). It has been tested many times in Hardware-In-the-Loop (HIL) simulations and in real operations with a fleet of 6 UAVs. The 4G offers up to 4 Mb/s (for the whole fleet) at 1.5 km that is enough for the datalink and some pictures for 6 UAVs via the FTP.

For the future evolutions, we will use the results obtained for this fleet and network configurations. Indeed, our communication system is already at TRL6, but relies on a private 4G LTE network and a dedicated mean of communication for video feedback. During C4D we will investigate how to change the communication mean and merge the uses. This investigation will start by several Proof-of-Concept systems, will then be implemented and finally tested until it reaches TRL6 again.

During our tests we experienced some troubles with the 4G throughput which dramatically decreases with the distance between the fleet and the 4G antenna. Due to the limited throughput of our private 4G network, a dedicated FPV (First Person View) streamer module had to be used to send the video stream to the pilots, instead of using the 4G network. In the future, our objective is to deploy fully autonomous fleets, without pilot. Thus, the need of throughput for the video will be smaller.

The software part of the communication system has been approved in simulation for 12 UAVs. Minor changes should answer the needs for a fleet of a 25-50 UAVs.

In order to dispense with our private 4G network we performed some experiments with the public 4G networks. The advantage is that we don’t need to transport and setup our own 4G antenna, and that we may cover a larger area by using all the different available antennas around the mission area. Also, most of public 4G providers generally offer a better throughput than our private 4G network, but there is no guarantee of service.

Using public 4G networks also means that we could benefit of the internet network which permit to offer new features and services, but it is also a big threat for the fleet (cybersecurity). Being connected to internet will enable the UAVs to share their status and data to the different end users, wherever they are. It also permits to the UAVs to download new mission data or software updates.

To use these public 4G networks, it is mandatory to setup a Virtual Private Network (VPN) in order to make the fleet communicating. Our experiments were based on the StrongSwan VPN software that we deployed on a virtual dedicated server hosted on internet. The results were mitigated, and we have to do more experiments to improve the solution. Indeed, the network latency was much higher using the VPN layer (150 ms to 200 ms) than the private 4G network (50 ms). Also, the software configuration of the VPN network was hard to maintain and was not adapted to a fleet composed of a dynamic number of agents.

In the meantime, we started to perform a state of the art of the existing communication means in order to choose the most adapted technology according to the different projects and missions.

In a long-term view, the communication system for SCALIAN should:

- Be quickly deployable on the field and with an autonomous network configuration.
- Provide fleet monitoring and data sharing to end-users connected through internet.
- Be compatible with a fleet of 25 to 50 agents. It should support the data communication, with notion of message priority for critical communication.
- Permit to operate on “large” areas.
- Offer on-demand video-stream from UAV without using a dedicated streaming device.

3.2.3.2 Plans for the year 2 and the year 3

Today the system is at a state of TRL 6. But as presented above, we aim at developing a new communication mean from the ground up. We shall start with proof-of-concept tests and finish with a validated system reliable enough to ensure safe operations. Henceforth we need to reach TRL6 level after all our exploratory, development and test work.

For the year 2, SCALIAN wants to integrate and test different technologies:

- Use the public 4G network to deploy the fleet on a VPN and permit to monitor the fleet and mission from internet.
- New physical support, such as Wi-Fi Long Range.
- A mesh network, with UAVs acting as relay to cover more surface and block spot.

For all these technologies, will be tested:

- the throughput and bandwidth;
- the quality of service;
- the network covering area;
- the ease to deploy the solution on the field.

For the year 3, the goal is to validate the use of one technology and to take it for TRL 5-6 with further tests on a fleet and further development for integration.

We plan to measure component characteristics, and compute metrics and KPIs according to the Table 3.4, and so to contribute to evaluate the achievement of the project objectives O1 and O3.

Table 3.4: KPI and metric of the component WP5-03-SCALIAN

№	KPI	#	Metric	MO
1	Ease of integration	1	Required volume to store the communication hardware in a vehicle/warehouse (excluded the modules embedded in each agent).	MO1.1 MO1.3
1	Ease of integration	2	Weight of the communication hardware (excluded the modules embedded in each agent).	MO1.1 MO1.3
1	Ease of integration	3	Time to physically setup the communication means on a new area of operations.	MO1.1 MO1.3
1	Ease of integration	4	Time to configure the system to integrate a new agent to the fleet network configuration (e.g. modification of a configuration files, generation of certificates).	MO1.1 MO1.3 MO3.3
2	Fleet network performance	1	Maximal number of agents on the fleet network.	MO3.1 MO3.2
2	Fleet network performance	2	Mean latency from an agent to another agent (e.g. UAV to GCS).	MO3.1 MO3.2
3	Operating range	1	Maximal range for an agent datalink without video stream.	MO3.1 MO3.2
3	Operating range	2	Maximal range for an agent datalink streaming a video of a least 720p@20fps.	MO3.1 MO3.2
4	Users from internet	1	Refresh rate of the agent status (position, state, health) during a live mission.	MO1.1 MO3.3
4	Users from internet	2	Latency of the update of the agent status (position, state, health) during a live mission.	MO1.1 MO3.3
4	Users from internet	3	Maximal number of users from internet monitoring a live mission.	MO1.1 MO3.3

№ = Component wide KPI number KPI = KPI description
# = KPI wide metric number Metric = Metric description
MO = Measurable Outcome (see Table 1.5) that the metric refers to and refines

3.3 Adaptive video coding and compression (WP5-06-AI)

The Table 3.5 provides the synopsis of the component WP5-06-AI.

Table 3.5: Component WP5-06-AI

Identifier: WP5-06-AI	**Partner:** AI	**Expected TRL:** 4
Name: Adaptive video coding and compression		
License: Proprietary	**Owner:** Aitek	**Contact:** sdelucchi@aitek.it
Description: Aitek is working on a methodology to control video compression frame rate and downsampling according to communication channel bandwidth available to provide the best possible imagery at any given system status and to achieve a higher overall system flexibility.		
Improvement: Proposed video compression and coding solution is characterized by the following strong points • It is particularly effective for bandwidth demanding application, like HD video flow transmission. • It is largely interoperable because it enables to considers lower layers as black box, working only at the application layer. • It is adaptive with respect to bandwidth reduction due to poor channel conditions and/or interference.		
Contributor: AI	**Task:** T5.1	
Use Case: • UC5 — Agriculture **Demonstrator:** • Crop monitoring		

3.3.1 Architecture context and interfaces

The diagram in Figure 3.3 shows the logical architecture for the acquisition and the transmission systems.

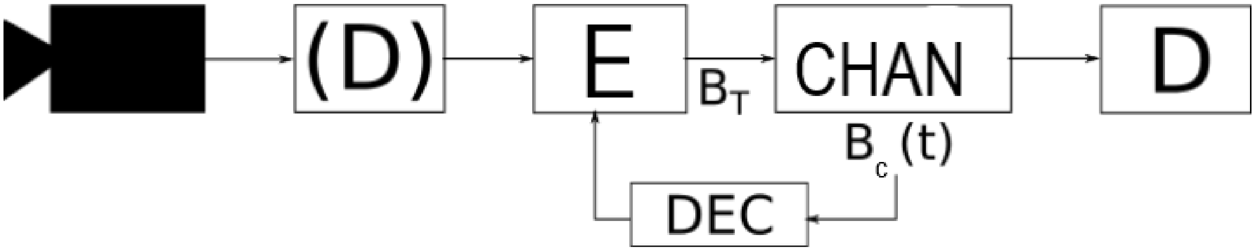


Figure 3.3: Logical architecture for the acquisition and the transmission systems

The components are as follows, from left to right:

- IP cameras.
- Decoder (D), used whenever the IP camera acquires encoded video transmissions, in order to have RAW images.
- Encoder (E), used to encode video transmission, given a suitable threshold of compression.
- Decision maker (DEC), this module is in charge for performing two main functions: measure/estimate the channel bandwidth or their variations and pilot the encoding operation accordingly by means of applying the video coding and compression configuration algorithms.
- Channel (CHAN), the channel used by the IP Camera to send the video stream
- Transmission Bandwidth (Bt).
- Channel Bandwidth over time (Bc(t)).

3.3.2 Internals and technologies

During the project Aitek will evaluate different methodologies to reduce video transmission bandwidth in order to reach optimal values for the transmitting channel.

1) One type is based on direct manipulation of the bandwidth output from the encoder by setting the desired value which will be obviously lower than the channel bandwidth but as close to it as possible. The encoder acts autonomously by reducing the number of bits used to represent the frame thus to reduce its quality. In other words, this kind of algorithm fixes the transmission bandwidth and the encoder sets the image quality accordingly.

2) Another type of solution to manage compression related to channel bandwidth acts on encoding parameters which affect the transmission bandwidth and thus to define the quality of the video stream in order to be able to set the encoder accordingly. These parameters are Frame rate per second (FPS) and the frame resolution, which heavily impact the VCA (Video Content Analysis) performance and the quality of the displayable video. The methodologies that are belong to this group are the following:

    - Resolution reduction per each frame. The Viewer has to update the video resolution before reproduction. When tuning this parameter, a minimum threshold value has to be considered to be able to correctly display some details within the frame.

    - Reduction of N frames per second. If the frame rate is too low, the video stream can experience a scattered playback. When setting this parameter, the constraint imposed on the minimum quality must be considered in relation to video fluidity.

3) The last group of methodologies that imply bandwidth reduction are related to other encoder elements, such as the periodicity of the "intra" type frames and the coding standard used (this kind of image manipulation technique analyses a single frame and looks for similar zones in order to reproduce the picture with less data thus to eliminate bit redundancy). More specifically:

    - Reduction of the “intra” frame frequency compared to “inter” frame which is not usable when applying MJPEG encoding (unlike the above described, this encoding technique are based on algorithms which analyse a sequence of N frames in order to identify changes and thus to eliminate pixel redundancy).

    - Transmitting encoder variation. Different encoders determine different compression levels and thus different transmission bandwidth. This technique is not applied in real time but in a transparent way to the end user. However, it is necessary to wait for an intra frame, after which the new encoding is activated and the previous one is ended. When receiving, once the decoder detects the change of standard, updates and uses more suitable flow decoding standard; the lower the intra frame frequency the lower is the bandwidth occupation but the higher is the time required in order to be able to apply an encoder switch. It is therefore useful to find a compromise between these two necessities or otherwise to apply this technique separately from variation of “intra” frame frequency. The following standards are applied, sorted by achievable compression level:

        a) H265
        b) H264
        c) VP8
        d) H263
        e) MPEG4
        f) MPEG2

In order to apply the above-mentioned techniques, it is necessary to investigate solutions in order to estimate the bandwidth usage and to set accordingly the encoding and compression parameters.

As regard the bandwidth usage, algorithms could be applied in order to verify if the available bandwidth is sufficient to transmit the video stream. These kinds of algorithms do not perform an absolute bandwidth estimate but rather a dynamic comparison of the quantity available on the channel in relation to the one that is needed to transmit the video stream- Therefore the procedure is based on the analysis of the characteristics of the video stream and of the transmitter - receiver pair. It is possible to perform an assessment of the estimation methods based on their functionalities:

- *Detection method, active or passive*:
  - Active: estimation performed by sending small quantities of data in the channel and thus obtaining the necessary information, analysing the dynamics of propagation and data reception
  - Passive: the information is collected by analysing the traffic regularly transmitted on the channel, without adding new ones.
- *Presence or absence of feedback within the measuring system:*
  - With feedback: the source estimates the channel bandwidth, by analysing the information received from the destination
  - Without feedback: the source retrieves the required information locally without any data from the destination
- *Measured quantity:*
  - Channel bandwidth measurement
  - Measurement of the channel bandwidth variations

As regard the algorithms to automatically set the encoding and compression parameters, an example of methodology is described whether the scope is to increase or decrease the bandwidth of the video stream by respectively increasing or decreasing the encoded stream quality. Based on the video content, it is possible to distinguish between two possible scenarios:

- A Scenario: poorly dynamic with small and distant targets that are moving at low speed.
- B Scenario: highly dynamic with fairly average targets that are moving at high speed.

More specifically both the VCA and the operator that is monitoring the video stream are more sensible to resolution in A scenario (even if the dynamics of the scenes are limited, the small targets make it is necessary to prioritize resolution over frame rate in order to be able to detect the required information) or frame rate in B scenarios (with higher frame rates a higher resolution is preferable).

It is therefore necessary to choose the best parameter onto act on by considering the video type and the kind of scenario that has to be monitored. The Table 3.6 provides a summary of actions based on the above described algorithms.

Table 3.6: Estimation of Channel Bandwidth Algorithms based on defined scenario

Channel Bandwidth estimator	Scenario	Primary Action	Secondary Action
Necessity to reduce bandwidth	Scenario A	FPS reduction	Resolution reduction
	Scenario B	Resolution reduction	FPS reduction
Chance to increase the bandwidth	Scenario A	Resolution increase	FPS increase
	Scenario B	FPS increase	Resolution increase

3.3.3 Component roadmap

3.3.3.1 Results at M10

Preliminary study of the scientific literature related to video coding SOTA techniques that could be applied within the project operating context.

3.3.3.2 Plans for the year 2 and the year 3

M12-M24: adaptive video coding and compression algorithms analysis and definition.

M25-M36: Final Report and guidelines for the reference UCs.

Moreover, this contribution is not related to any component implementation and thus neither integration nor testing activities are planned for theoretical and methodological guidelines purposes.

3.4 GPS Spoofing Detection Module (WP5-07-MODIS)

The Table 3.7 provides the synopsis of the component WP5-07-MODIS.

Table 3.7: Component WP5-07-MODIS

Identifier: WP5-07-MODIS	**Partner:** MODIS	**Expected TRL:** TRL4
Name: Detection of navigation system failures due to signal hijacking or system malfunction.		
License: Proprietary	**Owner:** Modis	**Contact:** Modis
Expected Outcome: New embedded autonomous management system for drones, backed with artificial intelligence and machine learning capabilities		
Description: Navigation system failures due to signal hijacking or system malfunction detection and reaction. Watchdog with emergency operation capabilities. The component WP5-07-MODIS will be designed as an embedded software module that will rely on commands received from the platform, geographical position derived from geomagnetic D-SLAM, GPS position and mission polygon to detect possible GPS signal hijacking/spoofing.		
Satisfied Requirement: IDENTIFIER — DEFINITION • UC5-DEM1-WP5-FNC-01 — Reliable GPS Messages: Detection of navigation system failures due to GPS signal hijacking or system malfunction. • UC5-DEM1-WP5-FNC-01 — Watchdog module: reaction to GPS Signal hijacking (e.g. start/enable geomagnetic based D-SLAM).		
Key Enabling Technology: KET — CATEGORY • KET: Security — U-Space Capabilities • KET: Communication, Navigation and Surveillance — CATEGORY: U-Space Capabilities • KET: Intelligent Vehicle System Monitoring — CATEGORY: System Functions • KET: Network Centric Communications Systems — CATEGORY: System Functions • KET: Over the Horizon Communications — CATEGORY: System Functions		
Improvement: AI-based algorithms to detect GPS-Spoofing attacks will increase drone availability.		
Contributor: Modis	**Task:** T5.4	
Use Case: • UC5 — Agriculture **Demonstrator:** • D1 — Crop monitoring		

3.4.1 Architecture context and interfaces

The Figure 3.4 depicts the architecture context of the component WP5-07-MODIS.

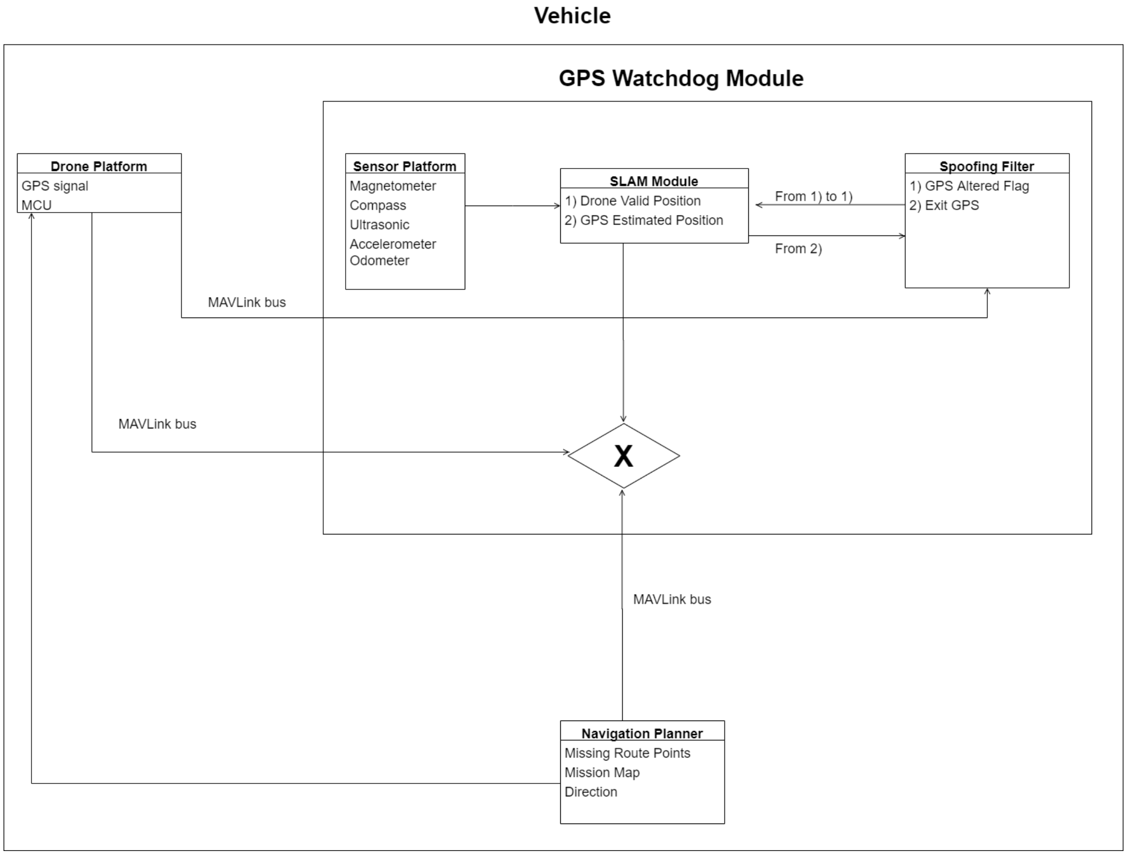


Figure 3.4: Architecture context of the component WP5-07-MODIS

3.4.2 Internals and technologies

Functional Description: The core of the provided component will be a system for spoofing detection of the GPS signal. The detector will rely on machine learning and signal processing algorithms. Specifically, the system will implement the following stylized pipeline:

1) Acquisition of the GPS signal.
2) Feature extraction.
3) Detection/Classification.
4) Watchdog.

The GPS signal is acquired by the GPS receiver, framed in time series of fixed length and treated by appropriate pre-processing before extracting the relevant features. As for the feature extraction and the detection, a range of solutions will be explored.

Feature-Based Supervised Learning: Classical supervised learning approaches (like SVMs and Neural Networks) require appropriate representations for the items to be classified. Such representations are given usually in terms of features vectors. Features are quantities measured on the object to classify and their choice heavily impact the classification performances. For spoofing detection, we plan to investigate the adoption of signal-processing-theoretic features like autocorrelation, amplitude, phase and so on. These features are already been explored in a recent work [22] and seem to allow for efficient collection and good classification performances.

Potential shortcomings of this approach include the high demand for labelled spoofing attacks (which might be difficult to collect) and the need of designing additional features to captures the rich pattern of abnormal behaviours. As an alternative for real data, several synthetic datasets have been provided in the literature [26] [27] [28], although their representativeness needs to be evaluated.

Time Series Classification: Time-series classification approaches alleviate the system designer from the need of carefully designing a good set of features, as the GPS signal is classified directly after the pre-processing [23]. These approaches have been successfully employed in other signal classification applications. The rough idea is to think about the input space as the set of all possible sequence of a fixed maximum length and define a (pseudo)-metric on it. Such a metric will define similarities between inputs: signals which are close (far) according to the metric, will be considered similar (dissimilar). Clearly, the choice of the metric will play a similar role to the choice of the features for classical supervised learning. A popular choice which has been proved to be effective in application the Dynamic-Time-Warping (DTW) which given two signals, roughly looks for their maximum matching alignment and then take Euclidean difference. Once a metric is chosen, the classification task can be performed with a mellow learner like K-NN which has guaranteed near-optimal performance in such dimensionless scenario.

Potential shortcomings of this approach include (1) the inefficiency of computing DTW among the object to classify and each pair of signals in the training data which involve running a slow dynamic programming algorithm for signal alignment, (2) the demand for labelled data.

Outlier Detection: In order to bypass the high demand of labelled examples, unsupervised approaches can be adopted. Here the idea is to rely only on normal examples to build a model for the regular behaviours [24]. There are essentially two main approaches to outlier detection: model based and clustering algorithms. The former assumes the normal data obeys a predefined distribution and fits the parameters of this distribution to the data. Once a model has been learnt, for each signal to classify its probability is computed and if below a given threshold it is classified as outlier. The latter are purely algorithmic and build clusters of the given data. Once a new signal needs to be classified, if it does not conform to any of the clusters, it is classified as outlier. Both the approaches can be feature-based or sequence-based.

Potential shortcomings of these approaches are the presence of one or more parameters to validate without labelled data and hardness in assessing their performances.

Regardless of the specific approach to the detection problem, the functional architecture of the system will follow the diagram in the Figure 3.5, where the Feature Extraction block is optional.

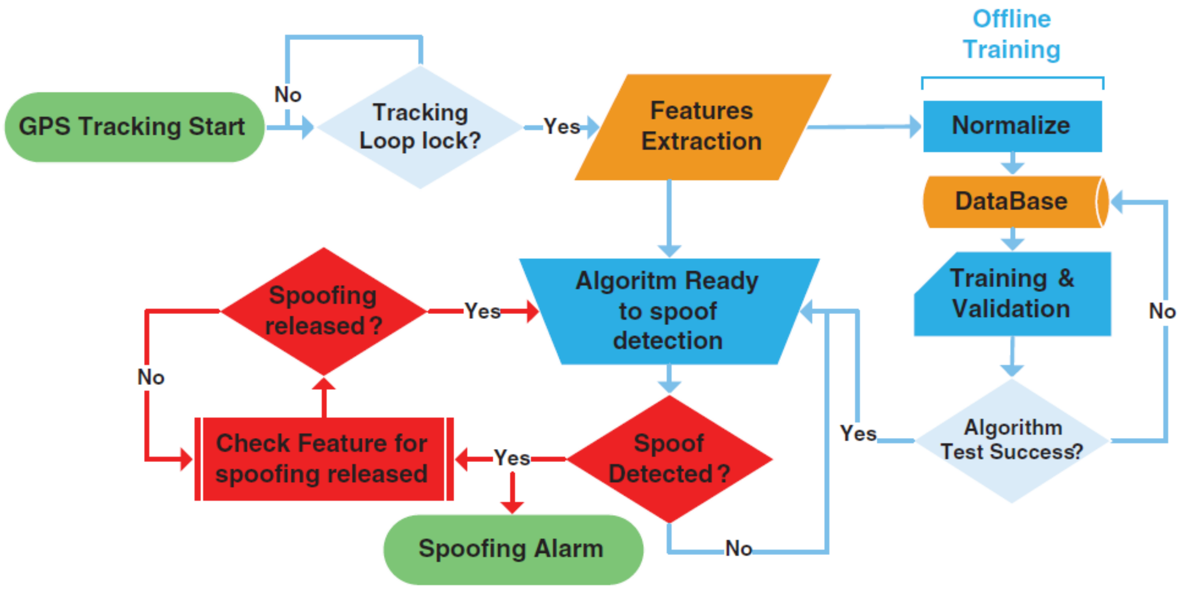


Figure 3.5: System architecture (from [22])

3.4.3 Component roadmap

3.4.3.1 Results at M10

Reviews of state-of-art approaches to the detection of spoofing attacks for GPS signals. High-level design of the component in terms of fundamentals building blocks. Pre-processing of existing datasets for training Machine Learning algorithms and evaluation.

3.4.3.2 Feedback to the other WPs

Feedback will be given to WP1 specifically for UC5 – Demonstrator 1. Moreover, there will be relation with WP3 and WP4 where HW/SW architectures for drones are defined and components for providing the drone of autonomous decision capabilities are designed. For example, results from this component will be used along with the SLAM module designed in WP4 to allow GPS free navigation: even if the GPS signal is attached or is lost due to natural interference, the drone will continue with its mission. Similarly, depending of the final algorithmic solution, some specific GPS receivers might be needed, impacting the architecture proposed in WP3.

3.4.3.3 Plans for the year 2 and the year 3

M12-M24: We plan to start the development of the component up to its first prototype. We also plan a preliminary experimental evaluation aimed at evaluating the correctness of the component.

M25-M36: We plan extensive experiments aimed at validating the component behaviour in more realistic scenarios. We will also design a number of alternative solutions in case modification or improvements will be needed. Finally, we plan a final embedded deployment for the use case.

We plan to measure component characteristics, and compute metrics and KPIs according to the Table 3.8, and so contribute to evaluate the achievement of the project objective O3.

Table 3.8: KPI and metric of the component WP5-07-MODIS

№	KPI	#	Metric	MO
1	Increase of drone availability	1	Increased Mean Time Between Failures	MO3.3
2	Spoofing detection capabilities	1	Attack classification accuracy	MO3.3

№ = Component wide KPI number
KPI = KPI description
# = KPI wide metric number
Metric = Metric description
MO = Measurable Outcome (see Table 1.5) that the metric refers to and refines

3.5 Autonomic management framework (WP5-17-FB)

WP5-17-FB component is a Generic Autonomic Management Framework.

Table 3.9: Component WP5-17-FB

Identifier: WP5-17-FB	**Partner:** FB	**Expected TRL:** TRL4
Name: Generic Autonomic Management Framework		
License: Open source	**Owner:** FB	**Contact:** FB
Description: The Generic Autonomic Management Framework (GAMF) will be used for developing autonomic managers for smart and secure drone-based applications in a vineyard management. This will be based on an extension of existing work with domain specific aspects.		
Satisfied Requirement: IDENTIFIER — DEFINITION • DEM9-INT-05 — The self-adaptability framework shall be generic • DEM9-INT-06 — The self-adaptability framework shall provide generic control mechanisms • DEM9-INT-07 — The self-adaptability framework shall provide a set of interaction interfaces • DEM9-INT-08 — The self-adaptability framework shall provide system adapters • DEM9-PRF-02 — The self-adaptability framework should be lightweight • DEM9-USB-01 — The self-adaptability framework should be open source		
Key Enabling Technology: KET — CATEGORY • KET: Autonomic Computing based on MAPE-K feedback loop — CATEGORY: System Functions		
Improvement: GAMF provides generic control mechanisms based on the autonomic control loop (MAPE-K) and a set of interfaces to allow the interaction between control mechanism and system specific management components, the system adapters. We will extend it to support domain specific aspects.		
Contributor: FB	**Task:** T5.3	
Use Case: • UC5 — Agriculture **Demonstrator:** • D2 — Wine production		

3.5.1 Architecture context and interfaces

The Generic Autonomic Management Framework (GAMF) architecture is a Java-based framework used to develop autonomic managers for any target system without having to (re)implement the generic control mechanisms. GAMF provides generic control mechanisms based on the autonomic control loop (MAPE-K) and a set of interfaces to allow the interaction between control mechanism and system specific management components, the system adapters. System adapters include event generators and effectors, which allow interaction of the control mechanism with the target system, as well as metric extractors and policy evaluators, which provide the means for computing a specific response determined by policies to an observed situation modelled by metrics. The information about how a specific system adapter is triggered is held in the system adapters registry.

The Figure 3.6 depicts the architecture context of the component WP5-17-FB.

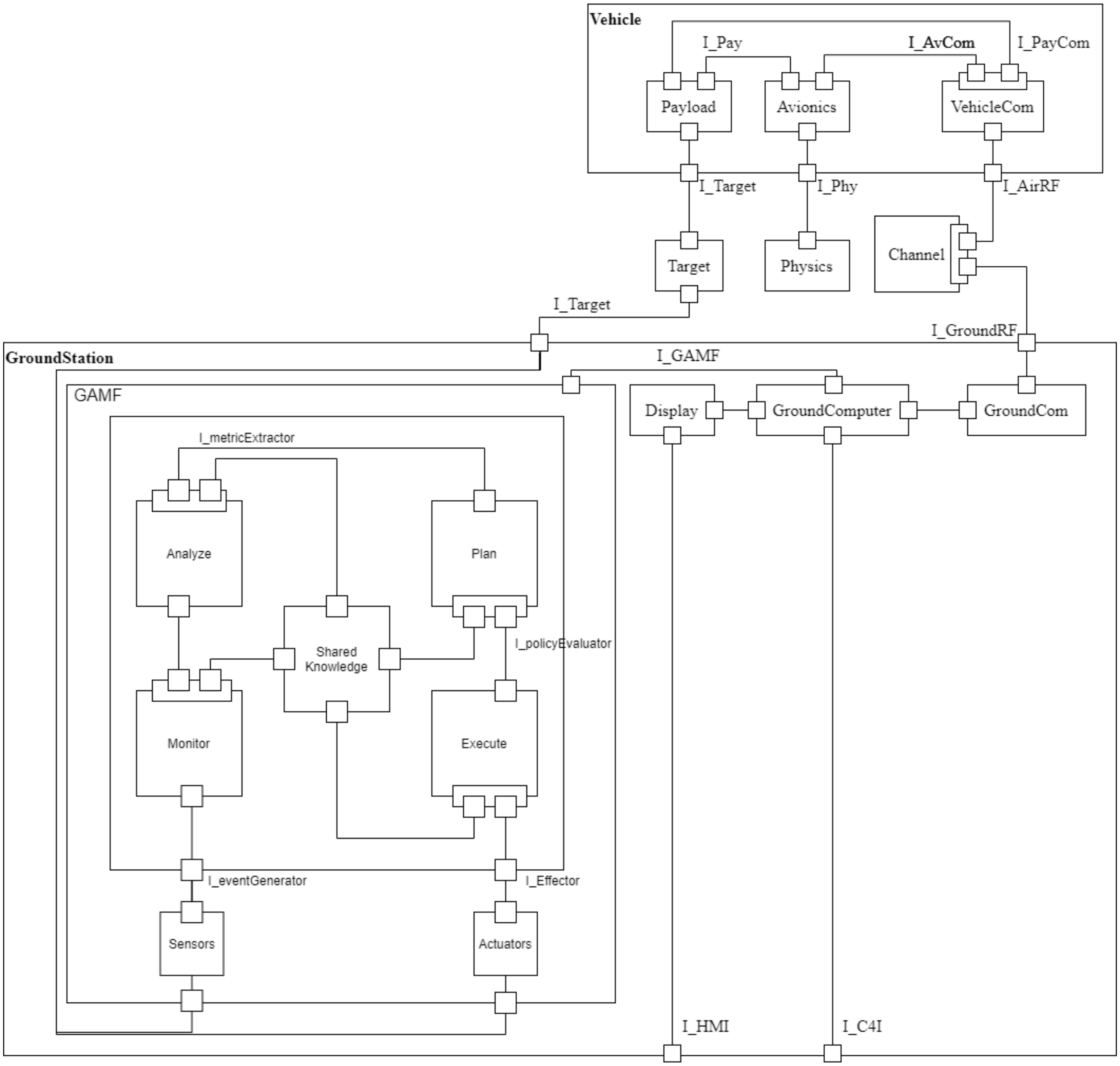


Figure 3.6: Architecture context of the component WP5-17-FB

3.5.2 Internals and technologies

Autonomic Computing: GAMF is a Java-based library with fundamental autonomic management functions and very little impact on the target system's runtime.

Service Oriented Architecture: From a Service Oriented Architecture perspective, Generic Autonomic Management is designed as a component-based REST service that can be invoked by different SOA-based frameworks without requiring a high adjustment effort [29]. Additionally, given its generic property, each component of the autonomic control loop has abstract interfaces that can be used by a number of application systems. This would reduce the software engineering effort since there is no need to (re)implement the generic control mechanisms for different application systems, only to properly define events, metrics and adaptation policies.

Monitor: The Monitor component constantly collects monitoring data from the sensor. The component performs a pre-analysis based on the incoming sensor data and context data stored in the SharedKnowledge. In case there is a significant delta an event is generated.

```
monitor[serviceID]
getSensorData(sensorData)
preAnalysis()
generateEvent()
updateSharedKnowledge()
```

Analyze: The Analyze component evaluates the events received from the Monitor component with respect to the requirements and context data in the SharedKnowledge. If the requirements cannot be satisfied a change request including a description of the metrics is send to the Plan component.

```
anayze[serviceID]
getRequired(require)
getContext(context)
getEvent(event)
extractMetric()
updateSharedKnowledge()
```

Plan: The Plan component is able to understand the metrics received from the Analyze component and to derive adaptation policies.

```
plan[serviceID]
getMetric(metric)
addResource()
releaseResource()
updateSharedKnowledge()
```

Execute: The Execute component receives the policies from the Plan component and executes the derived action via the GenericAutonomicManagement service.

```
execute[serviceID]
getPolicy(policy)
invokeNextAction()
updateSharedKnowledge()
effectorAdd[serviceID]
effectorRelease[serviceID]
```

In case the system cannot find a suitable adaptation solution (e.g. no additional resource is available) a user intervention is required to handle the issue.

3.5.3 Component roadmap

3.5.3.1 Results at M10

We have:

- investigated standards and best practice guidelines to extract controls for secure communication based on identified threats and vulnerabilities;
- discussed possible adaptation policies specific to the use case.

3.5.3.2 Feedback to the other WPs

We will provide feedback to WP1 – Use Case 5 (Agriculture) – Demo 2.

3.5.3.3 Plans for the year 2 and the year 3

In the second and third year of the project, in the course of WP5 FB will focus on developing autonomic managers for smart and secure drone-based applications in a vineyard management.

Additionally, FB will contribute its results in the following deliverables:

- M22: D5.5 APIs for Trusted Communication – first version (Lead: IFAT);
- M30: D5.6 APIs for Trusted Communication – final version (Lead: IFAT).

As previously defined, the targeted TRL of this component is TRL4 and it will be validated in Use Case 5 (Agriculture) Demonstrator 2 (Wine production).

We plan to measure component characteristics, and compute metrics and KPIs according to the Table 3.10, and so contribute to evaluate the achievement of the project objective O3.

Table 3.10: KPI and metric of the component WP5-17-FB

№	KPI	#	Metric	MO
1	Improve the trade-off between e.g. resource usage and security using autonomic elements	1	Number of autonomic elements in the field devices	MO3.3
1	Improve the trade-off between e.g. resource usage and security using autonomic elements	2	Number of autonomic elements in the backend	MO3.3
1	Improve the trade-off between e.g. resource usage and security using autonomic elements	3	Number of nested autonomic elements	MO3.3

№ = Component wide KPI number KPI = KPI description
# = KPI wide metric number Metric = Metric description
MO = Measurable Outcome (see Table 1.5) that the metric refers to and refines

3.6 Distributed Intrusion Detection System (WP5-01-CEA)

Despite the deployment of several security measures to prevent attacks, an attacker will always find a way in. That is why it is also important to enforce the security via reactive security mechanisms able to monitor and react against tentative penetration while the system is running, if a preventive measure failed or parts of the system is not enough protected. WP5-01-CEA is an Intrusion Detection system which provides real-time detection based on the network flow analysis.

Table 3.11 provides the synopsis of the component WP5-01-CEA.

Table 3.11: Component WP5-01-CEA

Identifier: WP5-01-CEA	**Partner:** CEA	**Expected TRL:** TRL4
Name: Distributed Intrusion Detection System with in-drone Machine Learning –based probes detection		
License: Proprietary	**Owner:** CEA List	**Contact:** baptiste.polve@cea.fr
Description: This component will provide a lightweight anomaly-based Intrusion Detection System (IDS) for drones. It will work on network traffic patterns and on carried data plausibility, for both drone to drone and drone to ground central station links. When possible, the IDS will extract information on the detected attacks to notify the experts and might propose some countermeasures if the feature is made available.		
Satisfied Requirement: IDENTIFIER — DEFINITION • UC3-DEM2-SEC-xx — The cybersecurity component should be able to detect both common and unknown cybersecurity attacks in real-time.		

- UC3-DEM2-SEC-xx — The cybersecurity component should provide list of actions to perform to mitigate a detected attack (and/or optionally perform the action by itself if the feature is integrated).

Key Enabling Technology: KET — CATEGORY

- KET: Security — CATEGORY: U-Space Capabilities
- KET: Communication, Navigation and Surveillance — CATEGORY: U-Space Capabilities
- KET: Intelligent Vehicle System Monitoring — CATEGORY: System Functions
- KET: Network Centric Communications Systems — CATEGORY: System Functions
- KET: Over the Horizon Communications — CATEGORY: System Functions

Improvement: The challenge is to being able to perform anomaly-detection, a technology able to detect known and unknown attacks on embedded systems such as drones or droids. The goal is to strengthen the security with a second line of defense in addition to the traditional preventive measures. Moreover, it will be important to make efforts on reducing the time required to make the system runnable. Indeed, as every machine-learning based system, the IDS requires a learning phase. A particular attention will be paid on tailorizing the IDS to drone- and droid-specific communication protocols as well as on the understanding of ongoing attacks.

Contributor: CEA	**Task:** T5.4

Use Case:

- UC3 — Logistics

Demonstrator:

- D2 — Logistics in 5G urban environment: clinical sample delivery in hospital campus

3.6.1 Architecture context and interfaces

The Figure 3.7 depicts the architecture context of the component WP5-01-CEA.

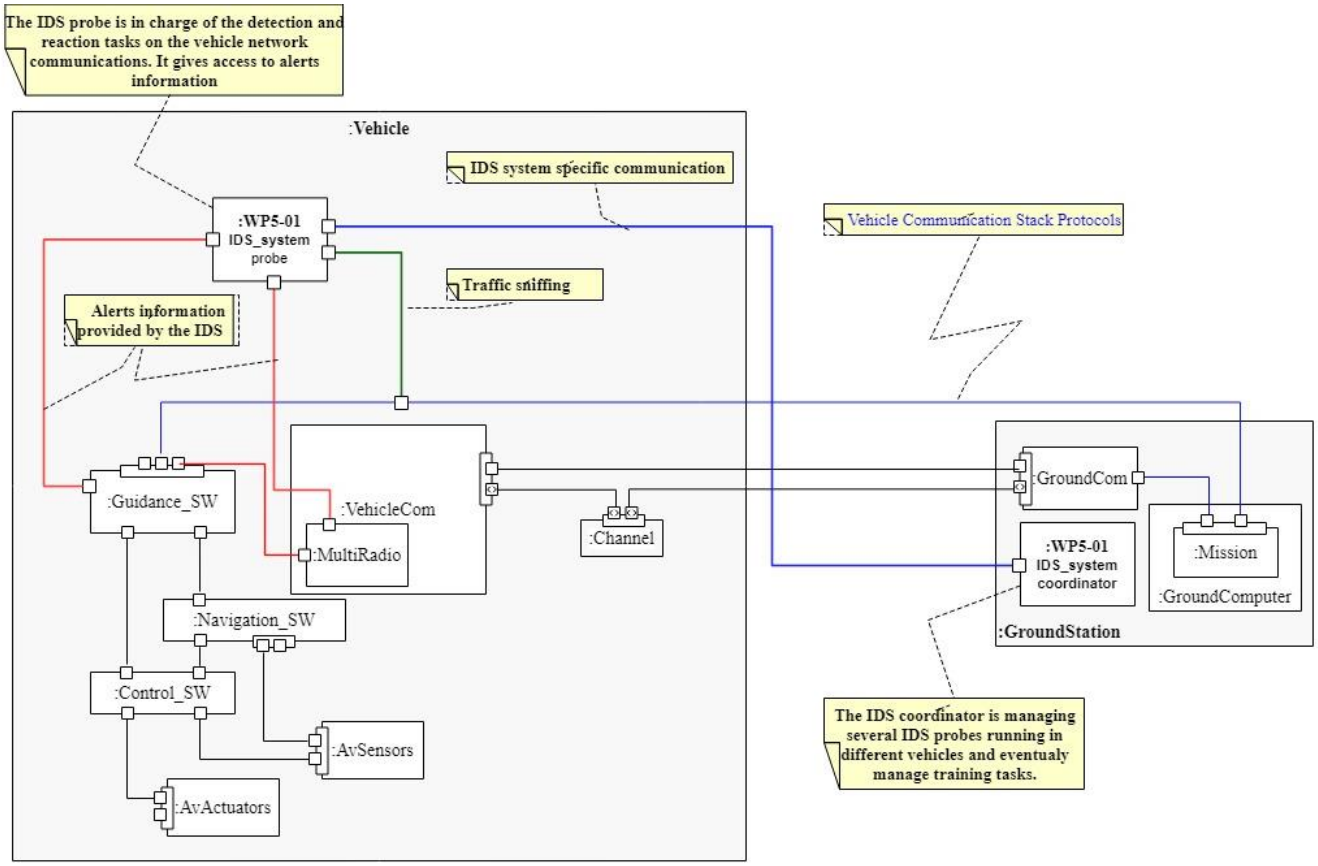

The IDS probe is in charge of the detection and reaction tasks on the vehicle network communications. It gives access to alerts information
:Vehicle
IDS system specific communication
Vehicle Communication Stack Protocols
:WP5-01
IDS_system
probe
Traffic sniffing
Alerts information provided by the IDS
:Guidance_SW
:VehicleCom
:MultiRadio
:Channel
:GroundCom
:WP5-01
IDS_system
coordinator
:Mission
:GroundComputer
:GroundStation
:Navigation_SW
:Control_SW
:AvSensors
:AvActuators
The IDS coordinator is managing several IDS probes running in different vehicles and eventualy manage training tasks.

Figure 3.7: Architecture context of the component WP5-01-CEA

The WP5-01-CEA component is an anomaly-based Intrusion Detection System (IDS) based mainly on the analysis of network packets. So, it requires access to the data plane (Encrypted or Decrypted packets). Moreover, the distributed part is done via the communication with an auxiliary module running closed to the GroundComputer. Finally, it can produce useful information about the status of the system; for example, those information can be the fact that there is an attack on an LTE link so it can be used by the Multi-radio component in order to redirect the traffic.

3.6.2 Internals and technologies

3.6.2.1 Distributed solution

The Distributed IDS system will be divided into two different components:

- IDS probes embedding detection capabilities with machine-learning (cf. Figure 3.7 WP5-01 IDS_system probe).
- An IDS system coordinator running in the ground station capable to communicate with all the embedded IDS probes (cf. Figure 3.7 WP5-01 IDS_system coordinator).

The IDS system coordinator will be able to report the status of the different drones and to manage them.

The IDS technology is based on the cognitive security system of CEA List.

3.6.2.2 Anomaly-based Intrusion detection

Each IDS probe will be able to analyse each packet according to their different protocols via a specific neural networks-based architecture. Moreover, different post processing methods will be developed in order to decrease the False Positive Rate.

A work-period will be dedicated to enrich the training efficiency in order to reduce the training time.

For the development of all the machine-learning parts, Python will be the preferred technology and will include the use of different Python packages: Keras over Tensorflow framework; scikit-learn and pandas.

3.6.2.3 Suggestive reaction system

When an alert is confirmed, the suggestive reaction system will try to explain the event and suggest different countermeasures.

3.6.3 Component roadmap

In order to design the best solution, we are focusing our development on the improvement of different aspect of the existing IDS solution and especially:

- To adapt the IDS to drones and droids network communications.
- To improve the training capabilities in terms of efficiency by studying different machine-learning approaches such as Federated Learning, Transfer Learning and classical cloud-based training.
- To adapt the IDS solution to the specific ROS2 messages
- To improve the real-time capability on resources constrained devices such as drones or droids.

3.6.3.1 Results at M10

From M4 to M10, we have done:

- A preliminary evaluation of the best machine-learning algorithm compatible with the envisaged available traffic.
- A preliminary version of a study regarding Federated-Learning and Transfer Learning mechanisms in order to increase the efficiency of our training phase.
- A first development of the detection capabilities.
- A preliminary version of a study to evaluate the possibility to detect attack over ROS2 messages, which depends on the common architecture.

- An evaluation of the possibility to offload part of the training phase.

3.6.3.2 Plans for the year 2 and the year 3

Between M10 and M22, the focus will be put on the development of the Distributed IDS with anomaly-detection capabilities. In the same time, the different architectural choices started between M4 and M10 will be finalized and considered.

Then, from M22 to M30, we will finalize the development of the reaction capabilities and integrate the developed Distributed IDS. Then, we will validate the performance.

We plan to measure component characteristics, and compute metrics and KPIs according to the Table 3.12, and so contribute to evaluate the achievement of the project objective O3.

Table 3.12: KPI and metric of the component WP5-01-CEA

№	KPI	#	Metric	MO
1	Secure Communications – Detection Capability	1	Accuracy: Ratio of well classified packets Objective: High Accuracy (> 90%)	MO3.3
1	Secure Communications – Detection Capability	2	Time Before Detection: Time between the beginning of the attack and its detection Objective: Low-latency alerts (~20 s)	MO3.3
1	Secure Communications – Detection Capability	3	False Positive Rate (FPR): number of normal packets classified as attacks Objective: Low FPR (< 2%)	MO3.3

№ = Component wide KPI number | KPI = KPI description
# = KPI wide metric number | Metric = Metric description
MO = Measurable Outcome (see Table 1.5) that the metric refers to and refines

3.7 Communication router (WP5-18-CEA)

The component entitled "Neon Drone Communication Router" is a network router based on the Software Defined Networking (SDN) architecture. It offers efficient, reliable and resilient communications by leveraging techniques such as bandwidth aggregation, real-time monitoring of the link quality and dynamic reconfiguration of the network interfaces.

The Table 3.13 provides the synopsis of the component WP5-18-CEA.

Table 3.13: Component WP5-18-CEA

<table>
<tr><td colspan="2"><b>Identifier:</b> WP5-18-CEA</td><td colspan="2"><b>Partner:</b> CEA</td><td colspan="2"><b>Expected TRL:</b> TRL6</td></tr>
<tr><td colspan="6"><b>Name:</b> NEON Drone Communication Router</td></tr>
<tr><td colspan="2"><b>License:</b> Proprietary</td><td colspan="2"><b>Owner:</b> CEA List</td><td colspan="2"><b>Contact:</b> Michael.Boc@cea.fr</td></tr>
<tr><td colspan="6"><b>Expected Outcome:</b> Embedded router with multiple radio interfaces capabilities able to (1) increase the available bandwidth through capacity aggregation, and (2) improve communication availability through seamless handover of traffic from an interface to another.</td></tr>
<tr><td colspan="6"><b>Description:</b> The component WP5-18-CEA provides communication capabilities from a drone to a pilot, to the cloud, and/or other drones in an efficient and reliable way. By aggregating the capacity of multiple radio interfaces, this component is able to increase the available bandwidth for applications. By using multiple radio interfaces, it offers the capability to switch the traffic from one interface to the other as soon as a disconnection or a drop of performance is detected.</td></tr>
<tr><td colspan="6"><b>Satisfied Requirement:</b> IDENTIFIER — DEFINITION
• UC3-DEM-COM-XX — Available bandwidth from multiple interfaces can be aggregated to ensure reliable transmission of voluminous data. Aggregation can be used for both drone-to-drone (e.g. Wi-Fi interfaces) and drone-to-infrastructure (e.g. cellular interfaces) communication.
• UC3-DEM-COM-XX — Quality of all available communication links will be monitored and assessed during operation.
• UC3-DEM-COM-XX — All network interfaces will be automatically reconfigured without the need for human intervention whatsoever. For example, addition of a new wireless interface must be plug-and-play. Faulty interfaces will also be automatically detected and discarded by the system.</td></tr>
<tr><td colspan="6"><b>Key Enabling Technology:</b> KET — CATEGORY
• KET: Communication, Navigation and Surveillance — CATEGORY: U-Space Capabilities
• KET: Command and control — CATEGORY: U-Space Capabilities
• KET: Intelligent Vehicle System Monitoring — CATEGORY: System Functions
• KET: Network Centric Communications Systems — CATEGORY: System Functions
• KET: Over the Horizon Communications — CATEGORY: System Functions</td></tr>
<tr><td colspan="6"><b>Improvement:</b> Accurately predict network congestion and other problems for proactive network management (common ML metrics such as MSE, RMSE, ROC etc. will be used for the evaluation).</td></tr>
<tr><td colspan="3"><b>Contributor:</b> CEA</td><td colspan="3"><b>Task:</b> T5.2</td></tr>
<tr><td colspan="6"><b>Use Case:</b>
• UC3 — Logistics
<b>Demonstrator:</b>
• D2 — Logistics in 5G urban environment: clinical sample delivery in hospital campus</td></tr>
</table>

3.7.1 Architecture context and interfaces

The Figure 3.8 show the architecture of the WP5-18-CEA component.

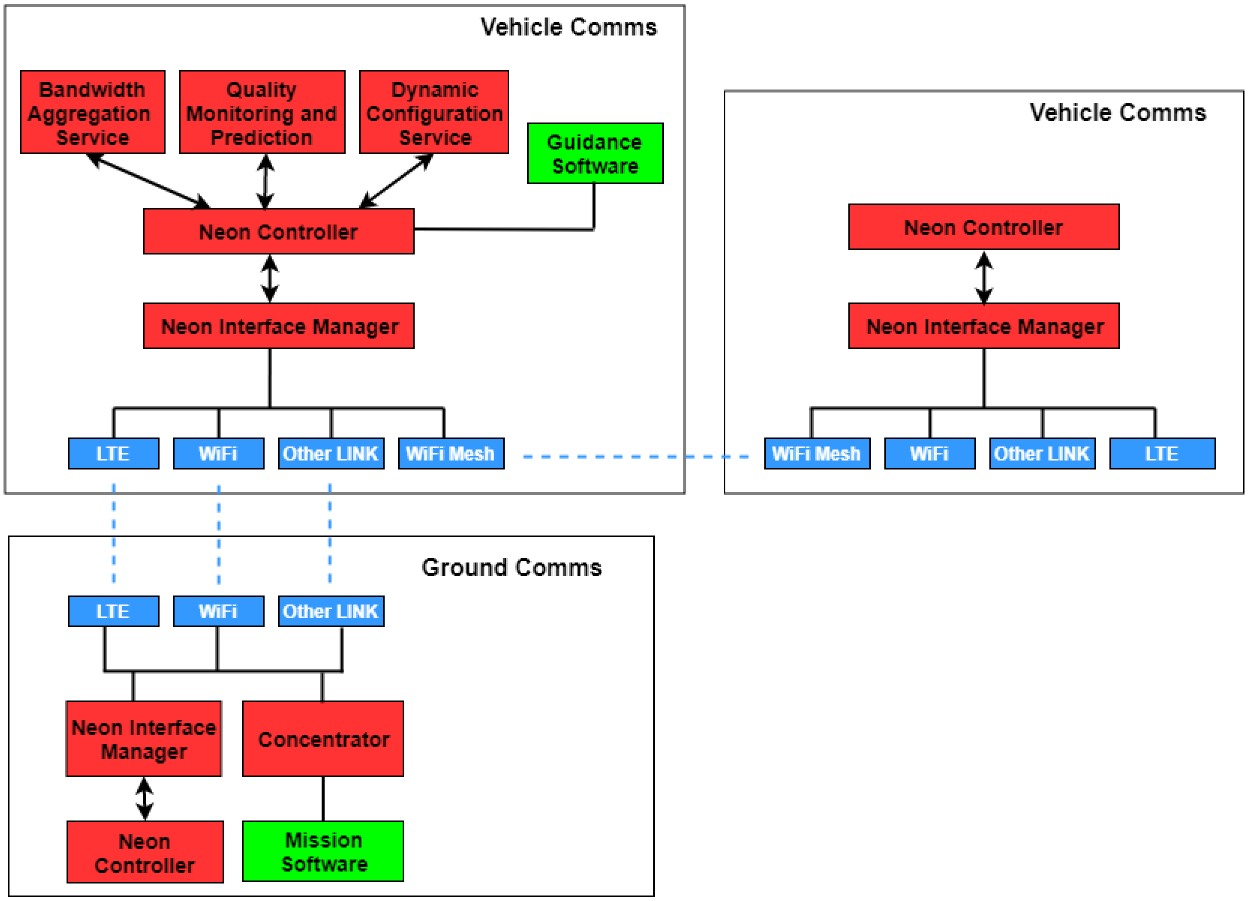


Figure 3.8: Detailed architecture of the WP5-18-CEA component

3.7.2 Internals and technologies

The component described in this section offers reliable, efficient and resilient communication from a drone to a pilot, to the cloud, and/or other drones in its range. It is a software system based on the SDN architecture running on an embedded hardware platform which will have to meet certain criteria (e.g. number of sockets for modems and Wi-Fi interfaces). The Neon Router can be deployed on the Vehicle as well as on the Ground Station. The system will provide three basic functionalities:

- **Bandwidth Aggregation**: The available bandwidth of multiple interfaces and multiple radio technologies can be aggregated to offer higher data throughput to the system than the capacity of each individual link. An obvious advantage is the leveraging of diversity; for example, LTE modems from multiple operators can be used to take advantage of the fact that some operators have better network coverage in an area than others.

- **Link Quality Monitoring and Prediction**: All available network links will be monitored real-time in order to assess their network quality. Based on the monitored results the system can decide for example to remove a link from the Bandwidth Aggregation process (if for example its estimated bitrate is too low) until its quality reaches an acceptable level again. Furthermore, Machine Learning regression and classification techniques will be used to evaluate future network quality indicators (e.g. available bandwidth, link utilization, congestion) in order to proactively reconfigure the network.

- **Dynamic Configuration of Interfaces**: This service will be responsible of dynamically configuring network interfaces based on results from the Link Quality Monitoring service. It can also provide plug-and-play services (e.g. no need to configure manually a newly inserted Wi-Fi interface).

Some of the technologies used include:

- **Neon**: CEA List's proprietary SDN platform.
- **OpenVSwitch**: The OpenFlow protocol implementation ubiquitous in SDN networks.
- **Tensorflow, Scikit-learn**: Machine Learning platforms for the prediction part.
- **Pandas, NumPy**: Necessary python packages for data manipulation.

3.7.3 Component roadmap

3.7.3.1 Results at M10

- Preliminary tests will be carried out for the congestion prediction mechanism.
- Conception and preliminary development of the services.

3.7.3.2 Plans for the year 2 and the year 3

- **M22**: Integration of the developed SDN services with the SDN software platform. Preliminary testing of the services using the metrics described below. Main focus on the prediction mechanism and its performance. TRL: 5
- **M30**: Integration of the complete software platform (SDN controller, services, prediction mechanism etc.) with the hardware platform. Validation of its performance using the proposed metrics. TRL: 6

We plan to measure component characteristics, and compute metrics and KPIs according to the Table 3.14, and so contribute to evaluate the achievement of the project objective O3

Table 3.14: KPI and metric of the component WP5-18-CEA

№	KPI	#	Metric	MO
1	Bandwidth	1	Available Aggregated Bandwidth based on the number and type of network interfaces connected to the system	MO3.2
2	Prediction Performance	2	Difference between the predicted value of a metric (e.g. the future link utilization ratio) and the real one.	MO3.2

№ = Component wide KPI number KPI = KPI description
# = KPI wide metric number Metric = Metric description
MO = Measurable Outcome (see Table 1.5) that the metric refers to and refines

3.8 Lightweight cryptography (WP5-08-ROT)

The Table 3.15 provides the synopsis of the component WP5-08-ROT.

Table 3.15: Component WP5-08-ROT

Identifier: WP5-08-ROT	**Partner:** ROT	**Expected TRL:** TRL4
Name: Lightweight cryptography		
License: Proprietary	**Owner:** ROT	**Contact:** ROT
Expected Outcome: Appling lightweight cryptography and IDS (Intrusion Detection System) system on drones' communication.		
Description: The component provides two modules. The data that has to be sent is encrypted by the encryption module. The data that has been received is decrypted by the decryption module in order to be read. If someone tries to deliver dangerous information to the system is blocked by the IDS module based on topology check.		
Satisfied Requirement: IDENTIFIER — DEFINITION • UC5-DEM1-SEC-001 — It shall be developed a cryptographic module that, on each network node, shall be in charge of handling the data evaluated as critical after the risk assessment phase. All the modules should be designed to use less computational resources as possible due to the architecture of the component itself. • UC5-DEM1-SEC-002 — There shall be designed a module that shall guarantee the detection of unauthorized access to the network, in order to avoid the introduction of dangerous information or the data breach.		
Key Enabling Technology: KET — CATEGORY • KET: Security — CATEGORY: U-Space Capabilities • KET: Communication, Navigation and Surveillance — CATEGORY: U-Space Capabilities • KET: Intelligent Vehicle System Monitoring — CATEGORY: System Functions • KET: Network Centric Communications Systems — CATEGORY: System Functions • KET: Over the Horizon Communications — CATEGORY: System Functions		
Improvement: Lightweight cryptography is the SoTA for what concerns communication between resource constrained devices such as sensors, RFID and embedded systems. Power consumption is one of the most important challenges w.r.t. drones. Nowadays the usage of a lightweight cryptography module is the best way to achieve security and power efficiency in drone-to-drone communication without losing performance.		
Contributor: ROT	**Task:** T5.1, T5.4	
Use Case: • UC5 — Agriculture **Demonstrator:** • D1 — Crop monitoring		

3.8.1 Architecture context and interfaces

With the purpose of providing encrypted communication and providing an intrusion detection system, the ROT module will be implemented on the drone/rover involved in the UC5 Demonstrator 1 scenario and on a workstation that exchanges information with it.

Referring to Figure 1.1, the module will be installed both in Ground Station computer and in Vehicle Computer.

Figure 3.9 depicts the architecture context of the component WP5-08-ROT.

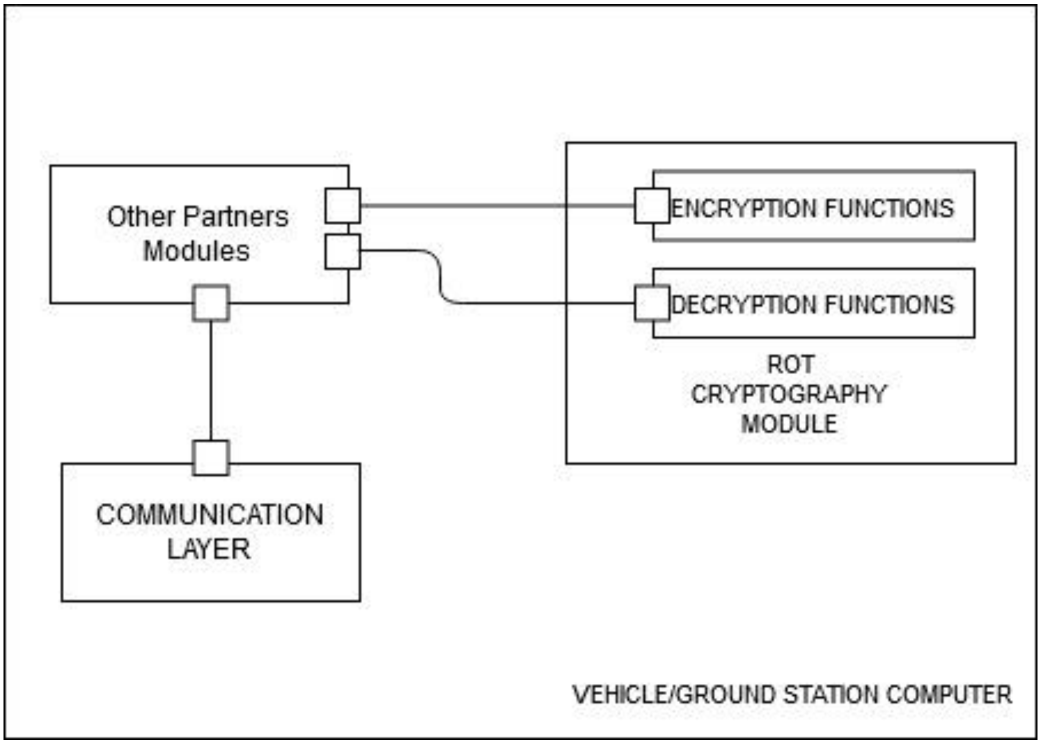


Figure 3.9: Architecture context of the component WP5-08-ROT

As shown in the figure, the other components that wants to encrypt their information will call Encryption module in order to send sensible data to outside the system. Once the encrypted data is received, in order to use the information, need to be exchange with the decryption module.

The module will be implemented as a library. The library must be included into the system in order to access the methods that it offers.

3.8.2 Internals and technologies

The Intrusion Detection System (IDS) will be based on network topology. Each node of the communication network will be configured in order to recognize if the encrypted data has been sent by a trusted node. A trusted node is a node equipped with a HMAC (Keyed-Hashing for Message Authentication) provided by system administrator.

HMAC is a specific type of Message Authentication Code, which is a short piece of information used to authenticate a message in order to improve confidentiality and integrity, generated through a cryptographic hash function and a secret cryptographic key.

A list of trusted HMAC is written inside all nodes belonging to the network. Once a node receive data as first step will control if the HMAC belongs to a trusted node. If the HMAC doesn't belong to a trusted node the data is rejected and the system is informed of the intrusion attempt.

3.8.3 Component roadmap

3.8.3.1 Results at M10

Within M10 we completed the study of the WP5-08-ROT component software architecture. We started the configuration of the environment for the development of software modules for drones. We started the study of open source autopilots in order to integrate new modules into one platform.

3.8.3.2 Plans for the year 2 and the year 3

For the year 2 the modules will be developed and tested in a laboratory simulated environment. For the year 3, the modules will be integrated and validated with other partners modules in the scenario of the Use Case 5 Demonstrator 1.

We plan to measure component characteristics, and compute metrics and KPIs according to the Table 3.16, and so contribute to evaluate the achievement of the project objective O3

Table 3.16: KPI and metric of the component WP5-08-ROT

№	KPI	#	Metric	MO
1	Light Encryption/Decryption	1	Communication performance degradation max	MO3.1
2	Intrusion Detection	1	Intrusion detection rate: number of packets detected as sent from not authorized source.	MO3.3

№ = Component wide KPI number KPI = KPI description
# = KPI wide metric number Metric = Metric description
MO = Measurable Outcome (see Table 1.5) that the metric refers to and refines

3.9 Anti-jamming and anti-spoofing features in geo-referencing system (WP5-11-ACO)

The Table 3.17 provides the synopsis of the component WP5-11-ACO.

Table 3.17: Component WP5-11-ACO

<table>
<tr><th><b>Identifier:</b> WP5-11-ACO</th><th><b>Partner:</b> ACORDE</th><th><b>Expected TRL:</b> TRL4</th></tr>
<tr><td colspan="3"><b>Name:</b> Navigation system with anti-jamming and anti-spoofing features</td></tr>
<tr><td><b>License:</b> Proprietary</td><td><b>Owner:</b> ACORDE</td><td><b>Contact:</b> ACORDE</td></tr>
<tr><td colspan="3"><b>Expected Outcome:</b> A navigation system with anti-jamming and anti-spoofing capabilities based on the integration of modern GNSS receiver.</td></tr>
<tr><td colspan="3"><b>Description:</b> The georeferrenced position and attitude system is in charge of providing a trustable positioning and attitude. It is based on the fusion of muti-antenna/multi-receiver data and of additional sensors like accelerometers, gyroscopes and a barometer. The system supports an extensive set of parameters whose values can be adapted to the characteristics of an UAV scenario and its expected dynamics. Data fusion and profile adaptation enable the system to provide the autopilot reliable and accurate navigation data for a safer drone operation. It can be also exploited on the payload side, e.g., for digitisation missions.
The WP5-11-ACO component comprises jamming and spoofing detection, and enables counter-measures at some extent which increase the integrity (i.e., trustability) of the navigation data, which can be compromised both by unintended jamming signals, or by malicious (jamming and spoofing) attacks.</td></tr>
<tr><td colspan="3"><b>Satisfied Requirement:</b> IDENTIFIER — DEFINITION
• UC2-DEM1-SEC-01 — Integrity vs shadows, interferences, and malicious attacks of the attitude and position data for digitisation.</td></tr>
<tr><td colspan="3"><b>Key Enabling Technology:</b> KET — CATEGORY
• KET: Indoor Positioning — CATEGORY: System Functions</td></tr>
<tr><td colspan="3"><b>Improvement:</b> In WP5, the georeferrenced position and attitude system will be enabled to support anti-jamming and anti-spoofing capabilities. This will enable more integrity of navigation data on different scenarios, e.g. unintended jamming, and also malicious jamming or spoofing attacks.</td></tr>
<tr><td colspan="2"><b>Contributor:</b> ACORDE</td><td><b>Task:</b> T5.4</td></tr>
<tr><td colspan="3"><b>Use Case:</b>
• UC2 — Construction
<b>Demonstrator:</b>
• D1 — Digitalization of civil infrastructure construction</td></tr>
</table>

3.9.1 Architecture context and interfaces

The Figure 3.10 depicts the architecture context of the component WP5-11-ACO.

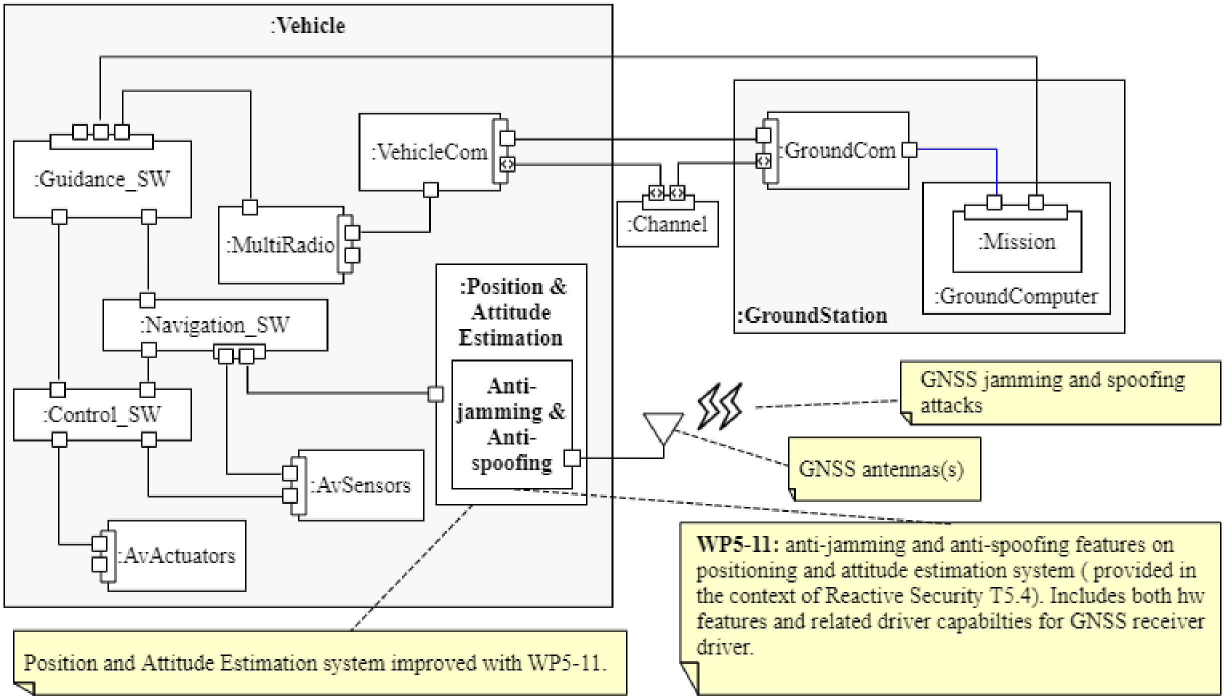

:Vehicle
:Guidance_SW
:MultiRadio
:VehicleCom
:Channel
:GroundCom
:Mission
:GroundComputer
:GroundStation
:Navigation_SW
:Position & Attitude Estimation
Anti-jamming & Anti-spoofing
:Control_SW
:AvSensors
:AvActuators
GNSS jamming and spoofing attacks
GNSS antennas(s)
WP5-11: anti-jamming and anti-spoofing features on positioning and attitude estimation system (provided in the context of Reactive Security T5.4). Includes both hw features and related driver capabilties for GNSS receiver driver.
Position and Attitude Estimation system improved with WP5-11.

Figure 3.10: Architecture context of the component WP5-11-ACO

As the picture shows, the anti-jamming and anti-spoofing features serve to ensure the reliability of any position and attitude estimation passed to the Navigation SW. In the first case, in addition to some basic hardware mitigation measurements, the presence of jamming is continuously monitored and its effect over the solution is estimated according the input power level detected, and, in the second case, a set of typical spoofing events can be detected.

With regard to the output interface of the position and estimation system, the picture provides a simple interface and architecture solution, in the sense that, any spoofing or jamming detection will eventually become an impact on the validity and quality information encompassing the position and attitude information at the output of the Position and Attitude Estimation system.

The component provides notifications to the guidance SW about the jamming presence and its effect in the reliability of the solution and about spoofing detection events. This way the guidance SW can either autonomously or by delegation to the ground station make a higher-level decision upon this information, e.g. to come back home or landing.

3.9.2 Internals and technologies

In this task, ACORDE carries out integration activities related to both to HW and to SW aspects. At the HW level, the main aspect is the re-design of the systems sensors board (see the **COMP4DRONES** report D3.1 [5] for the description of the HW platform of the ACORDE position and estimation system), which integrates a new model of GNSS receivers with anti-jamming and anti-spoofing capabilities. An additional improvement of the original HW design has been done to include a SAW filter for protection against out-band jamming.

At software-level, ACORDE is developing a specific driver for handling the GNSS receiver, in order to support all the configuration and receiver information retrieving capabilities available in the versions before **COMP4DRONES**, plus the newer anti-jamming and anti-spoofing configuration and information retrieving capabilities.

3.9.3 Component roadmap

3.9.3.1 Results at M10

By M10, the new GNSS receiver model has been already defined, as an aspect eminently associated to WP3 activity, but with necessary consideration of the of WP5 requirements to support anti-spoofing and anti-jamming. In addition, a first assessment on the communication interface with the new GNSS receiver model supporting anti-jamming and anti-spoofing has been done. Experiments on configuring and retrieving status information on the anti-jamming and anti-spoofing capabilities have been conducted.

3.9.3.2 Feedback to the other WPs

There is a clear coupling with WP3 activities. ACORDE is developing an abstract API for handling the new GNSS receivers. This API will push an easier migration to new receivers’ models with improved performance and cost in the future. For its initial development, the proprietary protocols of the old GNSS model used in the previous platform design and by the new model are used. As they are receivers from different manufacturers, using different proprietary protocols, this API is considered a first good reference. A main part of the information retrieved (such as PVT information, ionospheric corrections, raw data) is greatly common. However, some other aspects are not that common, either because the different degree of support on decoding all the information emitted by the GNSS signals, or, as it is the case in WP5 activity, because the previously used model does not support anti-jamming and anti-spoofing features. In that sense, the activity developed in WP5, to support these on the API will provide new feedback to WP3 activity. Moreover, since there is a lack of reference, the search for a general, standardisable API in WP3 will necessarily provide SoA feedback from WP3 to WP5 activity.

3.9.3.3 Plans for the year 2 and the year 3

For the time remaining, ACORDE will complete the new GNSS receiver driver, including the anti-jamming and anti-spoofing capabilities for the Y2 activity. For the Y3, ACORDE will complete the in-lab tests (TRL4).

We plan to measure component characteristics, and compute metrics and KPIs according to the Table 3.18, and so contribute to evaluate the achievement of the project objectives O1 and O3, specifically, in relation to the security of the drone navigation data, and therefore of the security of the drone, its payload and all the persons and elements in its operational environment,.

Table 3.18: KPI and metric of the component WP5-11-ACO

№	KPI	#	Metric	MO
1	Trustable Geo-Referencing system	1	Availability of anti-jamming capabilities, expressed as an integer number (the higher, the better): ● 0: Not available ● 1: Available ● 2: Available and jamming event can be signalled to the user	MO3.2 MO3.3
		2	Availability of anti-spoofing capabilities, expressed as an integer number (the higher, the better): ● 0: Not available ● 1: Available, based on continuity ● 2: Available, based on continuity and spoofing events can be signalled to the user ● 3: Available, not based on continuity ● 4: Available, not based on continuity and spoofing events can be signalled to the user	MO3.2 MO3.3
2	Easy and flexible integration of Trustable Geo-Referencing	1	Support or not of transparent integration, i.e., if it supports integration without requiring any handling or configuration of anti-jamming and anti-spoofing features (enabled by default), expressed as a Boolean: ● 1: Support transparent integration ● 0: Does not support transparent integration	MO1.3
		2	Support the configuration of the anti-jamming and anti-spoofing capabilities, expressed as an integer number (the higher, the better): ● 0: No kind of configuration ● 1: Hard-coded configuration (firmware) ● 2: From configuration file ● 3: From configuration file and user interface ● 4: From configuration file and user interface, supporting incremental configuration (variants on top of default configuration)	MO1.3
		3	Time required for the integration and configuration of trustable geofencing	MO1.3

№ = Component wide KPI number KPI = KPI description
# = KPI wide metric number Metric = Metric description
MO = Measurable Outcome (see Table 1.5) that the metric refers to and refines

3.10 Robust and enriched communication for Indoor Positioning System (WP5-19-ACO)

The Table 3.19 provides the synopsis of the component WP5-19-ACO.

Table 3.19: Component WP5-19-ACO

<table>
<tr><th><b>Identifier:</b> WP5-19-ACO</th><th colspan="2"><b>Partner:</b> ACORDE</th><th><b>Expected TRL:</b> TRL5</th></tr>
<tr><td colspan="4"><b>Name:</b> Robust and enriched communication among beacons, and among beacons and drone, enabling an improved indoor positioning</td></tr>
<tr><td><b>License:</b> Proprietary</td><td colspan="2"><b>Owner:</b> ACORDE</td><td><b>Contact:</b> ACORDE</td></tr>
<tr><td colspan="4"><b>Expected Outcome:</b> Robust and enriched communication among beacons, and among beacons and drone, enabling an improved indoor positioning.</td></tr>
<tr><td colspan="4"><b>Description:</b> The IPS provides a georeferenced, high rate position to the drone platform in indoor scenarios where GNSS is not available. It is based on the estimation of accurate ranges from a mobile tag to statically deployed anchors. The mobile tag can either provide the ranges to the anchors for being externally fused with other data sources (default use in the construction use case, indoor demo, UC2-D1) or its auto-calculated position. A robust and enriched communication scheme between anchors and tags makes possible to provide an optimal positioning solution.</td></tr>
<tr><td colspan="4"><b>Satisfied Requirement:</b> IDENTIFIER — DEFINITION
• UC2-DEM2-FUN-10 — An indoor positioning system will provide real-time geo-references position to drone for enable autonomous, waypoint-based flight within the indoor infrastructure.
• UC2-DEM2-INT-01 — The indoor positioning system shall provide real-time navigation information at the drone side, that can be integrated by the autopilot, including raw data (ranges).
• UC2-DEM2-PRF-01 — The system shall provide real -time navigation data with sub metric accuracy in all the infrastructure.
• UC2-DEM2-PRF-02 — The system shall imply an affordable and scalable cost for the infrastructure (< 3 k€ for 1 km tunnel).
• UC2-DEM2-OPE-01 — The system shall provide real-time navigation data, with sufficiently accuracy for autonomous, way point based navigation within the indoor flying volume (without obstacles).
• UC2-DEM2-OPE-02 — The indoor position navigation accuracy shall allow to track a sufficiently accurate route to allow effective indoor geofencing (without obstacles).
• UC2-DEM2-OPE-03 — The indoor position navigation accuracy shall have resilience against the presence of objects within the indoor infrastructure (machinery, etc.) in operational conditions.
• UC2-DEM2-OPE-04 — The indoor position navigation accuracy shall allow to track a sufficiently accurate route to avoid obstacles provided they position is known in advance (without obstacles).
• UC2-DEM2-USA-01 — The indoor position navigation accuracy shall allow to track a sufficiently accurate route to avoid obstacles provided they position is known in advance (without obstacles).</td></tr>
<tr><td colspan="4"><b>Key Enabling Technology:</b> KET — CATEGORY
• KET: Indoor Positioning — CATEGORY: System Functions</td></tr>
<tr><td colspan="4"><b>Improvement:</b> In WP3 and WP4, a custom tag and anchor platform will be developed. In WP5, a specific effort will be done for a more robust and enriched communication. The goal is to successfully tackle a scenario with possible obstacles and to offer enriched accuracy. The customization of the communication protocol for the data exchanged both among anchors and anchor-tag will be tackled. Specific extensions for the auto-positioning of the anchors, and for a better positioning of the tag will be done. Specific work related to the antenna configuration/integration on the drone platform will be also done. All this piece of work is likely to lead to benefits on the fulfilment of all UC2-DEM2 requirements on the Indoor Positioning System.</td></tr>
<tr><td colspan="2"><b>Contributor:</b> ACORDE</td><td colspan="2"><b>Task:</b> T5.2</td></tr>
<tr><td colspan="4"><b>Use Case:</b></td></tr>
</table>

- UC2 — Construction

Demonstrator:

- D2 — Monitoring underground infrastructure construction process

3.10.1 Architecture context and interfaces

The Figure 3.11 depicts the architecture context of the component WP5-19-ACO.

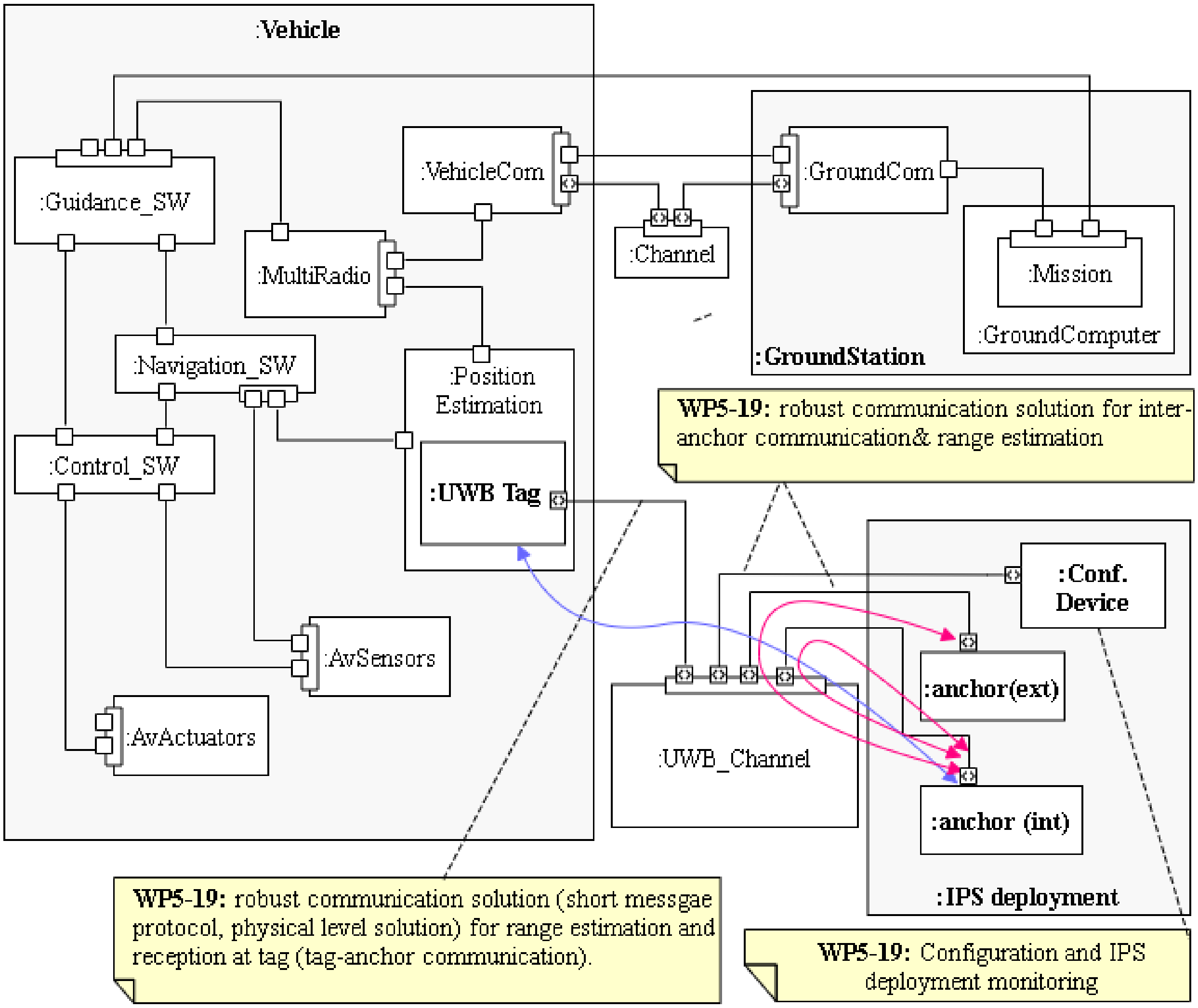

:Vehicle
:Guidance_SW
:VehicleCom
:MultiRadio
:Navigation_SW
:Position Estimation
:UWB Tag
:Control_SW
:AvSensors
:AvActuators
:Channel
:GroundCom
:Mission
:GroundComputer
:GroundStation
WP5-19: robust communication solution for inter-anchor communication& range estimation
:Conf. Device
:anchor(ext)
:UWB_Channel
:anchor (int)
:IPS deployment
WP5-19: robust communication solution (short messgae protocol, physical level solution) for range estimation and reception at tag (tag-anchor communication).
WP5-19: Configuration and IPS deployment monitoring

Figure 3.11: Architecture context of the component WP5-19-ACO

As the Figure 3.11 shows, the ACORDE activity in WP5-19 deals with the communication links among the elements of the UWB (Ultra-Wideband) network of the UWB Indoor Positioning System:

- Tag ↔ (visible) Anchors communication;
- Anchor ↔ Anchor communication;
- Tag/Anchor ↔ Configuration Device communication.

Tag-Anchor communication (show as a blue two-sided arrow in Figure 3.11) enables the real-time positioning. Anchor-Channel-Anchor communication (represented as red arrows in Figure 3.11) enables the initial (and later recurrently updated) auto-positioning of anchors. Internal anchors are distinguished from external anchors (with their own geo-positioning capability). Finally, the aim is also to exploit the UWB network for the setup of anchor and tags, via a “Configuration Device”. This element is platform generic, in the sense that the target is that the configuration can be performed from different platform types, e.g. a laptop, a tablet, a cell phone. This means that eventually, it could also be integrated in the Ground Station. However, in the architectural diagram of the Figure 3.11 it is intentionally shown as a separate element, to show this flexibility, which is convenient as it enables a modular configuration solution, independent from the GS, a versatile, in the sense that different devices can be used for configuration.

3.10.2 Internals and technologies

The main purpose is to achieve a robust communication in order to improve the integrity of the positioning information in the indoor scenarios, and that can work on (or be less limited by) the presence of obstacles.

Therefore, a set of main aspects are being part of the finally adopted solution:

- antenna configuration and set-up within the drone platform (outcome as setup recommendations for the integrator);
- the (ISO) level-2 and level-3 communication protocol;
- the range estimation technique;
- the detection of blocked signals (obstacles disturbing anchor-tag visibility).

While the impact of antenna setup might be apparent, the performance analysis of level 2 and 3 protocols and range estimation techniques is relevant. As known, UWB technology is intimately linked to short message protocols, which is important for accurate range estimation. However, this is not enough. For instance, minimization and proper use of broadcast or multi-cast messages shall reduce the traffic and thus the latencies in range estimation.

The range estimation technique has also a tight relation (as explained in section 3.10.3.2) with the underlying communication protocols. Two-way-ranging and TDoA are the two basic range estimation techniques that we consider.

Other techniques needed for developing the IPS are the position estimation techniques. They are obviously impacted by the quality and constraints of the lower levels. The algorithms and solutions assessed there are in the scope of WP3.

3.10.3 Component roadmap

3.10.3.1 Results at M10

At the current time ACORDE has already acquired an Evaluation Kit to test the performance of a candidate UWB COTS. Moreover, it has already conducted some preliminary tests in an indoor environment, and relative positioning based on a very basic position estimation algorithm (not suitable to be used in the final solution) to test the basic 2D performance of that algorithms, and doing qualitative assessment of impact of antennas relative orientation and some obstacles.

ACORDE has also done a preliminary analysis of the communication protocols used in in this evaluation, not only to devise convenient variants, but also a means for instrumentation and log which allows to feed the analysis activities related to WP4 and WP6.

3.10.3.2 Feedback to the other WPs

The relation between WP5 and WP4 is bidirectional. Although the range estimation technique (mentioned in section 3.10.2) can be seen as an application level feature (developed in the framework of WP4) it has a direct impact on the design of the requirements brought to the communication protocols. When considering two basic, dominant range estimation techniques, like TWR and TDOA, the former requires the exchange of more messages, and thus involves more latency in range determination. This latency limits the update rate of the position information on the mobile tag and, therefore, its dynamic range. However, TDOA has other, non-negligible counterparts, as it requires fine overall synchronization mechanisms. ACORDE will enrichment the payload of short messages for better position estimation. This is expected to show specific trade-offs, mostly between accuracy, estimation rate, which need to be evaluated before fixing the final solution.

The evaluations done by ACORDE are also being doing on a specific evaluation kit, so that they are also having a clear impact on ACORDE activity on WP3, where a custom design of the tag and anchor platforms is being done.

At the same time, experiments on the performance of the message protocols, and on the actual platforms are considered of the application-level. This in turn is also directly related to WP6 activities. In WP6 (T6.3) ACORDE is developing a framework (IPS MAF) for modelling, the application level solution. In this model, the impact of the lower levels is also modelled and considered the analysis. In the aforementioned task, it is important that the framework can be capture the lower details as they are measured and progressively available (e.g. transmission delays on different environments, response delays as a result of node platform and message sizes) in a decoupled, structured, and scalable way. Eventually, these figures need to be collected from the measurements and assessment on WP5.

3.10.3.3 Plans for the year 2 and the year 3

For the second year, the assessment of communication protocols will be finalized and decided in concurrency and relation with the decision of the application level range algorithms. The platform design, and the validation of communication solution is expected to be validated on top of the custom platform in lab. For the third year, the communication solution will be validated on the relevant environment.

We plan to measure component characteristics, and compute metrics and KPIs according to the Table 3.20, and so contribute to evaluate the achievement of the project objective O3, specifically, in relation to the improvement of the communication between the drone and anchors (primarily for range and positioning purposes).

Obviously, these KPI and metrics are in a close relation to some requirements provided in the context of the tunnel construction use case (Use Case 2-Demo 2). Actually, most of the functional, performance, and operational metrics for the UC2-Demo1 directly depend on the KPI expressed in Table 3.20, e.g. UC2-DEM-Fun10: “An indoor positioning system will provide real-time geo-references position to drone for enable autonomous, waypoint based flight within the indoor infrastructure”; UC2-DEM2-PRF-01: “The system shall provide real-time navigation data with sub metric accuracy in all the infrastructure”; UC2-DEM2-OPE-01: “The system shall provide real -time navigation data, with sufficiently accuracy for autonomous, way point based navigation within the indoor flying volume (without obstacles)”—for details see the report D1.1 [3]).

The metrics of Table 3.20 are to be contextualized, i.e., whether taken on an indoor scenario without obstacles, or on an indoor scenario with obstacles representative of those that can be found during the construction phase of the indoor infrastructure.

Table 3.20: KPI and metric of the component WP5-19-ACO

№	KPI	#	Metric	MO
1	Robust Communication for positioning at indoor scenario	1	Ratio of success/lost messages (per tag-anchor link)	MO3.2
		2	Time percentage with three or more anchors	MO3.2
		3	Variance of the estimated ranges	MO3.2

№ = Component wide KPI number KPI = KPI description
# = KPI wide metric number Metric = Metric description
MO = Measurable Outcome (see Table 1.5) that the metric refers to and refines

3.11 Cryptographic primitives and protocols (WP5-16-AIT)

For the communication between drones or drones and a base station it is important that the transmitted (meta)data is property protected. Strong protection guarantees can be achieved by means of cryptographic protocols that can provide confidentiality and authenticity of transmitted data as well as privacy protection for the entities involved in a communication.

WP5-16-AIT represents a collection of basic cryptographic building blocks (like public-key encryption) and protocols (like forward secret key-exchange or anonymous authentication) and typically replaces or augments other existing cryptographic primitives, e.g. basic mechanisms in the TLS (Transport Layer Security) protocol.

The Table 3.21 provides the synopsis of the component WP5-16-AIT.

Table 3.21: Component WP5-16-AIT

Identifier: WP5-16-AIT	**Partner:** AIT	**Expected TRL:** TRL4
Name: Cryptographic algorithms adapted for drones		
License: Open source	**Owner:** AIT	**Contact:** AIT
Description: This component represents a collection of cryptographic primitives and protocols whose characteristics are tailored to the use within drone environments (resource consumption, latency). In particular, the cryptographic protocols while providing means to satisfy low latency requirements will at the same time provide strong security guarantees (like full forward secrecy). Another important focus will be to provide long-term security and in particular resilience to quantum computers, i.e., post-quantum security.		
Satisfied Requirement: IDENTIFIER — DEFINITION • DEM9-SEC-07 — The key exchange mechanism used within TLS may provide low latency (zero round-trip or 0-RTT) and at the same time full forward secrecy. • DEM9-SEC-08 — The cryptographic primitives to provide trusted communication may be resistant against future powerful quantum computers.		
Key Enabling Technology: KET — CATEGORY • KET: Network Centric Communications Systems — CATEGORY: System Functions		
Improvement: State-of-the-art cryptographic protocols do not provide low latency as well as strong security properties such as full forward secrecy at the same time. Moreover, they lack security in the face of powerful quantum computers. All these aspects will be considered within this component, improving significantly over the state-of-the-art.		
Contributor: AIT	**Task: T5.3**	
Use Case: • UC5 — Agriculture **Demonstrator:** • D2 — Wine production		

3.11.1 Architecture context and interfaces

The AIT component (Figure 3.12) does not represent a stand-alone module, but can be viewed as being located within the “VehicleCom” and “GroundCom” modules, i.e., the modules in charge of the communication between vehicles or between vehicles and the ground station and adds security features to the communication (confidentiality, authenticity, privacy).

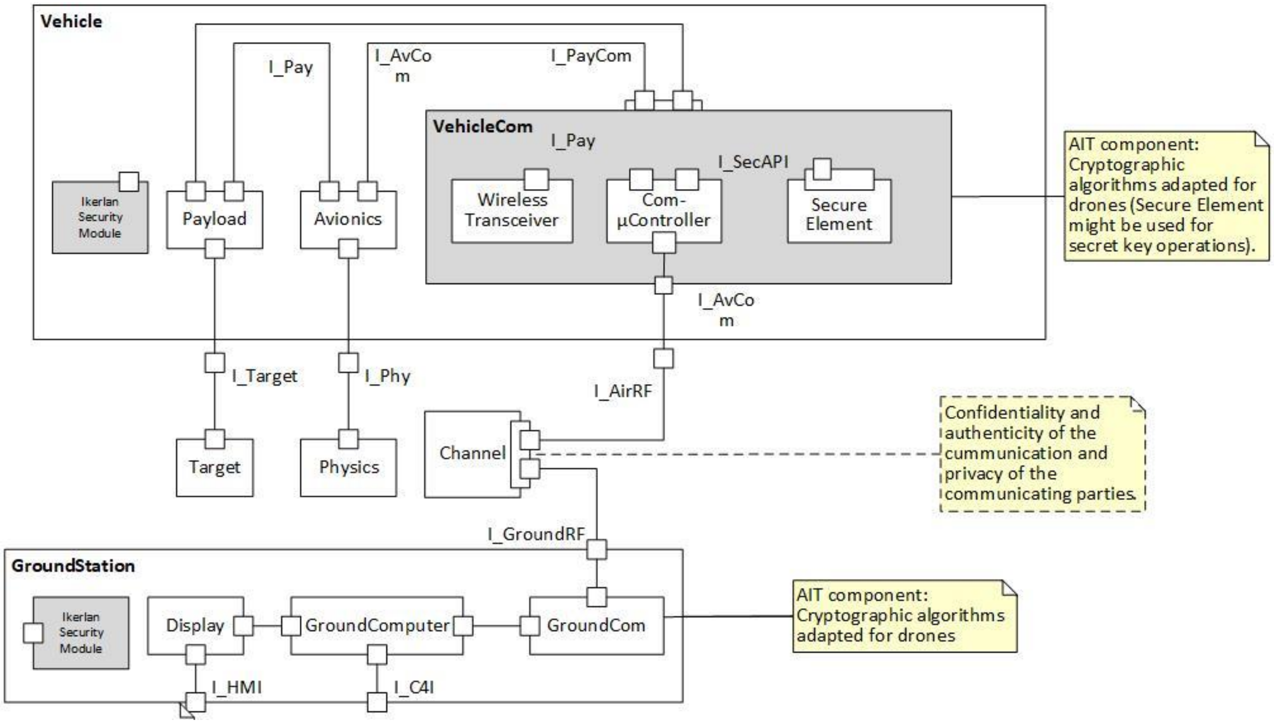


Figure 3.12: Architecture context of the component WP5-16-AIT

The component can be viewed as a collection of basic cryptographic building blocks (like public-key encryption) and protocols (like forward secret key-exchange or anonymous authentication) and typically replaces or augments other existing cryptographic primitives (e.g., basic mechanisms in the transport layer security (TLS) protocol). It runs on the Vehicle (VehicleCom), where it is intended to be used together with the “Secure Element” (SE) module to provide stronger security features (i.e., realize secret key operations within the SE). It also runs on the GroundStation and in combination it is possible for the Vehicle and the GroundStation to establish a secure communication (confidential and authenticated). Moreover, the component will provide features to protect the privacy of communication partners by means of anonymity.

3.11.2 Internals and technologies

We base the developments within this component on the previous works in the following domains:

(Low-latency) Key-exchange protocols: Our key-exchange protocols to achieve low latency (0-RTT) together with strong security properties are built upon so called puncturable encryption schemes (or key encapsulation mechanisms).

Privacy-preserving authentication: Depending on the required degree of privacy (only with respect to eavesdroppers or even communication partners like the base station), we focus on protocols either obtained from anonymous (authenticated) key-exchange protocols or primitives underlying anonymous attribute-based credentials or group signature schemes.

From an implementation perspective, depending on the maturity of the implementations (prototypes or libraries) we typically provide implementations either in Python or C.

3.11.3 Component roadmap

3.11.3.1 Results at M10

From M4 to M10, we have been working on:

- Improving cryptographic primitives to realize key-exchange protocols with low latency and in particular forward-secret zero round-trip time (0-RTT) key-exchange protocols. We have further improved the concept of Bloom Filter encryption (BFE) and investigated its practical applicability. These results are accepted to be published in the Journal of Cryptology [30].
- Work on post-quantum primitives for the use within key-exchange protocols. We have designed post-quantum instantiations of puncturable encryption primitives and efficient transformations for strong adaptive security of post-quantum KEMs [31].
- Work on privacy-preserving protocols (for guaranteeing anonymity of the involved parties) with a particular focus on ones that take advantage of the setting that involves a host and a secure element (where latter typically provides much stricter resource constraints then the former). We have designed a protocol and provide a prototypical implementation [32].

3.11.3.2 Plans for the year 2 and the year 3

On the larger timeframe we foresee to further improve our results from a cryptographic perspective, e.g., investigate additional aspects of (authenticated) key exchange protocols such as privacy [33] and provide implementations of selected primitives and protocols. Especially the work regarding the implementation will be specified in the next phase.

We plan to measure component characteristics, and compute metrics and KPIs according to the Table 3.22, and so contribute to evaluate the achievement of the project objective O3.

Table 3.22: KPI and metric of the component WP5-16-AIT

№	KPI	#	Metric	MO
1	Forward-security & low latency	1	Achieve fs 0-RTT for key exchange	MO3.3
2	Post-quantum secure crypto	1	Achieve post-quantum security	MO3.3
3	Anonymous authentication	1	Enable anonymous authentication for communication participants	MO3.3

№ = Component wide KPI number KPI = KPI description
# = KPI wide metric number Metric = Metric description
MO = Measurable Outcome (see Table 1.5) that the metric refers to and refines

3.12 Robust communication (WP5-20-TNL)

In some interesting use cases drones have to operate in difficult communication environments where there might not always be a (viable) direct connection between endpoints. Disruption Tolerant Networking (DTN) can help alleviate this issue by offering means to transfer data between endpoints using Store-Carry-Forward (SCF) message switching, eliminating the need for end-to-end connectivity. This software component aims to enhance drones with this robust communication capability.

The Table 3.23 provides the synopsis of the component WP5-20-TNL.

Table 3.23: Component WP5-20-TNL

Identifier: WP5-20-TNL	**Partner:** TNL	**Expected TRL:** TRL5
Name: Robust Communication		
License: Proprietary	**Owner:** TNL	**Contact:** Maurits de Graaf
Expected Outcome: Development of SW components to support robust delivery of data to end-users.		
Description: Software components for robust communications by means of store- and forwarding methods, using mechanisms from Disruption Tolerant Networking (DTN). Focus is on collection of sensors observations in areas where standard connectivity may be limited. Robust interrupt and tolerant communication system. Study of components, methods algorithms, leading to a small-footprint implementation.		
Satisfied Requirement: IDENTIFIER — DEFINITION • UC4-DEM-2-SEC-02 — Communication: robust against signal interruptions. • UC4-DEM-2-PRF-12 — Commercial Off-The-Shelf standards for communications: ROS2, MAVLINK, Wi-Fi. • UC4-DEM-2-SEC-06 — Typical communication ranges are 50 m. System should deal with hampered communication channel.		
Standard: • IETF standards — DEFINITION: Bundle Protocol Version 7 ▪ RFC 5050: Bundle Protocol Version 7 [34] ▪ RFC 6257: Bundle Security Protocol [35] ▪ RFC 7242: Delay-Tolerant Networking TCP Convergence Layer Protocol Version 4 [36]		
Key Enabling Technology: KET — CATEGORY • KET: Security — CATEGORY: U-Space Capabilities • KET: Communication, Navigation and Surveillance — CATEGORY: U-Space Capabilities • KET: Intelligent Vehicle System Monitoring — CATEGORY: System Functions • KET: Network Centric Communications Systems — CATEGORY: System Functions • KET: Over the Horizon Communications — CATEGORY: System Functions		
Improvement: Specific focus: quality of service methods (priority classes), routing algorithms (in drone context) (static, epidemic, spray-and-wait) and the possibility to create the functionality in embedded systems with small footprint.		
Contributor: TNL	**Task:** T5.1, T5.2	
Use Case: • UC4 — Surveillance and Inspection **Demonstrator:** • D2 — Fleet of multi robot navigating and mapping in an unknown environment		

3.12.1 Architecture context and interfaces

An architectural overview of the robust communication component WP5-20-TNL is shown in Figure 3.13. Blocks that are added to the base architecture presented in Figure 1.1 are indicated with the green plus sign. Each participant platform, ground station and vehicles, will run a Disruption Tolerant Networking (DTN) instance and a video and message proxy. These platforms connect to each other over a shared radio channel in a Mobile Ad-hoc NETwork (MANET) or Vehicular Ad-hoc NETwork (VANET).

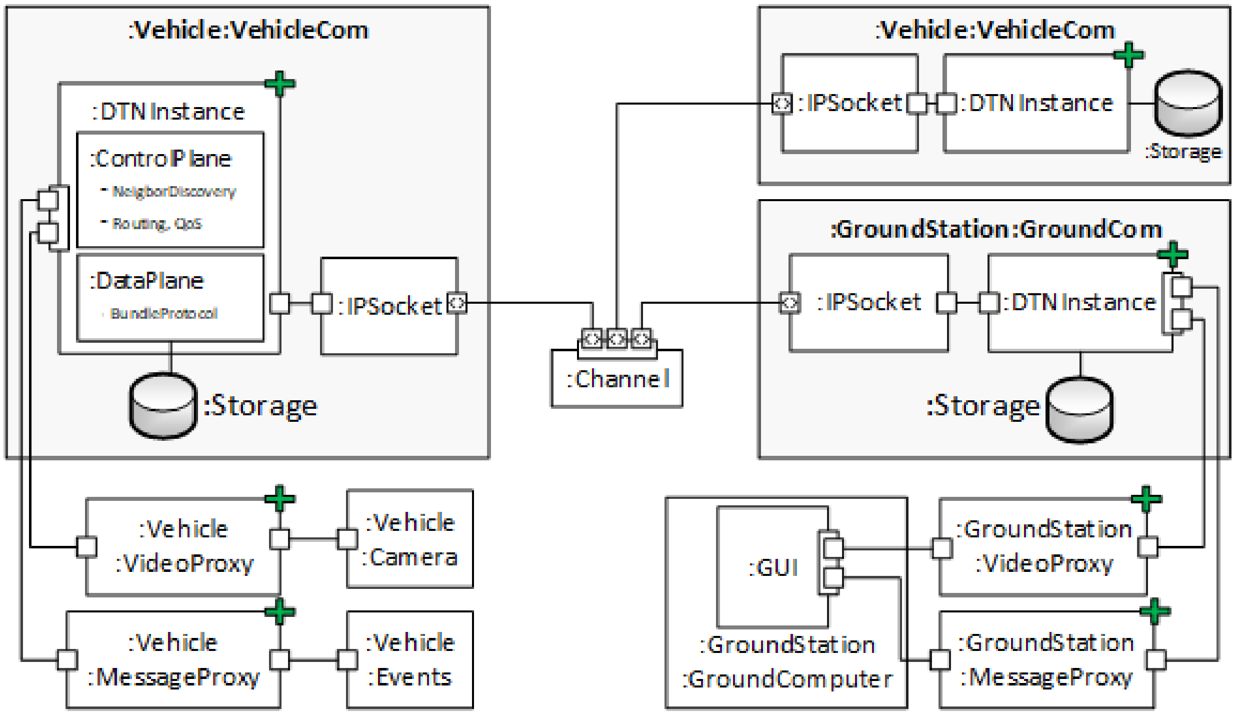
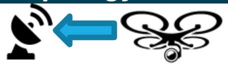
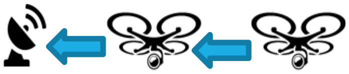
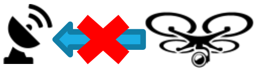
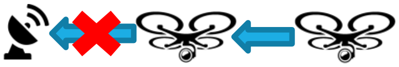


Figure 3.13: WP5-20-TNL Robust Communication Component Architecture

The DTN instance is divided in a control plane and a data plane. The control plane is responsible for detecting neighbouring nodes by means of the Neighbour Discovery Protocol and sharing of routing information with other nodes. The data plane is responsible for encapsulating / de-capsulating and sending / receiving data to and from neighbouring nodes following the BundleProtocol Version 7 RFC5050 [34]. Packets that are encapsulated by the bundle protocol header are referred to as bundles. DNT Instances communicate over both multicast and unicast IP sockets connected over a wireless IP or WLAN network.

The DTN instance is responsible for communication even if there is no direct link or path to the destination. In this case, the Store-Carry-Forward principle is applied where bundles are stored on the node itself until the destination becomes available again. The Table 3.24 what type of network scenarios are supported by the robust communication component.

Table 3.24: Robust Communication Network Scenarios WP5-20-TNL

Link Type	Topology	Description
Direct		There is a direct link from source to destination. Bundles are directly transferred to the destination
Multi-Hop		There is no direct link from source to destination. However, there is a path to the destination via one or more node. The Bundle is first transferred to the intermediate node(s) and then transferred to the destination.
No Link		There is no link nor a path from source to destination. Bundles are stored on the source node until a path to the destination becomes available.
Custody		There is no path to the destination but non-destination nodes (i.e. custody nodes) are reachable. Depending on the routing protocol, the bundle is transferred to the custody node and both stored on the source and custody node until the destination becomes available.

The message proxy and video proxy as depicted in Figure 3.13 are connected to the (Application Programming Interface) API of the DTN Instance and runs on the same computing platform as the DTN Instance. On the vehicles these proxies are responsible for translating sensor streams (e.g. video feeds) and messages (e.g. mission events) to DTN bundles and send them to the DTN Instance. These proxies provide a standardized interface for the sensor streams and messages. On the base station the proxies translate the received bundles back to the standardized interface and are then shown on a Graphical User Interface (GUI) on the base station computer.

For adapting the DTN protocol in a component that can be integrated on a drone platform we plan to use standardized interfaces between components. In the list below, we discuss what types of standards we consider to use for each interface corresponding to the interfaces depicted in Figure 3.13.

- *:Vehicle:VehicleCom:IPSocket ←→:GroundStation:Ground:Com:IPSocket / :Vehicle:VehicleCom:IPSocket.*
  Standard IP that supports TCP/IP communication over a wireless link. This wireless link could be based on MANET IEE802.11 (WiFi in ad-hoc mode), VANET IEEE802.11p/IEEE802.16 or 5G D2D (Device-To-Device) communication.
- *Vehicle:VideoProxy ←:Vehicle:Camera and GroundStation:VideoProxy→ GroundStation: GroundStationComputer:GUI*.
  The proxy will subscribe to the sensor using standardized streaming protocols like RTSP 2.0 (Real-Time Streaming Protocol) RFC7826. This allows us to steam not only video data but also data from other types of sensors like TOF (Time-Of-Flight) sensors or temperature sensors.
- *Vehicle:MessageProxy ←:VehicleEvents and GroundStation:MessageProxy→ GroundStation:GroundStationComputer:GUI.*
  The MessageProxy will be able to transmit asynchronous events based on the Publish-Subscribe design pattern. Protocols like DDS (Data Distribution Service) or architectural styles like REST (Representational state transfer) are considered.

3.12.2 Internals and technologies

Originally developed as a means to provide inter-planetary networking [37], DTN addresses some issues of modern mobile wireless networks which conflicts with some of the original assumptions of the internet. These issues include intermittent connectivity, long or variable delays, asymmetric data rates and high error rates. In intermittent connectivity, links between nodes appear and disappear, this can result in no end-to-end connectivity existing between two specific nodes, also called network partitioning. In addition to intermittent connectivity, long propagation delays between nodes and variable queuing delays at nodes in a multi-hop network can increase the end-to-end path delay significantly. This way, internet protocols relying on quick replies from other nodes will not function anymore.

To support connectivity between nodes communicating using a network suffering these issues, DTN introduces the Bundle Protocol (RFC5050 [34]). This protocol operates between the Application layer and underlying transmission layers, providing store-carry-forward (SCF) functionality to a node. Individual application messages will get stored in one or more bundles, which in turn will be transmitted using underlying transmission protocols. It is also in this bundle layer that DTN-specific routing occurs. In contrast to normal internet routing, messages, or fragments thereof, are stored in persistent storage at a node rather than a limited-size and -time buffer. Using the SCF principle explained earlier, even if no end-to-end connection exists due to e.g. network partitioning, messages are still able to be delivered to their destination. For the project we will restrict ourselves to the development of a solution for transmission of video and of sensor observations that may be transmitted multi-hop to the ground station.

3.12.3 Component roadmap

3.12.3.1 Results at M10

For M10 we have carried out a study of the relevant components of DTN. Specifically, in DTN routing is a challenge due to the fact that no end-to-end connection may exist, so the concept of ‘next hop’ becomes more difficult. Different approaches exist, for example: static routing (the ‘next hop’ is defined by configuration), epidemic routing (each node is a ‘next hop’), and ‘spray- and- wait’ (a selection of nodes is the ‘next hop’).

In addition, study has been made of different Quality of Service (QoS) approaches in DTN.

3.12.3.2 Plans for the year 2 and the year 3

For M22 we foresee the design, development and verification of a stand-alone vehicle demo that satisfies the requirements put forward in the use case. This includes the design, development and verification of the following components:

- A new DTN routing implementation.
- A new QoS implementation for efficient handling of data.
- A DTN video and message proxy using standardized interfaces.

For M30 we foresee integration in the demo 2 of UC4, where integration takes place with components of other partners. After integration we foresee verification and validation in the use cases. We aim for TRL5.

We plan to measure component characteristics, and compute metrics and KPIs according to the Table 3.25, and so contribute to evaluate the achievement of the project objectives O1 and O3.

Table 3.25: KPI and metric of the component WP5-20-TNL

№	KPI	#	Metric	MO
1	Ease of integration	1	Nr. of integration and test hours for an experienced software engineer.	MO1.1 MO1.3
2	Robustness of communication	2	Fraction of packets not arrived at destination relative to mission duration.	MO3.2
3	Lightweight communication	3	Added delay of the robust communication framework over a single hop.	MO3.1 MO3.2
3	Lightweight communication	4	Throughput of the robust communication framework over a single hop.	MO3.1 MO3.2

№ = Component wide KPI number KPI = KPI description
# = KPI wide metric number Metric = Metric description
MO = Measurable Outcome (see Table 1.5) that the metric refers to and refines

4 Conclusions

The safety of commercially deployed UAS relies on secure remote operations, which is feasible only if individual components as well as communication data links are secured. Additionally, communications need to provide sufficient reliability for timely transmission of avionics and command and control data in BVLOS scenarios.

This deliverable provides high-level designs for 21 components that can be used to build systems for trusted communication, with plans for further refinement of the design and implementation in the WP5 tasks, and for integration with the solutions the other work packages will provide. For each component, in addition to the technical design, the planned maturity level of the technology is stated (in terms of TRL levels) together with the intended measurable outcome for verification and validation.

The proposed system functions and U-Space components each provide for advancement over the state of the art within an architectural context adhering to aviation industry standards and best practices.

The designs presented in this document follow a common overall structure, referred to as the architectural canvas (Figure 1.1), which enables a shared understanding of the solutions to the challenges of designing systems for trusted communication, and which provides the basis for the further refinements of the designs to be carried out and reported in the deliverable D5.2 “Architecture for Communications and Security – Final version” that will be released at M30.

5 References

Ref.	Title
[1]	ECSEL ED 2018.181 RIA proposal: **COMP4DRONES** - SEP-210524490, 20/09/2018
[2]	ECSEL ED 2018.181 RIA proposal: **COMP4DRONES** - SEP-210524490, 03/12/2019 (Full Project Proposal amendment)
[3]	D1.1 Specification of Industrial Use Cases (**COMP4DRONES** report)
[4]	D2.1 Framework Specification (**COMP4DRONES** report)
[5]	D3.1 Specification of Integrated and Modular Architecture for Drones (**COMP4DRONES** report)
[6]	European ATM Master Plan: Roadmap for the safe integration of drones into all classes of airspace. SESAR Joint Undertaking. https://www.sesarju.eu/sites/default/files/documents/reports/European%20ATM%20Master%20Plan%20Drone%20roadmap.pdf
[7]	Parmenter, D. (2020). Key performance indicators: developing, implementing, and using winning KPIs (4th ed.). John Wiley & Sons.
[8]	L. F. Rivera, N. M. Villegas, G. Tamura, M. Jiménez, and Hausi A. Müller, “UML-driven automated software deployment,” In Proceedings of the 28th Annual International Conference on Computer Science and Software Engineering (CASCON ’18). IBM Corp., USA, 2018, 257–268.
[9]	C. Berger, B. Nguyen and O. Benderius, "Containerized Development and Microservices for Self-Driving Vehicles: Experiences & Best Practices," 2017 IEEE International Conference on Software Architecture Workshops (ICSAW), Gothenburg, 2017, pp. 7-12, doi: 10.1109/ICSAW.2017.56.
[10]	https://suricata-ids.org/
[11]	https://csrc.nist.gov/projects/security-content-automation-protocol
[12]	https://wazuh.com/
[13]	https://www.elastic.co/
[14]	https://hive.apache.org/
[15]	https://www.docker.com/
[16]	https://docs.docker.com/engine/swarm/
[17]	https://kubernetes.io/
[18]	Shames, Peter M., and Marc A. Sarrel. "A modelling pattern for layered system interfaces." *INCOSE International Symposium*. Vol. 25. No. 1. 2015.
[19]	Koubaa, Anis, et al. "Micro air vehicle link (mavlink) in a nutshell: A survey." *IEEE Access* 7 (2019): 87658-87680.
[20]	“STANAG 4586 Ed.3 Nov 2012, Standard Interfaces of UAV Control System (UCS) for NATO UAV Interoperability, NATO Standardization Agency (NSA), 2012” available from http://everyspec.com/NATO/NATO-STANAG/download.php?spec=STANAG_4586_ED-3_09NOV2012.051642.pdf
[21]	ISO/IEC, 25010:2011 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SquaRE) — System and software quality models, Int’l Organization for Standardization, 2011.
[22]	Shafiee E., Mosavi M. R., Moazedi M. *Detetion of Spoofin Attacks using Machine Learning based on Multi-Layer Neural Network in Single-Frequency GPS Receiver*. The Journal of Navigation, pp 1-20, 2017.
[23]	http://tarekamr.appspot.com/pdfs/tsdm2012.pdf
[24]	https://towardsdatascience.com/a-brief-overview-of-outlier-detection-techniques-1e0b2c19e561

[25] Novak A., Havel K., Bugaj M. Measurements of GNSS Signal Interference by a Flight Laboratory. Transportation Research Procedia, pp. 271-278, 2018.

[26] Khan A.M., Iqbal N., Khan M.F. Synthetic GNSS Spoofing Data Generation Using Field Recorded Signals. MethodsX, pp. 1272-1280, 2018.

[27] Lemmenes A., Corbell P., Gunawardena S. Detailed Analysis of the TEXTBAT Datasets Using a High Fidelity Software GPS Receiver. International Technical Meeting of Satellite Division of the Institute of Navigation, 2016.

[28] Humphreys T., Bhatti J., Shepard D., Wesson K. The Texas Spoofing Test Battery: Toward a Standard for Evaluating GPS Signal Authentication Techniques. Radionavigation Laboratory Conference, 2012.

[29] Maksuti, S., Tauber, M. and Delsing, J., 2019, October. Generic Autonomic Management as a Service in a SOA-based Framework for Industry 4.0. In IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society (Vol. 1, pp. 5480-5485). IEEE.

[30] D. Derler, K. Gellert, T. Jager, D. Slamanig, C. Striecks. „Bloom Filter Encryption and Applications to Efficient Forward-Secret 0-RTT Key Exchange”, Journal of Cryptology, Springer 2020.

[31] V. Cini, S. Ramacher, D. Slamanig, C. Striecks. „CCA-Secure (Puncturable) KEMs from Encryption With Non-Negligible Decryption Errors”, In submission.

[32] L. Hanzlik, D. Slamanig. “With a Little Help from My Friends: Practical Anonymous Credentials”, In submission.

[33] S. Schäge, J. Schwenk, S. Lauer. „Privacy-Preserving Authenticated Key Exchange and the Case of IKEv2”, 23rd IACR International Conference on Practice and Theory of Public-Key Cryptography, Lecture Notes in Computer Science 12111, Springer 2020.

[34] K. Scott, and S. Burleigh. "RFC5050 Bundle Protocol Specification." (2007)

[35] S. Symington, S. Farrell, H. Weiss, and P. Lovell. "RFC6257 Bundle Security Protocol Specification." (2011)

[36] M. Demmer, J. Ott, and S. Perreault. "RFC7242 Delay-Tolerant Networking TCP Convergence-Layer Protocol." (2014)

[37] Mahoney, Erin. "NASA - Disruption Tolerant Networking." (2016), url: https://www.nasa.gov/content/dtn