AIDA study case

Architecture description
System version : V4.5

IRT Saint Exupéry

www.irt-saintexupery.com

Table

1. Introduction

This document aims at synthetizing all the system architecture activities for the AIDA system.

It is organized according to the different layers of the Arcadia method deployed in the Capella tool. Whenever it is relevant, diagrams from the Capella model of the AIDA system are integrated and commented.

This document does not contain exhaustively the information represented in the model. For a better understanding, we recommend to open the Capella model aside of this document.

1.1. Abbreviations and acronyms

AIDA	Aircraft Inspection by Drone Assistant
BVLOS	Beyond Visual Line Of Sight
EasyMOD	
MOISE	
PDRA	Pre-Defined Risk Assessment
PFC	Pre-Flight Check
S2C	System and Safety Continuity
STS	Standard Scenario
UAS	Unmanned Aircraft System
VLOS	Visual Line Of Sight

1.2. Reference documents

[1] Easy Access Rules for Unmanned Aircraft Systems: https://www.easa.europa.eu/document-library/easy-access-rules/online-publications/easy-access-rules-unmanned-aircraft-systems

1.3. AIDA system

AIDA is a study case used by the Systems Engineering department of IRT Saint Exupéry as a concrete example for the methods and tools developed in the frame of various projects. It has been created for the needs of the MOISE project and will be used for other projects such as S2C and EasyMOD.

This document will not present the contents and results of these projects. The aim is only to present the system architecture and to explain the design choices. It is a complement to the open-source Capella model.

The proposed system aims at assisting civil aircraft Pre-Flight-Checks with a drone-based system (or UAS).

Although the aim is not to develop a real drone system, the architecture and design aim at being as realistic as possible. However, specialized studies (performances, mechanical,…) will not be conducted, unless it shows interest in the context of a particular project.

1.4. Versions management

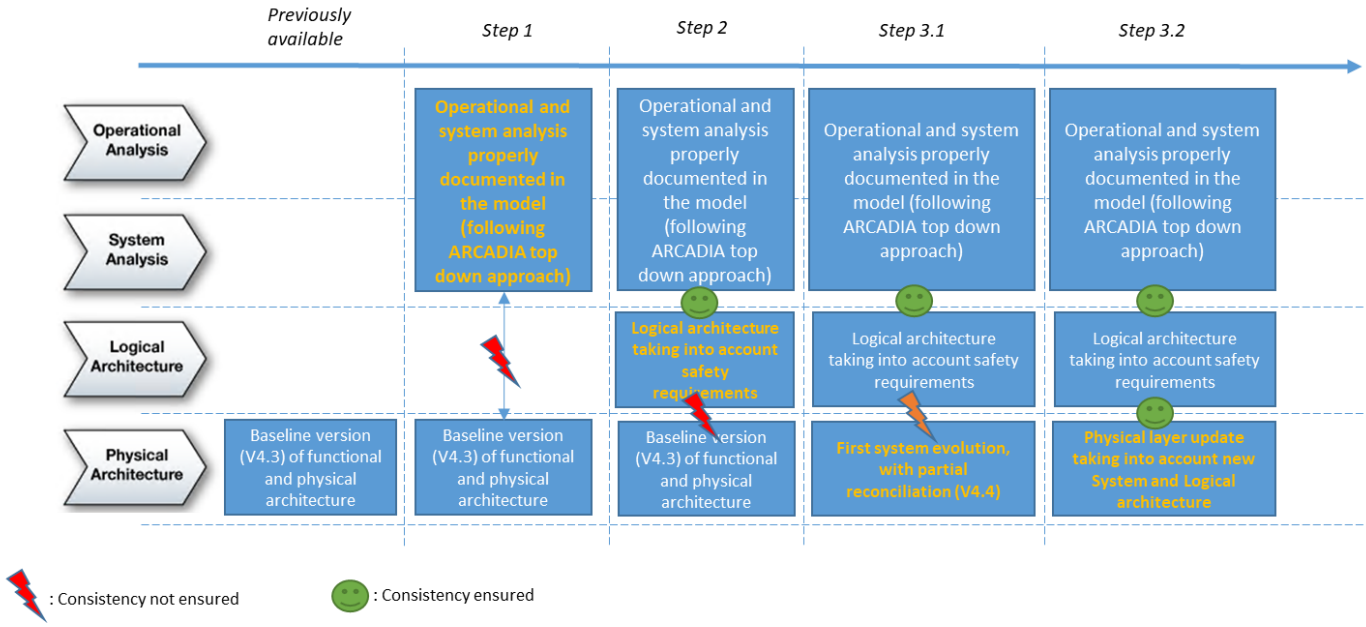
The creation of this document corresponds to the realization of the “top-down” analysis as recommended by the Arcadia method : Operational Analysis →System Analysis →Logical Architecture →Physical Architecture. However, the physical architecture already existed, as an inheritance of the MOISE project. Indeed, the current system version when this document has been created was V4.3.

Two approaches were possible :

- Perform the « Bottom-up » analysis to fill the upstream layers that correspond exactly to the existing physical architecture.
- Perform the « Top-down » analysis without taking into account the existing architecture.

In order to stay in the “spirit” of the Arcadia method, the second approach has been chosen, but keeping in mind the high level characteristics of the existing architecture : the system is constituted of a quadri-rotor electric drone carrying a payload, a remote control and a control desk. In this way, the results of the first modeling phases are not too far from the existing architecture, and the reconciliation will not be too difficult. This reconciliation will be considered as “system” evolutions, leading to new versions of the system.

This strategy is summed up on the table below :



Currently, the step 3.1 has been performed. Therefore, the physical layer (in version V4.5) is closer to the other layers than 4.3**, but the full consistency is not ensured.**

2. Operational analysis

Consistently with the Arcadia method, this chapter describes and characterizes the capabilities and activities that the various stakeholders realize, independently from the system of interest. The goal is to formalize the needs without focus on the solution implementation.

2.1. Pre-Flight Check operations

2.1.1.Civil aircrafts inspections

In order to ensure their airworthiness and safe operations, civil aircrafts are frequently and regularly inspected. Different kinds of inspection are planned along the aircraft lifecycle, ranging from the Pre-Flight Check (or Walk Around) visual inspection before each flight to the heavy “D check” that takes place every several years.

Drone-based solutions have already be commercialized to perform visual inspections during “heavy” check operations that are usually performed indoors, thus avoiding many regulation constraints (although it raises some technical issues, such as GPS unavailability).

The purpose of AIDA is rather to assist ground maintenance operators to perform the light systematic PFC that takes place before each-flight directly at the airport gate.

2.1.2.Pre-flight Check procedures

The Pre-Flight Check procedure consists in a systematic visual inspection of the aircraft before the flight, in order to detect any conditions that can compromise the safety of the flight. An example of PFC procedures for A320 is available below (or http://equicom.net/mcdu/blog20150714.php):

TheWalkAroundA320.docx

Several points of attention can be highlighted:

- The great majority of operations are visual inspection. However, a physical intervention is sometimes necessary (ex: ensure the engine fan can rotate)
- Current procedures only focus on « bottom » parts of the aircraft, or parts that are visible from the ground.
- Some checks necessitate an access to physically constrained areas (landing gears)

The PFC procedure allows also the detection of ice on the aircraft surfaces (wings…), in which case the de-icing of the aircraft before flight is requested to the airport ground teams.

PFC duration is of importance. although it may be performed in” masked time” (at the same time as other operations), it must not increase the Turn Around Time which is a critical phase of an aircraft operation. Any delay has direct impacts for airlines and airport (take-off / landing timeframes, flight connections, passengers financial compensations…).

2.1.3.Environment

The typical environment of PFC procedures is the airport gate, in which the aircraft is parked between flights. Many other operations take place on a reduced timeframe: passengers and cargo loading and unloading, fuel servicing, cleaning, food servicing, de-icing,… . It involves a potentially important number of persons and systems around the aircraft at the same time. Also, other aircrafts may be parked nearby and go through similar operations at the same time.

Of course the parking gate is an “outdoor” environment, meaning the PFC can be performed during the day or the night, possibly in adverse weather conditions (rain, wind, low temperature…).

Airports are also a busy airspace, with an important concentration of aircrafts evolving at low altitudes in critical flight phases (take-off, landing). Therefore the traffic of any flying object is strictly controlled and subject to high standard safety requirements. A trend to have more and more drone-assisted operations is probably to be expected.

Finally, security-related topics (mainly related to recent terrorism events) are a top-priority concern in airports environment.

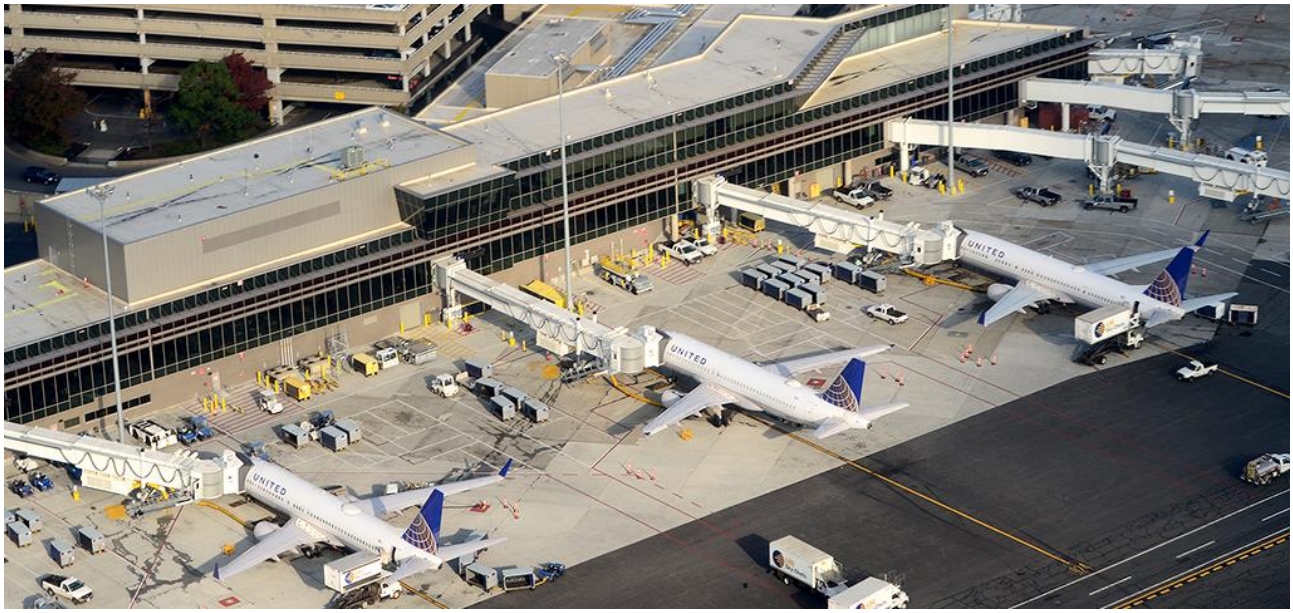


Figure 1: airport gates

2.2. Operational modelling

As the operational layer in Arcadia method consists in characterizing the mission and activities to be performed, and have a first overview of the environment, the system of interest does not appear in the diagrams.

The operational modelling results are the Operational Capabilities diagram the Operational Entities diagram, which show the stakeholders (persons or systems) and the associated capabilities and activities:

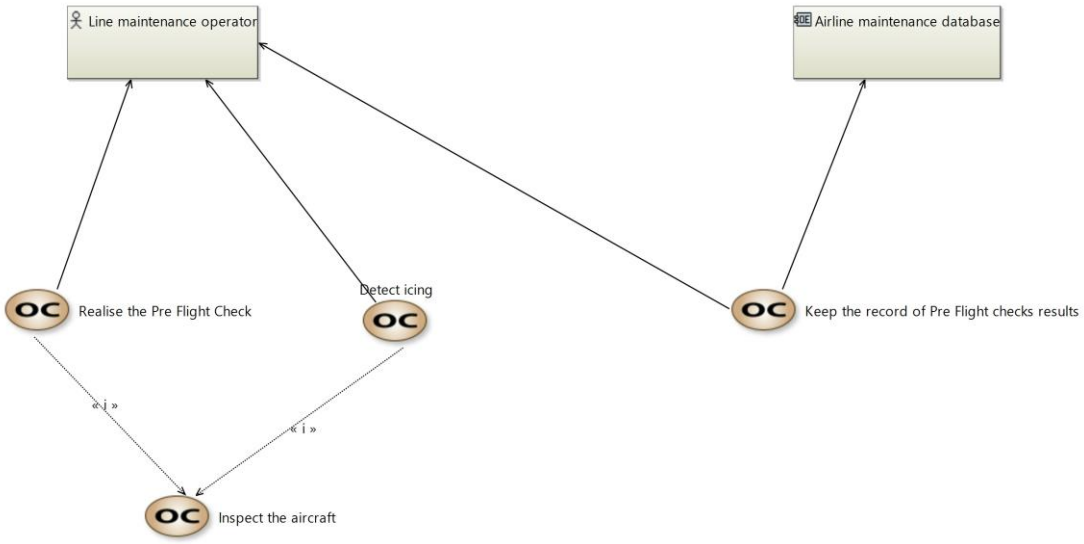


Figure 2: Operationnal capabilities diagram

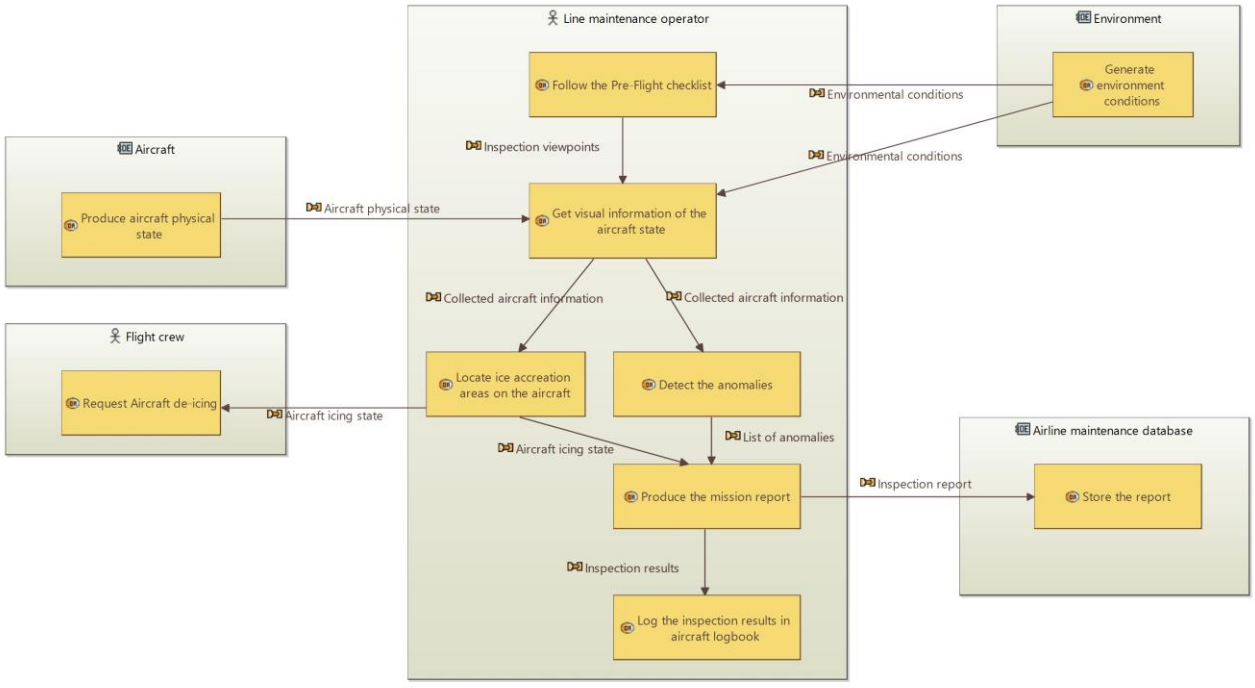


Figure 3: Operational entities diagram

The model also contains the association between activities and capabilities, which can be represented in the form of Operational Activities Interaction diagrams. These are part of the Capella model but are not included in this document as they act mainly as an intermediate step.

Comments:

- Without surprise, the diagram shows that the maintenance operator is the central entity of the PFC procedure. He is of course the « target » of the AIDA system, which will take in charge some of his activities.
- In some cases (small airport, reduced staff, airline policy…) the PFC may be performed directly by the Flight crew. However, the procedure is similar.
- It is not clear that current PFC procedures produce a specific report, or if deviations are only logged in the aircraft logbook. In the case of AIDA, it is very likely that a report is produced and stored for history and traceability.

2.3. Limitations of current procedures and interests for a UAS-based procedure

PFC are currently performed directly by humans (flight crew or maintenance crew). Taking into account human factors has been an important field of improvement in air transport safety. An assisted procedure is therefore of interest: the analysis is more objective and independent from human factors.

Practical interests are also foreseen: access to high parts of the aircraft (upper surfaces of lifting parts, upper probes,…), reduce operation time,…

2.4. Needs and Key Design Drivers

Establishing a complete set of needs for the AIDA system would be a time-consuming activity and would necessitate exchanges with stakeholders. However, this activity would be necessary in the frame of a real development of the AIDA system.

In order to give orientation to the design of the AIDA system and to make it somehow realistic, we give here a set of “Key Design Drivers”, or design objectives:

- Inspection time: compatible with current Turn Around Time, not worse than for current PFC procedures.
- Have at least the same inspection capability of inspection as a human. Capacity to improve procedures to be identified.
- Confidence in results: precise detection of defaults, limited rate of false detection.
- Operator skills: basic piloting skills (the operator is not a professional drone pilot), capacity to handle normal operations and abnormal situations which do not require advanced fault-management skills. Typical training: a few days.
- Safety: comply with related regulation (see System analysis chapter)
- Availability: no hard constraint, as the PFC can still be performed by a human. However, an interruption of the inspection may cause a delay to the aircraft operation, with possible financial consequences for the airline.

3. System analysis

In this chapter, we describe the system as a black box without focus on internal design and implementation. This analysis focuses on the system perimeter and interactions with external systems, stakeholders and environment.

Concept of operations are described in the form of scenarios, covering both normal and degraded operations. These scenarios support the preliminary risk analysis (SORA) which will help derive the safety requirements (failure conditions).

The purpose is to identify the high-level functions and modes of the AIDA system, which will be developed and allocated in the next layers (logical, physical). Associated requirements are also identified.

Because the concept of AIDA is to propose a drone-based system to assist Pre-Flight Checks operations, it is assumed that the AIDA system is a UAS (Unmanned Aircraft System) composed of a flying segment (the drone), a ground segment (the ground control system) and a communication link between the segments.

3.1. Lifecycle analysis

Although not an official part of the Arcadia method, a life-cycle diagram of the AIDA system is proposed below:

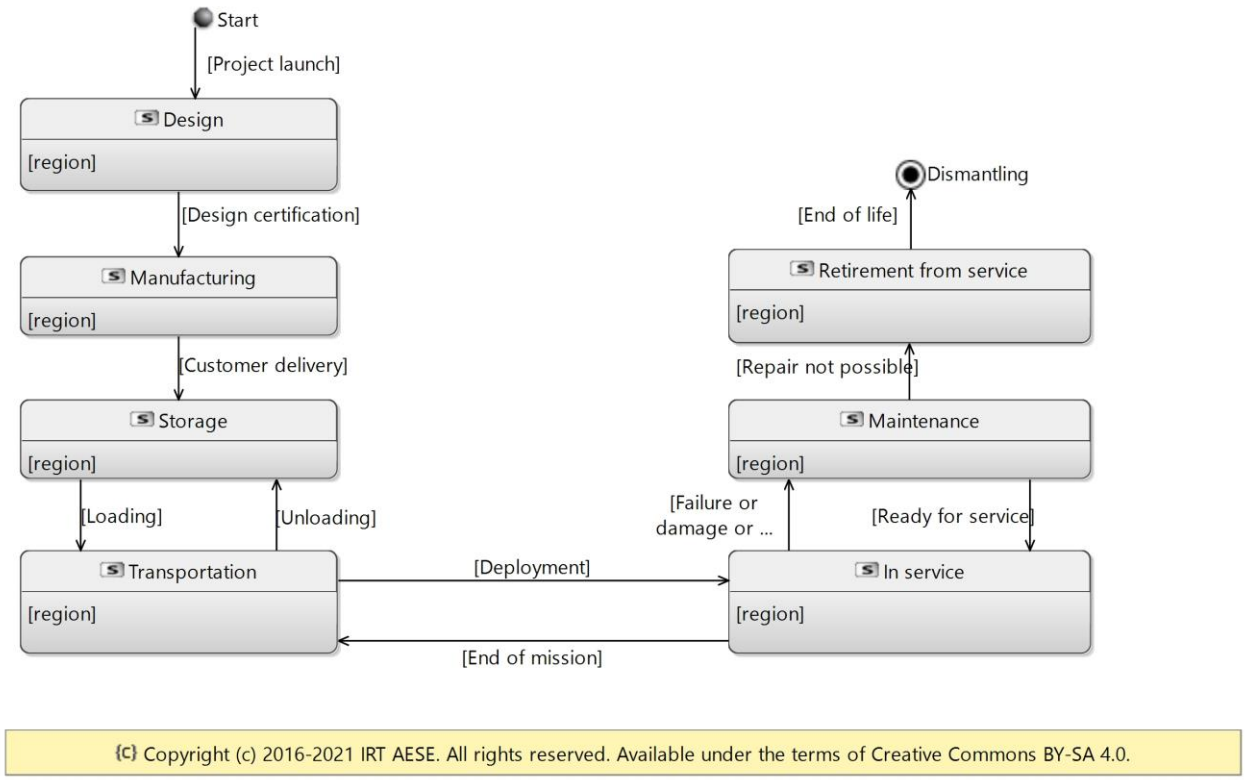


Figure 4 : lifecycle diagram

While the system is expected to fulfill its missions during the “In Service” phase, it is important to keep in mind the needs that could emerge from other life phases:

- Tests-related functionalities in Design and Manufacturing phases
- Storage and transportations constraints (mass and volume, assembly/dissassembly, conditioning, preservation…)
- Maintenance functionalities (auto-tests, diagnostic, repair capabilities,…)
- Retirement and recycling (forbidden materials…)

Apart from the Maintenance phase which can also induce functional requirements, the other phases will not be further studied.

3.2. Stakeholders and environment

A stakeholders diagram is proposed below. It identifies all the persons and entities that can have an interaction with the AIDA system in one or several life phases.

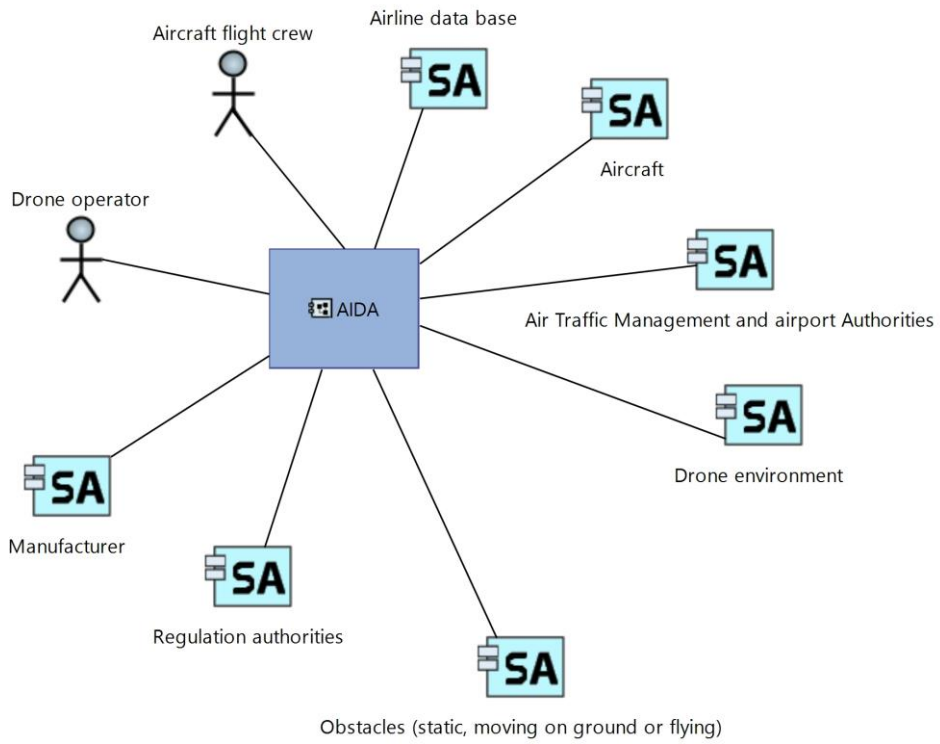


Figure 5 : stakeholders diagram

Drone operator: the person who will use the drone to accomplish the aircraft inspection. It is assumed that the drone operator is one of the ground operation team that usually perform the Walk-Around procedure. It is not a professional drone pilot, therefore only basic piloting skills are expected. However, it is assumed that sufficient training has been provided so that the operator knows how to operate the drone in its environment, covering normal and emergency operations. As a first hypothesis, for the AIDA system, the drone operator complies with the competency set described in Article 8-2 of the Easy Access Rules document (see [1]). The drone operator also intervene during maintenance phase to perform maintenance tasks on the AIDA system.

Aircraft flight crew: the flight crew is the ultimate “customer” of the AIDA system, as they will take decisions based upon the results of the inspection. The need for a direct interface between the

AIDA system and the flight crew (for example via the pilot notepad) will have to be defined, however this is not seen as a major architecture topic. So far, the flight crew does not intervene directly in the AIDA system operations.

Airline maintenance database: this database contains data that are used for inspection (flight plan definition, aircraft inspection history…) and stores the inspection report. A direct communication link between the AIDA system and this database is foreseen.

Aircraft: this is the aircraft inspected by AIDA. It "provides" visual information to be acquired by the system, and it is also a kind of "obstacle", as the drone must of course avoid any collision with it.

Air traffic management and airport authorities: being a flying object in an airport environment, it is very likely that some kind of communication will be needed between the AIDA system and ATM systems (radio communication between the operator and the control tower, identification within the ATM system, direct orders communicated from the ATM system to AIDA…). This will be further studied in the concept of operations.

Atmospheric conditions: the AIDA system is designed to be used outdoor, and therefore has to sustain possible adverse atmospheric conditions: wind, extreme temperatures, rain, snow and icing, thunderstorm… Probable operational limitations will be identified in further states of design.

Obstacles: many foreign objects can be expected in the airport gate environment. Three categories can be identified:

- Static objects: airport buildings, aircraft access gate, lamppost, static vehicles,…
- Objects moving on ground: passengers access ramps and buses, fuel truck, fret and luggage vehicles, food and servicing vehicles, de-icing vehicles, involved (mechanic/ground team, aircraft crew…) or uninvolved (passengers…) persons, other aircrafts,…
- Flying objects: other drones (although not very developed so far), flying aircrafts (not foreseen in the direct environment of the airport gate),…

The definition of Concept of Operations and procedures will be a key topic to formalize the expected capabilities of the system in terms of obstacle management.

Regulation authorities: they publish the regulation documents (see [1]) and authorize the system operations in the context of "specific" or "certified" operations.

Manufacturer: mainly involved during the design and manufacture life phases, the manufacturer can also intervene as back office all along the lifecycle.

3.3. Missions and Capabilities

Before performing the high level functional analysis, the Arcadia method proposes to establish the missions and capabilities of the system. As mentioned earlier, we will mainly focus on the "In service" and "Maintenance" life phases.

3.3.1. "In service" missions and capabilities

Three missions are identified for the "In service" life phase :

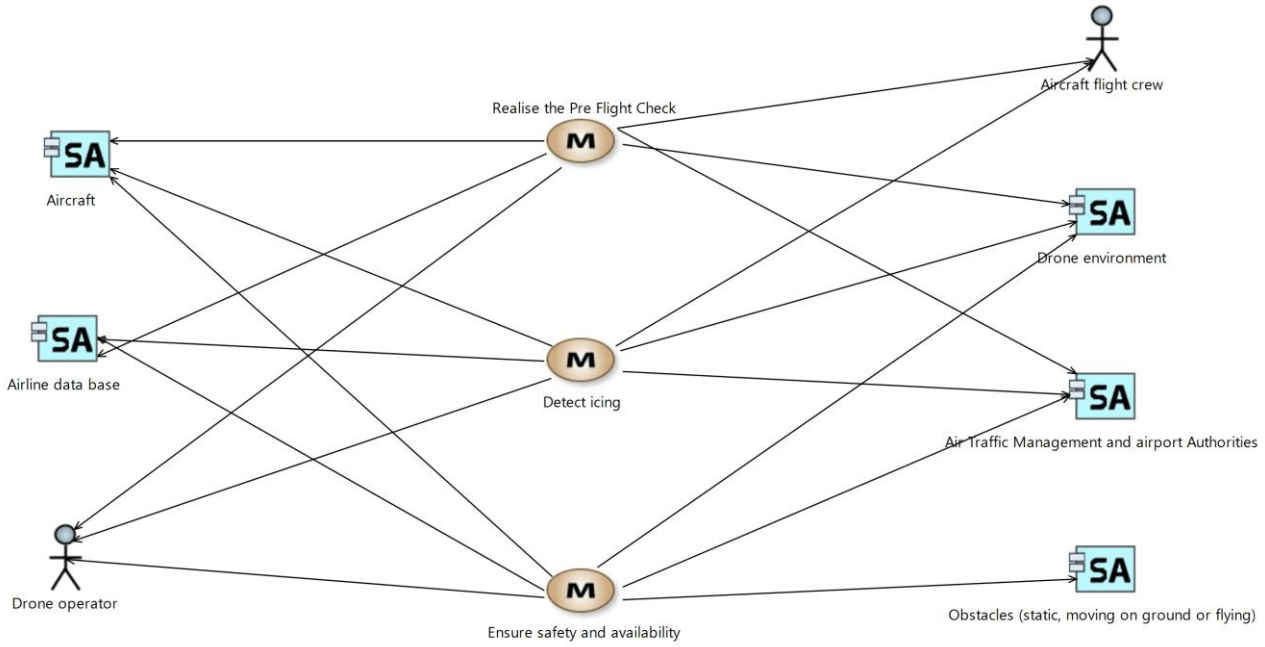


Figure 6 : "In service" missions diagram

The two first missions “Realise the Pre Flight Check” and “Detect icing” correspond directly to the same Operational capabilities (see “Operational modelling” section).

In this System analysis phase, a third mission is added : “Ensure safety and availability”. Indeed, it is expected at this point that a drone operation in an airport environment is subjected to safety constraints driven by the EASA regulation (see “UAS regulation analysis”). Also, the unavailability of the AIDA system may have operational impacts for the customer.

The capabilities associated to each missions are detailed in the Missions and Capabilities diagrams, that we will detail hereafter.

3.3.1.1. Mission “Realise the Pre Flight Check”

The capabilities of the AIDA system exploited in this mission context are the following :

- « Inspect the Aircraft » : the Drone Operator uses the AIDA system to acquire photos and videos of the aircraft, corresponding to the inspection viewpoints detailed in the Pre Flight checklist. Atmospheric conditions can have an impact on the drone operation
- « Detect the anomalies » : the Drone Operator uses the AIDA system to review the acquired photos and videos and detect anomalies on the aircraft.
- « Produce the mission report » : the Drone Operator uses the AIDA system to produce a report of the mission, which is used by the Aircraft Flight crew to take decisions (confirm or abort aircraft operation, request additional maintenance operation,…) and exported to the Airline maintenance database.

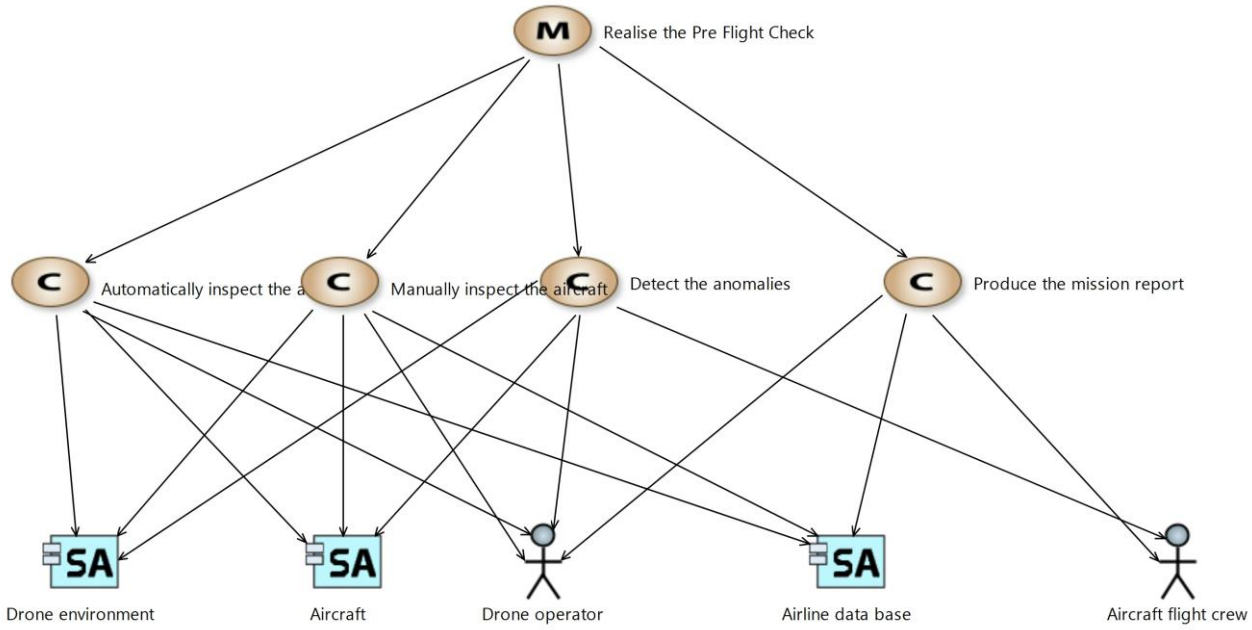


Figure 7 : "Realise the Pre Flight Check" mission and capabilities diagram

3.3.1.2. Mission "Detect icing"

This mission exploits almost the same capabilities as the "Realise the Pre Flight Check" mission. The following capability is added :

- « Locate ice accreation areas on the aircraft » : the Drone Operator uses the AIDA system to locate the areas of the aircraft which are iced and necessite a de-icing operation before the aircraft can take-off.

In this context, the mission report also contains the icing state of the aircraft, and is used by the Flight Crew to request a de-icing operation.

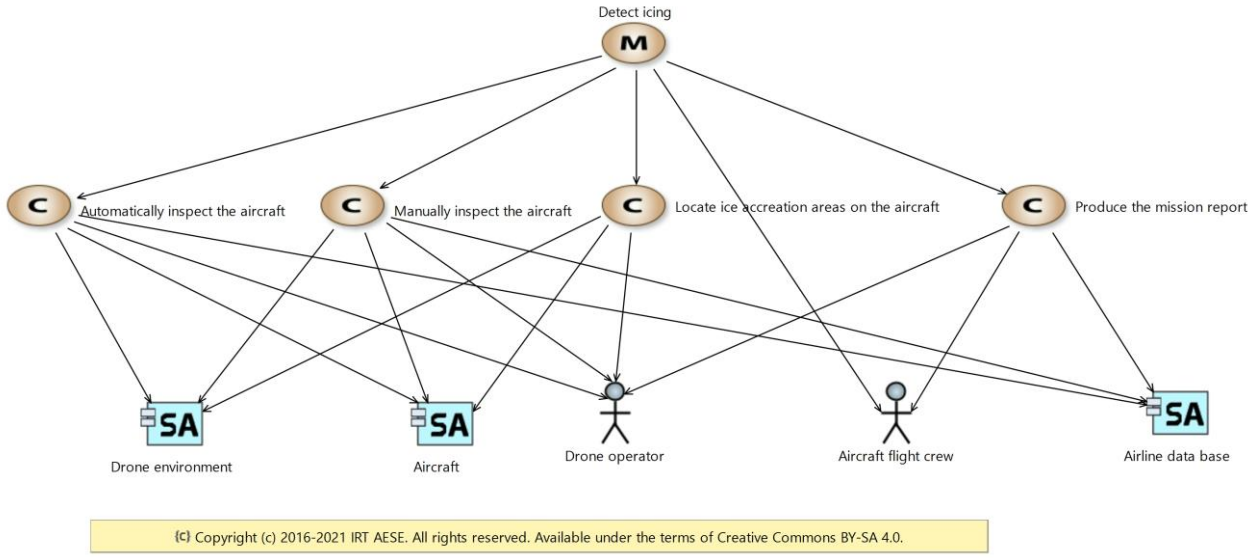


Figure 8 : "Detect icing" mission and capabilities diagram

3.3.1.3. Mission “Ensure safety and availability”

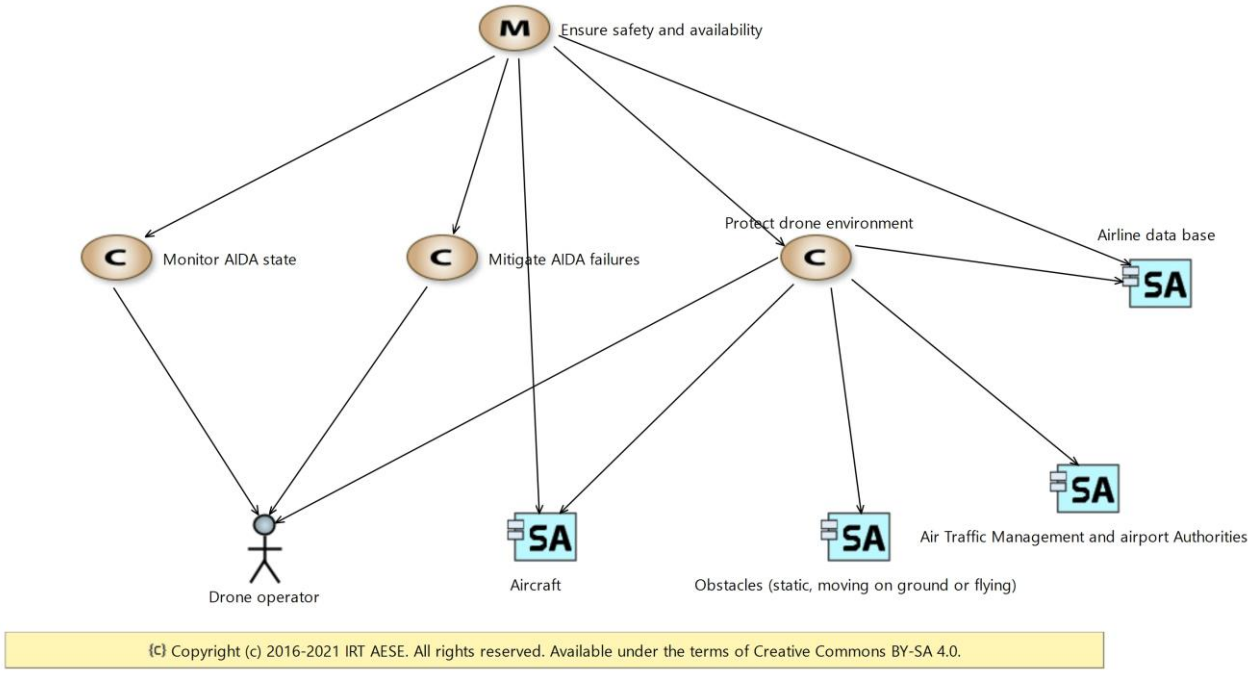


Figure 9 : "Ensure safety and availability" mission and capabilities diagram

The capabilities of the AIDA system exploited in this mission context are the following :

- « Monitor AIDA state » : the Drone Operator is involved in the monitoring of AIDA failures. He is informed of detected failures and contributes to detect failures that cannot be detected directly by the system. Detected failures are logged within the AIDA system to allow further diagnostic and maintenance operations.
- « Mitigate AIDA failures » : the Drone Operator is involved in the mitigation of AIDA failures. Depending on received alarms or unusual behaviour, the AIDA system is able to avoid undesired events or the Drone Operator takes the appropriate measures.

- « Protect drone environment » : the AIDA system operations must be safe for the drone environnement, it mainly consists in avoiding collisions with the Aircraft or with foreign obstacles (which can be involved or uninvolved persons). ATM system and airport authorities are involved.

3.3.2.« Maintenance » mission and capabilities

The mission and associated capabilities in « Maintenance » life phase are identified on the diagram below :

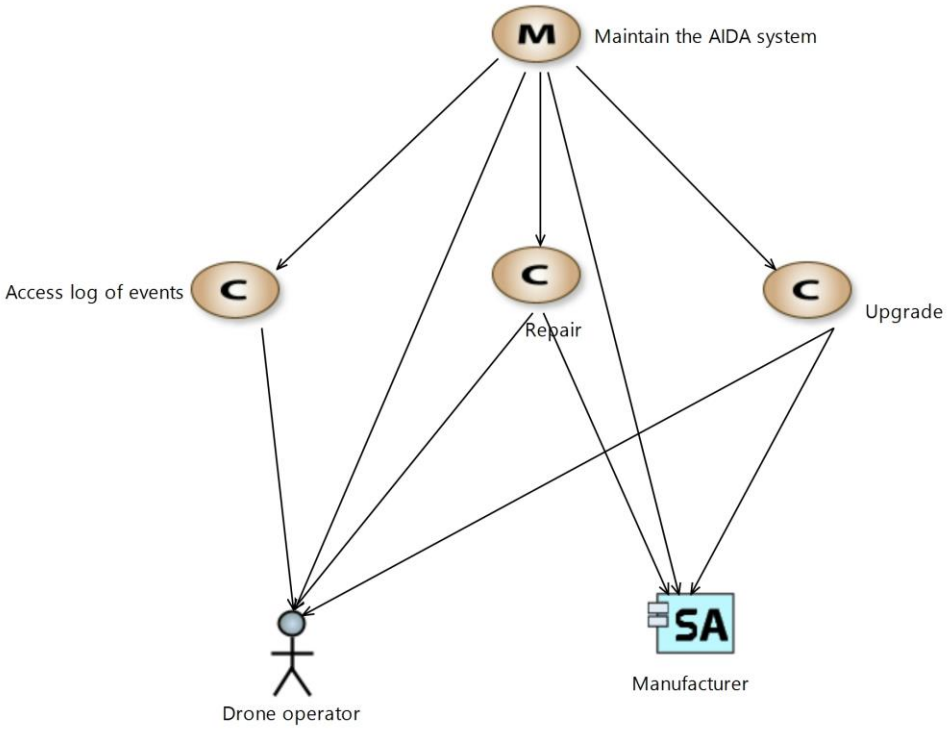


Figure 10 : Maintenance phase mission and capabilities diagram

The capabilities of the AIDA system exploited in this mission context are the following :

- « Access log of events »: the Drone operator access the internal log of events which contains the list of encountered detected failures. When the maintenance operation is performed, the operator cleans the log.
- « Repair »: the Drone operator repairs the AIDA system, which may requires spare parts and/or support from the manufacturer.
- « Upgrade »: the Drone operator upgrades the AIDA system by replacing old or obsolete components or software by new ones supplied by the manufacturer/supplier.

3.4. UAS regulation analysis

The purpose here is to identify technical constraints for the architecture and design that can be derived from the regulation texts (mainly the safety related requirements). One should note that beside the safe design of the UAS, the regulation texts cover a large variety of topics including operations, pilot training, etc.

UAS regulation are evolving quickly with the development of new UAS technologies and usages. An up-to-date and easy to read version of the EASA European regulation is available (see [1]).

According to the EASA view, UAS are classified under the three following categories associated to a risk level (see articles 3 to 6 for details): “Open”, “Specific” and “Certified”.

While the classification of AIDA in the Open category can directly be ruled out (airport environment, possible BVLOS, populated area,…), a more detailed analysis would be required to identify if a classification in the Specific category is possible or if the AIDA system shall be certified according to standard aeronautical processes. Indeed, AIDA does not fall directly under the criteria of the “certified category” (see article 6-1.):

- Although populated, the airport gate environment area with the person usually evolving within is not an « assembly of people » in the terms of article 2 definition
- AIDA does not transport people
- AIDA does not transport dangerous goods

However, article 6-2 states that a certification process may be needed if the risk assessment considers that the risk mitigation strategy is not sufficient.

For both of these categories, an Operational Risk Assessment is required in order to derive safety objectives and requirements. This risk assessment process will particularly help establish if a certification process is required. The SORA process is proposed in article 11.

Alternatively to the complete SORA process, EASA proposes the possibility to use Standard Scenarios and Pre-Defined Risk Assessment, when the Concept of Operation correspond to a common and well known situation and when the risk, although exceeding the “open” category, is considered low.

In the latest amendment (2020/639), the following STS and PDRAs are available :

- STS-01 : VLOS, below 120m in urban environment, with UA MTOM<25kg
- STS-02 : BVLOS, below 120m in sparsely populated area environment, with UA MTOM<25kg. Range <2km when using airspace observers (AO), otherwise <1km
- PDRA-01 : BVLOS, below 120m, in controlled airspace over sparsely populated area, with UA MTOM<25kg

It shows directly that AIDA operations are more at risk that the situation corresponding to these current STS and PDRA (BVLOS in airport environment, in possibly populated environment).

Link with standard manned civil aircraft safety assessment process:

Keeping in mind that the purpose of the AIDA study case is to illustrate the methods deployed in the frame of IRT-Saint Exupery’s projects, for which the main beneficiaries are the aerospace industry actors, the development of the AIDA system will be in anyway conducted according the usual aeronautical guidelines (mainly ARP4754 and ARP4761).

In the civil aeronautical context, we have several decades of history and it is assumed that airframers have a very good knowledge of aircraft functions and associated failure conditions. This is not the case for UAS, therefore it is interesting in the AIDA context to formalize clearly the Concept of Operations and to perform (at least partially) the SORA process to identify the high level failure conditions which will then be used as an input for the standard safety assessment processes (FHA, PSSA/SSA,…).

The pre-requisite for the SORA process is to define the Concept of Operations, which consists mainly in the foreseen operational scenarios in normal and abnormal conditions (failure, external threat, adverse conditions…). These scenarios will be formalized in the next chapter.

Preliminary list of functions derived from regulation

Although the AIDA system and operations are not compatible with current CE marking classes, the following functions (which are part of some classes requirements) are judged relevant for the AIDA system and will be considered in the design :

- Altitude limitation and geo-awareness function, which prevent the drone from exiting the predefined flight zone (C1 : (3) and (13) ; C2 : (3),(15) and (16) ; C3 : (2),(10) and (11)
- Direct Remote identification, which broadcast information about the drone (C1 : (12) ; C2 : (14) ; C3 : (9))
- Flight termination device, which shuts off the propulsion and ensure a safe « low-energy » landing. The usual solution consists in cutting off the electric power supply and triggering the opening of a parachute system (C5 : (5))

3.5. Concept of operations

This section focuses only on the “In Service” life phase. As mentioned in the Lifecycle Analysis section, other life phases must be considered when establishing the list of needs for the AIDA system, but the large majority of functionalities and constraints concern mainly this “In Service” phase.

Inspection modes :

Two main inspection modes are foreseen:

- Manually inspect aircraft exterior
- Automaticaly inspect aircraft exterior

In the manual inspection mode, the drone follows in real-time the pilot consigns: the pilot commands directly the drone displacements and triggers the photo and video acquisition. Video flow feedback is available so that the pilot is able to position the drone adequately.

Automatic operation concept :

Several philosophies are possible for automatic inspection:

- « Blind » drone: the drone follows a predefined flight plan (set of positions and viewpoints + trajectory to be followed between positions) and knows the predefined obstacles, but is not aware of new coming obstacles
- « Fully Autonomous » drone: the drone « knows » the « points of inspection », recognizes them directly on the aircraft and determine automatically his flight plan, while detecting and avoiding obstacles.
- Intermediate concept: the drone follows a predefined flight plan but is aware of its environment (obstacles) and can adapt the flight plan to select an alternate trajectory or a back-up position/viewpoint.

The “Blind” concept, selected as the current baseline of the AIDA system (V4.3) is technically realistic as it does not requires advanced real-time mapping, picture recognition and obstacle detection capabilities, but quite optimistic or very constraining from an operational point of view. It necessitates that the flight zone does not evolve all along the process, therefore any other operations around the aircraft must be suspended. This is not compatible with the usual environmental context in which the PFC is performed.

The “Fully Autonomous” concept seems to be very challenging technically, as it requires advanced sensing and algorithmic capabilities. In order to keep the AIDA system realistic compared to the State of the Art of such systems, it is not selected as the baseline concept.

The “intermediate” concept is proposed as the new baseline concept for the AIDA system. Compared to the current baseline (V4.3), it mainly consists in adding an obstacle detection and avoidance functionality.

Scenarios are represented under the form of sequence diagrams, directly in the Capella model. The model contains both Function scenarios and Exchange scenarios, however for the sake of visual simplicity only the Exchange scenarios are presented in this document.

3.5.1.Manual inspection

We describe here the sequence of events during a manual inspection operation assisted by the AIDA system.

System state: At the beginning of the operation, the system is supposed to be assembled, in perfect functioning state (no detected or undetected failure), completely energy loaded. The drone is set on the ground, in a ready-to-take-off position.

Power-up and initialization phases are not described here, as they are complex to model and not very interesting for the architecture study.

Environment:

- On ground: normal airport gate environment, with possible moving vehicles and uninvolved persons
- In air: normal airport traffic. No expected flying aircraft in the direct environment of the gate (ensured by radio contact between the drone operator and the Traffic management authority, so that no other flying objects will be encountered inside the flight zone).

Normal sequence:

- The drone is powered and ready to take-off.
- The system starts broadcasting Remote Identification data which are received by the Airport Control Tower to identify drone operations.
- The operator retrieves the flight zone in the external database and uploads it to the drone (see “Flight zone containment” paragraph below).
- The drone checks that its current location is contained in the flight zone. Powering the actuators is forbidden as long as the flight zone is not uploaded and validated.
- When the drone is ready for take off, the operator contacts the control tower and request the authorization to fly the drone.
- The operator triggers the drone take-off and moves the drone to the first viewpoint. Video flow is provided to the operator on a screen display. When the operator releases the commands, the drone keeps its position and altitude (« Loiter mode »).
- The operator triggers video/photo acquisition.
- The operator moves the drone to the next viewpoints and triggers the video/photo acquisition.

- When the last viewpoint has been reached, the operator brings back the drone and lands it to its launch position.
- The operator contacts the control tower to indicate the end of the drone operation.
- The AIDA system analyses the photos and videos and detects the failure and the icing state of the aircraft, and produce the inspection report.
- The AIDA system stores the inspection report in the airline database.

Flight zone containment:

Given the criticity of the airport environment, a geo-caging functionality is considered, assuming for example that the coordinates of the flight zone corresponding to the particular airport gate in which the system is operated are available in the database (external or internal) and can be retrieved before the mission. Several level of alarms can be identified:

- Alarms to the operator when the drone approaches the fence
- Automatic drone stop when the fence is reached:
    - Operator commands in a direction that leads the drone outside the fence are ignored. Other operator commands are taken into account
    - The fence is defined with a « safe buffer zone » to cover the distance travelled by the drone before it stops and get back to the fence (therefore depending on the drone max speed)

Obstacle management :

In manual mode, the obstacle detection and avoidance is managed directly by the pilot.

Data analysis and report generation :

Several solutions (from the most manual to the most automatic):

- The operator reviews directly the picture and indicate directly to the Flight Crew and in the logbook the results. Photos and videos are stored in the database.
- The operator associates each photo or video to a predefined inspection point, then the system analyses the photo to detect faults. A report is issued (and stored in the database along with the photos and videos) to synthetise:
    - The completeness of the inspection (« are all the inspection points covered ? »)
    - The results of the analysis: list of detected defaults, ice detection.
- The system automatically detects the inspection point associated to each picture and analyses the picture to detect faults. A report is issued (and stored in the database along with the photos and videos), with the same content as described above.

Because it has no direct impact on AIDA architecture, the topic of photos and videos post-processing and inspection report building is not studied in deep at this stage.

The global sequence is represented on the sequence diagram below (referenced sequences are presented below the global sequence) :

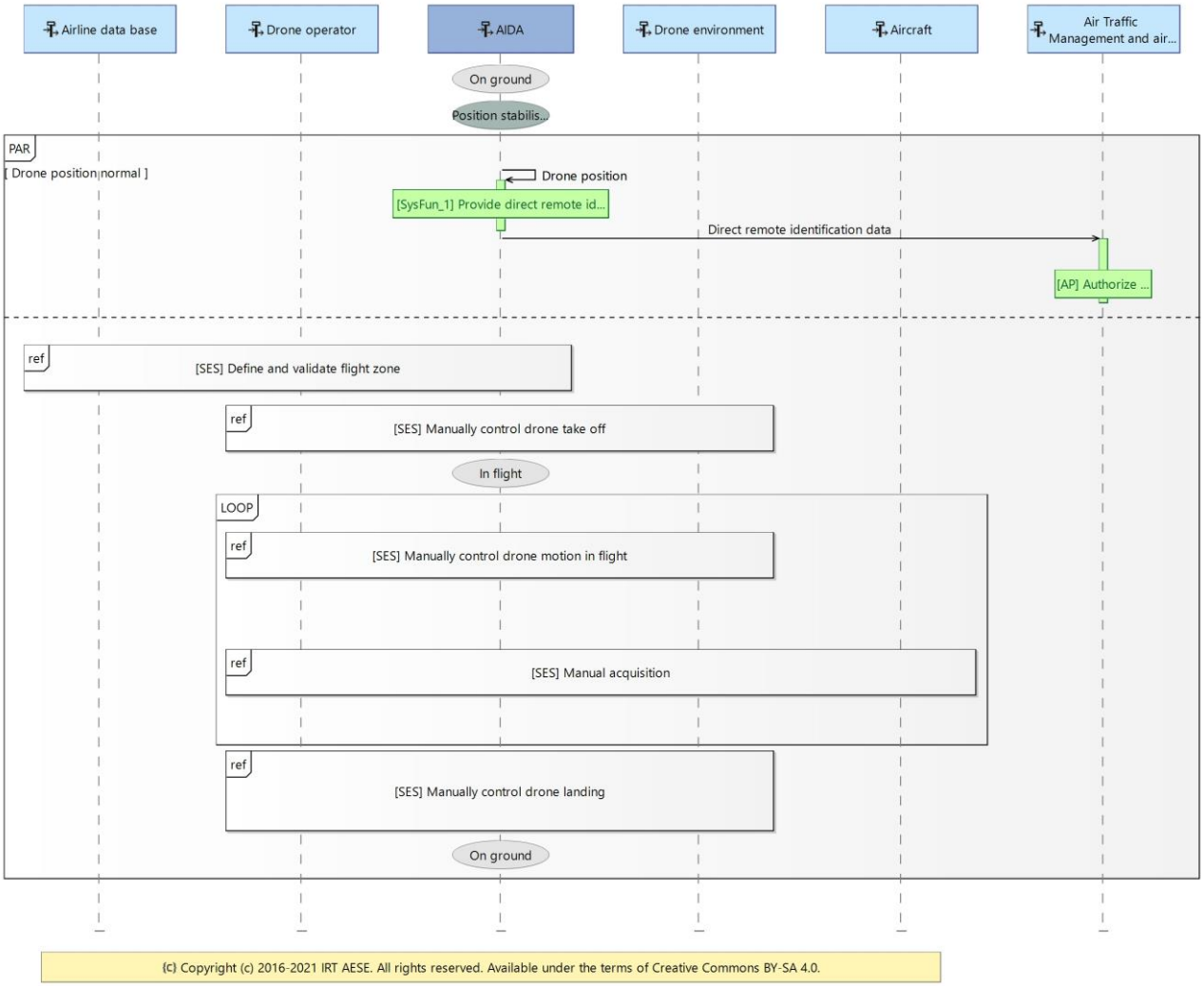


Figure 11 : Manual inspection sequence

3.5.1.1. "Define and validate flight zone"

In this sequence, the operator first selects the appropriate flight zone associated to the airport gate, which the AIDA system retrieves in the airline database, then confirms that the correct flight zone has been selected.

Then, the system checks that the drone is situated inside the flight zone.

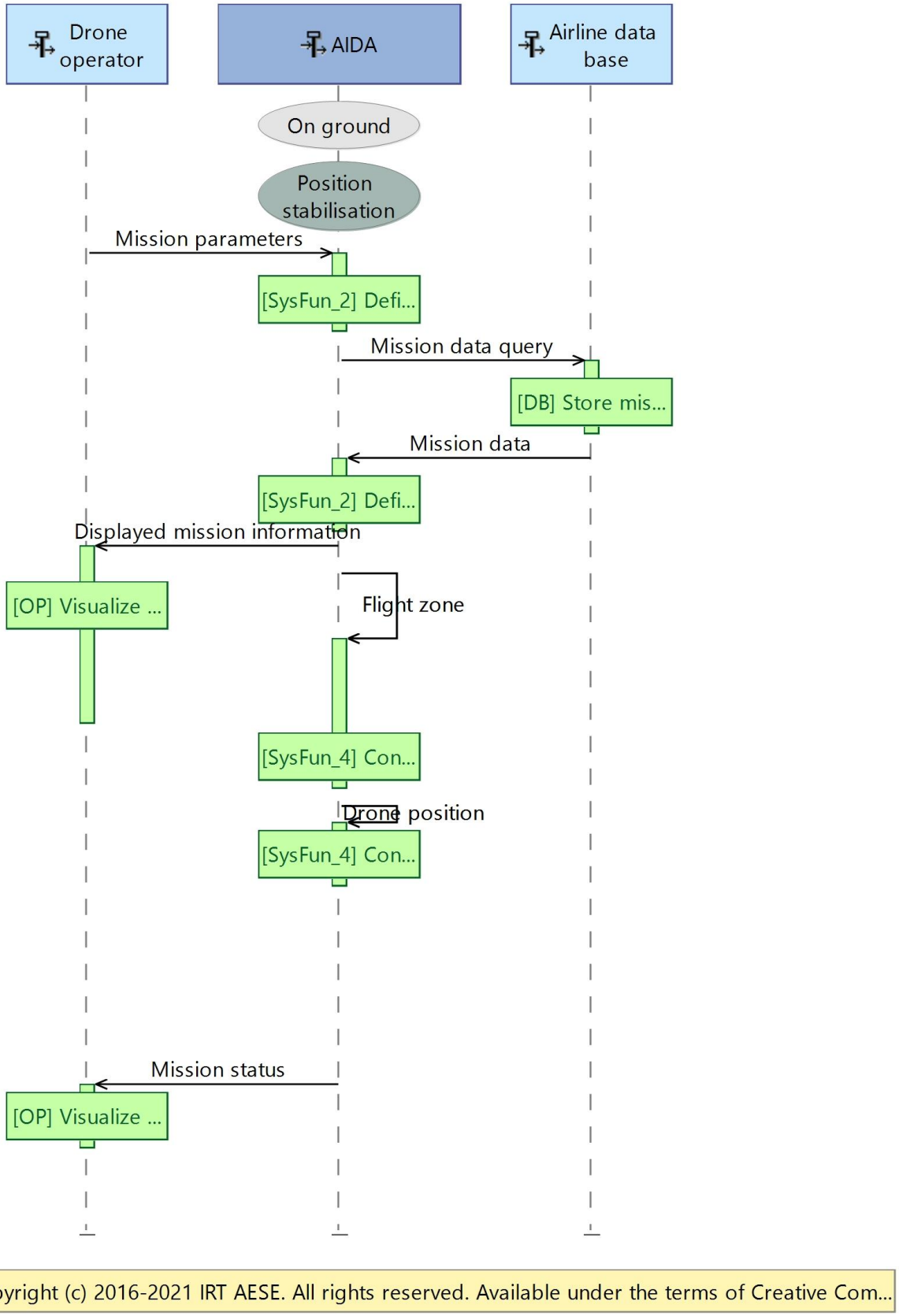


Figure 12 : "Define and validate flight zone" sequence

3.5.1.2. "Manually control drone take off" sequence :

After the "Flight zone definition and validation" sequence, the drone is ready for take off. The operator contacts the control tower to inform them of the beginning of the flight and waits for clearance (this is not represented here, as it does not involves directly the AIDA system).

After initialization, the "Position Stabilisation" (corresponding to manual position control) is selected. The pilot sends commands to trigger the drone take-off, which involves drone attitude and position information. At this stage, atmospheric conditions (wind mainly) can have an impact on the drone control. Assuming that the wind is not too strong, the "position stabilization" is able to correct wind-induced drift.

The system provides constant feedback to the operator : drone position and altitude, current control mode, detected failures.

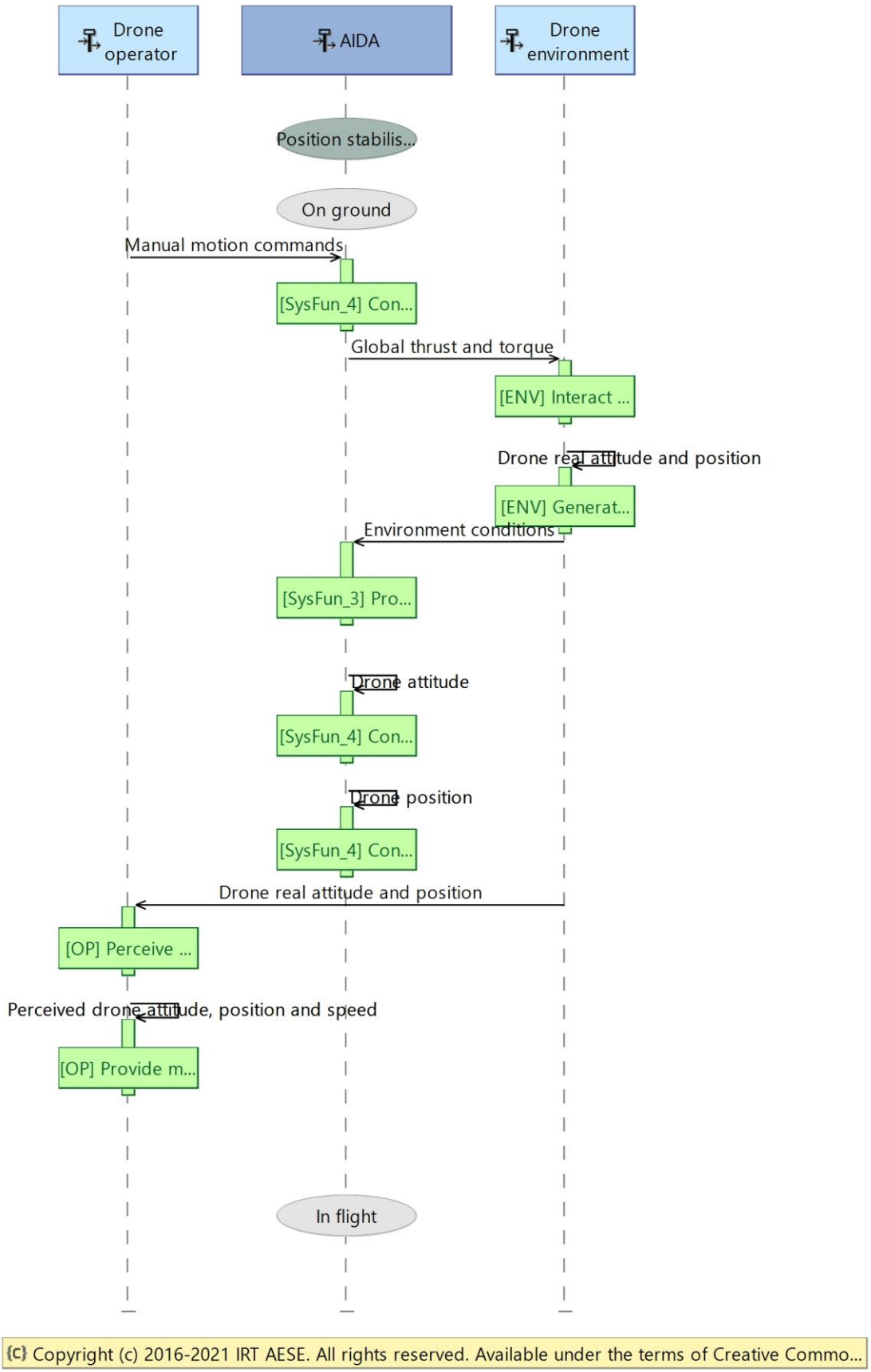


Figure 13 : "Manually control drone take off" sequence diagram

3.5.1.3. “Manually control drone motion in flight” sequence

This sequence is quite similar to the take off sequence. Additionally, the system provides constant live video feedback of the drone point of view, in order to help the operator to position adequately the drone before acquiring photos and videos.

In the global sequence, this sequence is repeated (with the visual information acquisition sequence presented after) so that the drone operator performs the whole inspection checklist.

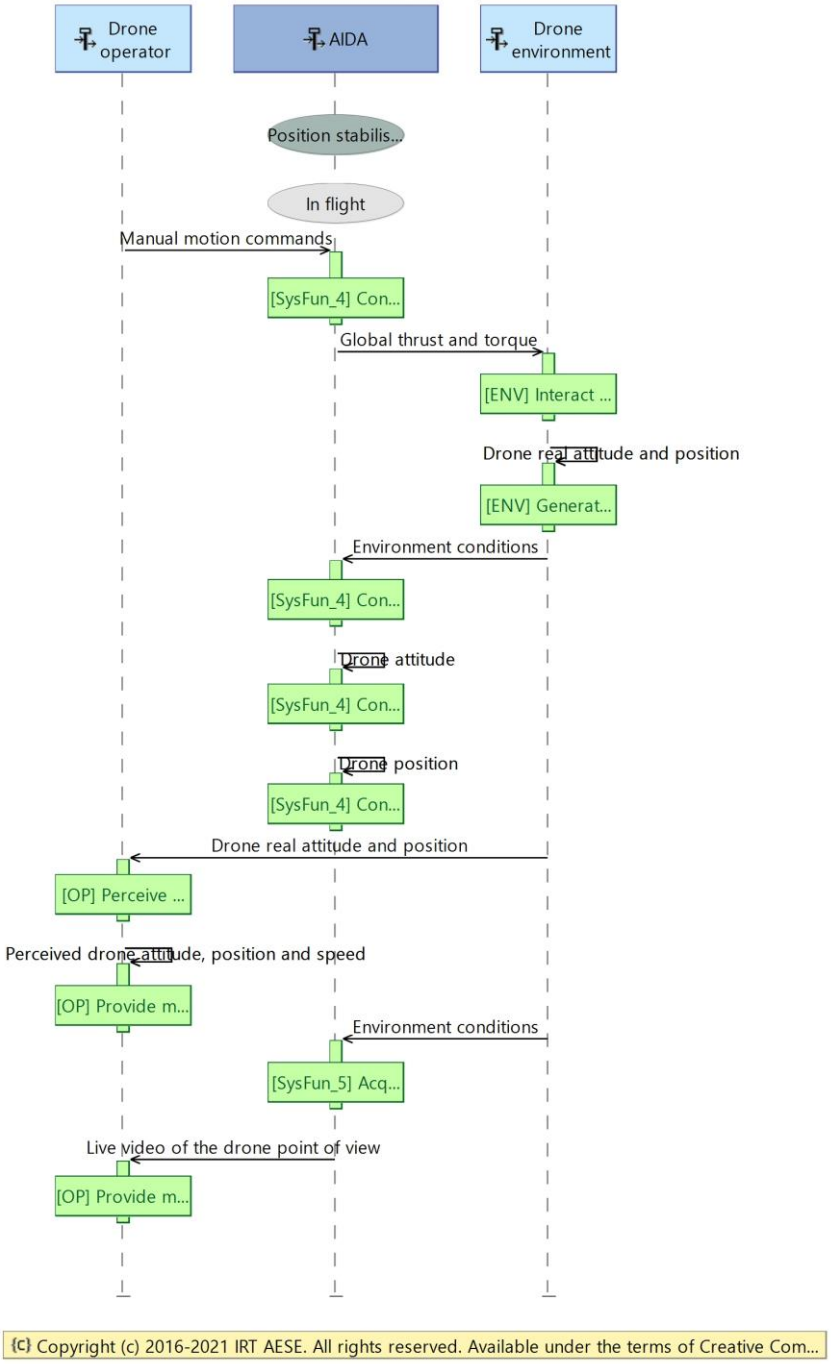


Figure 14 : "manually control drone motion in flight"

3.5.1.4. "Manual acquisition" sequence

In this sequence, the operator sends an acquisition command. The systems acquires photos or videos of the aircraft physical state. Again, the atmospheric conditions can have an impact on the acquisition (light exposure, precipitations,…).

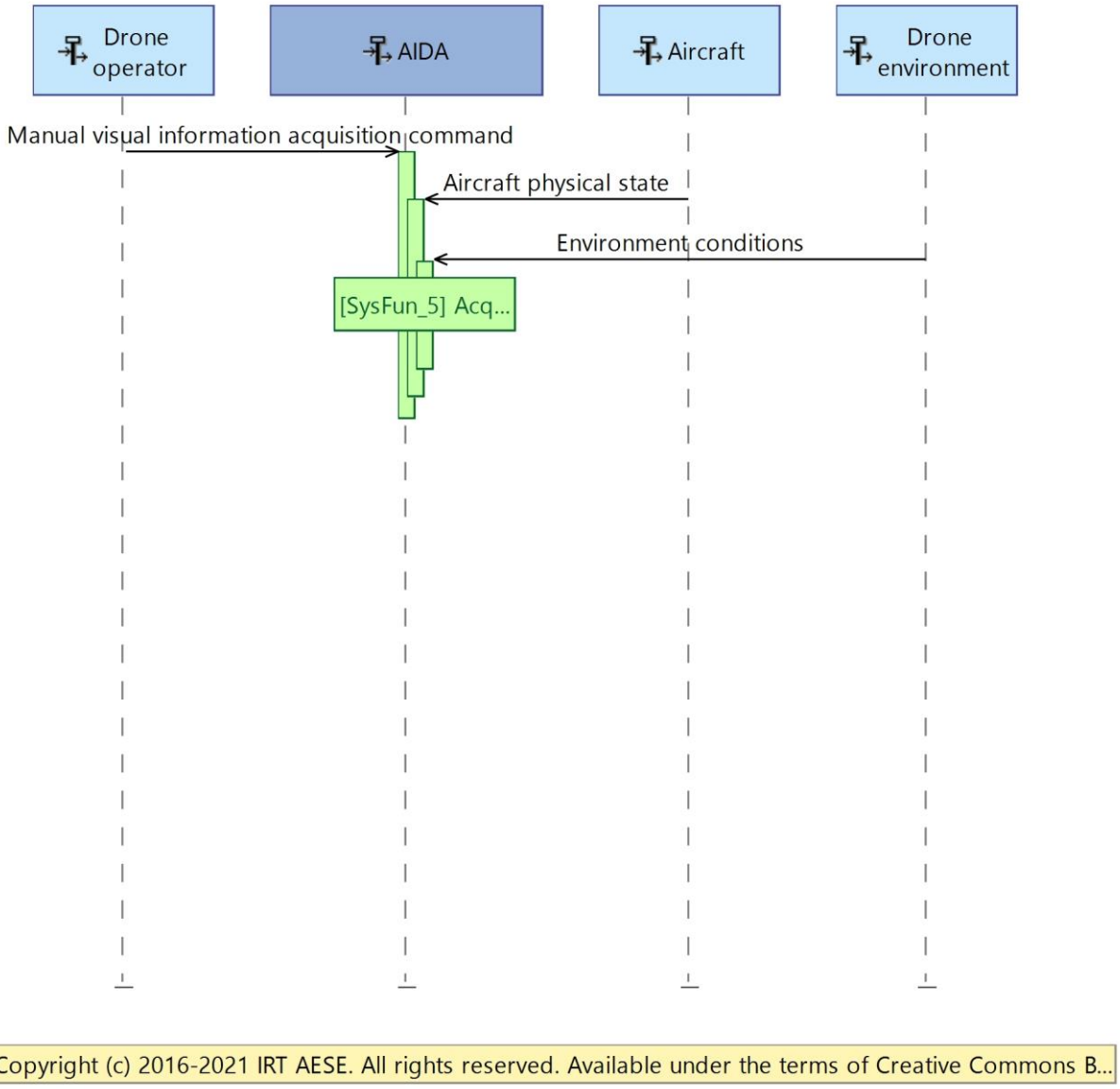


Figure 15 : Manual acquisition sequence

3.5.1.5. "Manually control drone landing"

Again, this sequence is similar to the manual take off sequence. At the end, the drone is on the ground.

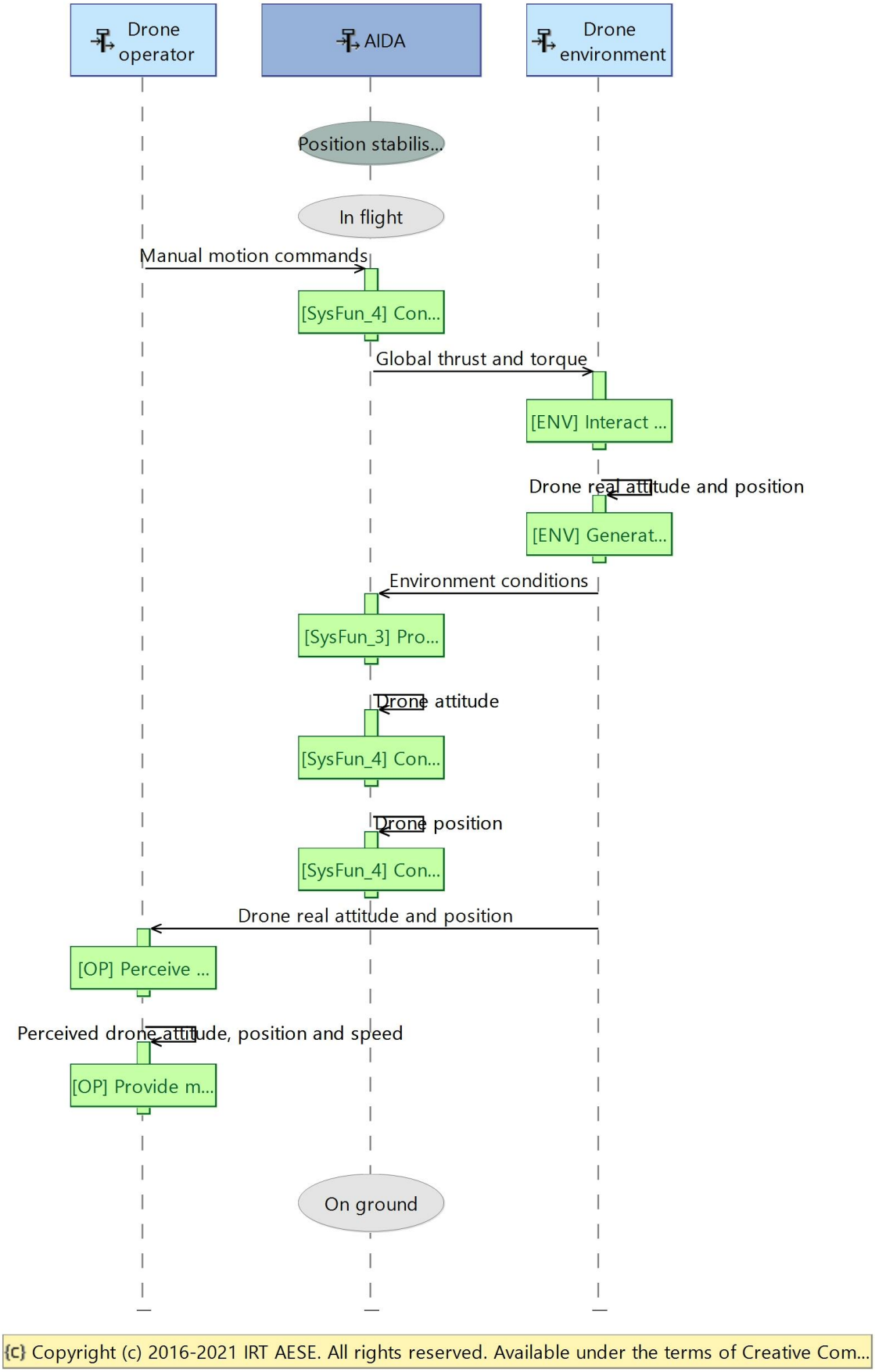


Figure 16 : "Manually control drone landing" sequence

3.5.2.Automatic inspection

We describe here the sequence of events during an automatic inspection operation assisted by the AIDA system.

System state: At the beginning of the operation, the system is supposed to be assembled, in perfect functioning state (no detected or undetected failure), completely energy loaded. The drone is set on the ground, in a ready-to-take-off position.

Environment:

- On ground: normal airport gate environment, with possible moving vehicles and uninvolved persons
- In air: normal airport traffic. No expected flying aircraft in the direct environment of the gate

Normal sequence:

- The drone is powered and ready to take-off.
- The system starts broadcasting Remote Identification data which are received by the Airport Control Tower to identify drone operations.
- The operator retrieves the flight zone in the external database and uploads it to the drone (see “Flight zone containment” paragraph below).
- The operator retrieves the inspection plan, consisting in viewpoints coordinates and associated sight angle, relatively to the aircraft position. Default trajectory paths to go from one viewpoint to another are also provided.
- The drone checks that its current location is contained in the flight zone. Powering the actuators is forbidden as long as the flight zone is not uploaded and validated.
- When the drone is ready for take off, the operator contacts the control tower and request the authorization to fly the drone.
- When authorized, the operator launches the automatic take off sequence and the “airplane detection” sequence : the drone takes off vertically to a given altitude and detects visually the aircraft position. It stays at the same position until the pilot launches the automatic inspection sequence.
- The operator validates the aircraft position detection.
- The drone computes its flight plan (i.e. the absolute coordinates of the viewpoints).
- The operator launches the “automatic inspection” sequence.
- The drone moves to each viewpoints and take the associated photos and videos.
- When the last viewpoint has been reached, the drone flies back and lands to its launch position.
- The drone operator launches the “automatic landing” sequence.
- The operator contacts the control tower to indicate the end of the drone operation.
- The AIDA system analyses the photos and videos and detects the failure and the icing state of the aircraft, and produce the inspection report.
- The AIDA system stores the inspection report in the airline database.

Flight zone containment:

The automatic inspection sequence (list of waypoints and trajectory path between the waypoints) must ensure that it will not lead the drone out of the flight zone. Also, the geo-fencing is still activated.

Obstacle management :

In automatic mode, obstacles are managed as follows :

- When an obstacle is detected on the trajectory path to join to next viewpoint, the drone tries to find an alternate path
- When a viewpoint is not reachable, the drone selects an alternate viewpoint (which are also part of the flight plan)
- If all the main and alternate viewpoints for a given inspection point cannot be reached, an alarm is sent to the operator and the drone keeps its position while waiting for a new pilot instruction (skip the next inspection point, return to base,…)

Data analysis and report generation:

Same as manual inspection.

The global sequence is represented on the sequence diagram below (referenced sequences are presented below the global sequence) :

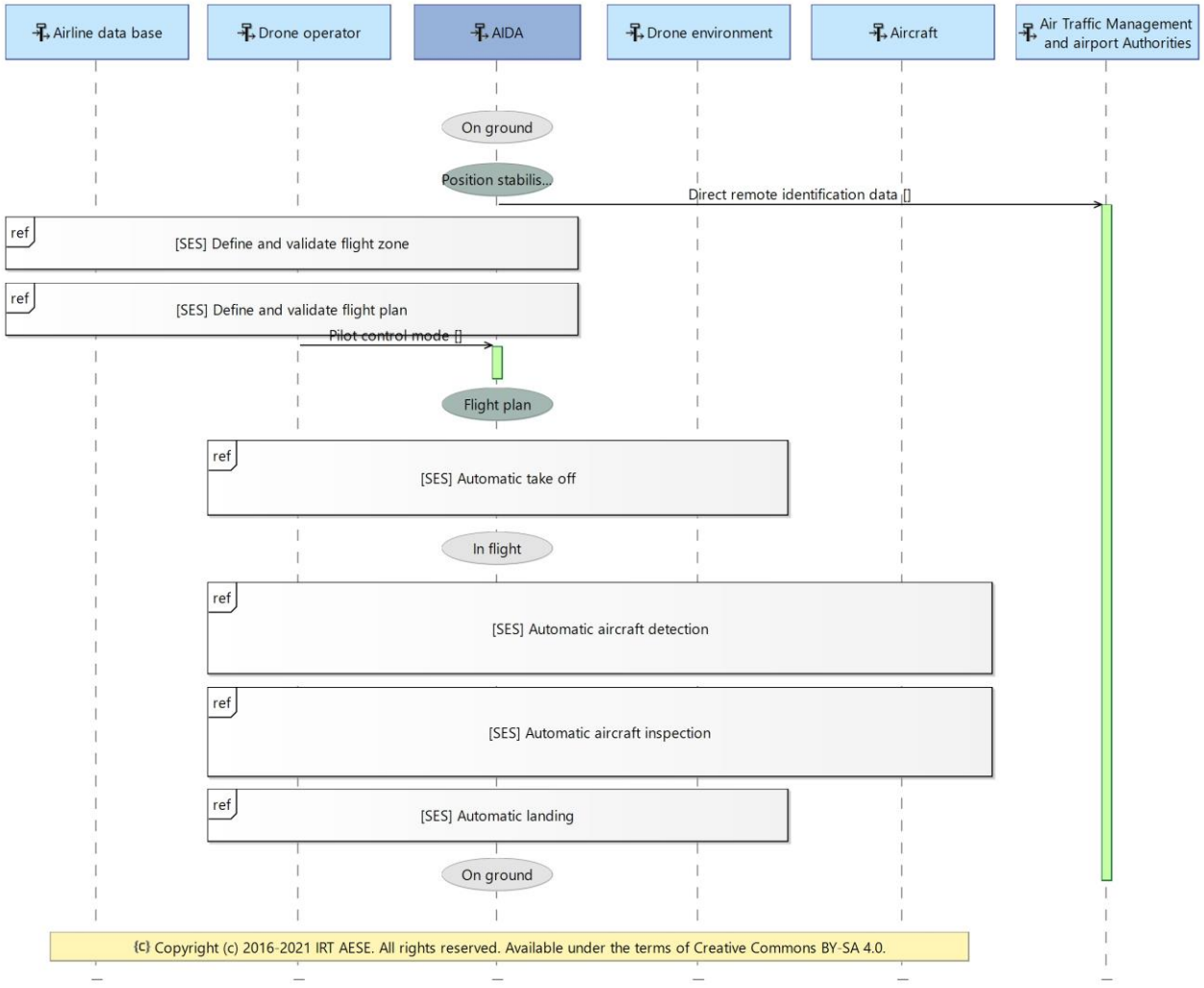


Figure 17 : "Automatic inspection sequence" diagram

The sequence “Define and validate flight zone” is the same as for the Manual inspection sequence, therefore it is not presented again in the following sections.

3.5.2.1. “Define and validate flight plan”

In this sequence, the operator selects a mission plan to be executed (the mission plan can be defined according to the type of aircraft, the type of inspection,…), which is retrieved by the AIDA system in the airline database. The mission plan consists in a list of viewpoints of the aircraft defined relatively to the aircraft position.

The operator can then visualize the mission plan and adapt it if necessary.

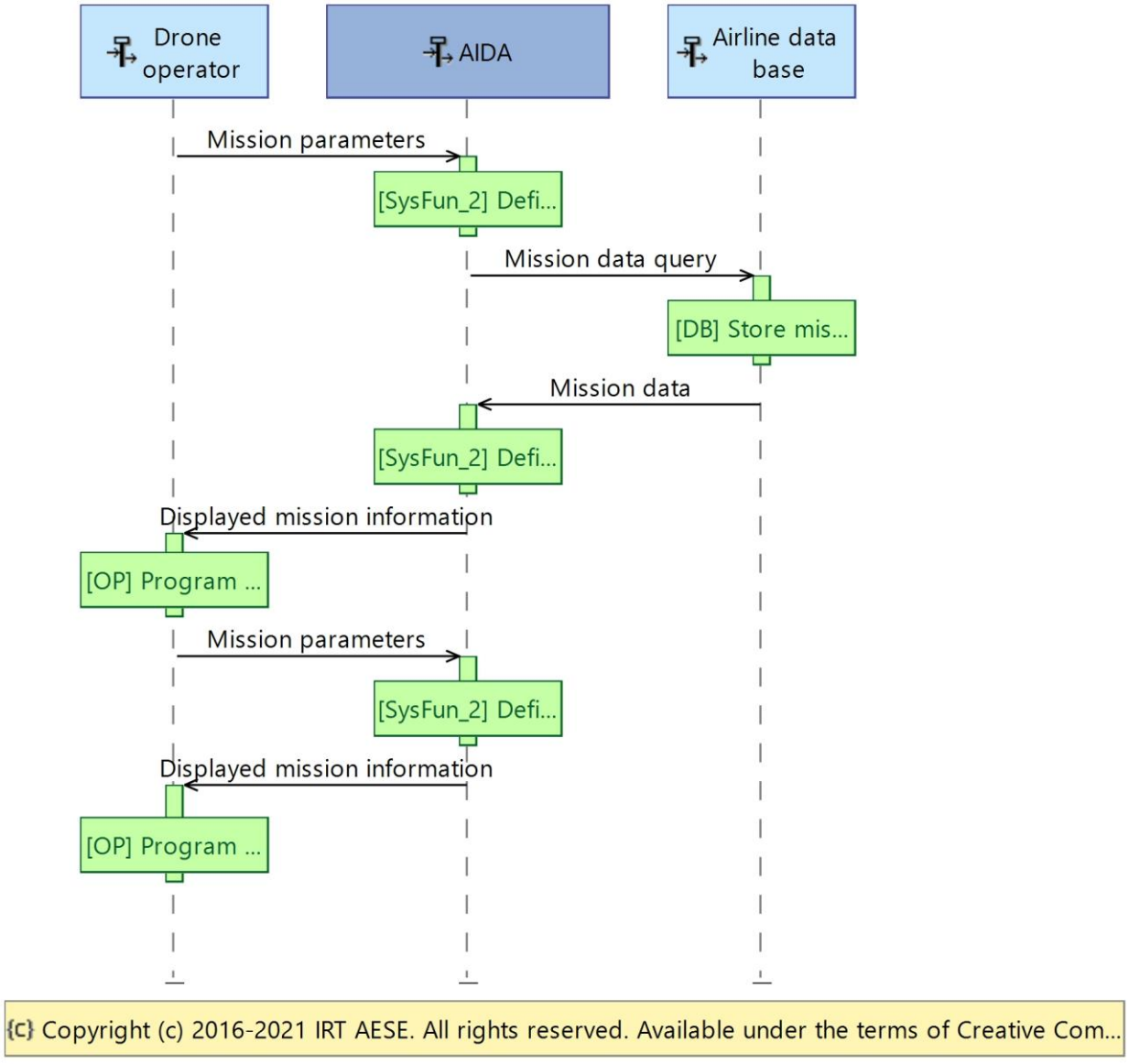


Figure 18 : "Define and validate flight plan" sequence diagram

3.5.2.2. “Automatic take-off” sequence

Before executing this sequence, the operator selects the “Navigation auto” control mode (see the global inspection sequence).

Then, he selects the auto sequence corresponding to take-off and engages the Auto-Pilot which leads the drone to execute the flight plan (which is a list of waypoints defined in absolute coordinates). The take-off sequence simply corresponds to the rise of the drone to a predefined altitude.

As in “Position stabilization” mode, both the attitude and position are required and atmospheric conditions may have an impact on the control.

The system provides constant feedback to the operator : drone position and altitude, current control mode, detected failures. It warns the operator when the drone has reached the last waypoint.

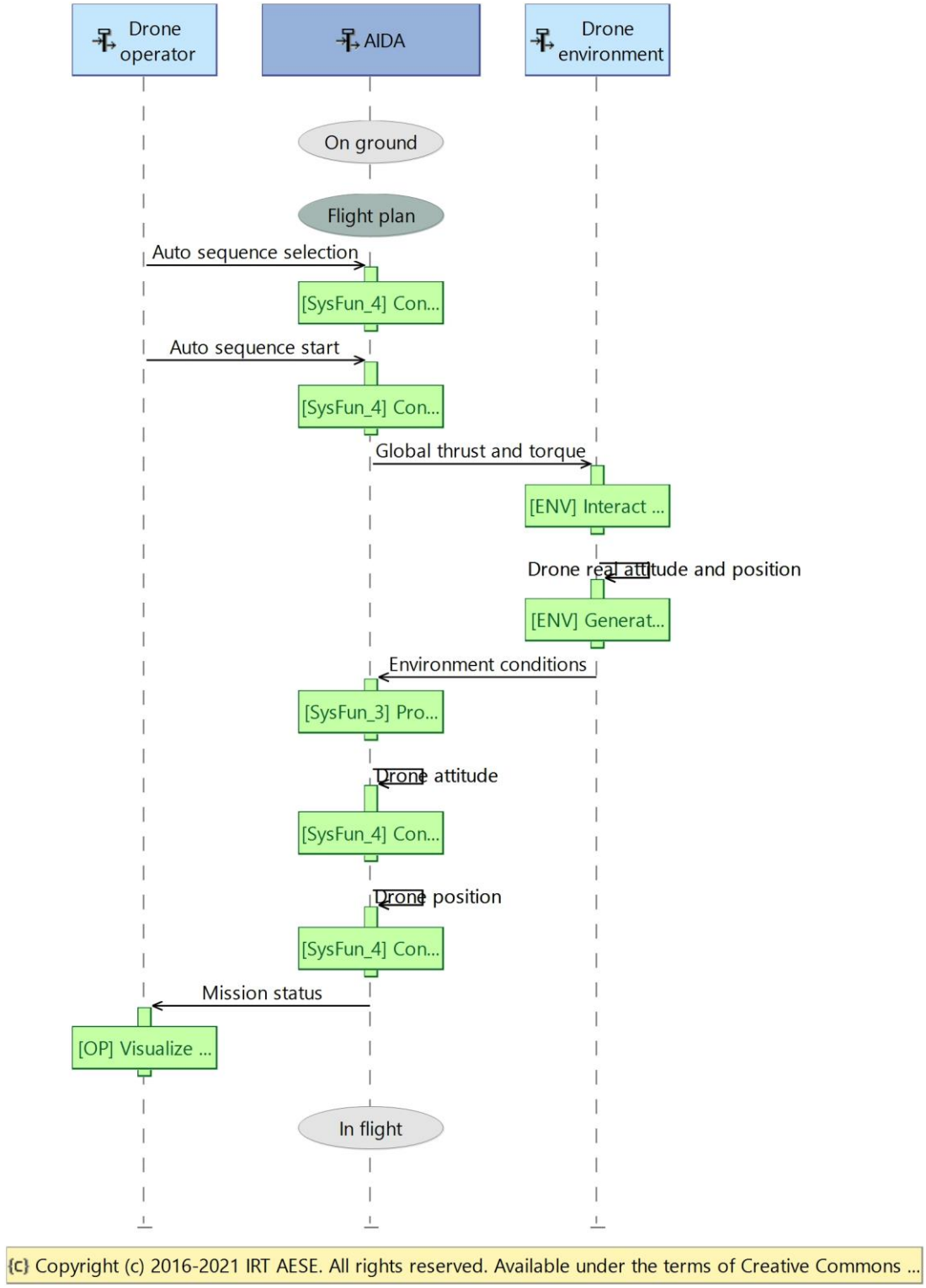


Figure 19 : "Automatic take-off" sequence diagram

3.5.2.3. *"Automatic aircraft detection"*

As explained before, the flight plan retrieved in the airline database is defined relatively to the aircraft position (so that it does not depend on the aircraft position which is not known exactly at the beginning of the operation).

In this sequence, the system detects the aircraft position and is then able to compute the flight plan in absolute coordinates, so that the drone can execute it.

This corresponds to another automatic sequence in which the drone detects the aircraft (one or several points of view may be needed, this will be defined later). The aircraft position detected by the drone is displayed and validated by the operator.

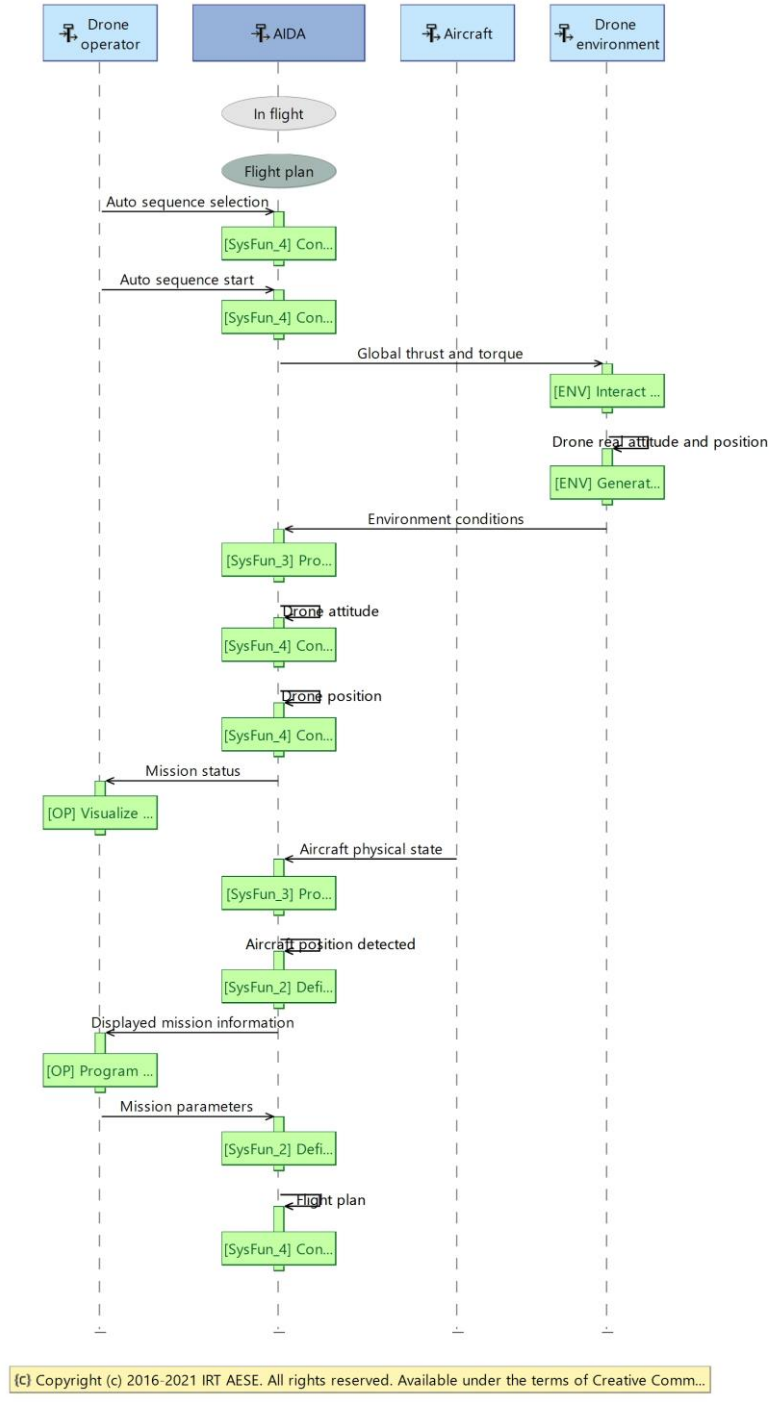


Figure 20 : "Automatic aircraft detection" sequence diagram

3.5.2.4. *"Automatic aircraft inspection" sequence*

Once the system has computed the absolute flight plan, the operator can launch the inspection sequence. As described earlier, he selects first the appropriate automatic sequence, then engages the Autopilot which executes the flight plan.

The inspection flight plan consists in moving the drone from one waypoint to another and automatically acquire photos and videos of the aircraft (see "automatic acquisition" sequence for details). This is represented by the loop in the diagram below.

Figure 21 : "Automatic aircraft inspection" sequence

3.5.2.5. “Automatic acquisition” sequence

In this sequence, the autopilot triggers the acquisition of photos and videos of the aircraft. A confirmation of the good acquisition is sent to the autopilot so that it can move the drone to the next viewpoint.

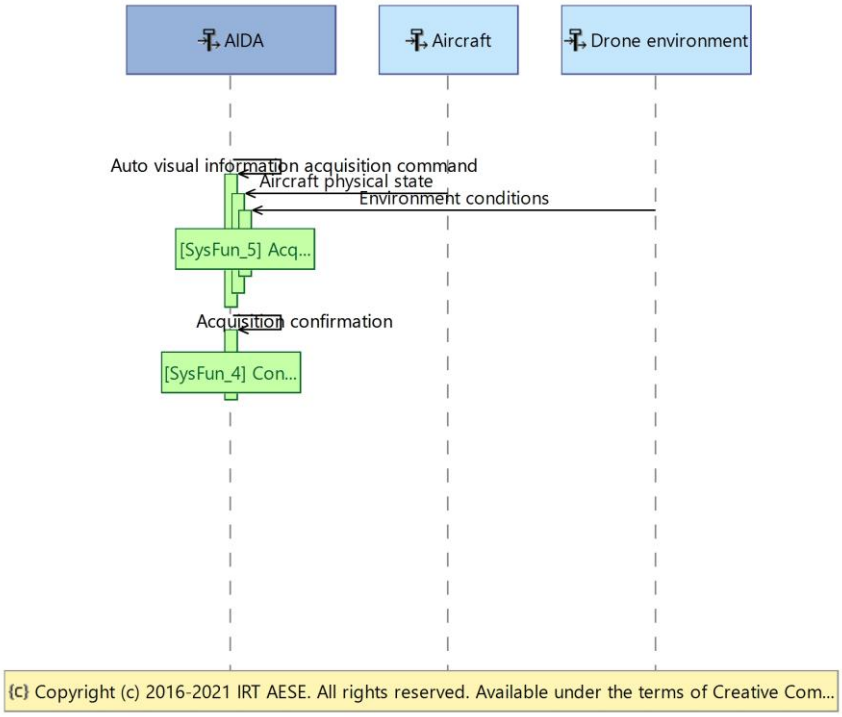


Figure 22 : "Automatic acquisition" sequence

3.5.2.6. “Automatic landing” sequence

This sequence is similar to the take-off sequence. At the end, the drone is on the ground.

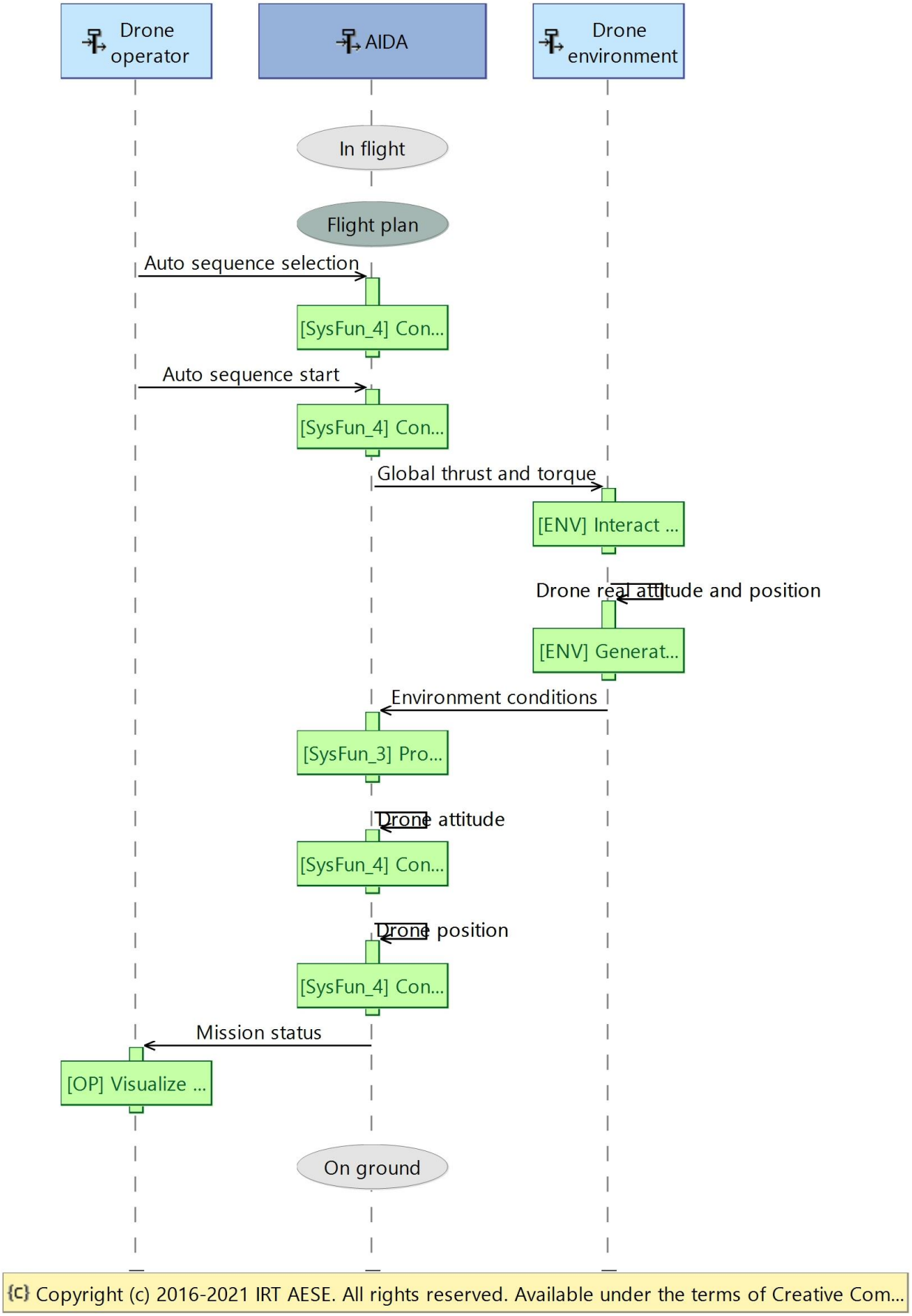


Figure 23 : "Automatic landing" sequence diagram

3.5.3.Failure scenarios

In order to complete the concept of operations, several high level failure scenarios are considered at this stage, corresponding to the loss of the main functions of a standard drone :

- Loss of communication between the drone and the operator : the operator is not able to send commands to the drone.
- Loss of position information : the drone is not able to control automatically its position, but attitude stabilisation is still ensured.
- Loss of attitude information : the drone is not able to stabilise automatically its attitude.

3.5.3.1. Communication loss

As explained earlier, it is known at this point that the AIDA system is a UAS, which consists in a flying segment (the drone), a ground segment and a communication between both parts. This is a small infringement of the Arcadia method in which, during the System Analysis phase, no hypothesis are supposed to be done on the internal architecture of the system.

However, we do not make here any hypothesis on the drone and ground segment architecture, functional breakdown between both segments and technological choices.

In case of communication loss, the drone does not receive control commands from the pilot. If this occurs while the system is in manual control mode (Position Stabilisation), the mission must be aborted. The drone switches to automatic navigation mode and engages in a “Return To Launch” automatic sequence :

- Stop horizontal displacement and climb to a safety altitude
- Move to the initial launch position
- Automatically lands at its launch position

The failure detection is announced to the drone operator, which can take appropriate actions (traffic management warning,…)

If the failure occurs while in automatic control mode (Navigation auto), we consider that the mission must also be aborted, as the pilot has no feedback on the drone state and is not able to take back the control if needed. The same sequence is executed.

This is reprensented by the sequence diagram below.

This diagram presents the case of a communication failure occurring while the system is in “Position stabilization” mode, however the behavior while in “Navigation auto” mode would be similar.

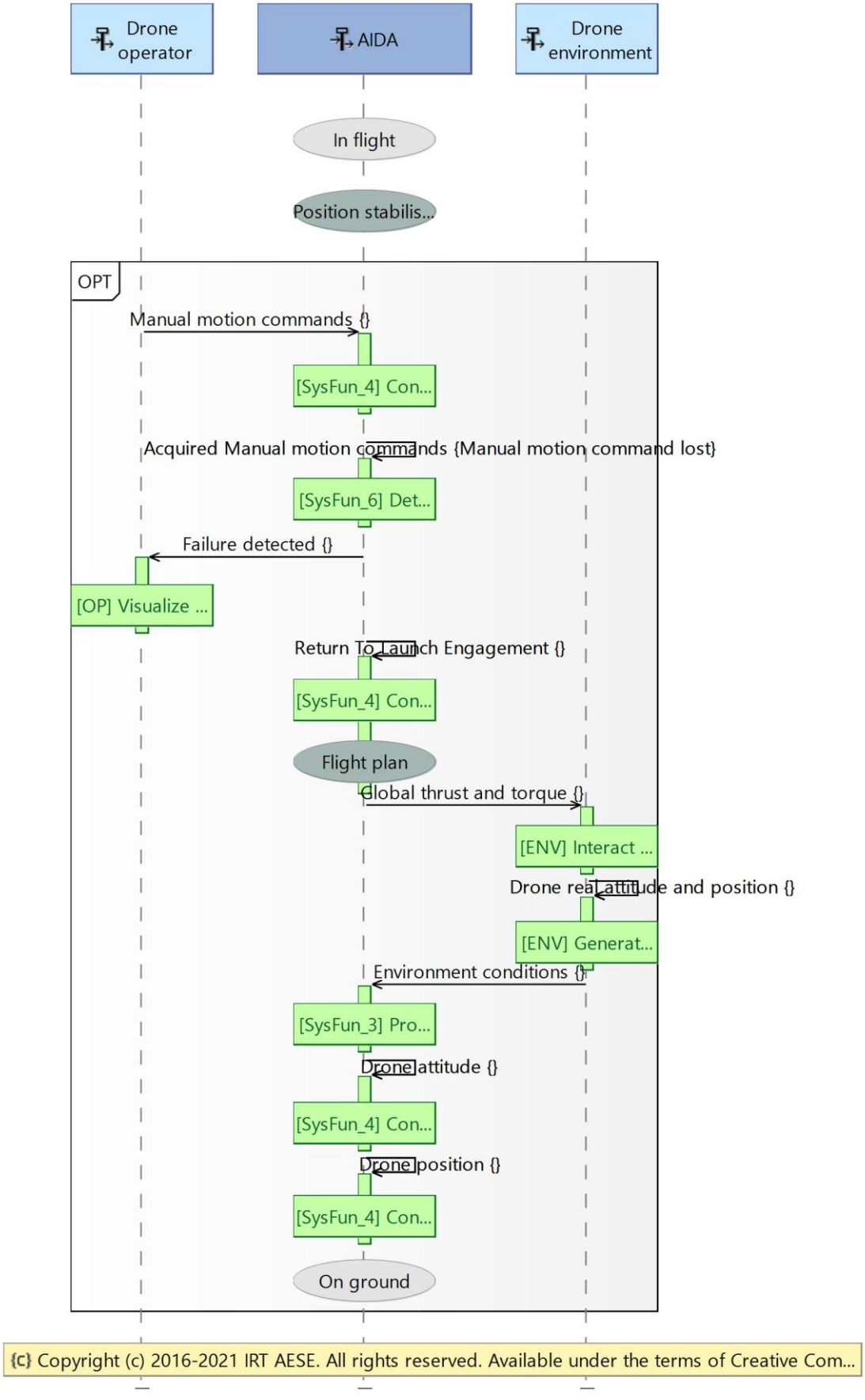


Figure 24 : "Communication loss" sequence diagram

3.5.3.2. Position loss

In case of position information loss, both normal control modes cannot be ensured : position stabilization is not possible without position information, and automatic navigation and guidance cannot be ensured. However, attitude stabilization is still possible. The drone switches to “Attitude stabilization mode” and alert the pilot that he must ensure the safe landing of the drone.

This is represented in the diagram below :

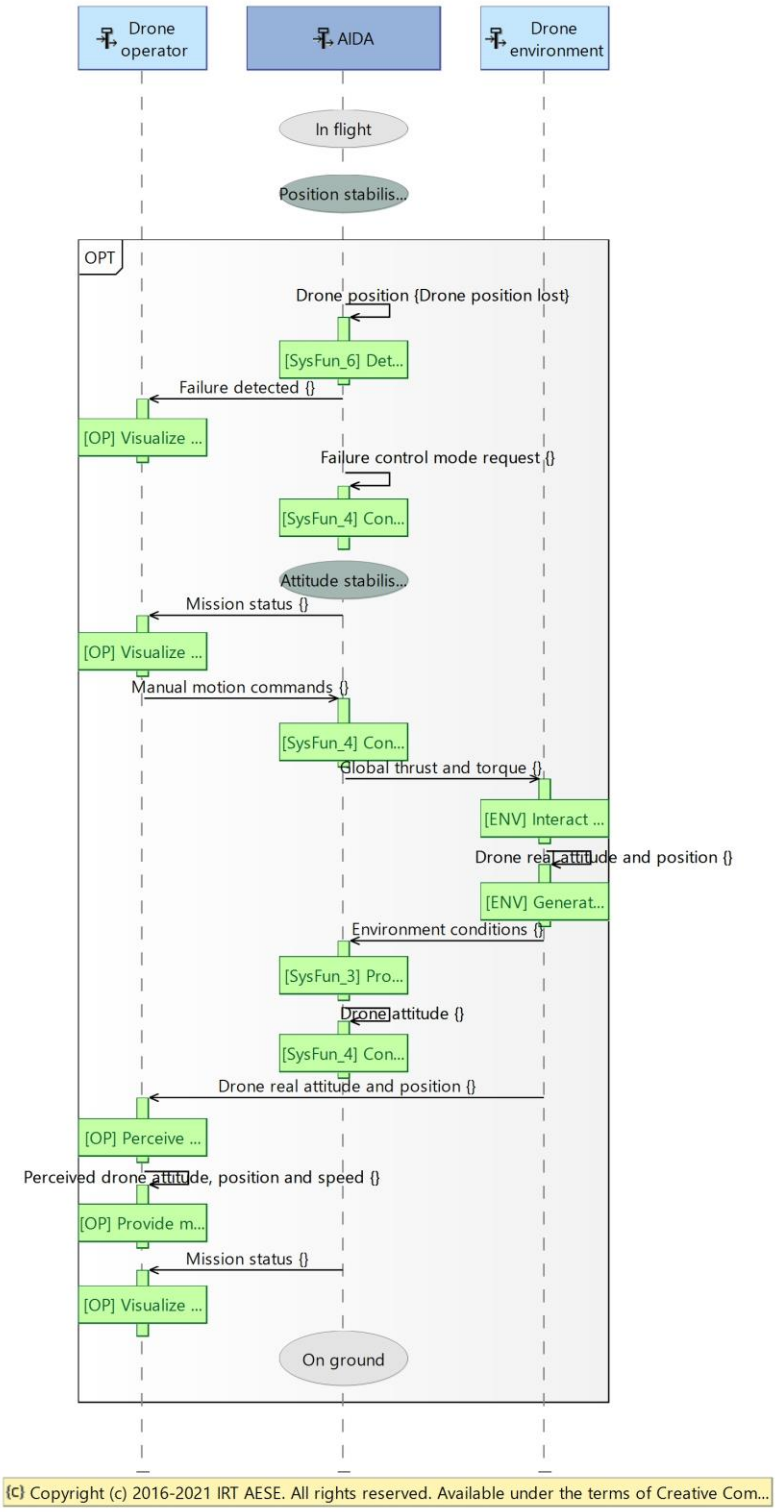


Figure 25 : "Position loss" sequence diagram

3.5.3.3. Attitude loss

In case of attitude information loss, the drone can only be control manually by the pilot without attitude stabilization. This requires advanced drone piloting skills in the complex environment of the airport gate, which is not compatible with the expected skills of the drone operator. The solution proposed is then the activation of the “Flight termination device”, which consists in cutting off the drone power supply and triggering the deployment of a parachute to ensure a “low-energy landing”.

This scenario is presented on the sequence diagram below :

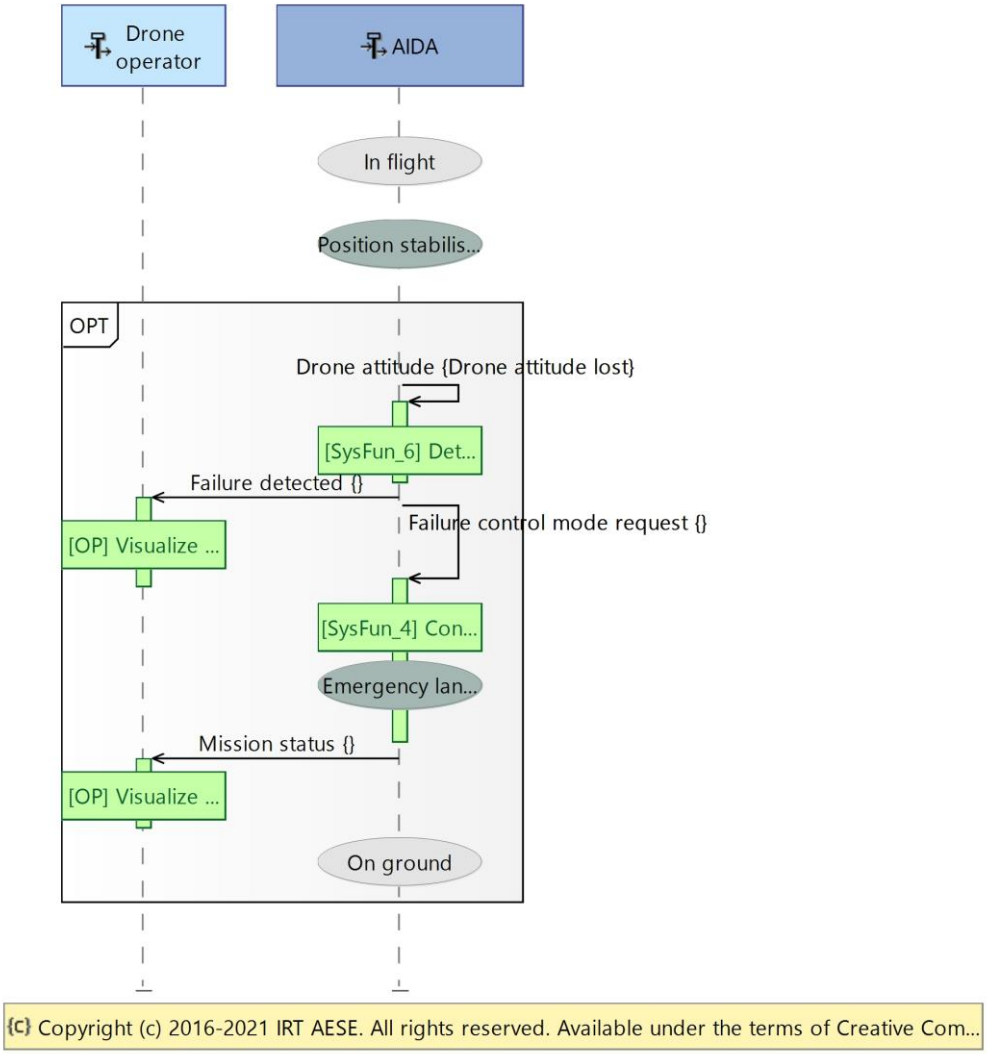


Figure 26 :"Attitude loss" sequence diagram

3.5.4.Maintenance scenarios

In the Maintenance life phase, the operator may need to access the maintenance status of the AIDA system (consisting mainly in the list of internal failures detected during previous flights, and possibly usage counters if limited life parts are considered further in the design). He can send a specific request to review the maintenance status.

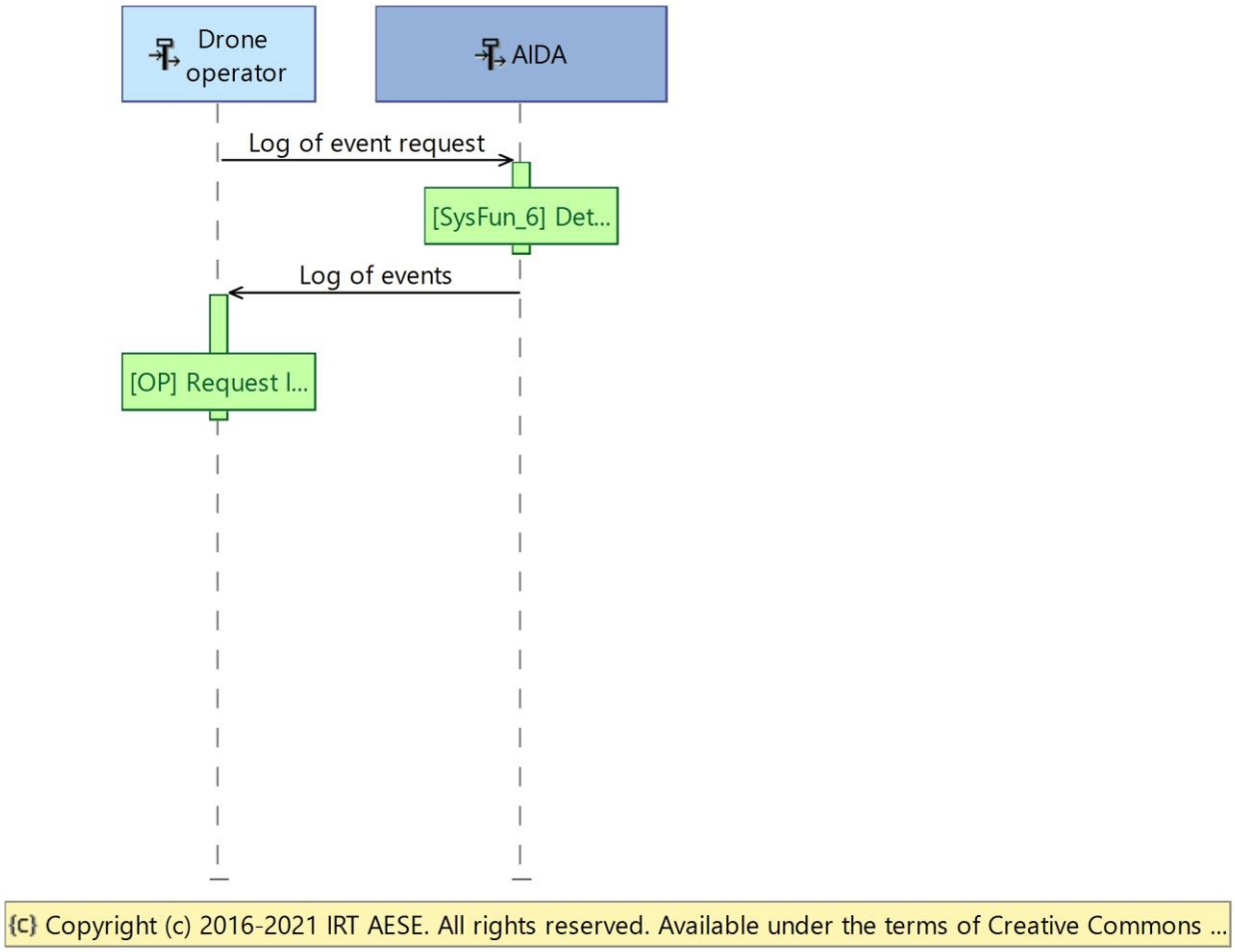


Figure 27 : "Access log of event" sequence diagram

Note : this scenario could be enriched with a "reset" request.

3.6. High level functional analysis

All the scenarios presented earlier allow to identify the modes and states of the AIDA system, and the high level functional architecture.

3.6.1.Modes

The AIDA system modes are presented on the mode machine diagram below :

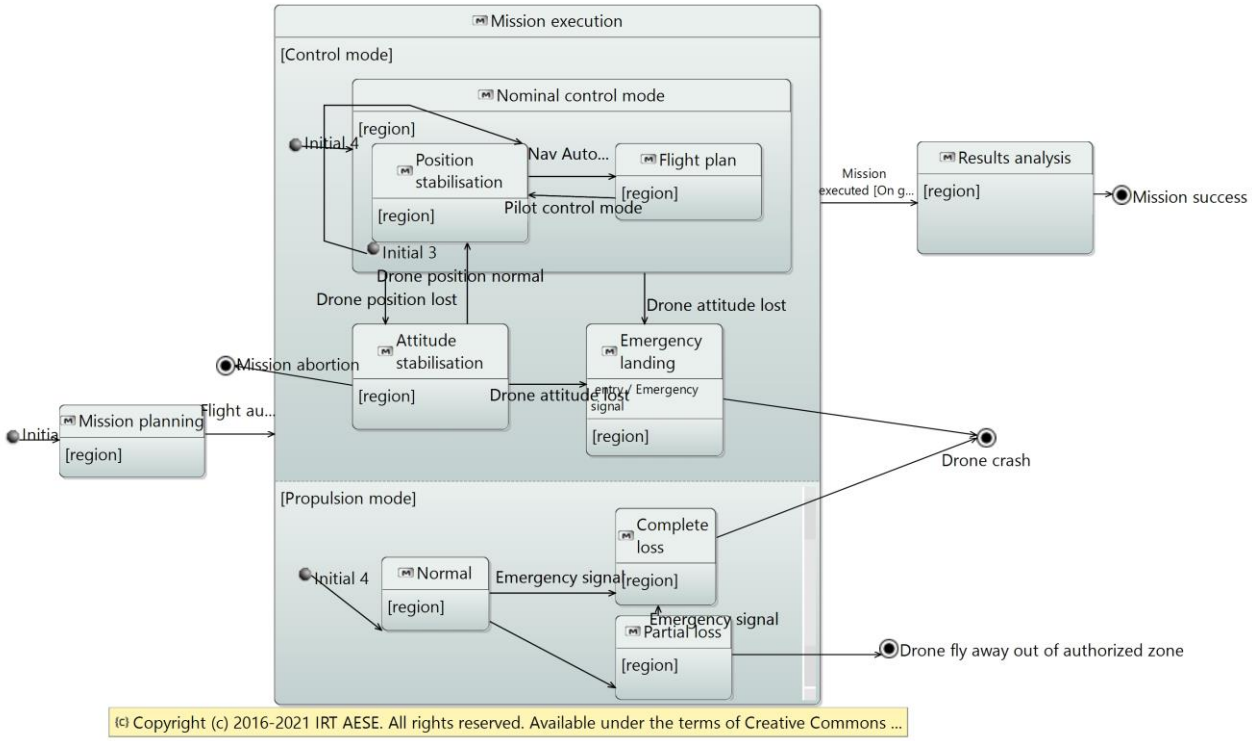


Figure 28 : AIDA system modes

The operations of the AIDA system can be split into three phases, which are represented by the following modes :

- Mission planning : this mode consists in defining the flight zone and the flight plan. A valid flight zone is required before authorizing the mission execution, as it ensures that the drone remains within the authorized zone.
- Mission execution : in this mode, the drone executes its mission. Two parallel sub-modes machines are defined :
    - One is related to the control mode
    - The other one is related to the propulsion activation
- When the mission has been executed, the results are analyzed by the system.

The control modes machine is defined as follows :

- In nominal situation, two modes are available :
    - Position stabilisation : the drone position is controlled manually by the operator. In this mode, the operator commands correspond to horizontal and vertical displacements, and drone yaw orientation. When the operator does not send any command, the drone maintains its current position (this requires the availability of a position information)
    - Navigation auto : in this mode, the drone follows a pre-defined flight plan. The actual execution of the flight necessitate the selection of this mode and the engagement of the Autopilot. When the Autopilot is not engaged, the drone stays in its current position. As mentionned in the sequence diagrams, several pre-defined flight plans are possible : automatic take-off or landing, aircraft detection, Return To Launch and Inspection flight plan (the last one corresponding to the flight plan retrieved from the database).
- In order to manage failure cases, two other control modes are proposed :

- Attitude stabilization : in this mode, the drone is only stabilized in attitude. This mode is activated when the position control is lost (because of the loss of position information for example). The drone can then move under the effect of the wind. The pilot then has to bring back and land the drone, the operator commands corresponding directly to attitude commands.
- Emergency landing : in this mode, the power supply is cut off and the parachute deployement is triggered. The drone is actuelly not controlled anymore and falls « slowly » to the ground (which limits the impact consequences). This mode is activated when the drone cannot be controlled, either automatically or manually (for example, when the attitude control is lost).

Without specific considerations on the drone architecture and propulsion technology choices, three propulsion modes are defined :

- Normal : the drone propulsion behaves normally
- Partial loss : the drone propulsion is partially lost, which may lead the drone outside the authorized zone (considered as a Catastrophic Event)
- Complete loss : the drone propulsion is completely loss, either because it failed completely or because the « emergency landing » control mode has become active, leading to the desactivation of the propulsion.

3.6.2.Flight phases

To ease the modelling, another state machine have also been defined, corresponding to the flight phases of the drone. So far, only two states have been identified : On Ground and In Flight. If necessary, the “In Flight” phase will be detailed in the logical or physical architecture phases.

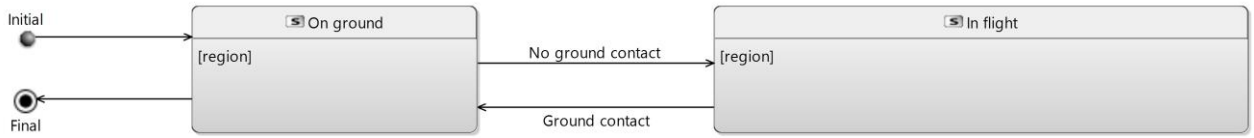


Figure 29 : flight phases diagram

3.6.3.High level functions and architecture

The high level functions identified during this System analysis phase are presented on the System Functionnal Dataflow diagram below :

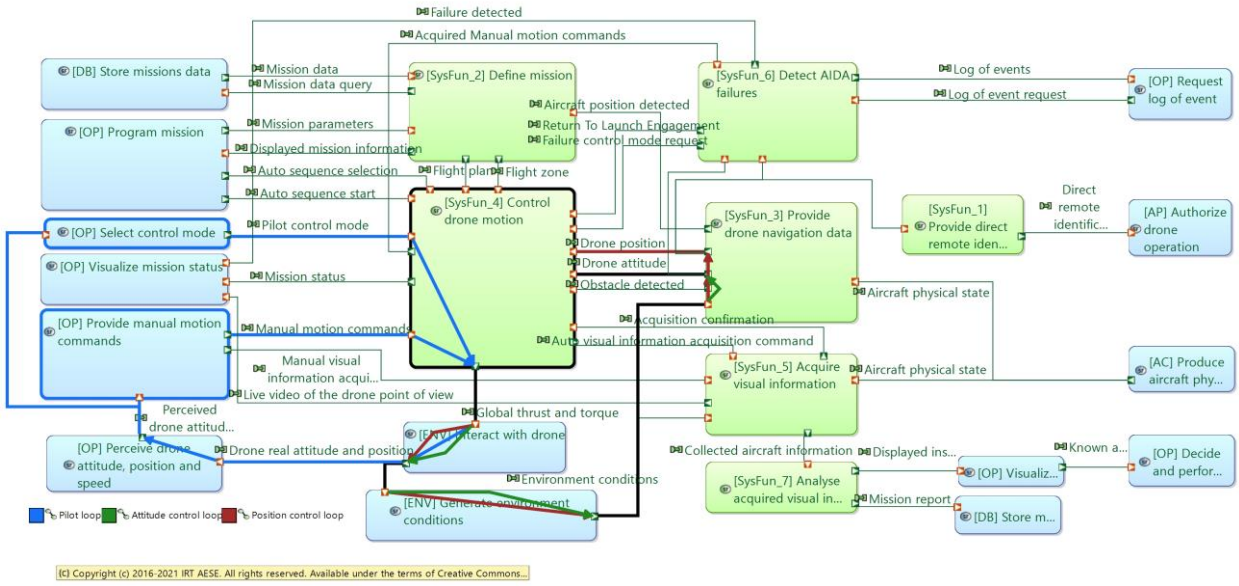


Figure 30 : AIDA System functional dataflow diagram

Seven high level system functions of the AIDA system are identified and synthetized in the table below :

ID	Function name	Description
SysFun_1	Provide direct remote identification information	This function broadcasts information about the drone as requested in UAS regulation texts
SysFun_2	Manage mission	This function enables the operator to define the mission, computes the flight zone and flight plan.
SysFun_3	Sense drone state and environment	This function senses the drone attitude, position and surrounding obstacles
SysFun_4	Control drone motion	This function manages the drone control modes and controls the drone motion according to the selected control modes, inputs from the operator and from other functions. It provides feedback to the operator about the mission progress
SysFun_5	Acquire visual information	This function realizes the acquisition of photos and videos of the inspected aircraft
SysFun_6	Detect AIDA failures	This function detects the failures of the AIDA system and informs the operator and other functions of the detected failures so that appropriate actions can be performed.
SysFun_7	Analyse acquired visual information	This function analyzes the acquired photos and videos in order to detect aircraft anomalies or icing

Modelling principles : in order to visualize the various control loops, the external function "Interact with drone" is added, allocated to the drone environment. This function "receives" the global thrust and torque created by the drone, and provides the real position, speed and attitude of the drone as a result of the interaction with the environment (ambient air, earth gravity and magnetic field, inertial reference,…). The pilot then visualizes the drone position, speed and attitude within the environment. Similarly, the system perceives the drone position, speed and attitude "generated" by

the environment (SysFun_3). Functional chains are used to represent these loops on the dataflow diagram.

These functions (internal and external) are then allocated to the AIDA system or to the relevant external actor, leading to the Architecture diagram below :

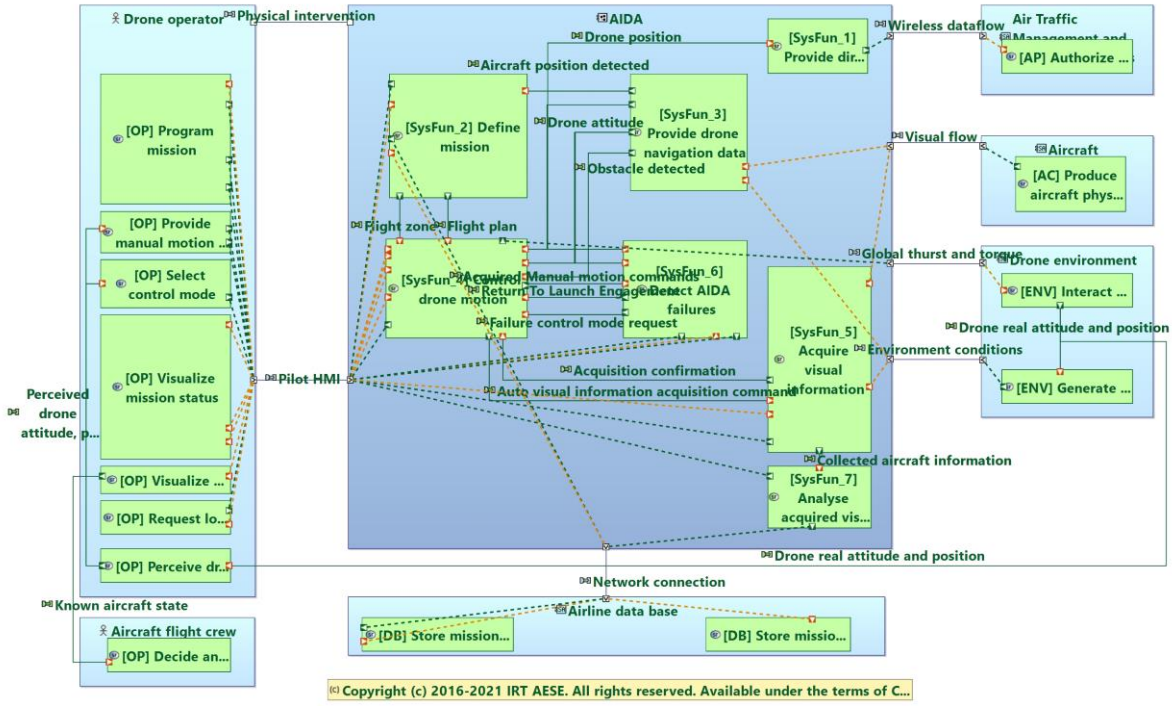


Figure 31 : AIDA System functional architecture diagram

3.6.4.External Interfaces

The functional exchanges between the AIDA system and the external systems and actors are allocated to components exchange. The diagram below represents the identified components exchanges, which represents the interfaces between AIDA and external systems :

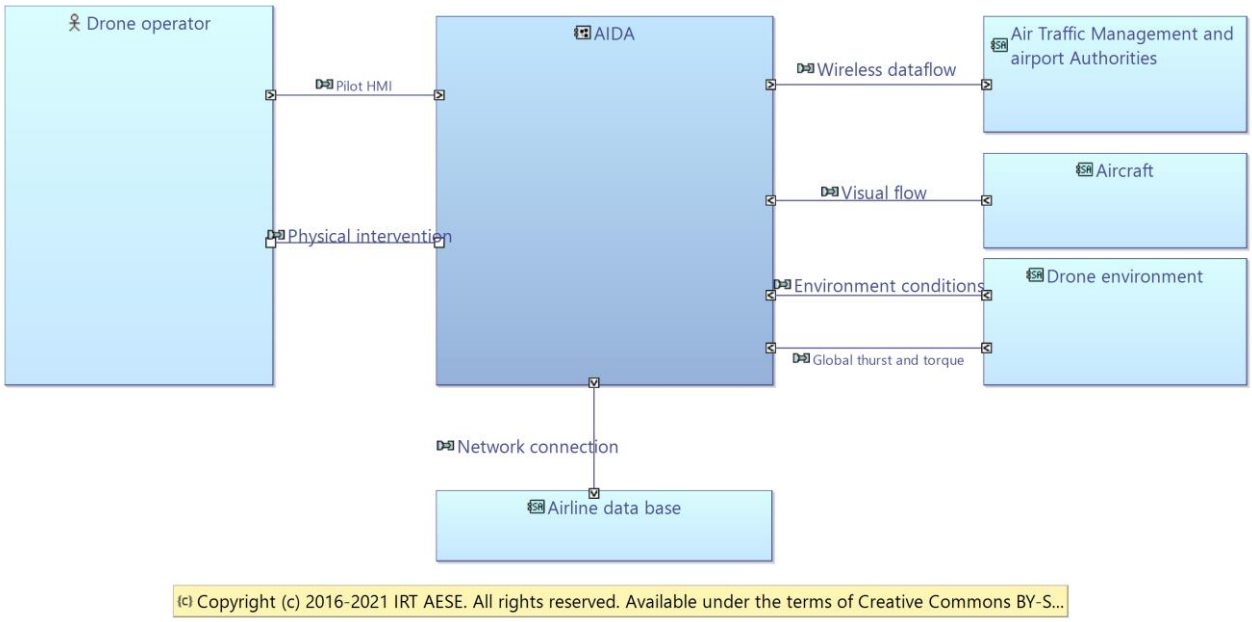


Figure 32 : AIDA System architecture diagram

These interfaces are described in the table below :

External system	Component exchange	Description
Drone operator	Pilot HMI	Interface between the pilot/operator and the electronic systems
	Physical intervention	Physical intervention of the operator for assembly, repair,…
Airline database	Network connection	Connection to an external network to enable communication with the airline database
ATM and airport authorities	Wireless dataflow	Wireless protocol communication with airport systems
Aircraft	Visual flow	Visual information of the inspected aircraft
Atmospheric conditions	Environment conditions	Atmostpheric conditions around the system

4. Logical architecture

In the Arcadia method, the logical architecture answers to the question “how the system will work to fulfill expectations”. It is an intermediate step between the high level functional analysis (“what the system has to accomplish for the users”) and the detailed physical architecture (“how the system will be developed and built”).

The purpose is to start defining the logical components that will work together to fulfill the high level objectives, taking into account non-functional constraints but without too much focus on technological solutions. Ideally, the result of the logical architecture step of the Arcadia method is supposed to be quite constant along the development process, while the physical architecture evolves taking into account physical constraints.

Several drivers for the decomposition into logical components can be identified : industrial split between the suppliers of the sub-systems, qualitative safety requirements (DAL and independence),…

4.1. Initialization of logical architecture

The main input to perform the logical architecture activities are the list of functions of the AIDA system, as defined during the System analysis phase. These functions are then broken down to allow allocation to sub-systems.

In Capella, each design layer has its own list of functions and functional architecture. However, the objects of the different layers are linked through “Realization links”.

For AIDA, the high level functions for the logical layer are directly inherited from the system layer as follows :

System function	Realising logical function
[SysFun_1] Provide direct remote identification information	[LogFun_1] Provide direct remote identification information
[SysFun_2] Manage mission	[LogFun_2] Manage mission
[SysFun_3] Sense drone state and environment	[LogFun_3] Sense drone state and environment
[SysFun_4] Control drone motion	[LogFun_4] Control drone motion
[SysFun_5] Acquire visual information	[LogFun_5] Acquire visual information
[SysFun_6] Detect AIDA failures	[LogFun_6] Detect AIDA failures
[SysFun_7] Analyse acquired visual information	[LogFun_7] Analyse acquired visual information

Note : these realization links can be visualized in Capella using the traceability matrix of the Logical architecture layer. The realization links also cover the “external” functions allocated to external actors of the system.

4.2. Candidate logical architecture

In the case of AIDA, which is supposed to be a UAS, we propose the following components breakdown :

- The drone, which can be decomposed as follows :
  - Flight control and monitoring system
  - Propulsion system
  - Payload
  - Structure
- The remote control

- The control desk

The proposed logical components architecture is represented on the diagram below :

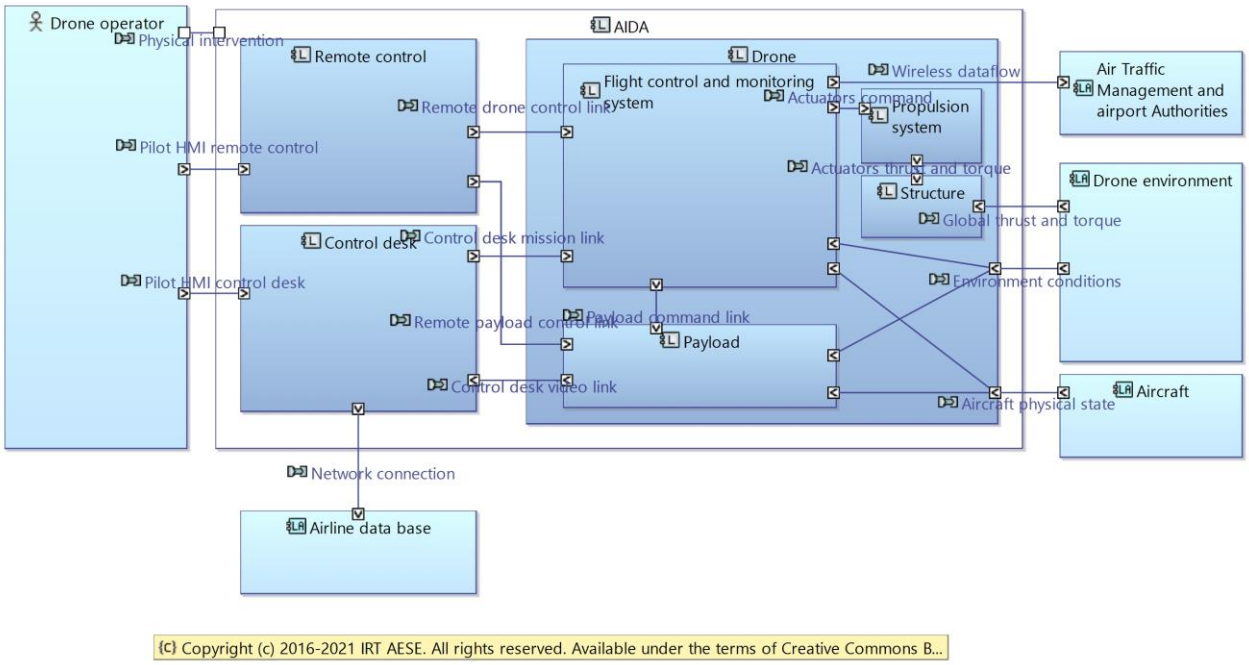


Figure 33 : first level of logical architecture

New components exchanges are created :

- The « Pilot HMI » component exchange represented in the System analysis layer is decomposed into two logical components exchanges : Pilot HMI remote control and Pilot HMI control desk
- Several links are created for communication between the remote control and the drone, and between the control desk and the drone.
- Internal drone links are created for exchanges between the drone components.
- New delegated exchanges are created for Environment conditions and Aircraft physical states

The proposed logic for the decomposition of each function is the following :

- The drone embeds the functions which are required to ensure its operations, including the safe Return To Launch in case of loss of communication with ground systems
- The drone operator does not have any direct interface with the drone. All functional flows from the drone operator transit either through the remote control or the control desk.
- The remote control is used for real-time control of the drone operations : all the functional flows involved in real-time control of monitoring operations of the drone transit through the remote control. While using the remote control, the pilot can keep on an eye on the drone and its environment.
- In order to limit the operator workload during the drone operation, the functional flows which are not directly involved in real-time operations transit through the control desk. When using the control desk, the pilot cannot watch the drone at the same time, therefore the usage of the control desk must be limited while the drone is in the air.

4.3. Functions refinement and allocation

4.3.1.[LogFun_1] Provide direct remote identification information

This function is directly allocated to the flight control and monitoring system and does not require breakdown. The figure below represent the contextual architecture diagram of this function :

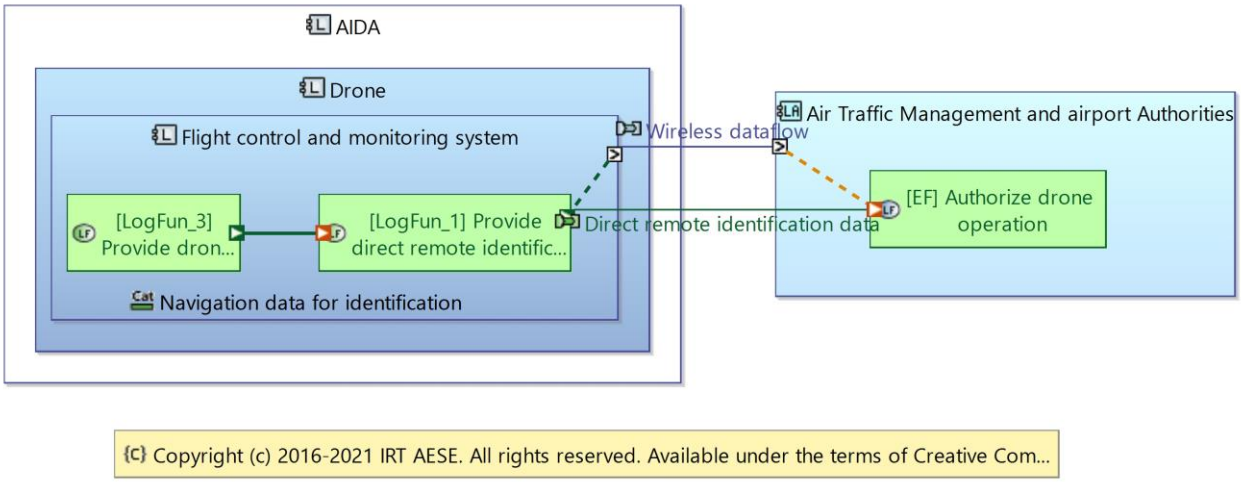


Figure 34 : contextual logical architecture of LogFun_1

4.3.2.[LogFun_2] Manage mission

This function is directly allocated to the control desk and does not require breakdown. The figure below represent the contextual architecture diagram of this function :

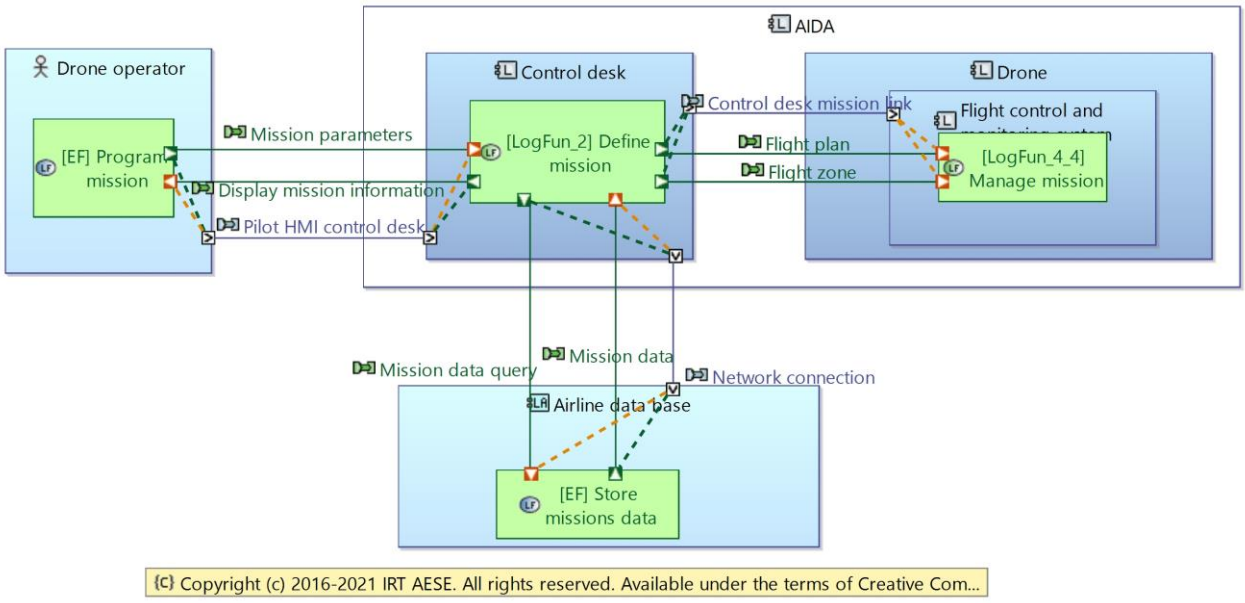


Figure 35 : contextual logical architecture of LogFun_2

This function is implemented in the control desk, it provides the flight plan and the flight zone boundaries to the drone, through the control desk link between the control desk and the drone.

To allow the computation of the flight plan, the drone provides the detected aircraft position to the control desk (see “Automatic aircraft detection” sequence in the System analysis phase”)

4.3.3.[LogFun_3] Sense drone state and environment

This function is directly allocated to the drone and does not require breakdown. The figure below represent the contextual architecture diagram of this function :

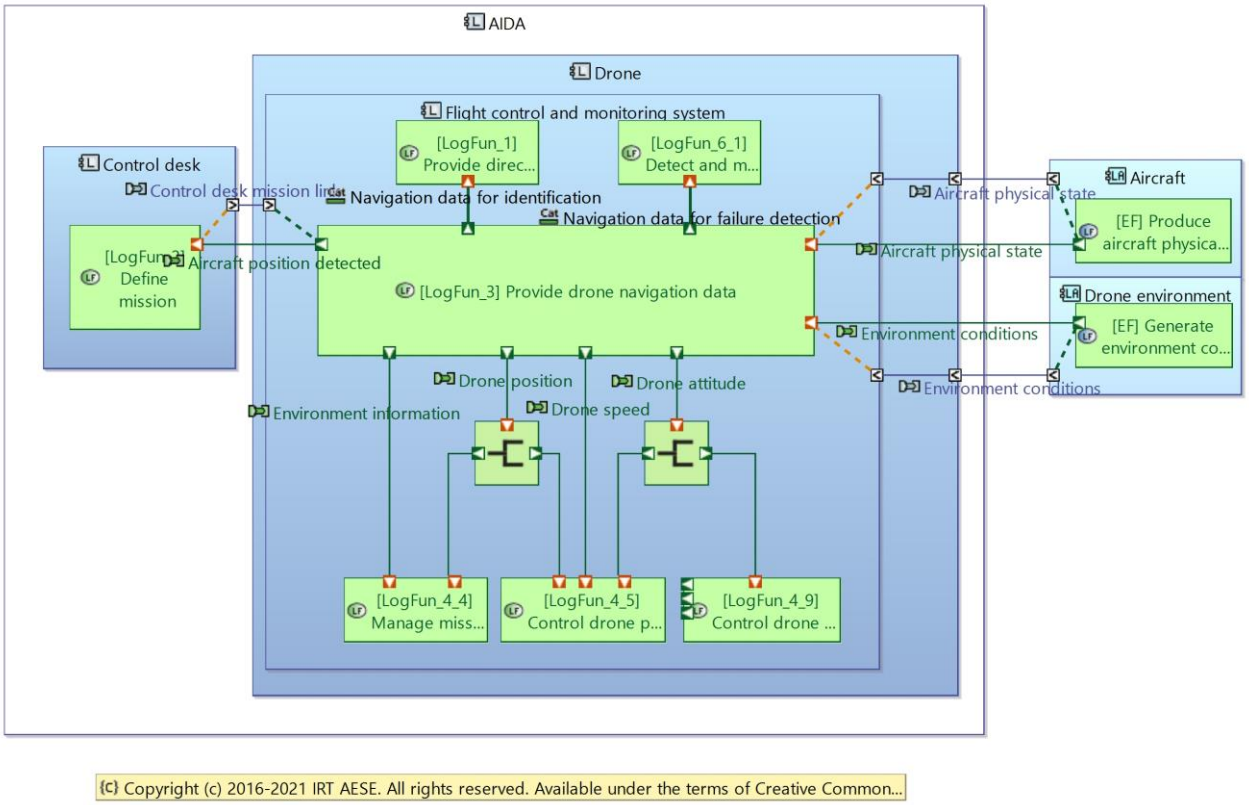


Figure 36 : contextual logical architecture of LogFun_3

This function is implemented in the drone. It provides information about the drone attitude and position to the other functions that requires it. It also provides the aircraft detection position to the control desk through the control desk link.

4.3.4.[LogFun_4] Control drone motion

As this function has many functional exchanges with the drone operator, it requires refinement and allocation between the drone and the ground systems (control desk and remote control). A possible allocation is proposed (see diagram below), in which the remote control and the control desk acquire the required information and transfer it to the drone through the appropriate links (remote control link and control desk link).

The choice of using the remote control or the control desk for each functional interface with the drone operator is justified as follows :

Functional exchange	Allocation	Justification
Manual motion commands	Remote control	These commands are required for real-time drone operation in Position Stabilization or Attitude stabilization mode
Auto-sequence selection	Remote control	The drone operator must be able to select the automatic flight sequence directly from the remote control.
Pilot Control mode	Remote control	The drone operator must be able to switch between control modes directly from the remote control

AP start	Remote control	The drone operator must be able to engage the execution of the Flight plan directly from the remote control
Mission status	Control desk	This status represents the progress of the flight plan execution (when using the Navigation Auto mode). This information is not directly required by the pilot in real-time

Four main sub-functions allocated to the drone are also identified :

- Control drone navigation : in Navigation Auto mode, this function provides manages the execution of the selected flight plan (automatic take-off/landing/aircraft detection/aircraft inspection) and provides the associated position and speed demands, along with the photo/video acquisition commands. It also selects the control mode, based on the drone operator request.
- Control drone attitude and position : this function realises the control of the drone attitude and position, depending on the selected mode. It provides the actuators commands to realise the required position and attitude. It uses information from LogFun_3.
- Control drone actuators : this function ensures the control of the actuators.
- Reconstitute global thrust and torque : this fonction represents the behaviour of the drone structure, which « gathers » the loads created by all the actuators into a global thrust and torque exchanged with the environment

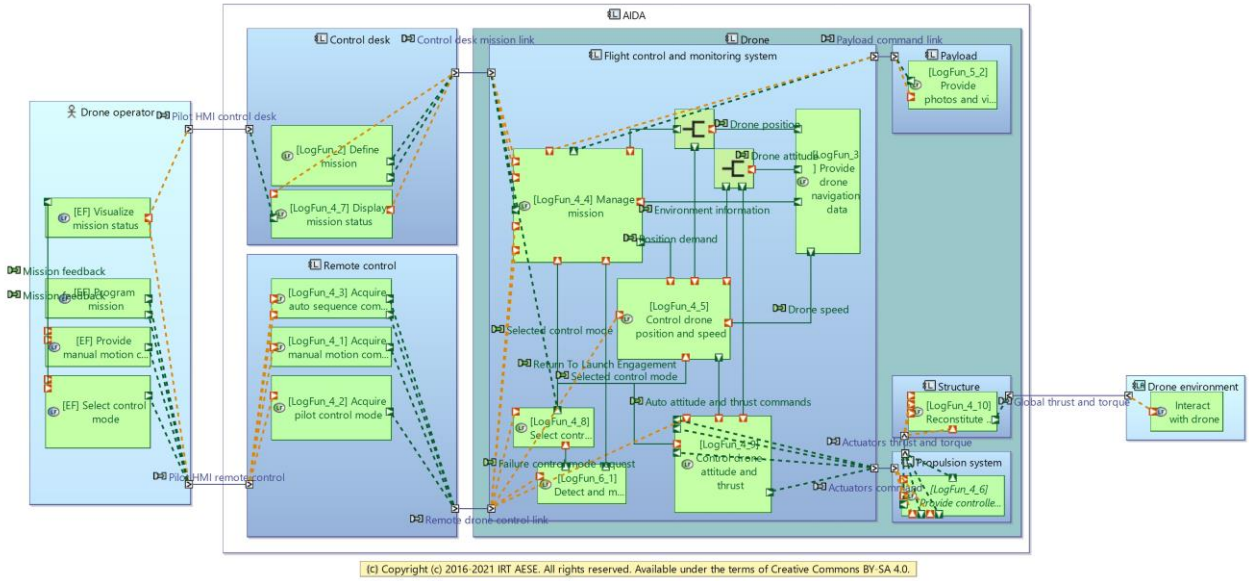


Figure 37 : contextual logical architecture of LogFun_4

4.3.5.[LogFun_5] Acquire visual information

This function has several interfaces with the drone operator, therefore it requires refinement and allocation : we propose a possible allocation (see the diagram below), in which the remote control realizes the acquisition and display functions.

The choice of using the remote control or the control desk for each functional interface with the drone operator is justified as follows :

Functional exchange	Allocation	Justification
Manual visual information acquisition command	Remote control	These commands are required for real-time drone operation in Position Stabilization or Attitude stabilization mode
Displayed drone point of view	Remote control	This visual feedback of the drone point of view is used in real-time to position adequately the drone in Position Stabilisation mode.

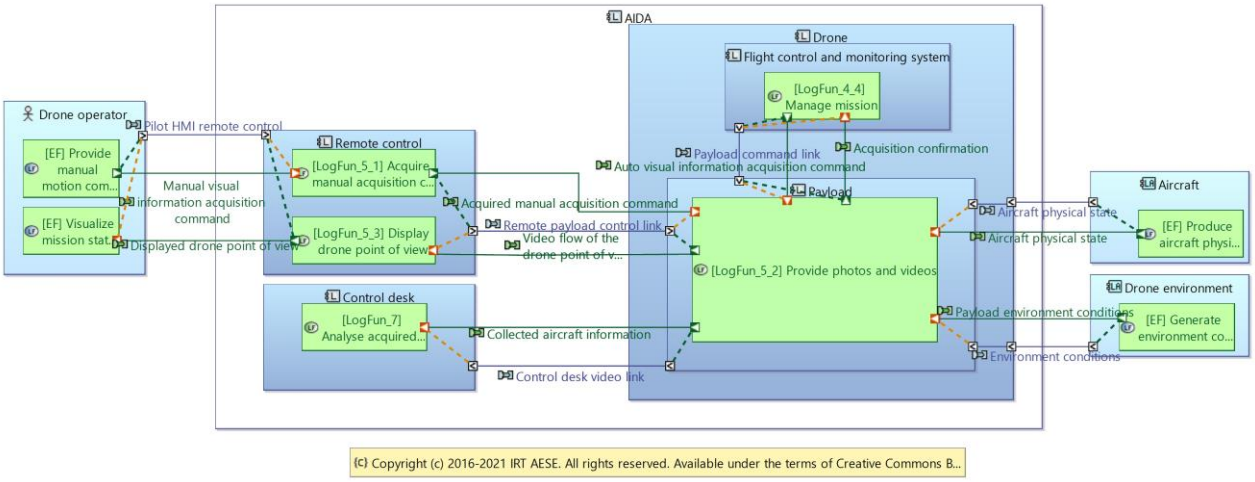


Figure 38 : contextual logical architecture of LogFun_5

4.3.6.[LogFun_6] Detect AIDA failures

This function has several interfaces with the drone operator, therefore it requires refinement and allocation : we propose a possible allocation on the diagram below.

The choice of using the remote control or the control desk for each functional interface with the drone operator is justified as follows :

Functional exchange	Allocation	Justification
Displayed detected failure	Remote control	The drone operator must be directly informed of the drone state and detected failures.
Log of event request	Control desk	This request is used to access the complete list of internal failures of the drone. It is used mainly during maintenance phases, not for real-time drone operation
Log of events	Control desk	This is the result of the previous request, which is not used for real-time operation.

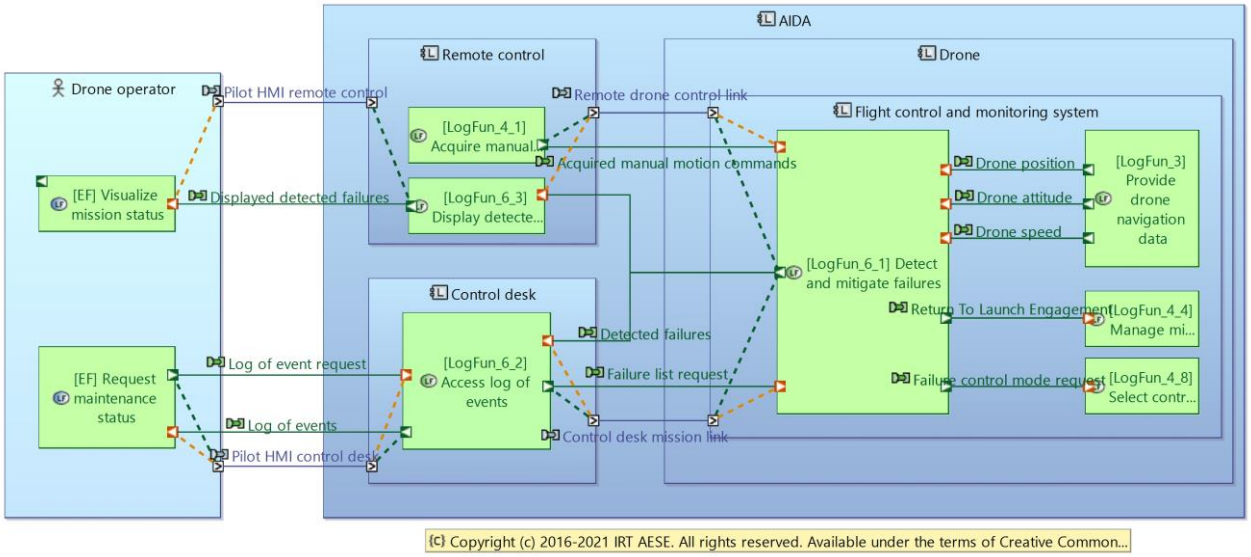


Figure 39 : contextual logical architecture of LogFun_6

4.3.7.[LogFun_7] Analyse acquired visual information

This function is directly allocated to the control desk and does not require breakdown. The figure below represent the contextual architecture diagram of this function :

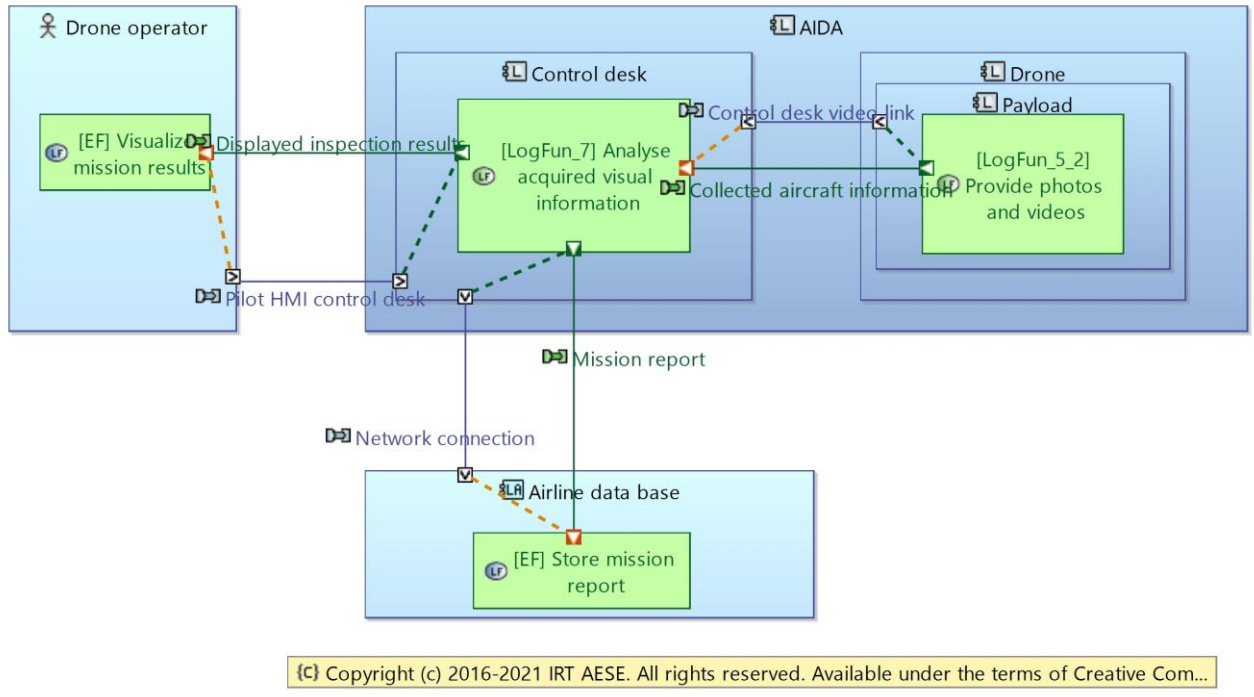


Figure 40 : contextual logical architecture of LogFun_7

4.4. Resulting architecture

The global view of the resulting architecture is represented on the diagram below :

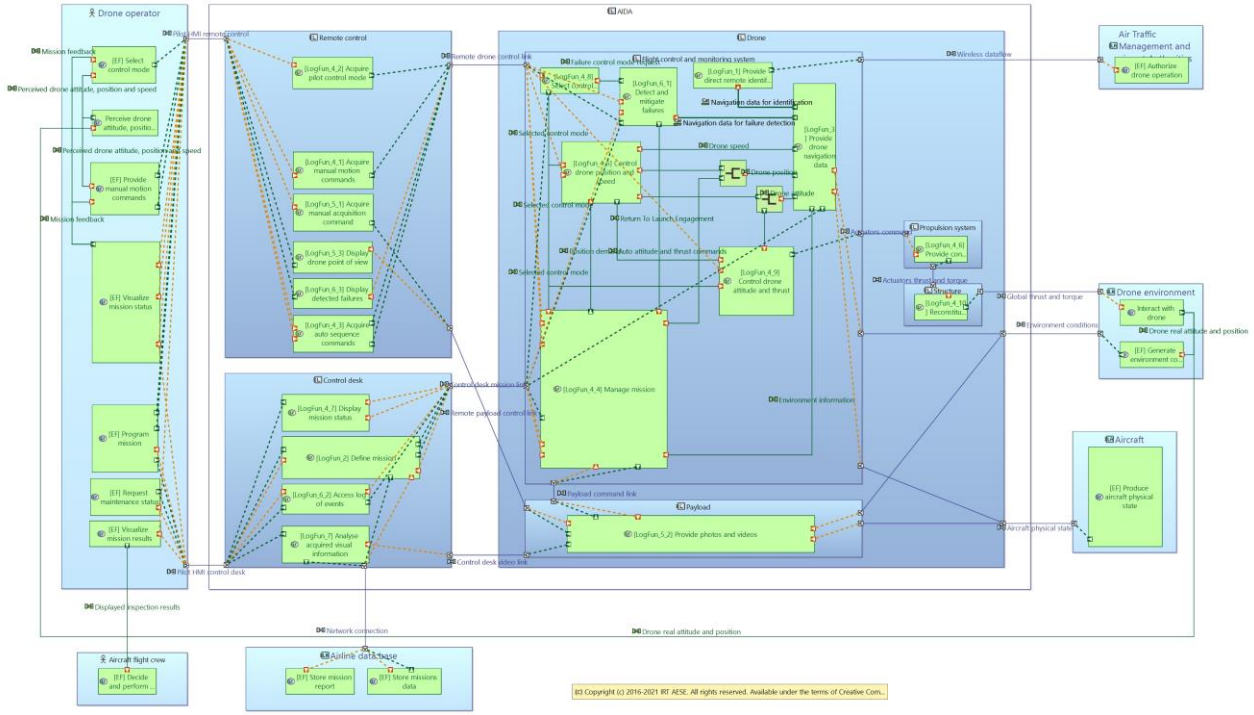


Figure 41 : logical architecture diagram

A synthesis of the functions allocated to each component is available in the table below :

Remote control functions	Control desk functions	Flight control and monitoring system
[LogFun_4_1] Acquire manual motion commands	[LogFun_2] Manage mission	[LogFun_1] Provide direct remote identification information
[LogFun_4_2] Acquire pilot control mode commands	[LogFun_4_7] Display mission status	[LogFun_3] Sense drone state and environment
[LogFun_4_3] Acquire auto-sequence selection	[LogFun_6_1] Access log of events	[LogFun_4_4] Manage mission
[LogFun_5_1] Acquire manual acquisition command	[LogFun_7] Analyze acquired visual information	[LogFun_4_5] Control drone position and speed
[LogFun_5_3] Display drone point of view		[LogFun_4_8] Select control mode
[LogFun_6_3] Display detected failures		[LogFun_4_9] Control drone attitude and thrust
		[LogFun_6_2] Detect and mitigate failures

Payload functions	Propulsion system fonctions	Structure fonctions
[LogFun_5_2] Provide photos and videos	[LogFun_4_6] Provide controlled actuators thrust and torque	[LogFun_4_10] Reconstitute global thrust and torque

4.5. Modes

The global mode machine in the System analysis layer is only an abstract view, and results in fact from the combined behavior of modes machines implemented in the various components.

4.5.1.Control desk modes

In the current view, the mission modes are implemented in the control desk, which is responsible for mission planning, flight authorization and results analysis. It results in the following modes machine

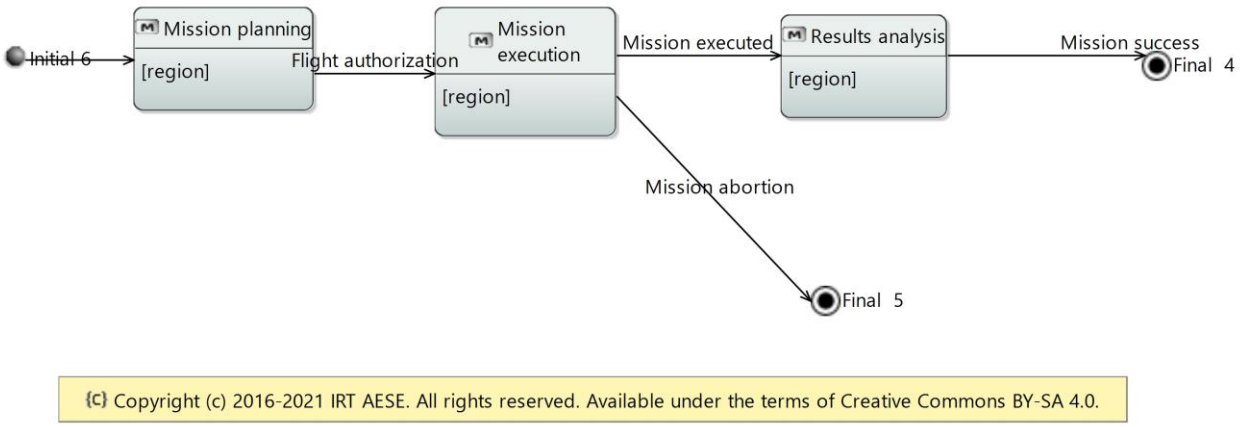


Figure 42 : mission modes diagram

4.5.2.Flight control and monitoring system modes

The flight control and monitoring system manages the control modes :

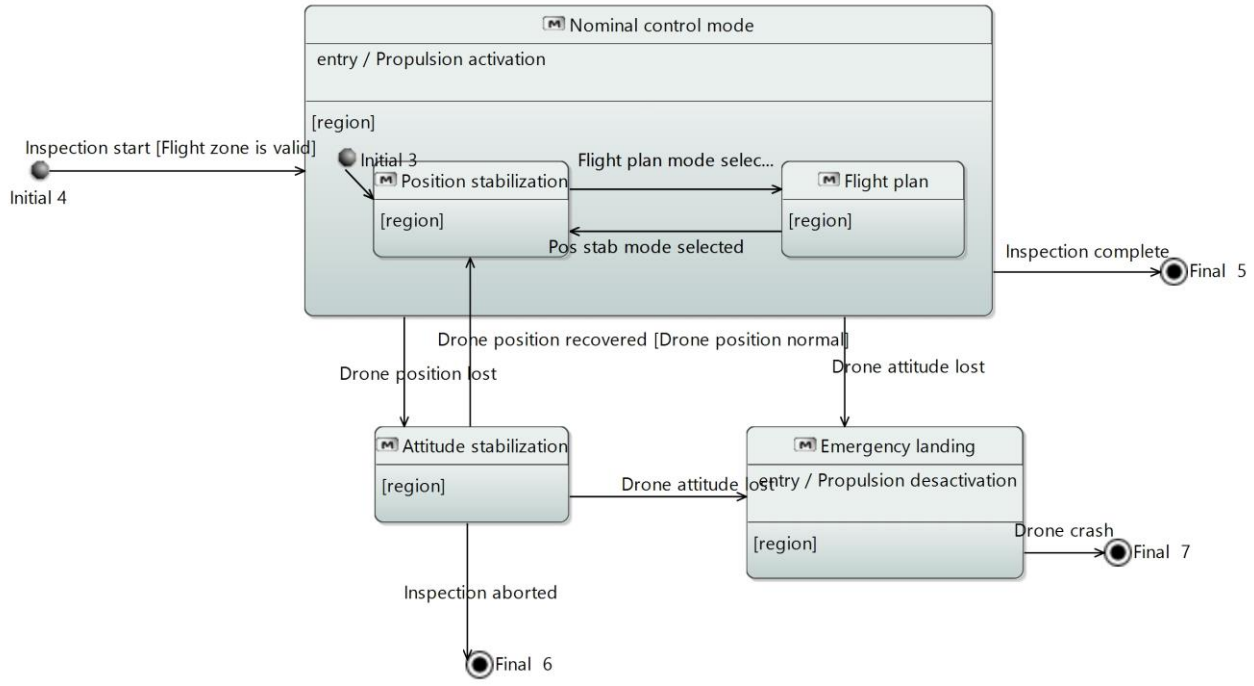


Figure 43 : control modes diagram

4.5.3.Propulsion modes

The propulsion modes are related to the propulsion system :

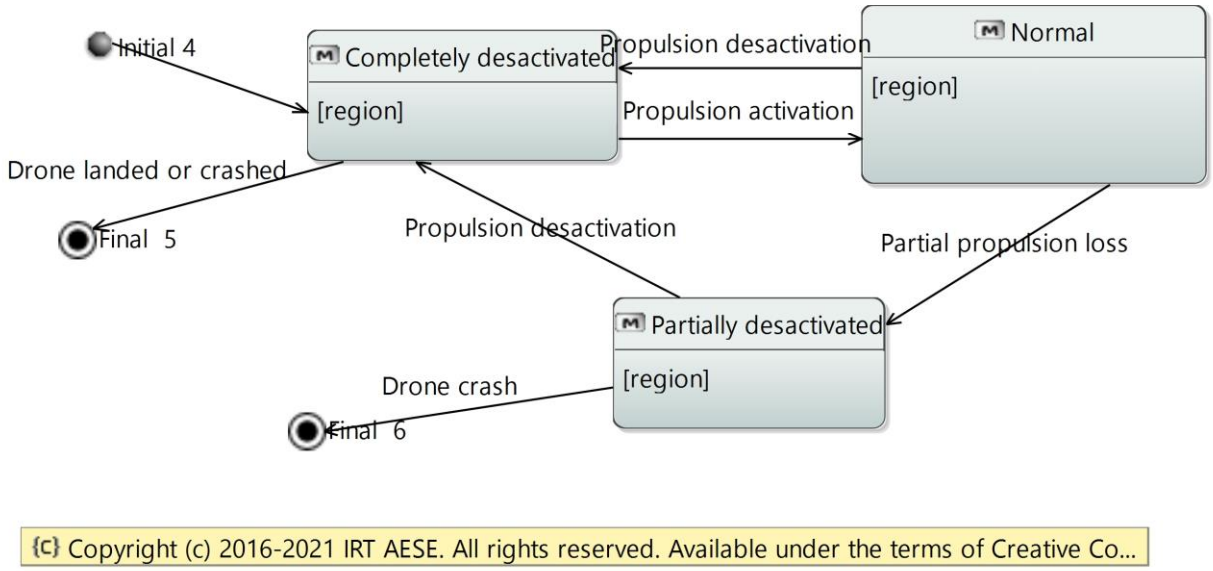


Figure 44 : control modes diagram

5. Physical architecture

Warning : as explained in 1.4, the physical architecture layer existed before the rest of the analysis was performed. Therefore, some inconsistencies remain between the architecture described in the previous chapters and the content of this chapter.

5.1. General architecture concept

We describe here the main physical architecture concepts. Detailed architecture of sub-systems will be detailed in the next section.

The chosen drone concept is a classical quadri-rotor multicopter drone. This is the most common architecture for images acquisition drones. The advantages of this architecture are widely discussed in the research literature and will not be discussed here. In the critical context of AIDA operations, this architecture is expected to offer good reliability characteristics, mainly because of its mechanical simplicity (no complex mechanism or turning joints involved).

The relevancy of this architecture choice could be further studied, and alternative designs may be considered : various numbers of actuators (hexa or octo-copters), introduction of additional degrees of freedom, decoupling of lift and horizontal displacement,… .

5.2. Sub-systems architecture

The logical architecture layer resulted in the identification of the main components and sub-systems of the AIDA system, without focusing on the actual implementation and technological choices. Here, we get into the details of each logical component and define these implementation and technological choices.

The purpose is to present and justify one possible design. One must have in mind that the compliance of the presented architecture with system requirements (performances,etc…) has not been studied. Only the safety studies are performed, in the context of the S2C project. As the project is still going on, the results of these safety studies are not presented in detail here. We expect that these studies will identify possible flaws and weaknesses in the current design, and will therefore drive further system evolutions in order to improve the safety of the system.

5.2.1.Propulsion system

As explained before, the chosen drone architecture is a quadri-rotor multicopter drone. For industrial simplicity and easier control laws implementation, the propulsion system consists in four identical “propulsions units”. The expression “propulsion unit” designate the sub-system composed of the electrical engine, the propeller and the engine speed controller.

5.2.1.1. Functional breakdown

In the logical architecture layer, the propulsion system implements the function “[LogFun_4_6] Provide controlled actuators thrust and torque”. In the physical architecture layer, this function is realized by four identical functions that correspond to the four propulsion units : “[SF1.X] Control propeller X”, X=1..4.

The figure below, extracted from the PA layer of the Capella model, shows the breakdown of the SF1.X functions, and also the engine control and monitoring functional chains :

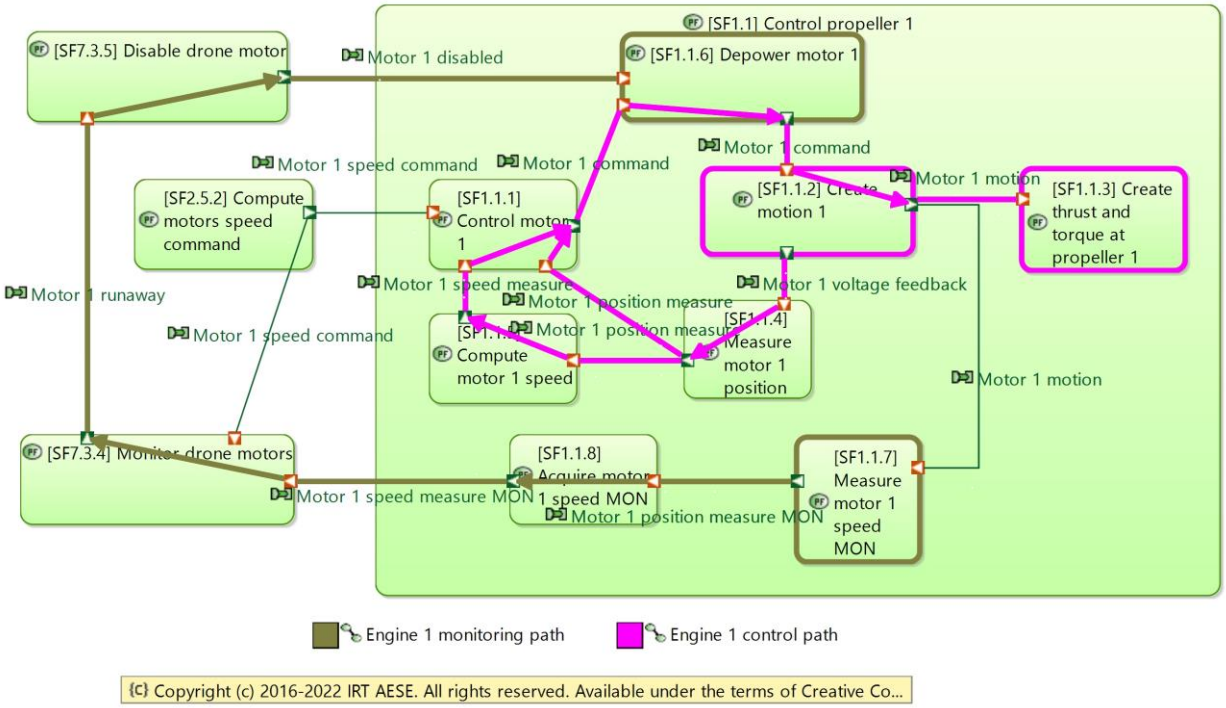


Figure 45 : SF1.X breakdown and engine control loops

The breakdown of SF1.X mainly consists in two functional chains :

- A control functional chain, corresponding to SF1.X.1 to SF1.X.5. This functional chain, in pink on the diagram above, goes from the motor speed command (produced by SF2.5.2) to the thrust and torque creation.
- A monitoring functional chain, corresponding to SF1.X.6, SF1.X.7 and SF1.X.8, which also involves sub-functions of SF7 (see 5.2.2). This functional chain consists in measuring the engine rotation speed, detecting discrepancies with the commanded speed and disabling the power supply if the engine is failied.

5.2.1.2. Safety considerations

The safety analysis has established that an erroneous thrust creation by one propulsion unit is a Catastrophic (CAT) event. Indeed, this could result in the loss of control of the drone, with drone flying out of the flight zone and the eventuality of a collision with a flying aircraft. Because of the CAT classification of this failure, it is required that a single failure does not result in an erroneous thrust creation by one propulsion unit.

In the previous versions of AIDA, it was considered that in this case, the engine shall be shut down, and that the operator can take the control of the drone in manual mode, in a degraded way (3 propulsion units left), and land the drone without further consequences. Further reflection on this topic have shown that the manual or automatic control of the drone with only three propellers is not realistic. Therefore, it is known that the current architecture does not comply with the safety requirements.

The foreseen architecture target to comply is the following :

- An erroneous motion generation (or thrust creation) is detected by the monitoring functional chain, which triggers the engine shutdown. Segregation between the control and monitoring paths is required.
- The flight control system then adapts the thrust allocation to the remaining propulsion units. It implies that :

- The flight control system is aware of the failed propulsion unit shutdown
- There are enough remaining propulsions unit to ensure the controlability of the drone (even with degraded performances) so that a safe Return-To-Home and landing can be ensured.

The current architecture (V4.5) complies with the first point, but not with the second :

- The thrust allocation does not take into account the actual state of each propeller (nominal or failed)
- There are not enough propellers remaining to ensure the safe return of the drone.

In the next version of AIDA architecture, evolutions are expected to take into account these considerations.

5.2.1.3. Physical implementation

The Arcadia method suggests to perform the functions allocations to components in two steps, and identify two types of components :

- The behavior components (in blue), that bear the functional behavior defined by the functions.
- The node components, which do not have a functional behavior on their own, but on which the behavior components are deployed. These node components provide the non-functional ressources (electrical power, casing,…) needed by the behavior components.

In our case, the “Propulsion unit behavior” consists in the following behavior components :

- The Motor controler, which ensures the control closed-loop of the motor rotation speed
- The Motor protection, which ensures the acquisition of the rotation speed measure for protection purpose and the motor de-powering
- The motor, which converts the control signals and electrical power into rotation motion, and includes the rotation speed measure function.
- The propeller, which convert the rotation motion into thrust and torque

See the diagram below for detailed functional allocation:

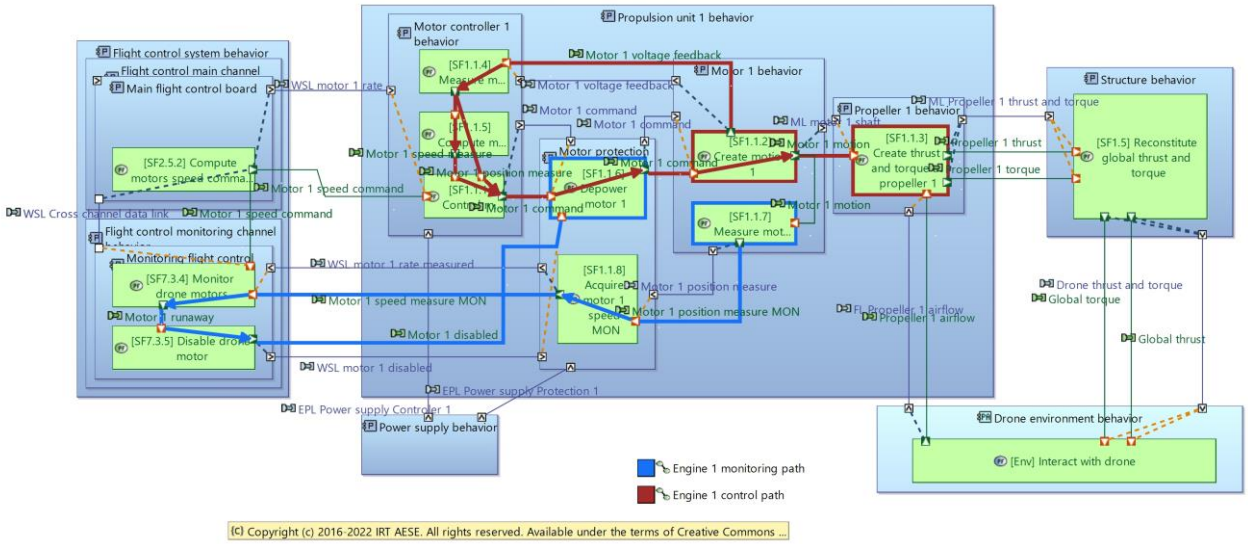


Figure 46 : Propulsion unit functional allocation

This diagram directly shows that the segregation between the control and the monitoring functional chains is ensured, on the condition that the motor controller and motor protection components are independent.

In further versions, we will probably modify the allocation of SF7.3.4 and 7.3.5 so that the whole monitoring function chain is realized by the motor protection component (and not by the flight control system).

The deployement of these behavior components into node components is represented on Figure 47. We can see here the complete deployement of the four Propulsion units, realizing the Propulsion system.

We made the choice to deploy the Motor control and Motor protection behavior components into a single node component (the Motor Control and Protection System, or MCPS), so that a single casing will englobe both control and protection functions. The independence constraint between those functions will be one of the safety requirements for the provider of the MCPS. The main reason for this choice is to limit the number of components.

The Figure 47 also shows :

- The communication principles between the Flight Control system and the MCPS (detailed in section 5.2.2)
- The power supply architecture for each Propulsion Unit detailed in section 5.2.4)
- The load paths between each propeller and the structure : the propeller is not attached directly to the structure, the thrust and torque generated by the propeller goes to the motor stator which is fixed on the structure (represented by the « Motor Support » physical links)
- Each MCPS is attached on the structure

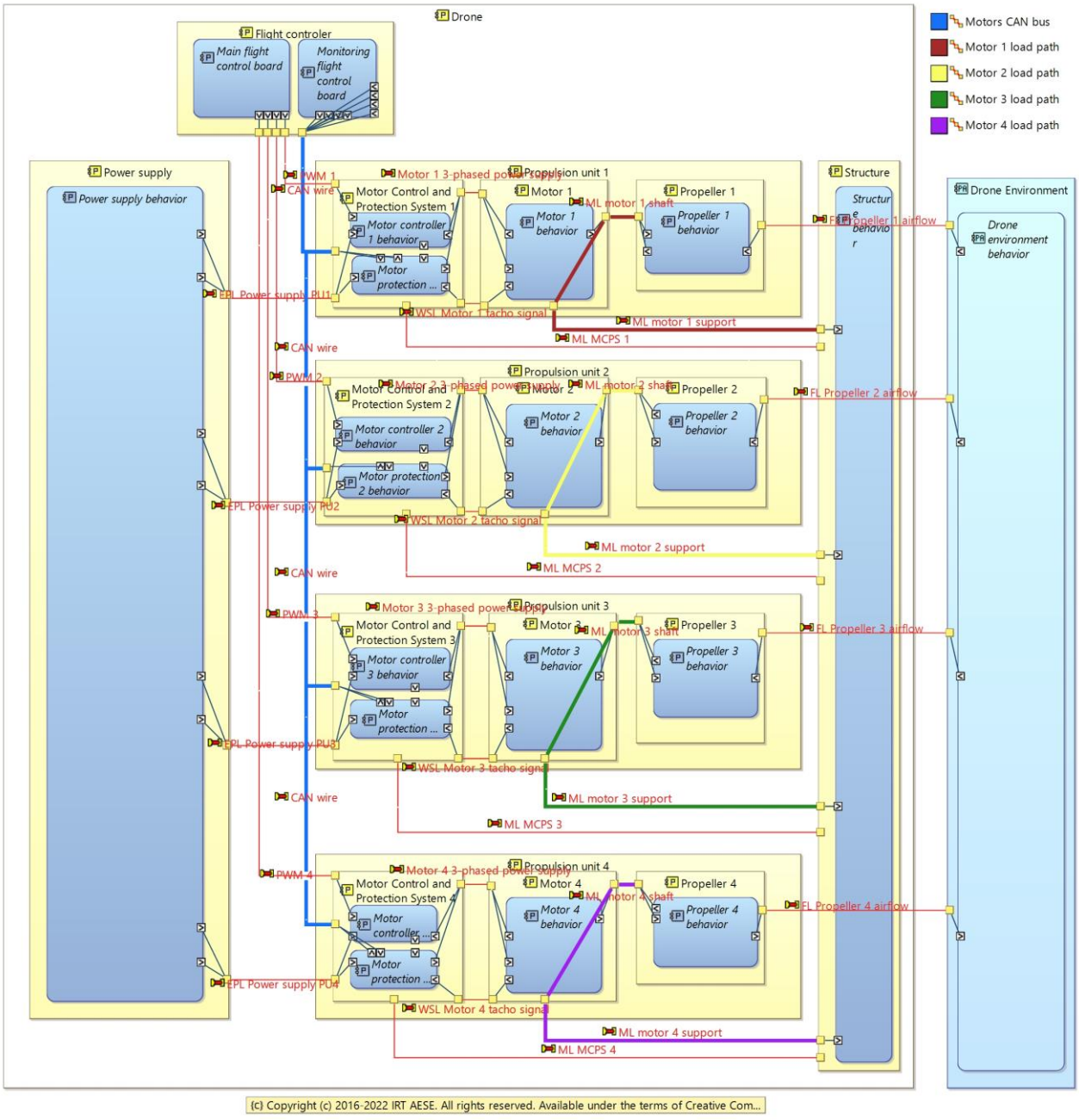


Figure 47 : Propulsion system behavior components deployment

5.2.1.4. Technological choices

The motors considered for the AIDA application are brushless DC motors. This is the most current practice for multi-copter drones.

These motors are powered by three-phased alternative currents. The frequency of the power supply must be consistent with the motor rotation speed. The control of these motors requires an angular position measurement, in order to ensure the adequate commutation and alternative currents generation. Several solutions exist :

- A sensor (resolver, hall effect,…) provides the angular position of the rotor
- The position is computed from the voltage measurement on one of the phasis (sensor-less solution).

Here, the second solution has been arbitrarily selected (with the position measure and speed computation being directly performed by the MCPSs). Deeper studies could lead to a more enlightened choice, for example taking into account weight, robustness, cost, performances, etc… of each solution.

Here, the sensor-less solution is chosen for the motor speed control (SF1.1.4 and SF1.1.5) : the control channel of the MCPS has direct access to the third phase and use it for the motor closed-loop control.

For the speed monitoring, a sensored solution is favoured : the motor embeds a speed sensor (SF1.1.7, ex : tachometer), which is acquired by the Motor protection channel (SF1.1.8) and sent to the Flight controller.

5.2.2.Flight control system

The Flight control system is the most complex sub-system of AIDA. It gathers all the drone control and monitoring functions, and is greatly impacted the safety constraints.

In the current version, only the high level principles of the drone control and monitoring are represented. Detailed behavior and performances studies, including simulation in dedicated tools, would be needed to detail and validate the control and monitoring algorithm.

A quite simple hardware architecture is proposed, taking into account some safety constraints.

5.2.2.1. Functional breakdown

Because the functional architecture of the flight control system is quite complex, dressing the full correspondence between the functions allocated to the Flight Control system in the Logical Architecture and the Physical architecture layers is not easy. However, the great principles are the same in both modelling layers.

Functions in Logical Architecture layer	Functions in Physical Architecture layer
[LogFun_1] Provide direct remote identification information	[SF8] Provide direct remote identification information
[LogFun_3] Sense drone state and environment	[SF3] Provide drone navigation data
[LogFun_4_4] Manage mission	[SF4.1] Acquire and store flight plan [SF4.2] Run automatic control
[LogFun_4_5] Control drone position and speed	[SF2.1] Automatically control drone position
[LogFun_4_8] Select control mode	[SF4.3] Select control mode
[LogFun_4_9] Control drone attitude and thrust	[SF2.3] Control attitude [SF2.4] Control altitude [SF2.5] Compute motor commands
[LogFun_6_2] Detect and mitigate failures	[SF7] Monitor drone control

The functional breakdown for all these functions will not completely detailed, however we will describe the principles of each control function.

5.2.2.1.1. Flight control modes

The control mode selection state machine implemented in the Physical Architecture layer is represented on the diagram below :

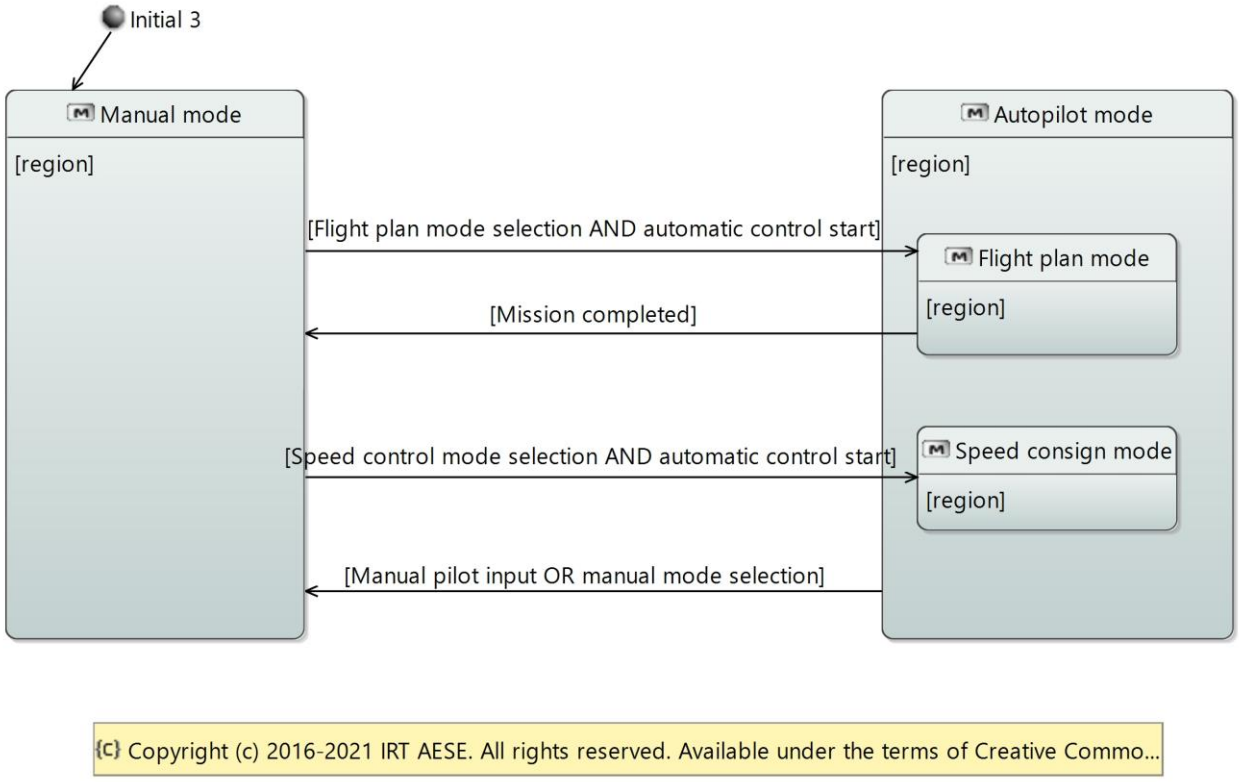


Figure 48 : control modes diagram in PA layer

It is quite similar to the one proposed in the Logical layer, with the following correspondance :

Control mode in Logical Architecture layer	Control mode in Physical Architecture layer
Nominal control mode	Autopilot mode
Flight plan	Flight plan mode
Position stabilisation	Speed consign mode
Attitude stabilisation	Manual mode
Emergency landing	*Not implemented in PA layer*

Several differences are identified, and may be resolved in future system versions :

- The emergency landing mode, which consist in cutting off the power supply to actuators and deploy the emergency landing device, is not implemented in the PA layer
- In LA layer, the transition to the Attitude stabilisation mode can be triggered by fault detection logic (loss of position data). This is not the case in PA layer, where the transition is only triggered by the operator

The control mode selection is performed by SF4.3.

5.2.2.1.2. Mission management

The mission management consists in :

- storing and executing the pre-defined flight plan when in Flight Plan mode
- computing the speed consign from pilot inputs when in Speed consign mode

It is realized by the functions SF4.1 and SF4.2. In manual mode, these functions are not exploited.

The diagram below shows the breakdown of SF4 :

- SF4.1 and SF4.2 related to mission management
- SF4.3 which realises the control mode selection (see previous section)
- SF4.4 which realises the acquisition of operator commands for autopilot modes (see control desk section)

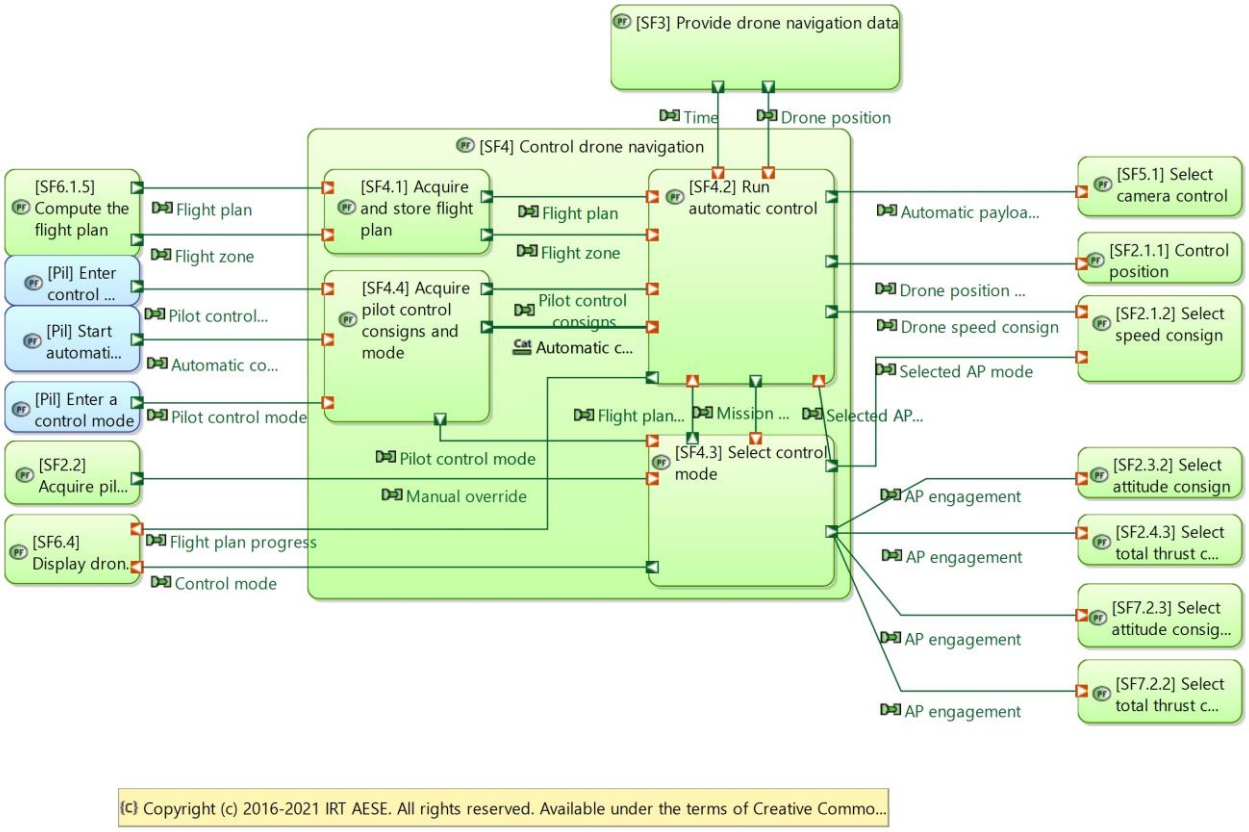


Figure 49 : SF4 breakdown

5.2.2.1.3. Drone navigation

SF3 performs the acquisition of navigation data, which are needed for the drone control. It includes the following data :

- Position and time
- Speed : ground speed and vertical speed
- Attitude : yaw, pitch, roll
- Attitude rate : yaw rate, pitch rate, roll rate

The principles for measuring and computing these data is quite common: various physical values are acquired individually, then fused by a more or less complex algorithm (usually a kind of kalman filter) to compute the navigation data.

The diagram below shows the breakdown of SF3 and gives an exhaustive view of all measured physical phenomenon and computed data.

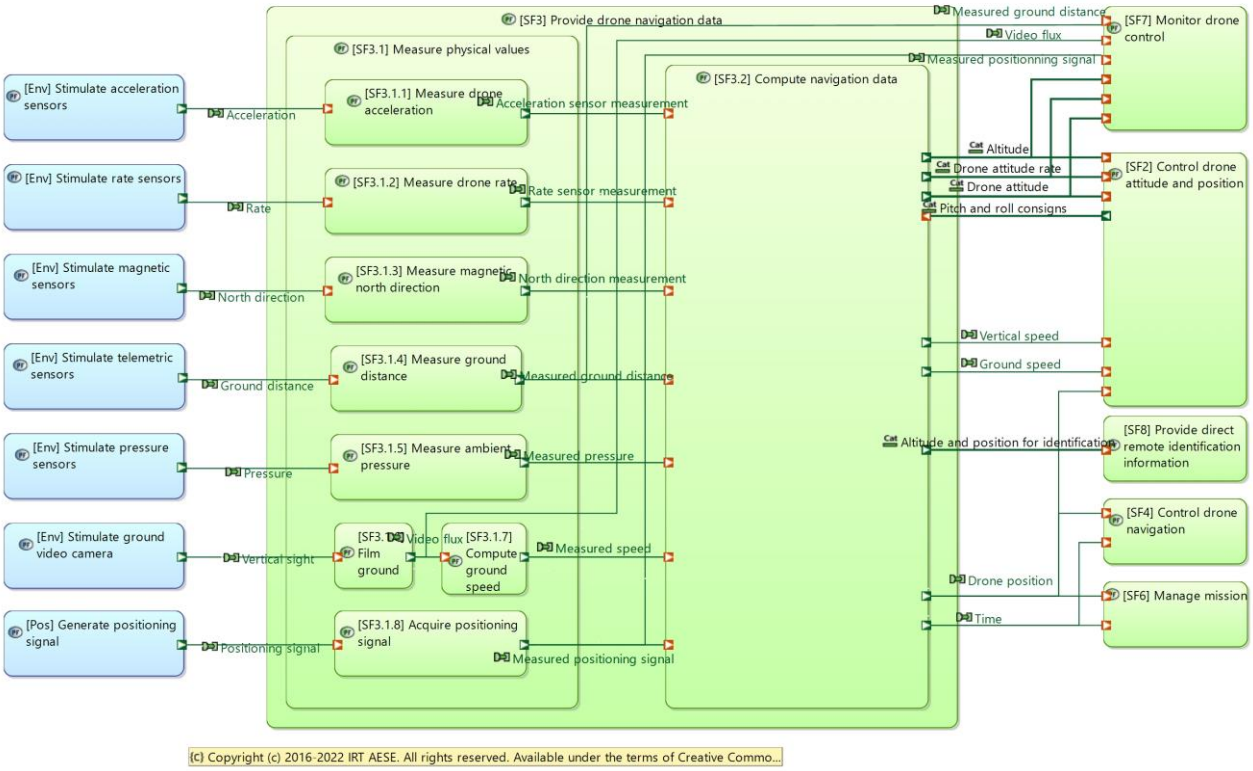


Figure 50 : SF3 breakdown

5.2.2.1.4. Drone position and speed control

The drone position and speed control, realized by SF2.1, consists in implementing the position and speed control closed-loop, and ensuring the adequate selection of the speed consign depending on the selected control mode. This function provides then the attitude and thrust consigns to be achieved by the drone in order to reach the required position and speed.

Again, this function is only executed in one of the autopilot modes.

The breakdown of SF2.1 is represented on the diagram below :

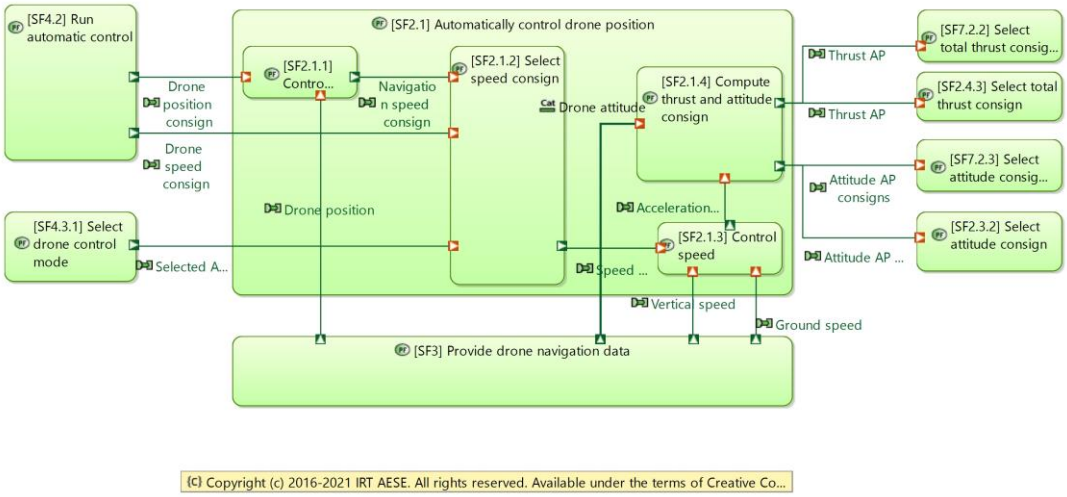


Figure 51 : SF2.1 breakdown

5.2.2.1.5. Drone attitude and altitude control

The control of drone attitude and altitude consists in :

- Selecting the appropriate attitude and thrust commands, coming either from the autopilot functions or directly from the operator through the remote control, depending on the control mode
- Implementing the closed-loop control on each attitude axis and on altitude to compute the torque required on each axis and the vertical thrust
- Computing the actuators command to achieve these required torques and thrust.

The functions SF2.3, SF2.4 and SF2.5 realise these operations. There are represented on the diagram below, which shows the breakdown of SF2 :

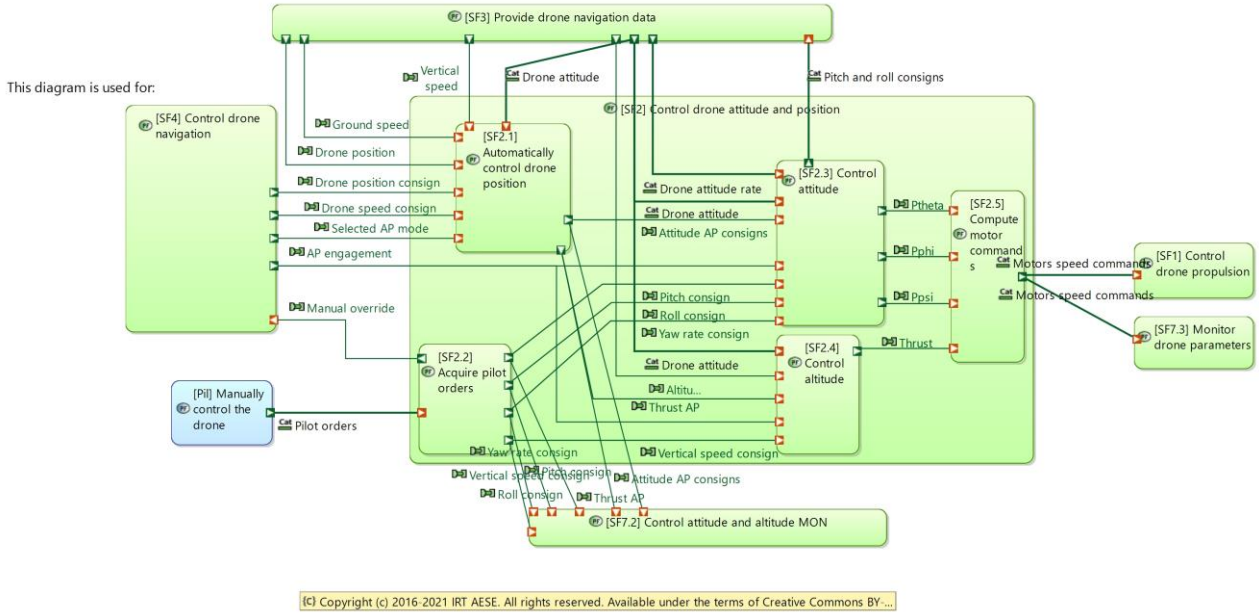


Figure 52 : SF2 breakdown

The internal breakdown of SF2.3, SF2.4 and SF2.5 is not detailed here. Refer to the model for more details.

5.2.2.1.6. Drone control monitoring

The drone control monitoring strategy is the following :

- Failures related to the automatic control of the drone position are detected visually by the drone operator, who triggers the switch to manual mode and lands the drone safely. This does not require any dedicated function
- Failures related to the attitude and altitude control of the drone cannot be mitigated by the operator in manual mode, because the related functions (SF3, SF2.3, SF2.4 and SF2.5) are exploited both in automatic and manual mode. Therefore, a monitoring solution to detect and accomodate the failures of these functions must be implemented. This monitoring functional chain consists in performing in parallel the same computation as the main control functional chain, comparing the results of both functional chains and triggering the cut off of all motors when a discrepancy is detected.

The function “[SF7] Monitor drone control” realises this monitoring. Its breakdown, represented on the diagram below, is the following :

- SF7.1 is similar to SF3. It acquires some physical parameters and computes the navigation data required for the other control functions. There is one subtility, related to the physical implementation : whereas the acquisition of inertial (linear accelerations and rotation rates), magnetic (north direction) and anemometric (ambient pressure) parameters is performed by dedicated sensors for the monitoring channel, the others (ground distance, optical flow and GPS) are provided by the same sensors than for the main control channel.
- SF7.2 perfoms the same computation as SF2.3, SF2.4 and SF2.5, based on the same inputs
- SF7.3 compares :
    - Navigation data from SF3 and from SF7.1
    - Actuators commands from SF2.5 and SF7.2
    - Actuators commands from SF2.5 and actuators real rotation speed from SF1 (see the details on motor monitoring in 5.2.1)
- And trigger a shutdown of one motor (if only this motor is failed) or all the motors (when a discrepancy is detected for navigation data or actuators commands computation)

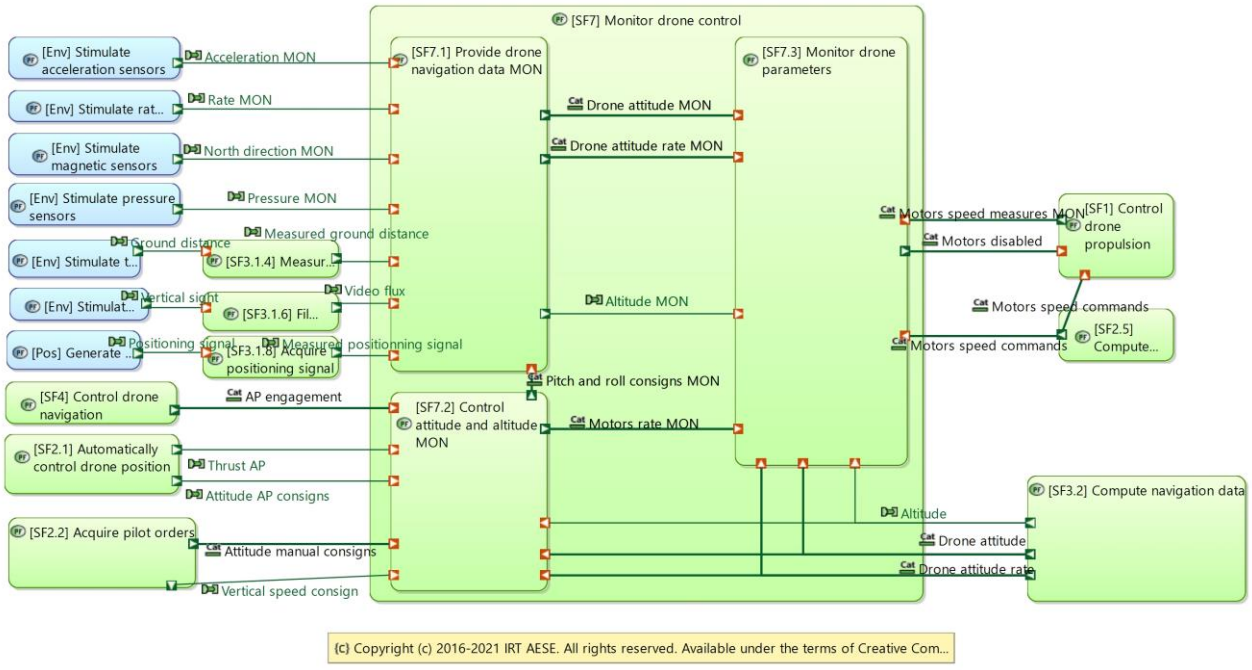


Figure 53 : SF7 breakdown

5.2.2.2. Safety considerations

As explained in the previous section, the failure detection and mitigation strategy is based on two principles :

- Failures leading to an erroneous position control in automatic mode are mitigated directly by the operator, who observes the drone and take back the control in manual mode. The hypothesis behind that is that an erroneous position control is « slow enough » so that the operator has the time to react before the drone goes out of the authorized area
- Failures leading to an erroneous attitude or altitude positions are mitigated directly by the drone itself, thanks to the drone control monitoring function (SF7). Here, a loss of drone control is a quick phenomenon which is not compatible with an usual human reaction time. In this case, the motors are shut down and the drone crashes inside the authorized zone (which is a Hazardous event).

Both types of failure can lead to a Catastrophic event : the drone may fly out of the authorized area, with a possible collision with a flying aircraft. Therefore, the architecture must be robust to single failures leading to :

- An erroneous position control in automatic mode combined with an incapacity to control the drone in manual mode (including the incapacity to switch to manual mode)
- An erroneous attitude or altitude control combined with the incapacity to detect this failure and shut down the drone motors.

In order to ensure this robustness inside the flight control system, these constraints can be translated as follows :

- The functional chains related to drone automatic position control (constituted of SF4.1, SF4.2 and SF2.1) and to the manual mode selection (constituted of SF4.3) shall be independant
- The functional chains related to the drone attitude and altitude control (constituted of SF3, SF2.3, SF2.4 and SF2.5) and to the drone control monitoring (constituted of the sub-functions of SF7) shall be independant

Note : these are only the high level principles for the integration of safety constraints in the flight control system architecture. The actual compliance of the proposed detailed architecture must be evaluated with the appropriate tool (an MBSA model in the context of the S2C project).

Other functions, involved in these functional chains but not allocated to the flight control system, will be studied in the related section. They concern the remote control and the control desk.

5.2.2.3. Physical implementation

To allow the correct allocation of functions on the flight control system components while taking into account the previously formulated safety constraints, several architecture principles are possible. Some of them are quite common in the aeronautic industrial world : COM-MON, full duplex, triplex,… .

Here, we made the choice of simplicity, which reduces the number of components : the flight control system is composed of :

- A flight control main channel, which contain the main flight control electronic board and integrated sensors : magnetometer, inertial reference sensors, pressure sensor
- A flight control monitoring channel, which is similar to the main channel
- Some external sensors, which are common for both channel : a LIDAR altimeter, a GPS receiver and an optical flow sensor.

The resulting architecture for the flight control system is detailed on the diagram below.

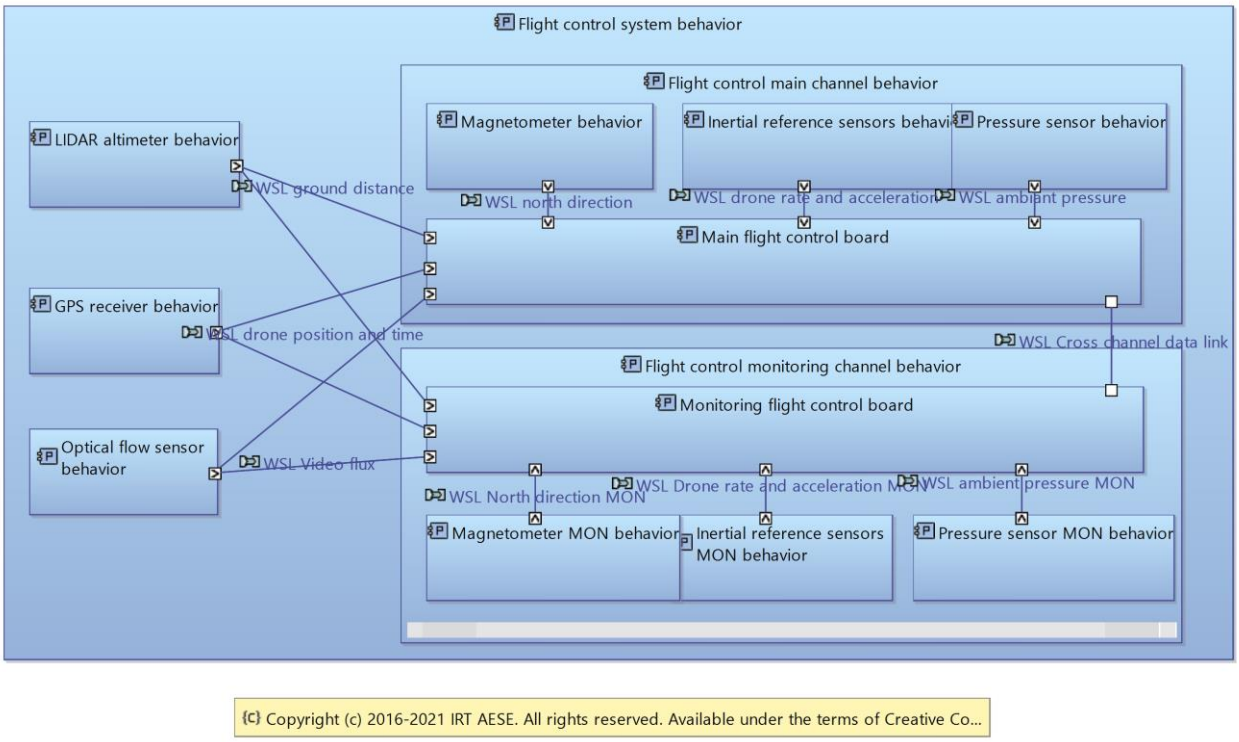


Figure 54 : Flight control system behavioral architecture

The functional allocation to this behavior architecture is a bit complex to visualize. The corresponding diagrams are represented below (with focus on each channel), but for better understanding it is advised to visualize them directly in the model. Here are the main principles :

- The functions related to the automatic position control are allocated to the Flight control main channel, and the functions related to control mode selection are allocated to the Flight control monitoring channel. In this way, the required independance is ensured.
- The functions related to attitude and altitude control are allocated to the Flight control main channel, and the functions related to the drone control monitoring are allocated to the Flight control monitoring channel. In this way, the required independance is ensured.

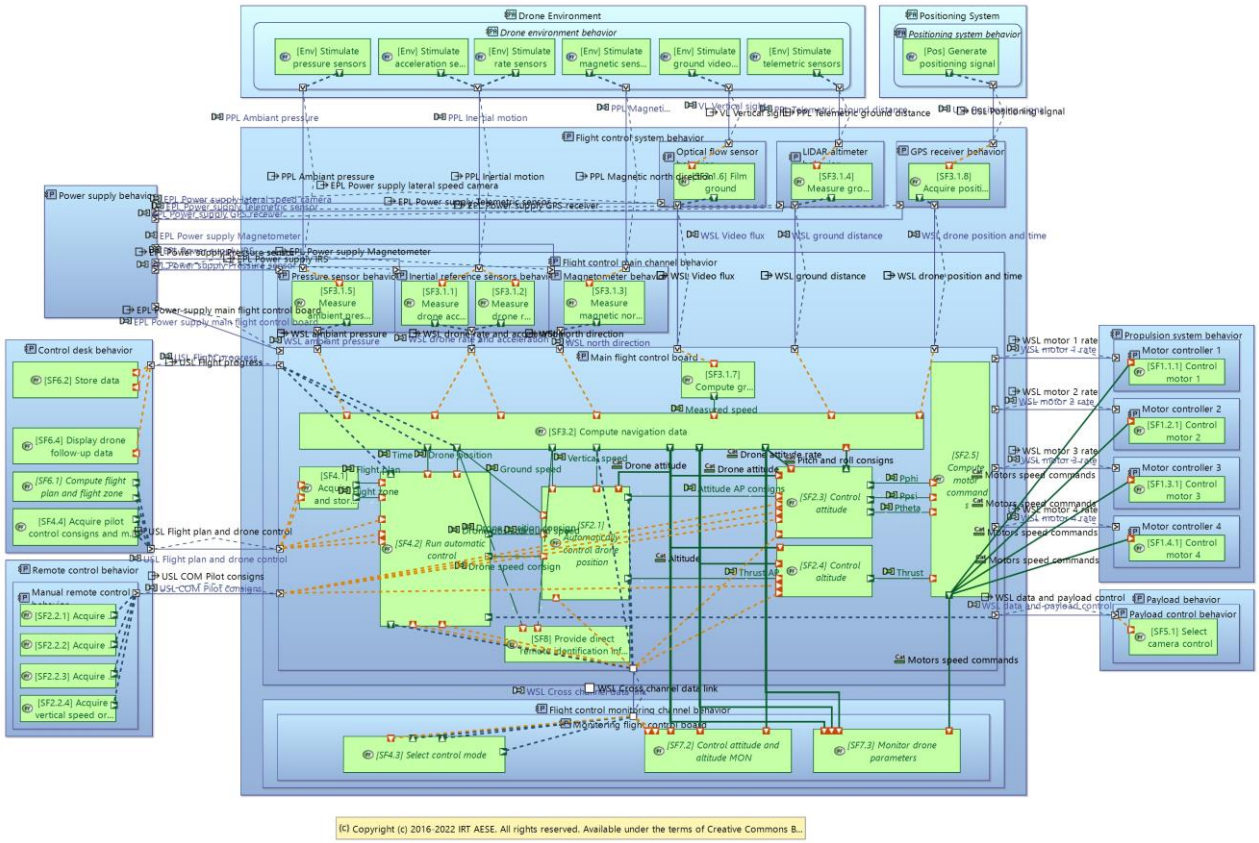


Figure 55 : Functional allocation on Flight control main channel

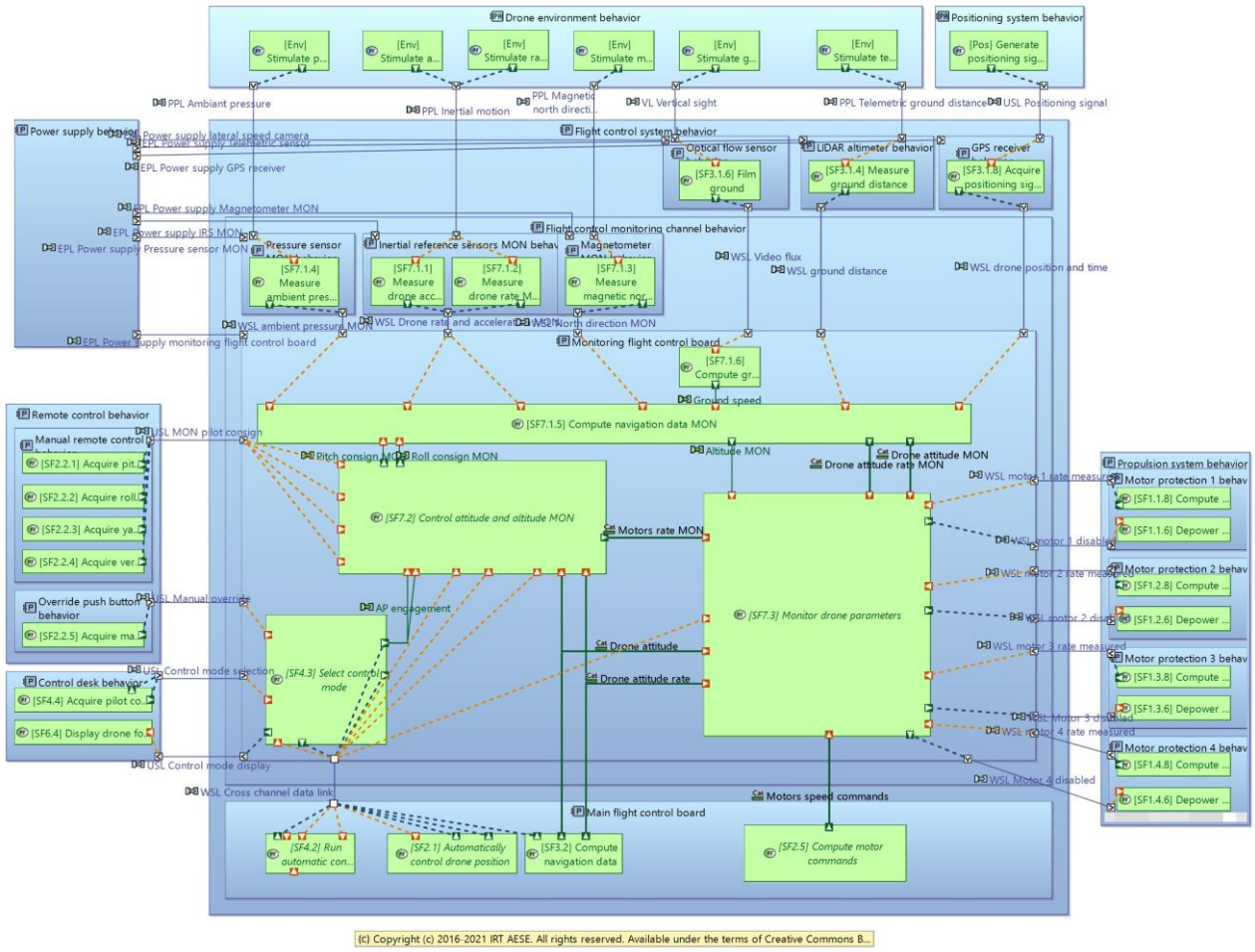


Figure 56 : Functional allocation on Flight control monitoring channel

The architecture choices performed here have been driven by simplicity, and many improvement could be proposed. One example is the robustness of navigation data consolidation :

- as the measured acquired by each channel are not exchanged and consolidated with the other channel, a single “erroneous” failure of one these measures on the main channel is not detected before the data fusion performed by SF3.2, and leads directly to an erroneous of computed navigation data (which is detected by the monitoring channel, and then triggers the shut off of the drone motors).
- similarly, a single « erronenous » failure of one of these measures on the monitoring channel leads to an erroneous shut down of the drone motors, while the drone control may still be possible

Quantitative safety studies have not been performed yet, but this will probably have an important impact on the availability of the system, and on the compliance to the Hazardous failure condition (drone crash inside the authorized area).

This could be greatly improved with the following principles :

- after acquisition by each channel (SF3.1 and SF7.1), the measured values are exhanged on the cross channel datalink
- measurement issued from both channels are compared, and the data is declared invalid if a discrepancy is detected
- the data fusion function takes into account this invalidity to compute, when possible, the required navigation data taking into account this invalidity status

The behavior components of the Flight Control system behavior are then deployed on the node components representing the real implementation of the flight control system. This is represented on the diagram below.

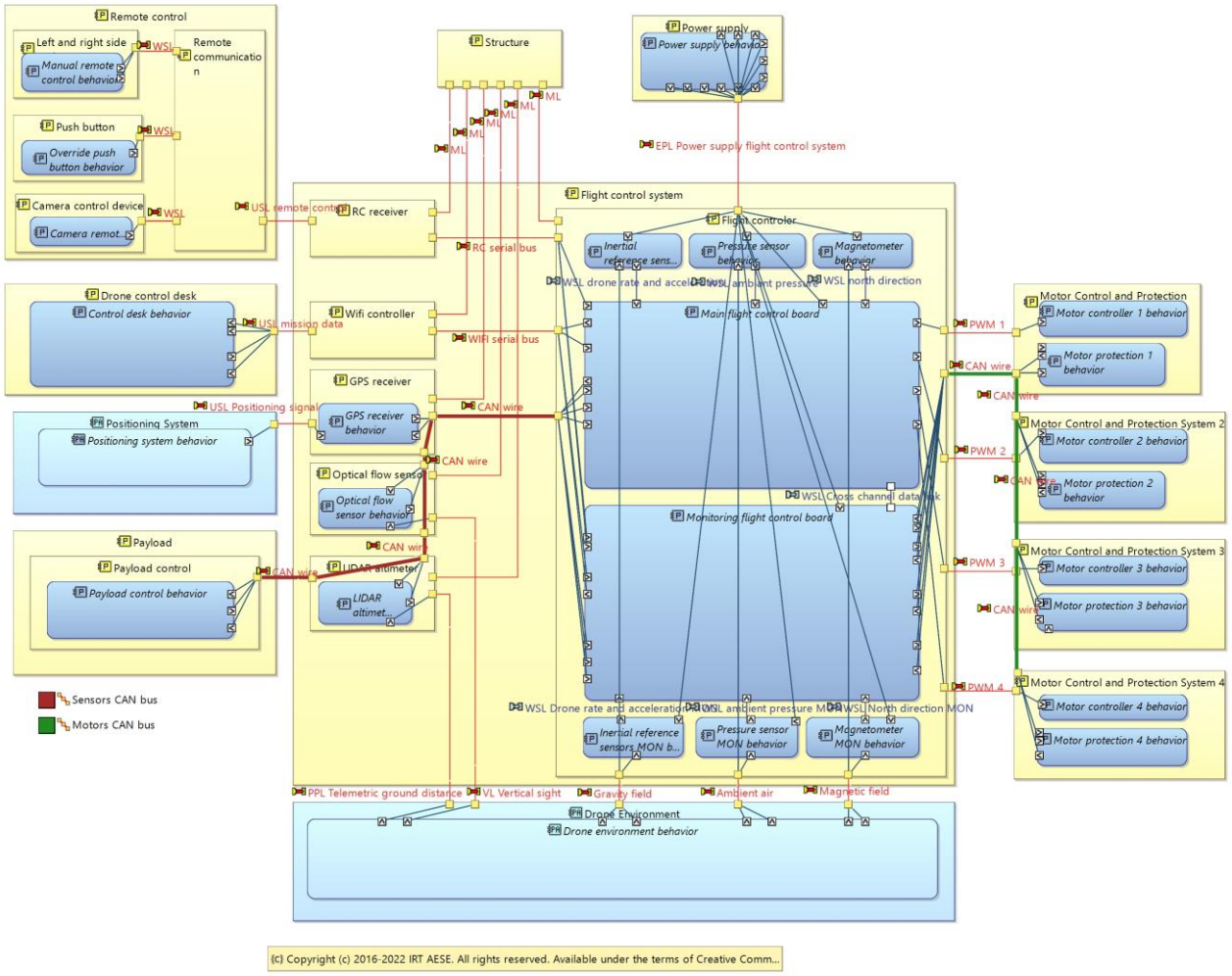


Figure 57 : Flight control system behavior components deployment

Here are the main features of this architecture :

- both the Flight Control Main channel (consisting in the Main Flight control board and associated sensors) and Flight Control Monitoring channel (consisting in the Monitoring Flight control board and associated sensors) are deployed in a single node component, the Flight controller. It means that we expect a single item containing both channels, and ensuring the independance requirements between them.
- This Flight controller distributes electrical power to all its components which require electrical power supply.
- The three external sensors are deployed on independant node components. The communication between the Flight Controller and these external sensors is ensured by a serial CAN bus (the Sensors CAN bu, which also ensure communication with the payload. As explained in 5.2.4, this CAN bus also provides the power supply for those external sensors.
- Another CAN (the Motors CAN bus) is implemented for the communication between the Monitoring channel and the Motor protection part of the MCPSs.

- The Motor commands computed by the Main channel of the Flight controller are sent to the Control part of the MCPSs through dedicated Pulse Width Modulation (PWM) signals
- Additional node components are added, to ensure communication with the control desk and the remote control : the RC receiver and the Wifi controller. In this version of the model, there are no functions allocated to these components (which could be discutable).
- All the node components constituting the Flight Control system are attached directly to the structure.

5.2.2.4. Technological choices

Many technological choices have been made here, deliberately or unintentionally. Here are those who come to mind :

- CAN communication buses : this is a common technology, which even have a dedicated standard for UAV (called UAVCAN). However, other solutions are possible. The relevance of the bus topology could be evaluated with regards to safety objectives, as there is a risk of common failure causes.
- PWM communication with MCPS : again, this is the common (and most simple) way of sending actuators commands to the MCPSs. However, this analogic technology is sometimes replace by digital communications between the Flight Controller and the motors controllers, enabling for example the Flight Controller to get some feedback about the health state of the actuators. This kind of solution could be considered with the possible evolutions of the Propulsion system
- As mentionned in 5.2.3, the choice of routing the data from the payload to the control desk through the Sensor CAN bus is probably discutable, regarding performances (transmission rate) and the risk of impact on the external sensors data used for the drone control.
- In this version, there is no « down link » from the drone to the remote control. However, if such a link is implemented for a live video feedback of the drone point of view, and for alarms display on the remote control, the RC receiver should be replaced by a more complex technology enabling this down link.
- The choice of sensors technology (LIDAR, optical flow, static pressure,…) is one possible mix that offers various possibilities of data fusion. Other technological choices could be considered. Also, having a dedicated navigation system (GADIRS, etc…) instead of having these functions realised by the Flight controller could be a solution to improve the navigation data robustness and availability.
- Other technological choices of positionning are possible : GNSS, Galileo, with or without SBAS/LBAS,… which offer various levels of robustness and service continuity.

5.2.3.Payload

For the AIDA application, the payload of the drone is a camera, which can provide photos and videos. This camera is mounted on an articulated support, which allows the adjustment of the camera sight axis elevation. The azimuthal direction of the camera correspond directly to the yaw axis of the drone.

5.2.3.1. Functional breakdown

In the logical architecture layer, the payload realizes the function “[LogFun_5_2] Provide photos and videos”. This logical component is generic enough to allow various possible implementations and technological choices.

In the physical architecture layer, this function is realized by several sub-functions of SF5, SF5.1 to SF5.4 (SF5.5 corresponding to [LogFun_5_1] and being allocated to the remote control.

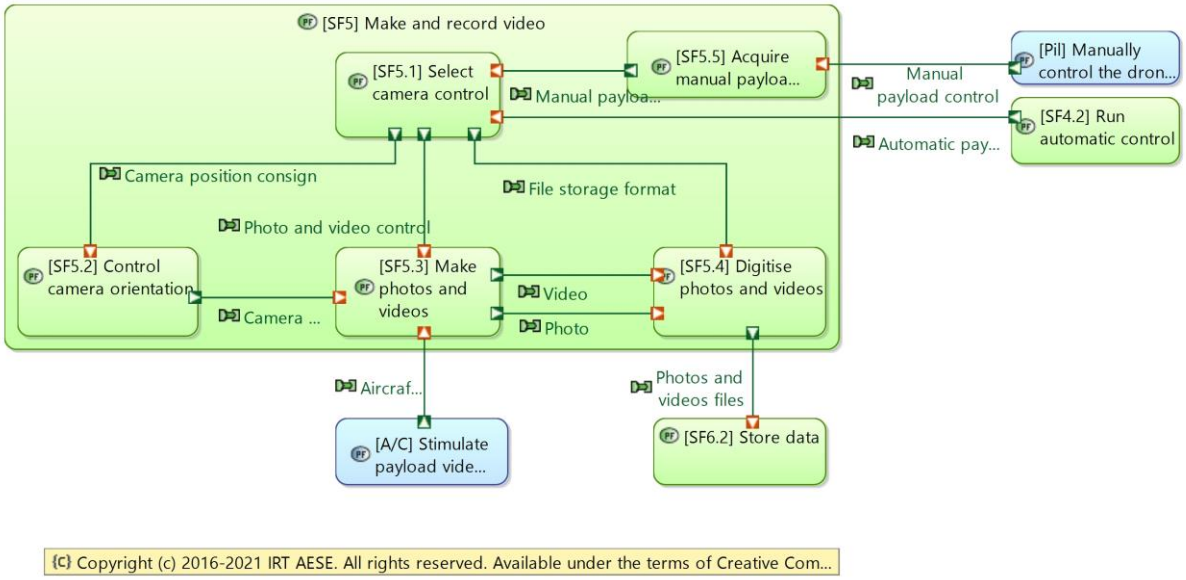


Figure 58 : SF5 breakdown

In the current version of the Physical architecture layer, the live video feedback to the operator on the remote control is not implemented. This is an expected evolution for next system versions.

5.2.3.2. Safety considerations

As the live video feedback is not implemented in this system version, a failure of the payload can "only" lead to the failure of the mission (erroneous or loss of capacity to acquire pictures) which is a Minor failure condition. It will not lead to a crash or fly away of the drone outside the authorized zone. Therefore, there is no strong safety related constraints for the payload architecture.

5.2.3.3. Physical implementation

The following behavior components of the payload have been identified :

- The Payload control, which ensure the control of the camera support and camera sensor, and realized the digitalisation (formatting, compression) of the acquired pictures and videos
- The Camera rotary support, which ensure the adjustement of the elevation of the camera sight axis
- The Camera sensor, which acquires the photos and videos.

This is represented on the diagram below :

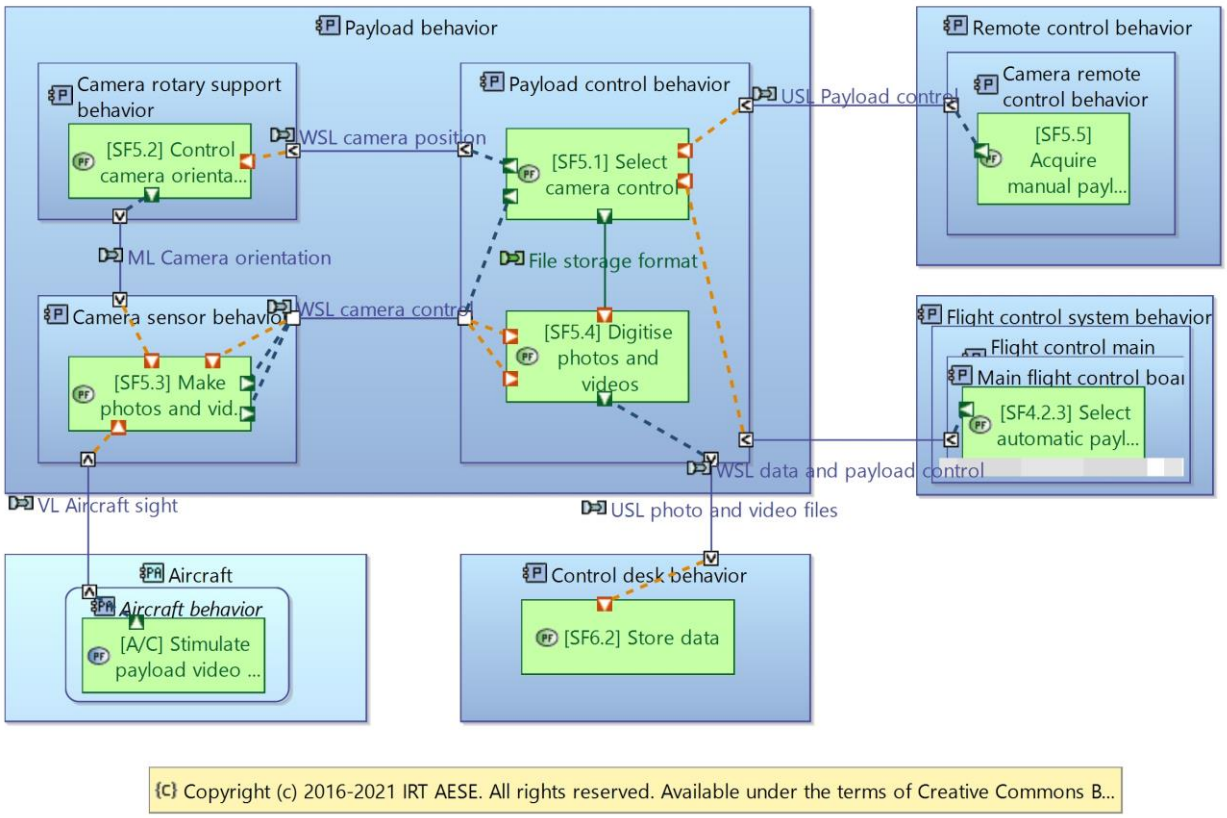


Figure 59 : Payload functional allocation

The deployment of behavior components on node components is represented on the diagram below :

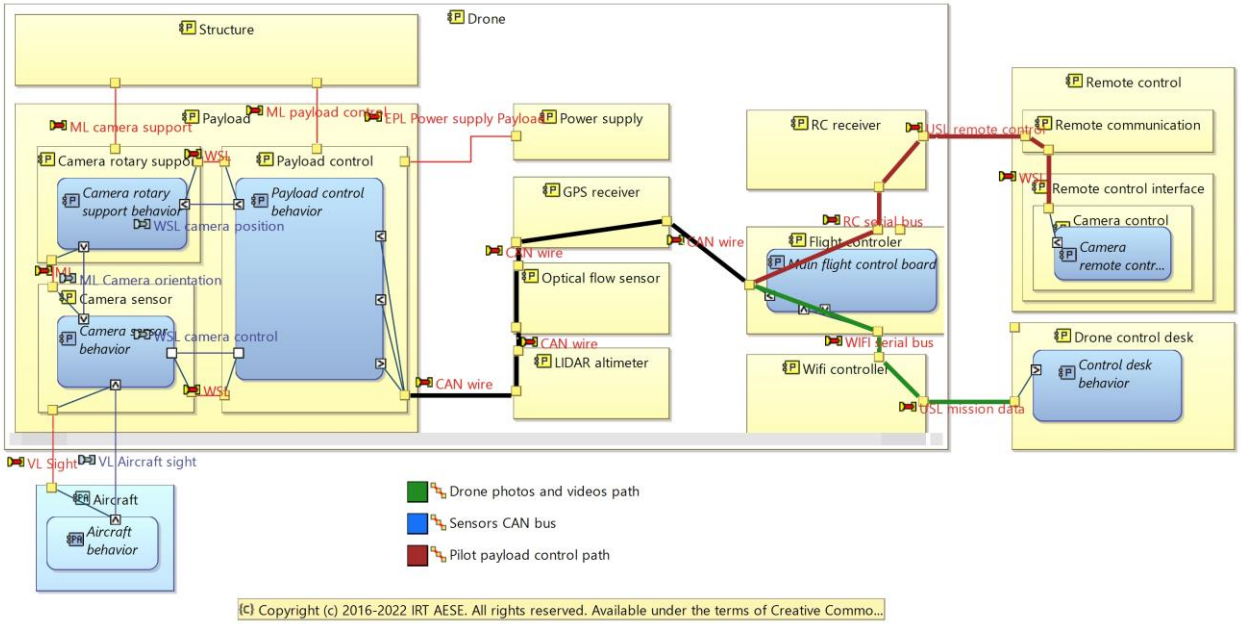


Figure 60 : Payload behavior components deployment

This deployment is straightforward : each behavior component is deployed on a dedicated node component. This choice is quite arbitrary, and represents the possibility for the payload to be composed of several components to be installed in different areas of the drone.

Another possibility would have been to consider the payload as a single node component on which the behavior components are directly deployed. This would represents the payload as a single “monolithic” component.

This diagram shows also the wiring choices :

- Power supply comming directly from the power supply system
- CAN wiring to the Flight controler, which then ensure communication with the remote control and the control desk (see 5.2.2 for details on the communication architecture).

5.2.3.4. Technological choices

The choice of having a single degree of freedom provided by the rotary support has been made for simplicity : this is the minimal configuration that allows adjustment of the elevation, the azimuthal angle corresponding directly to the drone yaw angle.

Another possible solution would be to install the camera on a 3D stabilized gimbal, which allows the pointing of the camera independently from the drone orientation. This would provide also better isolation from drone parasite movement and vibration, but with impact on complexity, weight and cost.

The choice of realizing the “Payload data path” with the main CAN bus also used for sensors could be discussed. It is probably not the most relevant choice regarding the volume of data to be transmitted, and possible interferences with signals from sensors used for the drone control. It could be considered for future version to implement a dedicated data link.

At this stage, no choice have been made regarding the sensor and electronic payload technologies.

5.2.4.Power supply system

Electrical power is the most common way to power drones, and by large. Only some very specific applications, based on standard winged aircraft architecture, may still use fuel powered engines.

This power supply system is not detailed in the current version, it could be a subject for future studies. It appears as a single component that provide DC power to the actuators and embedded electronics.

5.2.4.1. Functional breakdown

So far, no function are allocated to the power supply system.

5.2.4.2. Safety considerations

The failures of the power supply system can lead to the crash of the drone inside the authorized zone, which is an Hazardous event, or in the worst case to the drone fly away out of the the authorized area, if one or several (but not all) the propulsion units are depowered, which is a Catastrophic event. The failure condition corresponding to the last case, allocated to the power supply has been classified as CAT.

A detailed and quantitative safety analysis, performed on a more detailed modelling of the power supply system, could identify new safety related constraints:

- Is the availability of the power supply sufficient with regards to the quantitative objectives associated to an HAZ event ? Is there any redundancy required ?
- What are the constraints on the wiring architecture, so that a single failure leading the loss of power to one or several engines (but not all of them) does not lead to a CAT event ?

5.2.4.3. Physical implementation

In terms of behavior components, the Power Supply system is considered as a single component which provide directly power to all the components that require it :

- The propulsion units, through the motor controllers that modulate the DC power to control the motors
- The flight control system, including both control and monitoring channels with their embedded sensors, and external sensors (LIDAR, GPS and optical flow)
- The payload and its components.

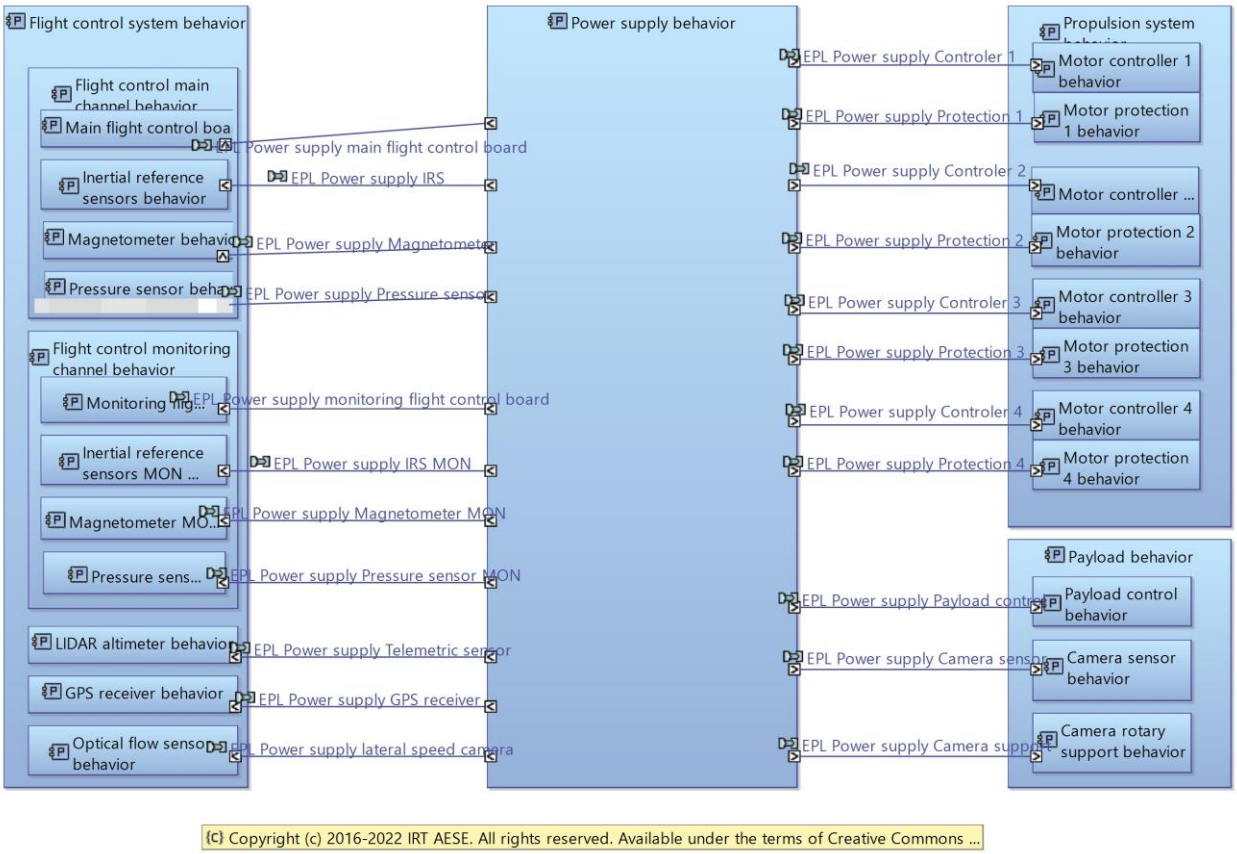


Figure 61 : Power supply behavioral architecture

The diagram below shows the deployment of the Power Supply: as for the behavioral architecture, a single node component appears, on which the behavior component is deployed.

This diagram also shows the power distribution architecture :

- For the flight control system, the power distribution to all the components is ensured by the system itself, i.e. there is no direct power wire from the power supply to the components. This distribution is associated to the serial communication wires between the flight controller and the other devices : the CAN bus for external sensors, the RC and WIFI serial buses for the RC receiver and the WIFI controller.
- For the propulsion system, a point-to-point wiring ensures the power distribution to each MCPS
- For the payload, the payload control ensure the power distribution to the rotary support and the camera sensor.

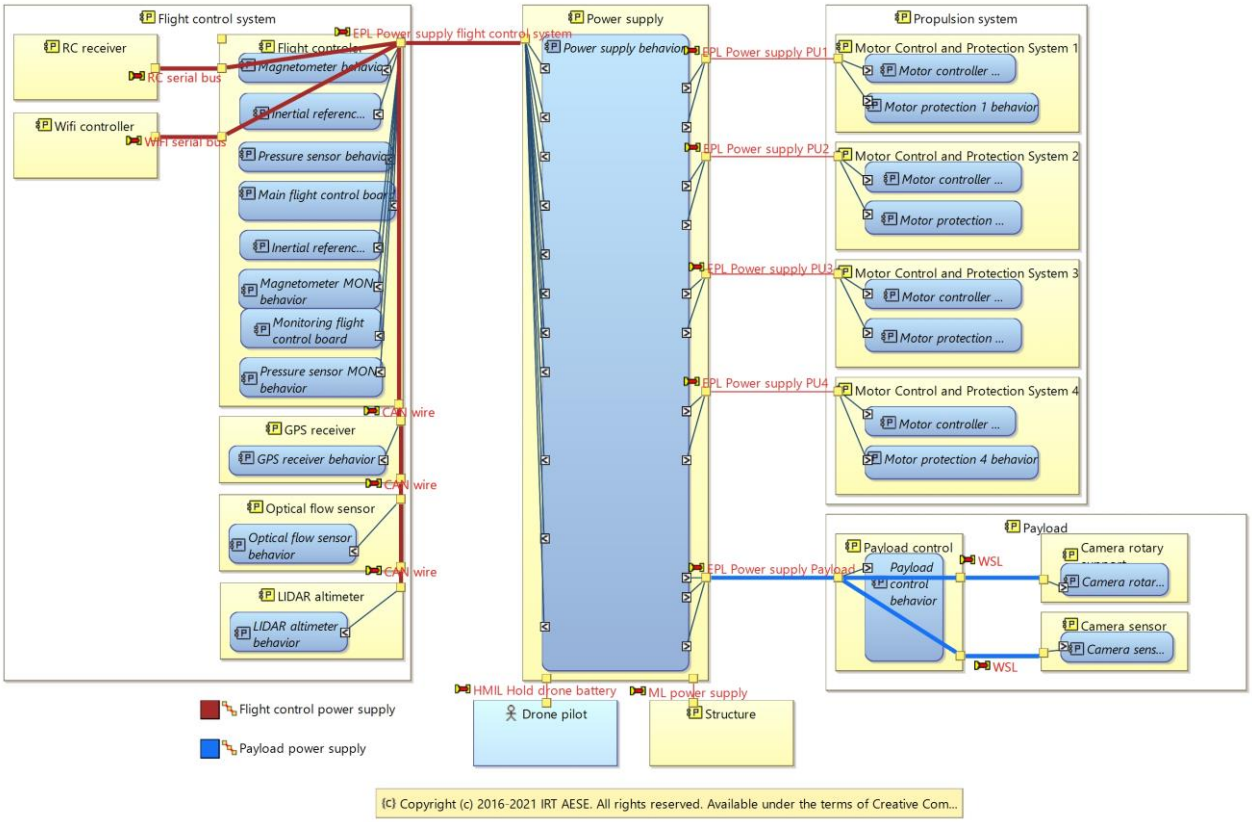


Figure 62 : Power supply behavior components deployment

5.2.4.4. Technological choices

Other than for the power distribution wiring, no technological choice have been made for the power supply. For future studies, different batteries technologies could studies, looking for the best compromise on autonomy, mass, recharge time, cost,… This could be associated to a more global study on the drone performances, autonomy, robustness,…

5.2.5.Structure

As an entire part of the AIDA system, the structure is present in the Capella model. The purpose is to identify the mechanical interfaces with all the other components. By default, and because this is the most common choice for quadri-rotor drones, we assume that the structure takes the form of a cross, with the actuators being installed at each end of the cross, and other components (flight controller, power supply, payload) on the central platform. However, no choice are made regarding the detailed geometry, the materials, etc… of this structure. The picture below shows a conceptual view of the drone structure.

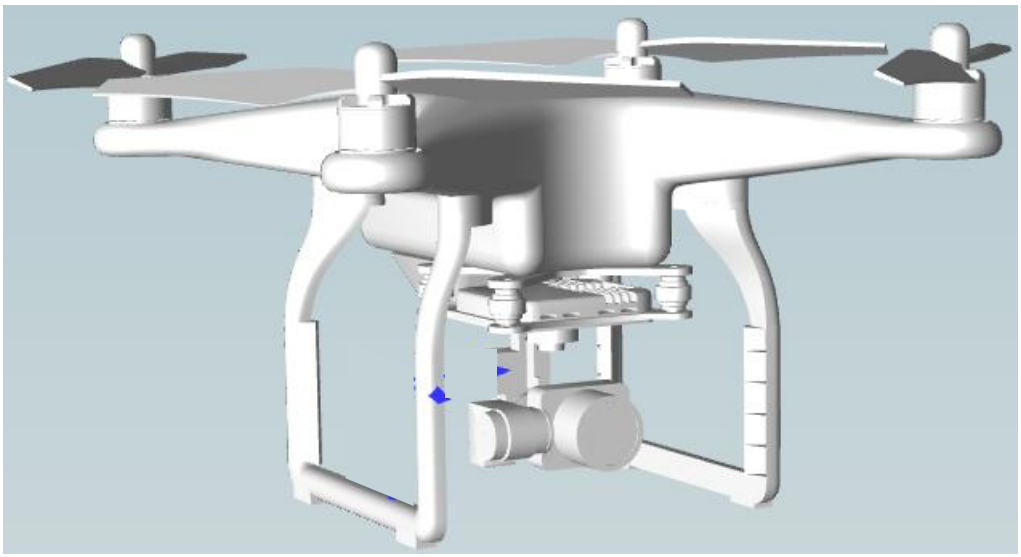


Figure 63 : drone structure conceptual view

5.2.5.1. Functional breakdown

As identified in the logical architecture layer, the function “[LogFun_4_10] Reconstitute global thrust and torque” has been allocated to the structure. In the physical architecture layer, this function is realized by the function “[SF1.5] Reconstitute global thrust and torque”. It represents the load paths from the actuators and the drone center of mass.

As the structure is not decomposed in several parts, this function does not require any breakdown.

5.2.5.2. Safety considerations

In the safety analysis, we consider that there are some failure modes of the structure that can lead to the Catastrophic event. In particular, the rupture of one arm can lead to the loss of control of the drone which can fly out of the authorized zone. A qualitative constraint for the structure would then be that a single failure of the structure should not lead to the loss of one or several actuators. However, the way to actually take into account this constraint on structural and mechanical parts is not explicit in the ARP4754 process. In particular, the Design Assurance Level concept applies only to software and complex hardware electronics.

Several possibilities to comply with this single failure constraints could be implemented :

- The definition of a process similar to the DAL concept, for materials and mechanical parts : depending on the consequences of the failure of the part, more or less constraining design and V&V process can be required, along with usage monitoring and limited life constraints. This way, we could consider that the feared failure mode is improbable enough
- Another possibility could be to design the structure with redundant or failsafe mechanical paths, so that a single failure (resulting for example from the propagation of a crack) cannot lead to the separation of a part of the drone. This can also apply to mechanical interfaces
- A third way could be the implementation of an adequate structure monitoring function which triggers the cut off of the power supply to all engines.

For the current system version, this has not been studied.

5.2.5.3. Physical implementation

The diagram below shows the allocation of SF1.5 to the structure behavior component. Here, only the component exchanges that implements the functional exchanges allocated to SF1.5 are represented.

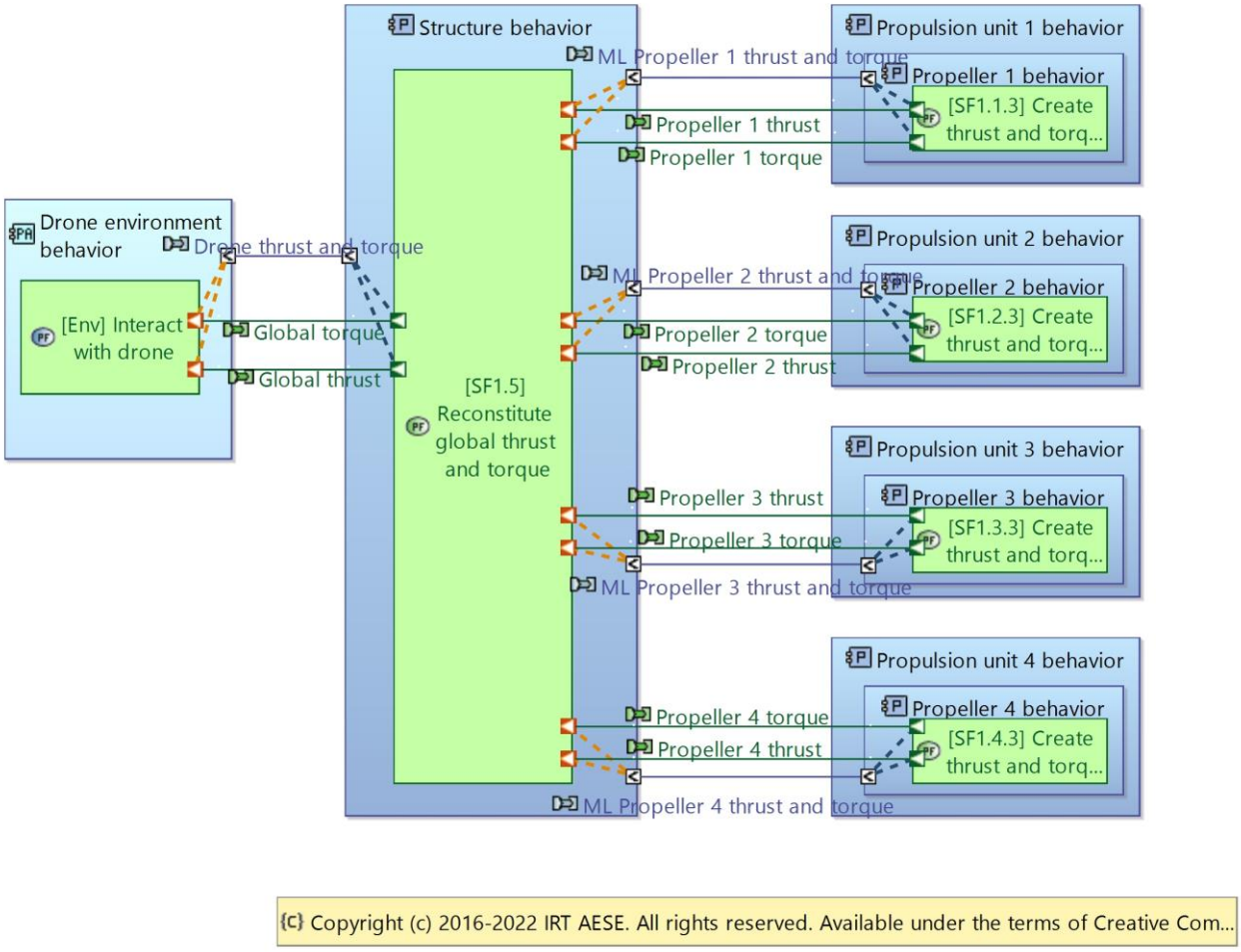


Figure 64 : Structure behavioral architecture

The next diagram show the deployment of the structure behavior component on the associated node component. Here, all the mechanical links with other components are represented, whether they implement a component exchange or not.

The actual load paths between the propellers and the structure, through the motor rotors and stators, are represented in the model with the physical path concept (which is similar to functional chains).

By default, except for the Propulsion system and the payload sensor, we represent direct mechanical links between the structure and all the components. 3D modelling and installation studies would be required to get a more precise view of the actual installation of components around the structure.

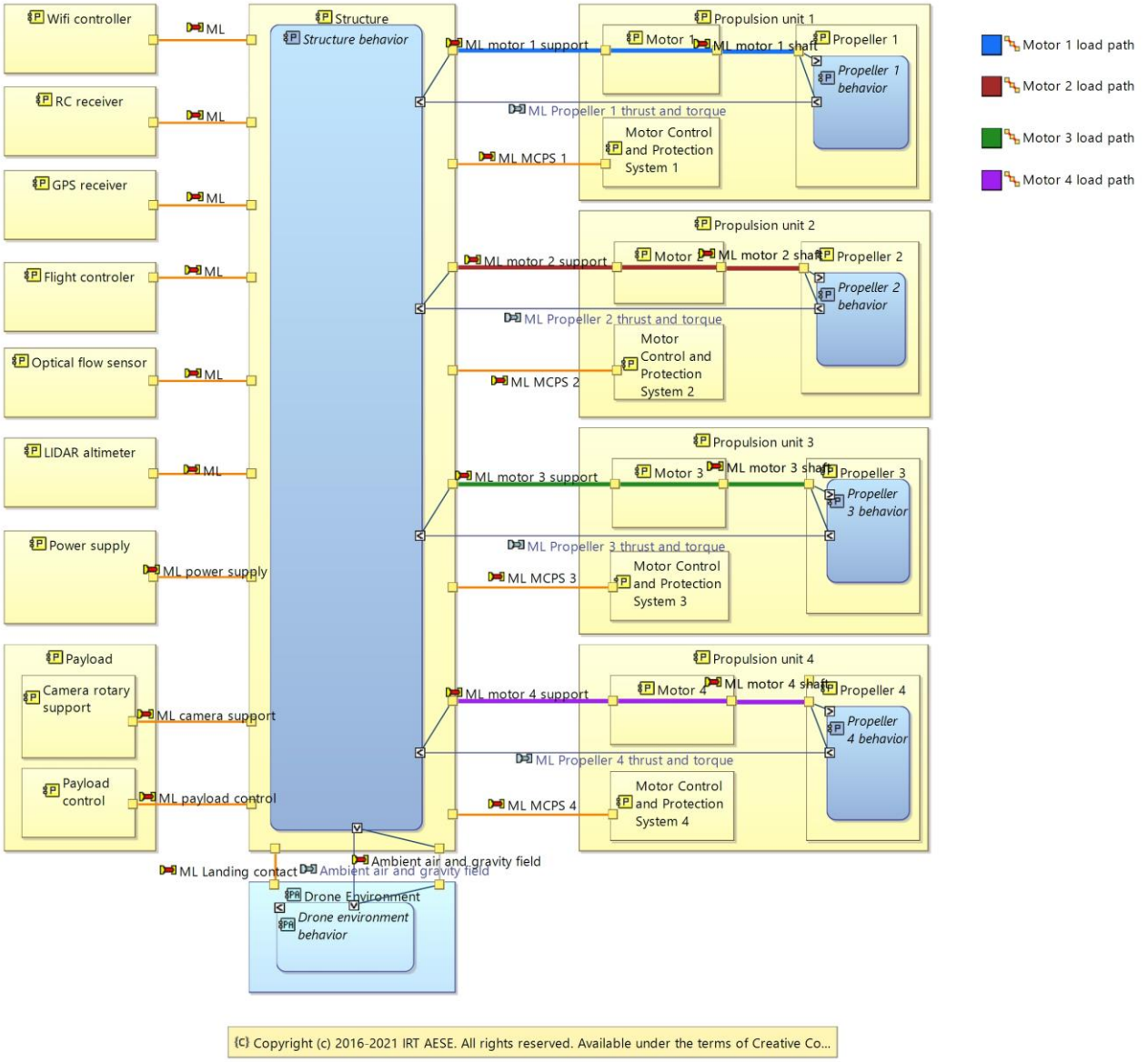


Figure 65 : Structure behavior components deployment

5.2.5.4. Technological choices

As explained earlier, the only choice made for the structure is the conceptual “cross-shaped” geometry, which is standard for quadri-rotor drones. Other geometries could be evaluated, along with material choices, in order to reach a better compromise in terms of mass, manufacturing constraints and cost, robustness, safety constraints, …

5.2.6.Remote control

In the Logical Architecture layer, the remote control provides the commands and feedback that are required for the drone control in real time by the operator. This is not exactly the case in the current version of the Physical Architecture layer, however there are still some similarities on the functions allocated to the remote control.

5.2.6.1. Functional breakdown

The matching between the functions allocated to the remote control in the Logical architecture layer and the Physical architecture layer is established in the table below.

Functions in Logical Architecture layer	Functions in Physical Architecture layer
[LogFun_4_1] Acquire manual motion commands	[SF2.2.1] Acquire pitch order [SF2.2.2] Acquire roll order [SF2.2.3] Acquire yaw order [SF2.2.4] Acquire vertical speed order
[LogFun_4_2] Acquire pilot control mode commands	[SF2.2.5] Acquire manual override order
[LogFun_4_3] Acquire auto-sequence selection	*Allocated to the control desk*
[LogFun_5_1] Acquire manual acquisition command	[SF5.5] Acquire manual payload control
[LogFun_5_3] Display drone point of view	*Not implemented in PA layer*
[LogFun_6_3] Display detected failures	*Allocated to the control desk*

The internal architecture of the remote control is not very detailed in the current version of the model. Therefore, no further functional breakdown is required.

5.2.6.2. Safety considerations

The combination of the following dysfunctions can lead to the Catastrophic failure condition :

- Erroneous manual mode selection « stuck on manual mode »
- Erroneous manual commands acquisition

In this scenario, the control mode is stuck to manual, but the operator cannot control properly the drone which may fly out of the authorized zone. As this failure condition is Catastrophic, no single failure should lead to these dysfunction at the same time.

5.2.6.3. Physical implementation

The previous safety constraint is taken into account as follows : the functions SF2.2.1 to SF2.2.4 on one side, and SF2.2.5 on the other side, are allocated to different components.

Also, in order to respect an usual good practice to separate control functions and payload functions (but which has not be taken as a strong constraint so far), SF5.5 is allocated to a third component.

The diagram below shows the functional allocation on behavior components for the remote control :

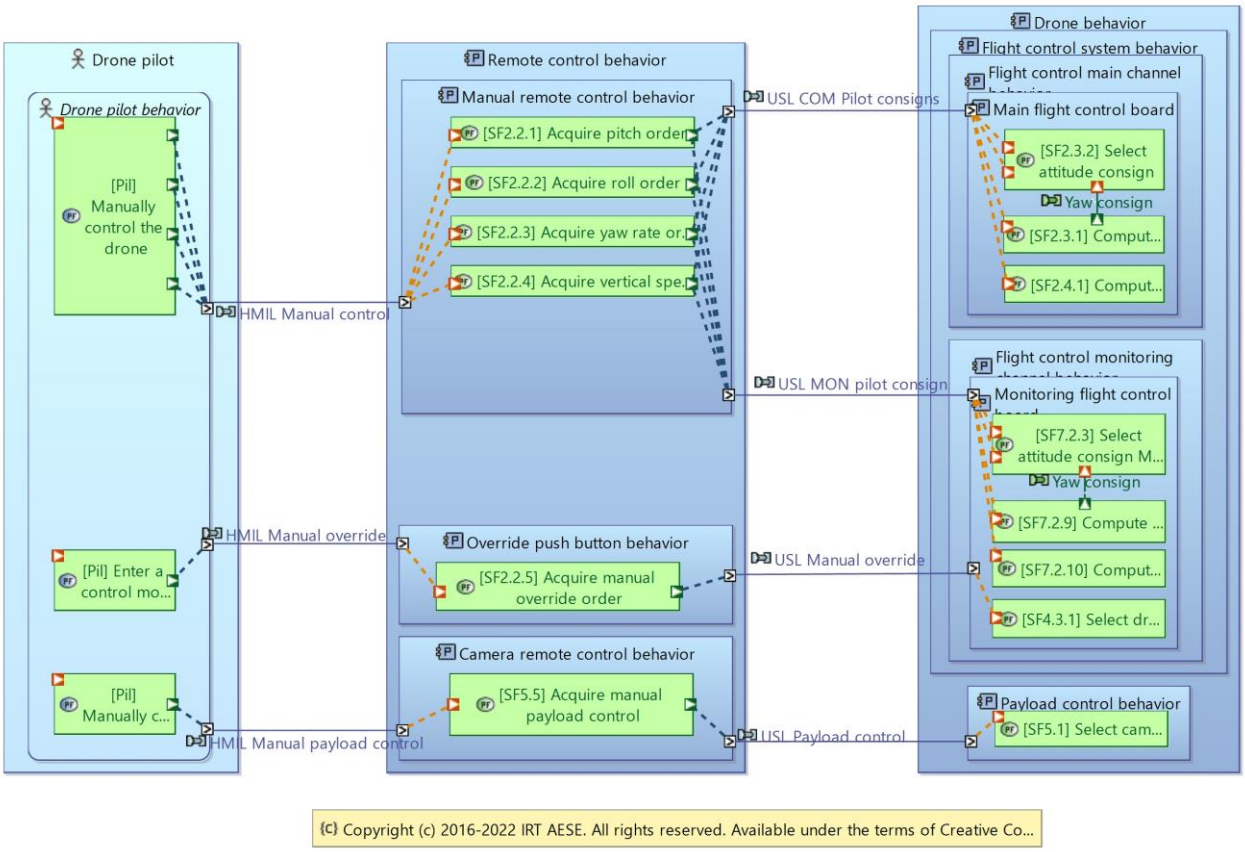


Figure 66 : Remote control behavioral architecture

At this stage, several component exchanges have been defined.

The deployment of behavior components on node components is represented on the diagram below :

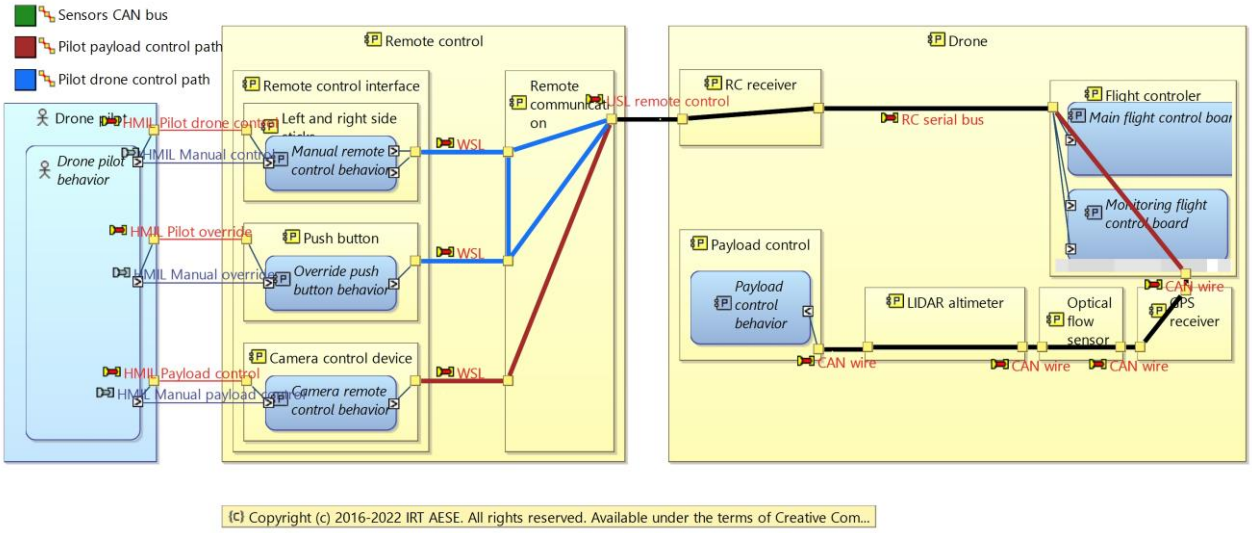


Figure 67 : Remote control behavior components deployment

We can make the following remarks :

- Two main components of the remote control have been defined : the interface, which gathers several sub-components on which the behavior components are deployed, and

the communication component, which ensures communication with the drone through its RC receiver. As for the RC receiver, no function are allocated to the communication component.
- All the components exchanges are allocated to the same physical link between the remote control and the drone. In order to respect the previously stated safety constraints, we make the hypothesis that an undetected erroneous transmission of information on this RC link is not possible, or at least is not the result of a single failure.
- The payload control signal transits from the remote control to the payload through the flight controller and the Sensor CAN bus

5.2.6.4. Technological choices

The main technological choice concerns the transmission link between the remote control and the drone. As stated before, we need a technology that ensure sufficient robustness of this communication, so that undetected erroneous signals are not possible (or not the result of a single failure). Error detection or correction codes can be employed, as well as redundant transmission on separated frequencies. In any case, solutions exist and the hypothesis stated in the previous section is realistic.

Preliminary technological choices have also been made regarding the pilot interface. The solutions presented here are the choice for simplicity :

- Manual control commands are realised with two 2D side sticks, as in common leisure remote control application
- The manual mode selection is ensured with a simple push button
- No particular choice on the camera control device.

Some further studies on the remote control could be performed, in order to propose a more mature design and technological choices. Also, in next system versions, some additional functions will be added to the remote control : the live video feedback display, and an alarm device to alert the pilot. It could be interesting to investigate other interfaces possibilities, such as touch screen, inertial commands, etc…, or the use of standard devices such as smartphones or tablets.

5.2.7.Control desk

In the Logical Architecture layer, the control desk provides the operator interface for all interactions which are not related to the real-time control of the drone during the flight. This is not exactly the case in the Physical Architecture layer, in which the philosophy was a bit different : here, all the commands related to the drone control in automatic mode are provided by the control desk.

This lack of coherence between layers will be resolved for next system versions.

5.2.7.1. Functional breakdown

The matching between the functions allocated to the control desk in the Logical architecture layer and the Physical architecture layer is established in the table below.

Functions in Logical Architecture layer	Functions in Physical Architecture layer
[LogFun_2] Manage mission	[SF6.1] Compute flight plan and flight zone
[LogFun_4_7] Display mission status	[SF6.4] Display drone follow-up data
[LogFun_6_1] Access log of events	*Not implemented in PA layer*
[LogFun_7] Analyze acquired visual information	[SF6.3] Analyze drone measurements
Not represented in LA layer	[SF6.2] Store data
Allocated to remote control	[SF4.4] Acquire pilot control consigns and mode

The internal architecture of the control is not detailed in the current version of the model. Therefore, no further functional breakdown is required.

5.2.7.2. Safety considerations

The control desk is not involved in safety critical scenarios, especially those related to the Catastrophic failure condition. In particular, the mode selection gives priority to the Manual Override signal from the remote control, so that the control desk cannot prevent the operator from selecting the manual control mode (unless several failures occurs at the same time). Therefore, there are no formal architecture constraints for the control desk.

5.2.7.3. Physical implementation

As stated before, the internal architecture of the control desk is not detailed in this system version. This could be the topic for further studies.

The functional allocation to the behavior components, and the behavior components deployment on node components are represented below :

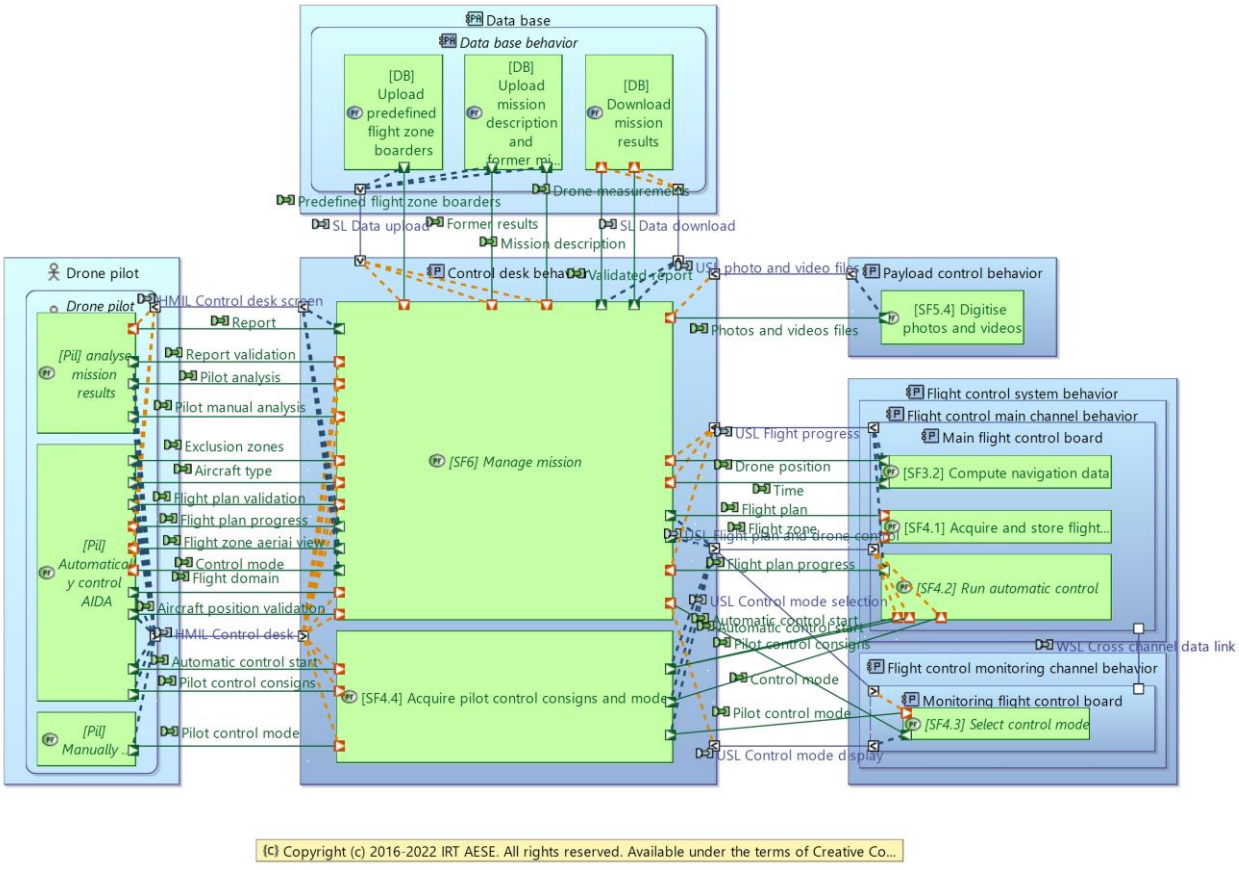


Figure 68 : Control desk behavioral architecture

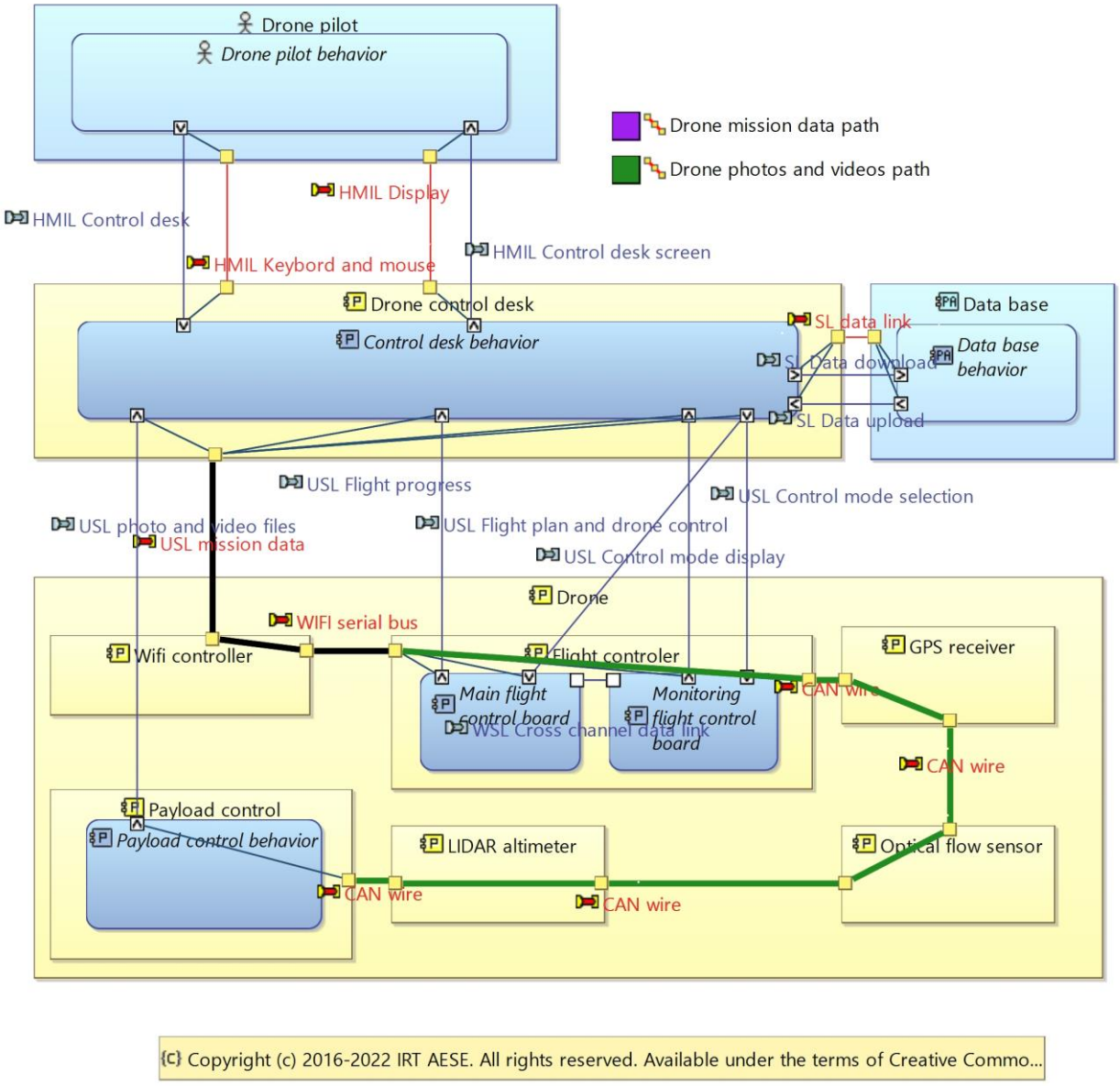


Figure 69 : Control desk behavior components deployment

5.2.7.4. Technological choices

The communication between the control desk and the drone is performed by a Wifi connection. More adequate connection could be considered, regarding the expected communication range and expected bandwidth.

The proposed operator interface is a common PC interface : display on a standard screen, inputs from a keyboard and a mouse. Again, other interfaces technologies could be investigated : touch screen,etc…

No hypothesis are made on the connection with the airline database, which is a standard external network connection and can take several forms depending on the environment and cyber-security contraints:

- Airport Wifi connection, if one is available
- 4G, if Wifi not available
- Satellite connection, for remote airports with no data connection available

5.3. Safety analyses feedback on the proposed architecture

The S2C project provided the opportunity to assess precisely the compliance to safety requirements, using analyses means defined in ARP4761, such as FHAs and PSSAs/SSAs. In particular, a full MBSA model corresponding to the physical architecture has been build, in order to assess the conformity to qualitative and quantitative requirements associated to CAT, HAZ and MAJ Failure conditions. These analyses are available in the aida-safety repository.

Without surprises, the current architecture proved to be highly non compliant to the safety requirements :

- Regarding the qualitative requirements associated to the CAT failure condition, the quadrirotor architecture is not robust to single failures. Indeed, the erroneous behaviour or loss of one motor directly leads to the CAT event. In order to improve the robustness to these single failures, it is necessary to modify the propulsion architecture in order to increase the number of propulsion units, and provide an adequate strategy to detect and mitigate the engines failures. Special care shall be taken regarding other common modes (power supply,…)
- Because of the existence of single failures for the CAT event, and the high number of them in the case of the HAZ event, the quantitative requirements are far from being fulfilled. Additional redundancies would probably be required in the Flight control system and remote control, along with a revision of the current strategy in which the propulsion is completely shut down as soon as single failures are detected.

6. Perspectives

This section closes the document (the EPBS layer of the Arcadia method being unused for AIDA). We identify here the possible next steps and further perspectives of studies around the AIDA study case.

6.1. Remaining inconsistencies

As explained in 1.4, there are remaining inconsistencies between this Physical Architecture layer and the previous layers (Operational and System Analysis, Logical Architecture). The resolution of these inconsistencies will be one main topic for the next system versions.

In the meantime, these main inconsistencies are summed up here :

- The control mode management is a bit different : in the Logical Architecture layer, an additional mode (Emergency Landing) is proposed, and the transitions between modes are more complete. In general, the modes and states machines could be improved, the use of simulation being a great help to validate the correctness of those machines
- The philosophy of functions allocation between the remote control and the control desk is different :
    - In the Logical layer, commands related to real-time drone control and monitoring are implemented on the remote control, so that the control desk is not needed during the drone flight. The control desk is used before the drone mission (for mission configuration) and after (for results analysis).
    - In the Physical layer, commands related to the manual mode are implemented on the remote control and commands related to the auto mode are implemented on the control desk
- Additional feedback on the remote control is proposed in the Logical layer, and not implemented in the Physical layer : live video feedback display, and alarms display

6.2. Candidates topics for next system versions

Besides these inconsistencies to be resolved, some topics are already identified as candidates for the next system versions :

- The propulsion system will evolve so that motor monitoring is performed entirely by the MCPS. The CAN motor bus and PWM signal scan then be replaced by full digital communication between the MCPS and the Flight controller.
- The MCPS will be able to provide feedback on the motors health state to the Thrust allocation function, so that this allocation can be adapted when a motor is failed. And probably, extra motors will be added to offer sufficient controlability in case of Motor failures.
- Some functions are mentionned in the system analysis phases, but are only implicit when the architecture is detailed. In particular, the Obstacle Management principles, flight zone protection and Emergency landing device could be developped more explicitely.

6.3. Opportunies for further details and domain specific studies

Until now, the model has mainly put the focus on the drone architecture, and safety related features. Some sub-systems would deserve further detailing :

- The power supply system appears only as a « black box », without any function allocated
- The remote control is basically detailed, only to make appear the safety constraints. A good way to improve its modelling could be to consider it as an entire system with its own model, and formalize properly the operational and system analysis with focus on the remote control only.
- This is also the case for the control desk which is not detailed at all

It could also be completed by other domain specific studies, such as simulation and performances, 3D and mechanical sizing, hardware/software detailed implementation…. Another relevant study to improve the realism of the study case could be the identification of standard off-the-shelf components for actuators, sensors, batteries….

6.4. Other possibles architecture concepts

Finally, the architecture presented here is only one possible solution to answer to the needs expressed for an enhanced Pre-Flight Check and aircraft inspection. In particular, the choice of a common quadri-rotor drone, although integrating safety constraints related to the operational context, could be discussed and other architecture investigated in architecture trades :

- Other numbers and/or combinations of rotors, including additional degree of freedom
- Decoupling the sustentation and displacement functions, using for example a light gas balloon to ensure the sustentation
- Using a more relevant device for inspection below the aircraft, moving on the ground instead of flying, with possible autonomous collaboration with the drone

END OF DOCUMENT