A Holistic Approach to Sustainable, Digital EU Agriculture, Forestry, Livestock and Rural Development based on Reconfigurable Aerial Enablers and Edge Artificial Intelligence-on-Demand Systems

CHAMELEON D.2.2. Technical Requirements and System Architecture v1

Work package	WP2: Conceptualization, use cases and system architecture
Task	Task 2.4: CHAMELEON Architecture and project outcomes
Authors	**Adrestia R&D**
Dissemination level	Public (PU)
Status	Final
Due Date	01/06/2023
Document date	13/06/2024
Version number	2.0
Funded by the European Union	Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Executive Agency. Neither the European Union nor the European Research Executive Agency can be held responsible for them.

Revision and history chart

Version	Date	Main author	Summary of changes
0.1	01/05/2023	ADR	Draft chapter 4
0.2	02/05/2023	ADR	Draft chapter 3
0.3	05/05/2023	ADR	Draft chapter 5
0.4	07/05/2023	ADR	Draft chapter 6
0.5	12/05/2023	ADR	Draft chapter 2
0.6	14/05/2023	ADR	Draft chapter 1
0.7	15/05/2023	ADR	Draft chapter 7
0.8	20/05/2023	ADR	Review partners' contribution
0.9	25/05/2023	ADR	Review partners' contribution
1.0	26/05/2023	ADR	Final submitted version
1.1	29/05/2023	Unparallel Innovation	Revisions after submission
1.2	31/05/2023	ADR	Final version after PC and QA review
2.0	31/05/2023	ACCELI	Revised submitted version (if required)
2.1	07/06/2024	ADR	Revised version based on reviewers' comments

Tables of Contents

Index of figures

Index of Tables

LIST OF ABBREVIATIONS AND ACRONYMS

Abbreviation	Meaning
ADSS	Agricultural Decision Support System
AI	Artificial Intelligence
API	Application Programming Interface
DEM	Digital Elevation Model
DFDPro	Delair Flight Deck Pro
DIP	Drone Innovation Platform
DSM	Digital Surface Model
DTED	Digital Terrain Elevation Data
EO/IR	Electro-Optical/Infra-red
FTS	Flight Termination System
GCC	Ground Canopy Cover
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
HD	High Definition
HMI	Human Machine Interface
KML	Keyhole Markup Language
PPC	Plug and Play Client
PPC	Plug and Play Client
REST	REpresentational State Transfer
RGB	Red, Green, Blue colour model
RPAs	Remote Piloted Aircraft systems
UAVs	Unmanned Air Vehicles
UI	User Interface
Wi-Fi	Wireless Fidelity
YOLO	You Only Look Once algorithm

1 EXECUTIVE SUMMARY

This document presents the first version of the CHAMELEON Architecture and its components. Initially, the conceptual view of the architecture, namely the main systems, activities, and stakeholders are outlined. Then, the functional and implementation views of the components that comprise the CHAMELEON System are described, providing component and sequence diagrams that illustrate the functional and technical specifications of each component. The Deployment View section presents the various software, hardware and network requirements needed to deploy the system on the pilots. Finally, the Use Case Scenarios are presented, to better justify the decisions taken when designing the architecture.

2 INTRODUCTION

In this deliverable, the architectural foundations of CHAMELEON are presented. CHAMELEON aims on an innovative software solution designed to deploy versatile software bundles on various Unmanned Aerial Vehicles (UAVs) to address a wide range of needs and industries. The outputs of task T2.1 “Stakeholders Use Cases, Requirements And Workshop”, and eventually the content of deliverable D2.1 “CHAMELEON conceptualisation, and use cases definition”, were considered in order to design an architecture based on the needs and requirements of the various stakeholders. Moreover, to ensure a comprehensive and well-rounded architecture, the widely recognized viewpoints, and perspectives from "Software Systems Architecture" by Rozanski and Woods [3], are adopted, emphasizing the Conceptual, Functional, Implementation, and Deployment views, which are described below. Additionally, the Scenarios views based on the "4+1" View Model of Software Architecture proposed by Philippe Kruchten [2] are incorporated. These views act as an enabler to analyse the system from multiple angles and provide a holistic understanding of CHAMELEON's architecture.

The Conceptual view provides an abstract representation of the system's high-level components and their interrelationships. The Functional view illustrates the system's functionalities and how they are organized to achieve the desired outcomes. The Implementation view focuses on the software's internal structure and the technologies employed to realize the system. The Deployment view presents the physical distribution and configuration of the software bundles across the UAVs. Lastly, the relevant scenarios that illustrate the practical usage of CHAMELEON are examined and the benefits it brings to different stakeholders are also showcased. By exploring these scenarios, it is demonstrated how CHAMELEON addresses the unique challenges and requirements of various industries while providing flexibility and adaptability in the ever-evolving landscape of UAV applications.

The purpose of this document is to introduce CHAMELEON’s conceptual, functional, development, deployment, and use case architectural viewpoints (Scenarios) in their initial form. It should be noted that the CHAMELEON architecture will be improved upon as work on the project advances, and the final architectural views will be presented in deliverable D2.5 CHAMELEON technical requirement and system architecture v3.

The document is organized as follows:

- Section 3 presents the conceptual architecture of the system.
- Section 4 presents the functional view of the various components.
- Section 5 describes the implementation view of the proposed system.
- Section 6 presents the deployment view of the CHAMELEON architecture.
- Section 7 concludes this document.

3 CHAMELEON PLATFORM TECHNICAL REQUIREMENTS

This section will elaborate upon the functional and non-functional requirements concerning the CHAMELEON Platform architecture. Components for this architecture were designed based on the needs of the stakeholders. This input is derived from the T2.1 "Stakeholders Use Cases, Requirements And Workshop" and D2.1 "CHAMELEON conceptualisation, and use cases definition". D2.1 also produced functional and non-functional requirements regarding the UAV bundle development. Those were also considered as part of this document's requirements. Finally, each stakeholder's role is further explained on the next section.

This initial set of architecture requirements is aligned with the end-user needs and throughout this document references to those requirements are made (by their ID) so that each architectural decision is justified accordingly.

Table 1 presents the created requirements. It contains the requirement number and description, as well as the rationale that resulted in the requirement.

Table 1: CHAMELEON Platform Architecture Functional and Non-Functional Requirements

RqID	Rq Type	Description	Rational
F01	Functional	The platform should support live video streaming to the end-user that handles the UAV	Live video streaming is identified as a need
F02	Functional	The platform should be able to redistribute the captured data onto another server for further post-processing/analysing.	Recording of the video and further post-processing off the UAV is also identified as a need
F03	Functional	The platform should support a network topology from which the UAV is connected and directly controlled	UAVs should be able to connect to a local (to the end-user) network
F04	Functional	The platform should have a User Interface where the end-user can view the results of the UAV's captures	UAVs will map a wide range of open outdoor spaces (with photos).
F05	Functional	The platform should provide results showcase User Interface.	Users want to view the results of the UAV's processing functions (or preview, in case of further pre-processing that needs to be done on another server).
NF01	Non-Functional	The platform's architecture should be designed in such a way to provide seamless connection between its	Users that lack the technological expertise requested that the

RqID	Rq Type	Description	Rational
		components and easy to manage data distribution between them	platform should be easy to use and easy to deploy their bundles on
NF02	Non-Functional	The platform should be implemented with hardware that supports BVLOS	Users identified that for the use-case pilots a flight where the UAVs are not visible by the naked eye will be performed (Beyond Visual Line of Sight - BVLOS).
NF03	Non-Functional	The platform should be designed in a way to support multiple language bundle implementations.	End-users might develop their bundles in different programming languages

4 CONCEPTUAL VIEW

In this chapter of the deliverable, the stakeholders, the main systems as well as the main activities of the project are being defined to provide context on the overall CHAMELEON system architecture.

4.1 STAKEHOLDERS

In this chapter, the key stakeholders of the project are categorised and presented.

4.1.1 DRONE OWNERS

Drone owners provide the Unmanned Air Vehicles (UAVs) that are being used in the project to improve the safety, efficiency, and accuracy of livestock, agriculture and forestry monitoring and management procedures.

4.1.2 DEVELOPERS

Developers construct the logic that will be implemented on the reconfigurable drones of CHAMELEON. The main interaction point between the CHAMELEON ecosystem and the developers is the Cloud infrastructure's User Interface through which they can upload their developed bundles.

4.1.3 LIVESTOCK ASSOCIATIONS

Livestock associations are end users of the CHAMELEON and main beneficiary of the bundles that regard livestock monitoring and management operations.

4.1.4 FOREST OWNERS

Similar to livestock associations, forest owners belong to the end user group of the CHAMELEON platform and are to be benefited from the overall work done under the scope of the project.

4.1.5 VINEYARD OWNERS

Vineyard owners are another end-user and beneficiary of the CHAMELEON platform. Bundles focusing on vineyard monitoring and management will be utilised by vineyard owners.

4.1.6 AUTHORITIES

Authorities include fire fighters, municipalities, and civil aviation and provide the law regulations to ensure lawful CHAMELEON activities.

4.2 MAIN SYSTEMS

Under the main systems chapter, key systems that comprise the overall system architecture are documented. This initial system architecture is created considering the CHAMELEON Platform end-user needs, and the main architecture requirements presented in Table 1.

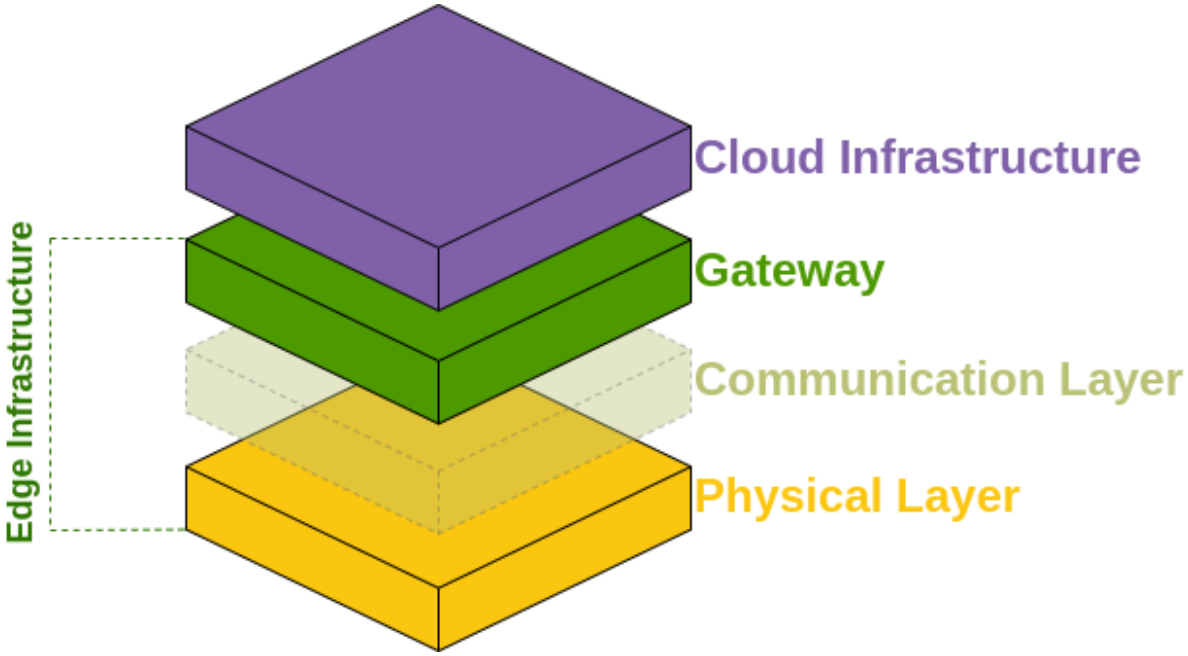


Figure 1: CHAMELEON Layered Architecture

4.2.1 CHAMELEON CLOUD

The cloud infrastructure hosts the bundles developed under the scope of CHAMELEON, the User Interfaces responsible for interaction with the end users, and the necessary functionality to send and receive information with the rest of the CHAMELEON ecosystem.

4.2.2 CHAMELEON EDGE

Under the CHAMELEON Edge infrastructure belong all the subsystems that need to be deployed in-place of the CHAMELEON Pilots. These systems include:

CHAMELEON GATEWAY

The CHAMELEON Gateway will serve as the middleware between the Cloud infrastructure and the rest of the CHAMELEON ecosystem (Communication layer, Physical layer) and contains the required functionality to make the communication between Edge and Cloud possible. The CHAMELEON Gateway is designed based requirements Req-F02 and Req-NF01.

COMMUNICATION LAYER

The Communication layer of CHAMELEON regards the necessary software and hardware means to transfer data between the Gateway and the Physical layer. This layer is designed based on requirements Req-F01, Req-F02, Req-NF01, and Req-NF02.

4.2.3 PHYSICAL LAYER

The Physical layer of CHAMELEON is comprised from the UAVs and Ground Sensors:

- The UAVs of CHAMELEON, equipped with the necessary hardware sensors (e.g., Lidar, RGB cameras and thermal cameras), will perform flights over areas of interest gathering data to be processed by the CHAMELEON bundles.
- The Ground sensors of CHAMELEON will complement the data gathering conducted by the UAVs, in an attempt to aid the accuracy and efficiency of data collection.

Physical layer is designed based on requirements Req-F03, Req-F04, Req-NF01, and Req-NF02.

4.3 MAIN ACTIVITIES

In this chapter, the main activities of the project are being documented. These activities are based on the stakeholders needs, as D2.1 "CHAMELEON conceptualisation, and use cases definition" has documented. Activities being conducted by key stakeholders of CHAMELEON will ultimately assess the effectiveness of the overall proposed CHAMELEON architecture.

Since those activities are created with the end-user in mind, they are mostly designed around the bundle handling and processing, and they are based on requirements Req-NF01 and Req-NF03.

4.3.1 BUNDLE UPLOAD

The process of uploading a developed bundle in the Cloud infrastructure of the project will follow specific guidelines to ensure the availability and its overall successful utilisation from the CHAMELEON ecosystem. Such guidelines include both authentication and authorisation procedures as well as specific configuration to enforce universality.

4.3.2 BUNDLE DEPLOYMENT

After the selection of one of the CHAMELEON bundles, the stakeholder can instruct its deployment through the User Interface provided from CHAMELEON Cloud. This deployment regards the dissemination of the bundle from the Cloud infrastructure to the Physical layer of the CHAMELEON. Specifically, the on-board system in the UAV, where the bundle can apply its constructed logic on the data received through the UAVs hardware adapter (Lidar, RGB Camera etc.).

4.3.3 ON-BOARD PROCESSING

Once a bundle has been deployed on-board the CHAMELEON UAV, it can initiate data processing operations. Specifically, by having defined the storage endpoint where the data streams will be saved UAV internally, the running Bundle can have access, process, and produce the computed results.

4.3.4 POST-PROCESSING

Some CHAMELEON Bundles may have heavier resource requirements as well as require multiple hours, even days of continuous processing. UAVs are limited in terms of computational power resources as well as battery life. Thus, such heavy computations will be conducted in the Cloud infrastructure of CHAMELEON.

5 FUNCTIONAL VIEW

This chapter presents an overview of the functional elements of the CHAMELEON system, including their respective responsibilities and interactions. As noted by Rozanski and Woods [1], the functional structure model of the Functional View typically consists of functional components, interfaces, and external entities/connectors. The CHAMELEON Platform consists of three main layers, the CHAMELEON Cloud, the CHAMELEON Edge, and the CHAMELEON Physical Layer. The CHAMELEON Cloud refers to all components running on the Cloud infrastructure of CHAMELEON. The CHAMELEON Edge is the middleware between the Cloud and the Physical Layer, namely the Edge Gateway and the Human Machine Interfaces (HMIs). The CHAMELEON Physical Layer consists of UAVs and ground sensors.

5.1 CHAMELEON CLOUD

This chapter presents the functional view of the components running on the CHAMELEON Cloud. The components described here are the following:

- **Bundles:** The bundles are services that will run on drones and are hosted on the cloud.
- **Drone Innovation Platform:** The main interaction point and interface for the end-users (technical and non-technical).
- **CHAMELEON Store:** The store will host the bundles and services to be used by the stakeholders. It will also provide filtering and matchmaking capabilities to assist in the bundle selection process.
- **Plug and Play Server:** The Plug and Play Server will be responsible for deploying the bundles on the UAVs through the CHAMELEON Edge Gateway.
- **Agricultural Decision Support System:** The Agricultural Decision Support System (ADSS) will assist the stakeholders in decision making by providing visualization and alerts.

5.1.1 CHAMELEON BUNDLES

Bundles are algorithms that will run on UAVs, designed for specific use cases. There are three main categories of bundles, for agriculture and rural areas, livestock monitoring and forestry. Each category will be used in different pilot cases of the project. Stakeholders will be able to select the bundle of their choice to satisfy their needs. The bundles' context is decided and refined upon the interviews and workshops that were held under D2.1 "CHAMELEON conceptualisation, and use cases definition".

FORESTRY BUNDLES

The forestry bundles will be responsible for characterizing the vegetation, with three fundamental objectives:

1. Quantify, classify, and monitor the health status of the existing species.
2. Prevent the accumulation of wood debris in rivers and mountain slopes.
3. Prevent and minimize the impact of fires by identifying areas at highest risk and developing plans for their early extinction. Of particular interest is the urban-forest interface, due to its proximity to the population.

As an example, Figure 2 shows the main functionalities of the module for preventing the accumulation of woody debris in rivers.

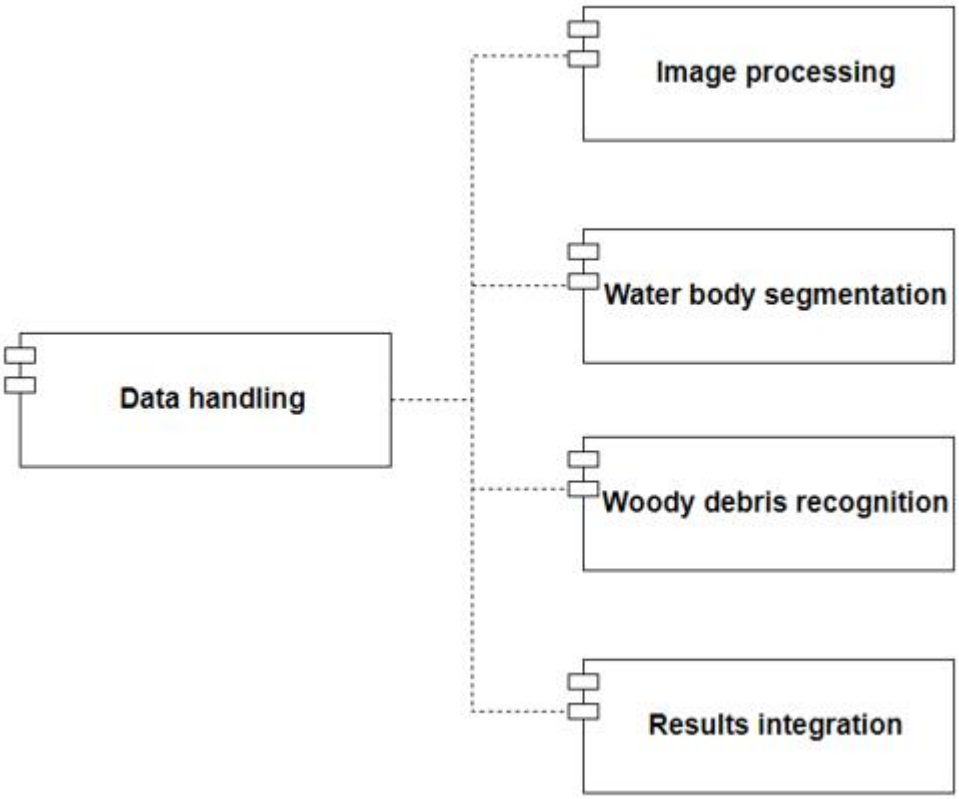


Figure 2: Wood debris functional view

Main functionalities of the Forestry bundles include:

- **Data Handling:** Data handling entails the management of the incoming video feed procured from the CHAMELEON UAV's RGB camera, and the outgoing data stream of woody debris detections. This process involves interfacing with the CHAMELEON UAV's storage system for local data redundancy, ensuring that the data is not lost or corrupted during transmission. The data is then sent to a cloud-based server where the processing takes place. This configuration enhances the robustness and reliability of the data handling procedure.
- **Image Processing:** The image processing stage makes use of advanced computational capabilities to process the video feed from the drone, frame by frame. This process involves several steps such as image normalization, noise reduction, and other pre-processing tasks that prepare the video frames for efficient water body segmentation and woody debris recognition. These procedures help to enhance the quality of the images and remove any potential distortions or anomalies that could interfere with the subsequent steps.
- **Water Body Segmentation:** Water body segmentation is carried out using a convolutional neural network model such as U-NET. This process involves separating

the water bodies from the rest of the image content, providing a clear and focused view of the rivers under observation. The U-NET model is specifically trained to segment the regions of interest, in this case, the water bodies, thereby enabling more accurate and efficient detection of woody debris.

- **Woody Debris Recognition:** Once the water bodies have been segmented, woody debris recognition is performed using a pre-trained deep learning model like YOLO. This model identifies and tracks wood debris within each frame of the video feed. By applying the segmented water body as a mask over the frame, the model is restricted to identifying woody debris within the water body only, thereby minimizing the chances of false-positive detections outside of the area of interest.
- **Results Integration:** The final step, results integration, involves collating the results from the water body segmentation and woody debris recognition stages and integrating them into a user-friendly format. This could include visual aids such as maps or charts that highlight the areas where woody debris was identified. The aim is to present the results in a way that is easily understandable by non-expert end users. The processed information is then transmitted to the end user, possibly through a report that provides a comprehensive overview of the woody debris detection in the surveyed rivers.

LIVESTOCK MONITORING BUNDLES

In the progressive field of agriculture, the enhancement of livestock management is significantly brought about by 'livestock bundles'. These integrated solutions, comprised of advanced hardware and software components, are built upon deep learning and object detection algorithms. The function they serve is the delivery of critical, real-time data about the positions and activities of a livestock herd. The task of livestock management, often marked by its complexity, is considerably simplified through the integration of such technology, making it easier to monitor and ensure animal health and productivity. A variety of these bundles are included in our offerings, each thoughtfully designed to address specific needs within the farming sector. To provide a tangible understanding of these advanced tools, a detailed discussion on a specific product from our extensive range, the 'Animal Detector Bundle', will be presented in the following section.

The Animal detector is a pivotal module within the CHAMELEON ecosystem, designed to effectively identify and locate animals in their natural habitats or when they have been lost through real-time object detection from drone videos. The primary objective of this module is to facilitate animal monitoring and tracking, contributing to conservation efforts and ecological research. It operates on NVIDIA Jetson platforms[1], which provide AI capabilities on the drone itself, enabling near real-time insights and location tagging.

Bundle's Main Functionalities:

- **Data Handling:** This function involves managing the incoming video feed from the drone's RGB camera and the outgoing data stream of animal detections. It includes

[1] https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/

interfacing with the UAV storage system for local data redundancy and utilizing the Plug and Play Client for data transmission to the CHAMELEON ecosystem.

- **Image Processing:** Using the high computational capabilities of the Jetson module, the real-time video feed is processed frame by frame. This includes tasks such as image normalization, noise reduction, and other pre-processing steps that prepare the video frames for efficient object detection.
- **Animal Detection:** With the pre-processed video frames, the animal detection bundle (which employs pre-trained deep learning models like YOLO [1]) identifies animals within each frame. It then has the potential to combine these identifications with GPS data from the drone to locate and track the animals.
- **Detection Validation:** This function involves verifying the accuracy of animal detections. A feedback loop reviews each detection and applies further checks or models (like a secondary verification model) to minimize false positives and ensure reliable detection results.
- **Alerts and Notifications:** This function could involve generating real-time alerts or notifications when animals are detected. These alerts could be crucial for rapid response in certain conservation scenarios.
- **Data Analytics and Visualization:** Post-processing the collected data can provide valuable insights. This could include tracking the movement patterns of detected animals, heat maps, or visualizing detection accuracy over time.

Figure 3 shows the functional elements (modules) of the animal detection bundle.

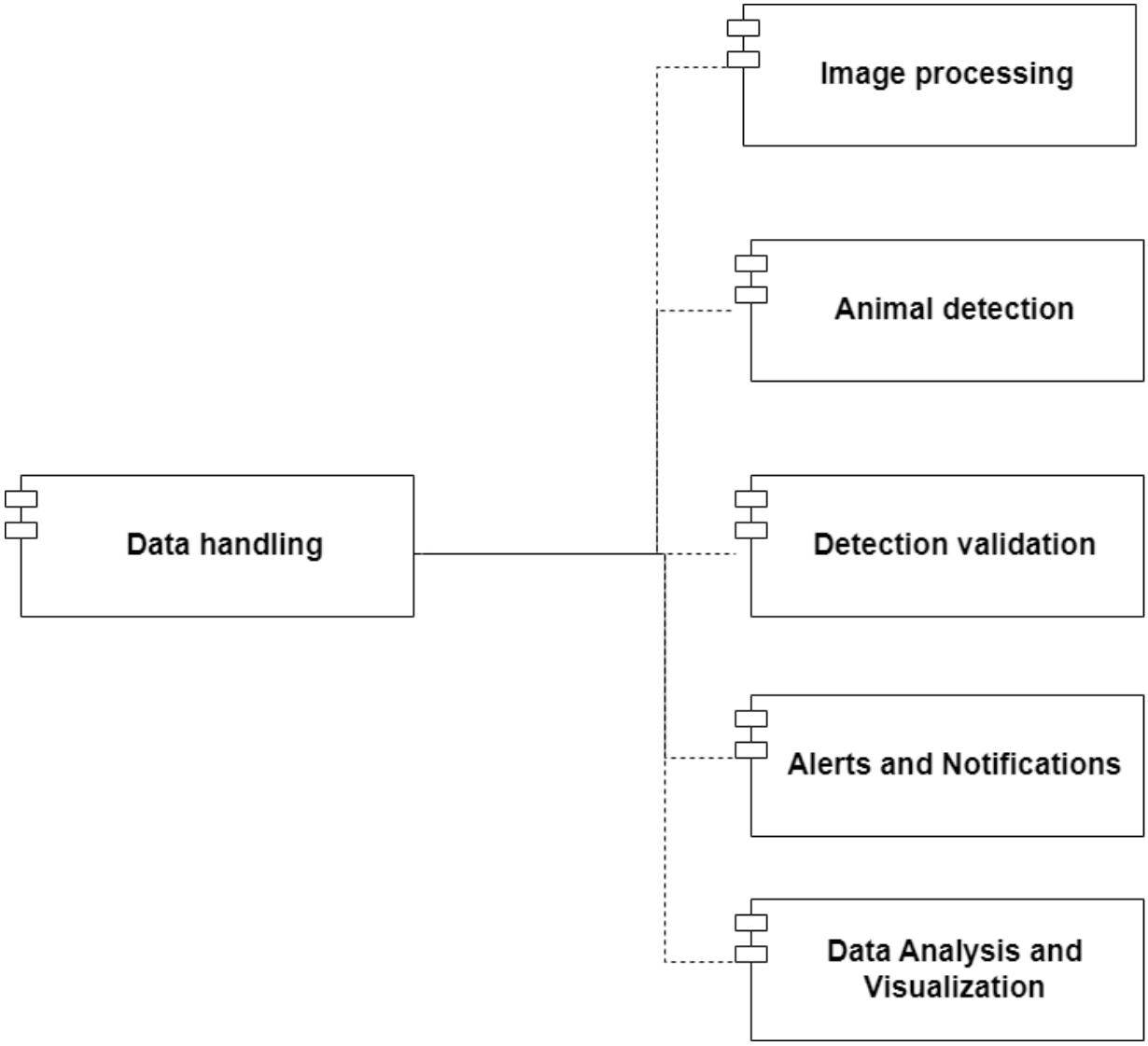


Figure 3: Animal detection functional view

AGRICULTURE AND RURAL AREAS BUNDLES

The agriculture and rural areas bundles will be responsible for crop and vegetation monitoring, primarily focused on vine crop, with three fundamental objectives:

- Crop growth and development monitoring, which is a key bundle to evaluate the evolution of the crop along the crop season and detect any abnormality in this evolution due to different biotic or abiotic factors. It integrates health and pest detection and water stress due to drought. Also, with this bundle, the status of flora at high altitude will be monitored to manage herbs.
- Evaluation of the effect of extreme events on crops, such as storms or damage caused by wild animals.
- Soil zoning, with different strategies that allow, after plot zoning, to implement precision agriculture techniques, such as precise irrigation, fertilization, and pest and diseases treatments.

As an example, Figure 4 shows the main functionalities of the module for crop growth and development monitoring.

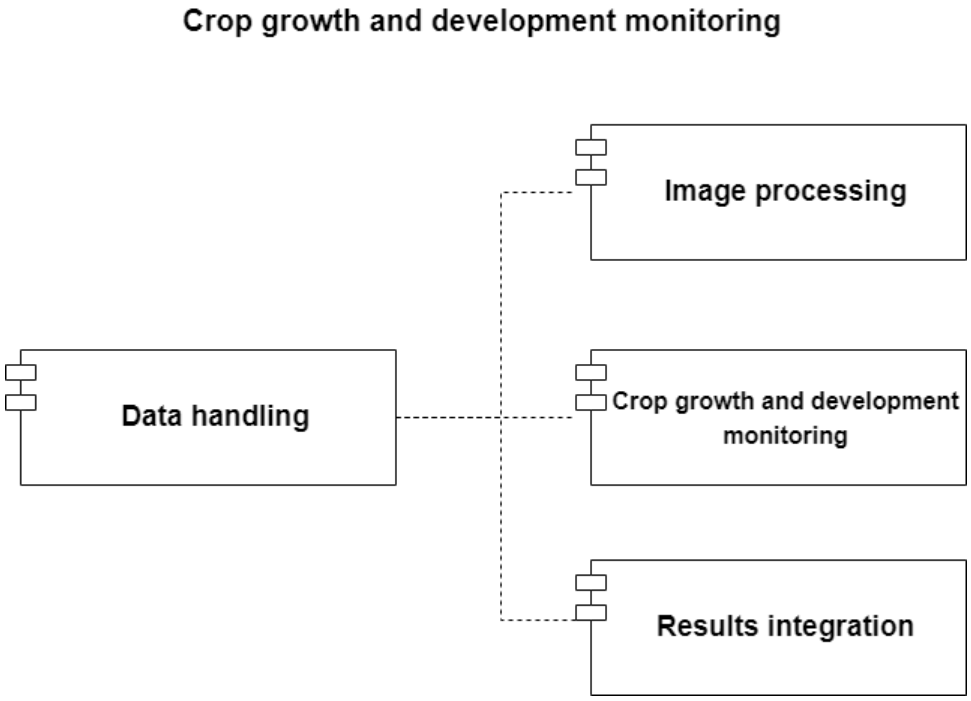


Figure 4: Crop growth and development monitoring functional view

Main functionalities covered by Bundles for Agriculture and Rural areas include:

- **Data Handling:** Data handling entails the management of the recorded images obtained from the CHAMELEON UAV's RGB, multispectral and thermal cameras. After downloading the images from the cameras, the data is then sent to a cloud-based server where the photogrammetry processing takes place. Also, field data relative to water stress, soil moisture sensors and others is sent to the cloud-based server to send the different bundles.
- **Image Processing:** in this case, image processing refers to a complete photogrammetry process with the aim to obtain high-quality orthoimages, digital surface models (DSM) and digital elevation models (DEM), and 3D point clouds, with the different captured images (RGB, multispectral and thermal). It is a post-processing step that would require high computation capabilities and storage capacity. Results of this process will be obtained after hours of processing and require alarm systems for final users to know when the information is available. This information is the base for the different bundles related to agriculture and rural areas.
- **Crop growth and development monitoring:** based on the geomatic products generated with the photogrammetry process, ground canopy cover (GCC) and volume occupied by the crops (individual vines in this case) is obtained. The process starts with the automatic detection of vines, based on the 3D point cloud, and generation of frames with the influence area of each vine. For these frames, using the DSM and DEM obtained with RGB products, GCC and volume is calculated for each flight, allowing the crop growth monitoring. Similarly, with multispectral information, different vegetation indices for each vine are calculated and alarms for erratic evolution of these parameters

are generated. Thermal information supplied key information for water stress monitoring that, related with field measurements, can determine water requirements.

- **Results Integration:** The final step, results integration, involves integration of parameters obtained with drones and field measurements integrating them into a user-friendly format. This includes visual aids such as maps or charts that define the evolution of the crop, even each individual vine, and generation of alerts in case erratic or abnormal trends are present. The aim is to present the results in a way that is easily understandable by non-expert end users. The processed information is then transmitted to the end user, possibly through a report that provides a comprehensive overview of the crop development of the analyzed plots.

5.1.2 DRONE INNOVATION PLATFORM

The Drone Innovation Platform (DIP), which is the front end of the CHAMELEON Cloud Platform, will be one of the main entry points through which users will be able to interact with the CHAMELEON ecosystem. DIP's architecture is created around requirements Req-F04, Req-NF01, and Req-F05.

Not only will the DIP have the CHAMELEON Store embedded, but it will also include dedicated graphical interfaces for each user type. In other words, providers will be able to register their UAVs and corresponding platforms, whereas consumers will use it to manage their orders. Figure 5 shows the functional elements (modules) of the DIP.

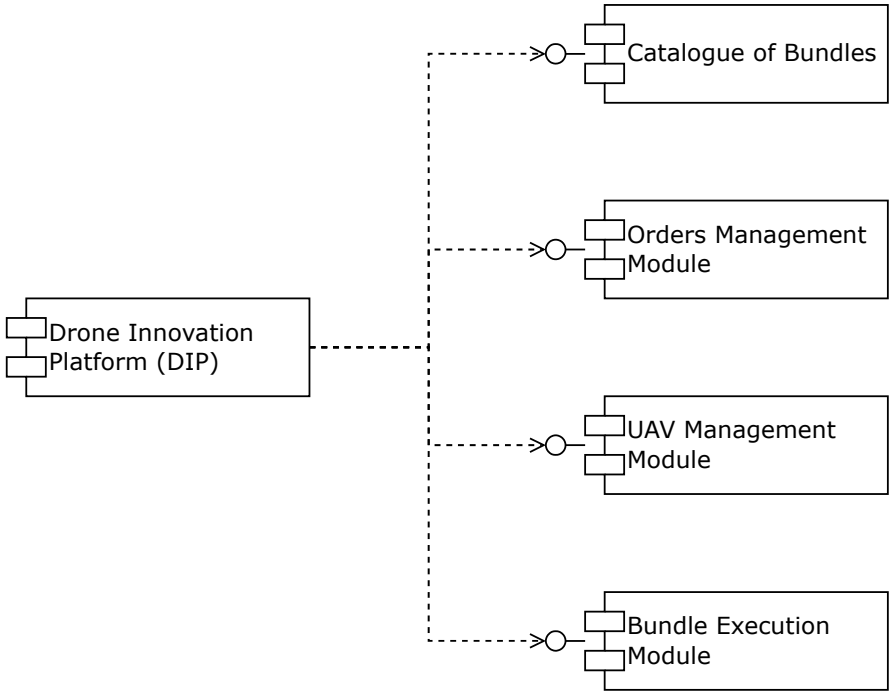


Figure 5: DIP Functional View

CATALOGUE OF BUNDLES

The Catalogue of Bundles will comprise all the bundles available in the CHAMELEON Store as well as the inherent functionalities, such as publishing a bundle, listing, selecting bundles for

deployment and showing their details. The searching filters and algorithms from the CHAMELEON Store will also be available.

ORDERS MANAGEMENT MODULE

The Orders Management module will allow end-users to interact with service requests depending on their role – consumer or provider. Consumers will be able to create new orders providing the required details (i.e., request one of the available services), monitor the status of their orders, and download the corresponding reports. On the other hand, Providers will be able to see all the pending orders and address them, to create the reports with clear, useful, and didactical solutions to be employed by consumers, i.e., customers.

UNMANNED AERIAL VEHICLES MANAGEMENT MODULE

The UAVs Management module will handle the registration and management of UAVs in the DIP along with the platforms that allow users to interact with them.

BUNDLE EXECUTION MODULE

Running the algorithms that output the results to be included in the final reports is an essential part of the CHAMELEON ecosystem. This is where the Bundle Execution module comes into play with its ability to gather input for the models (e.g., images) and show execution logs while the bundle is running, as well as the results once the execution has come to an end.

5.1.3 CHAMELEON STORE

The Store will host all the available solutions developed by the bundle owners. Stakeholders will interact with the Store through the DIP, which will be the frontend of the CHAMELEON Cloud Platform. They will be able to select the bundle of their choice to be deployed on a UAV. Furthermore, the Store contains the Broker that will recommend the appropriate services to the end-users. Figure 6 shows the functional elements (modules) of the CHAMELEON Store.

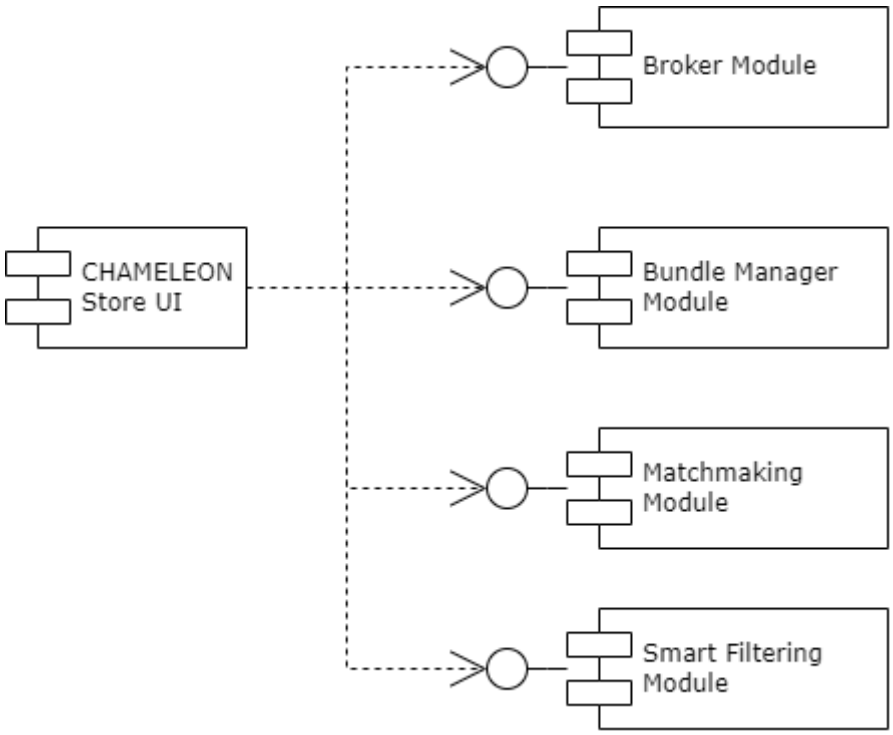


Figure 6: CHAMELEON Store functional view

BUNDLE MANAGER MODULE

The Bundle Manager will be responsible for all functionalities concerning the bundles. This includes publishing a bundle, listing, and selecting bundles for deployment. It will also provide information about a bundle like its deployment requirements, its description and functionality.

BROKER MODULE

The Broker will allow end-users to select the bundle that fits their needs best. This will be achieved by applying match-making algorithms, utilizing the Matchmaking module.

MATCHMAKING MODULE

The Matchmaking module will utilize matchmaking algorithms to find suitable bundles for a user. It will take into consideration different factors (e.g., cost, distance, time), to find the best possible solution.

SMART FILTERING MODULE

The Smart Filtering Module will apply filters to end-users' search for bundles. There will be different filters for bundles depending on the category they belong to (e.g., agriculture, forestry, livestock).

5.1.4 PLUG AND PLAY SERVER

The Plug and Play Server will be the main means of communication between the CHAMELEON Cloud and the CHAMELEON Edge Gateway. It will be responsible for deploying the bundles to the Edge Gateway and receiving data for post-processing by the cloud services. Figure 7 shows

the functional elements (modules) of the Plug and Play Server. The Plug and Play Server is solely designed around the need of users for seamless component connectivity while maintaining ease of use (Req-NF01).

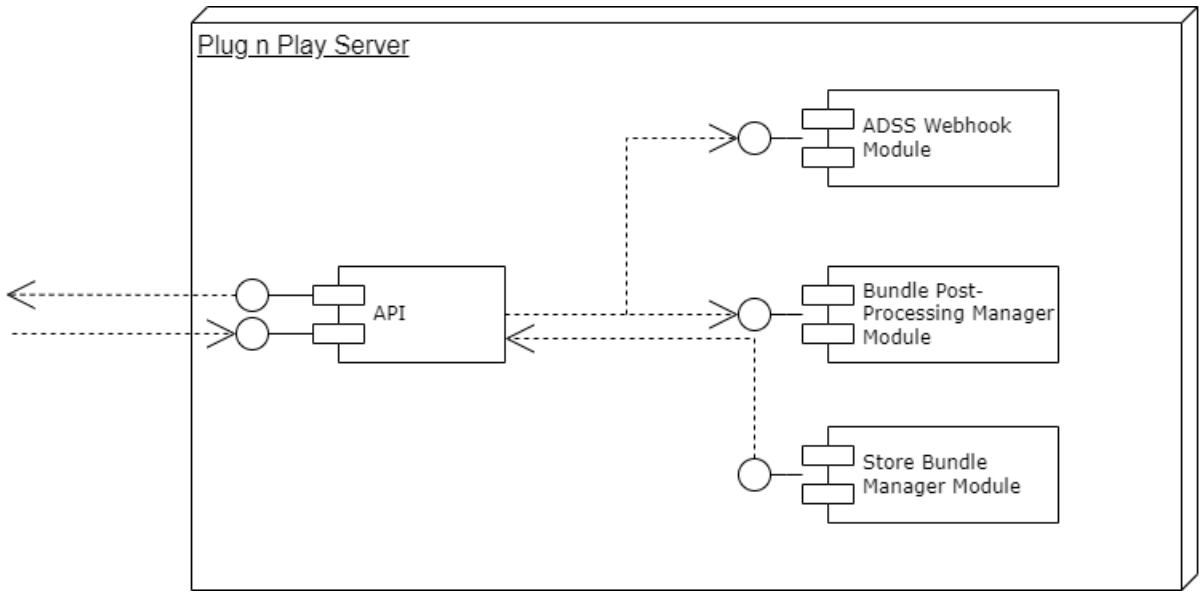


Figure 7: Plug-n-Play Server functional view

STORE BUNDLE MANAGER MODULE

The Store Bundle Manager Module will fetch the selected bundles from the CHAMELEON Store and deploy them to Plug and Play Client in the CHAMELEON Edge Gateway.

BUNDLE POST-PROCESSING MANAGER MODULE

The Bundle Post-Processing Manager Module will handle the post-processing for the bundles that require it. After a bundle is deployed, it will send data through the CHAMELEON Edge Gateway to the CHAMELEON Cloud for post-processing. Post-processing will be handled by the respective service of the bundle running in the cloud.

ADSS WEBHOOK MODULE

The ADSS Webhook Module will handle events coming from the UAV during flight time. The ADSS component will receive these events to produce alerts or visualize the results of the bundle running.

5.1.5 AGRICULTURAL DECISION SUPPORT SYSTEM

The Agricultural Decision Support System (ADSS) will help end users to take decisions. It will display data and the outcome of the bundles developed in T4.3 CHAMELEON, Bundles, services for agriculture and rural areas, T4.4 CHAMELEON, Bundles, services for livestock monitoring, and T4.5 CHAMELEON, Bundles, services for forestry as well as a digital twin of the area of interest to the end user. ADSS is based upon requirements Req-F02, Req-F04, and Req-F05.

Figure 8 shows the functional elements (modules) of the CHAMELEON ADSS.

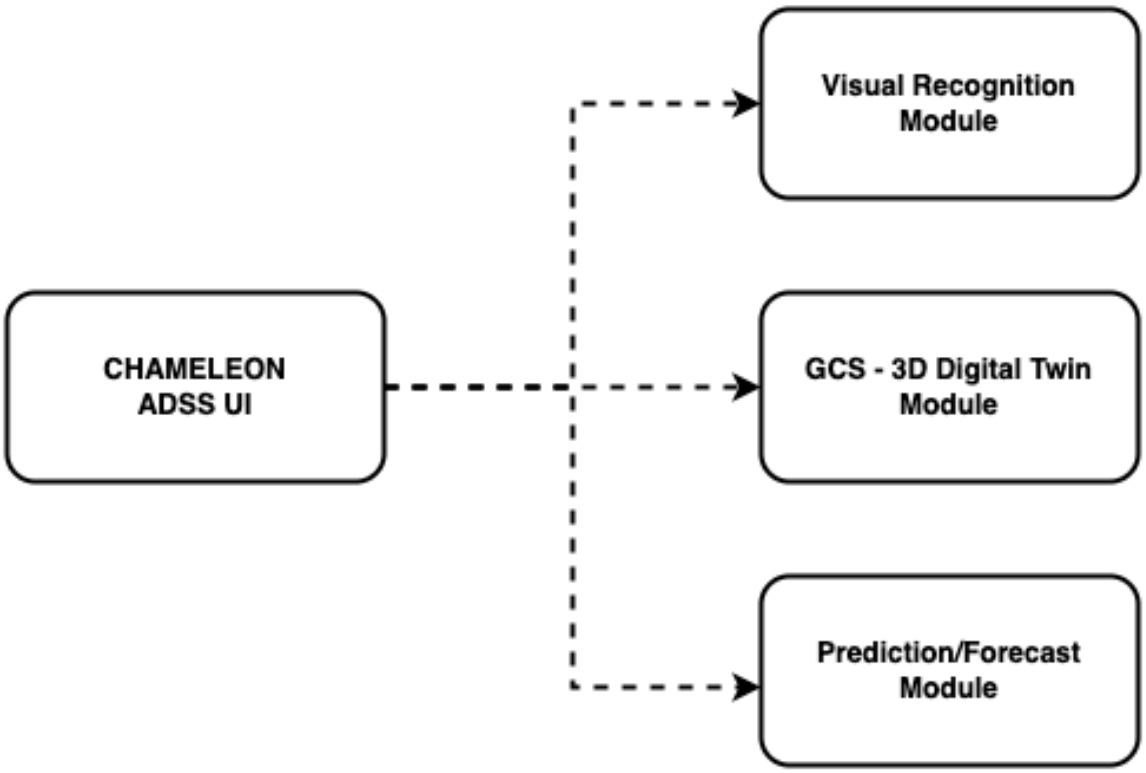


Figure 8: CHAMELEON ADSS Functional View

VISUAL RECOGNITION MODULE

The visual recognition module will show to the end user the results of computer vision algorithms implemented in the three modules developed in WP4, providing near real-time insights and supporting decisions. Using Computer vision, the modules developed in WP4 will provide information to the end-user in all three categories of agriculture and rural areas, livestock monitoring, and forestry. Depending on what is eventually developed in these three modules, the end-user will have a selection of features to select from in order to best suit their needs. The bundles that will potentially be developed will provide insights and information such as crop growth and development monitoring, field stress due to drought estimation, livestock-herd management, individual animal monitoring, hot spot identification for wildfires, and humidity of soil and plants assessing the risk of fire. Combining these results with alerts, infographics and other UI friendly components, the end user can make decisions assisted through data and Computer Vision algorithms.

GCS – 3D DIGITAL TWIN MODULE

The GCS - 3D digital twin of the area of interest will be displayed to the end users, helping them with their decisions. The GCS will provide a map with the area of interest, in order to better organize and orchestrate the different actions that will be available through the bundles for the end user. By providing information through a friendly UI platform such as those provided through the Google Maps Platform or OpenStreetMap, the user can connect the data and results of the different WP4 bundles with the actual locations of the user's area of interest.

PREDICTION/FORECAST MODULE

The Prediction/Forecast module will present predictions and forecasts to the end users, generated by the three bundles developed in WP4 which will use real-time as well as historical data. This way the user will be able to make decisions with AI-based decision support.

5.2 CHAMELEON EDGE

This chapter presents the functional view of the components comprising the CHAMELEON Edge:

- **CHAMELEON Gateway:** Middleware between the CHAMELEON Cloud and the UAVs. Handles the communication between the Cloud and the Physical Layer.
- **Human Machine Interfaces (HMIs):** Interfaces providing access to UAVs' functionalities.

5.2.1 CHAMELEON GATEWAY

Serving as a middleware between the physical layer of CHAMELEON (UAVs, Ground Sensors) and the Cloud infrastructure, the CHAMELEON Gateway will undertake the following objectives:

- Data handling:

  The CHAMELEON Gateway will handle data coming from the physical layer of CHAMELEON. This data will regard either the results produced from the Bundles, running on the UAVs, or information collected by the Ground Sensors. The CHAMELEON Gateway, depending on the source of this data, will be able to communicate them to the corresponding component of CHAMELEON. Specifically, data coming from the Ground Sensors will be pushed to the on-board running Bundles whereas information coming from the running Bundles will be pushed to the Cloud.

- Bundle deployment:

  Being the main connection point between the Cloud and Physical layer of CHAMELEON, the CHAMELON Gateway will be able to handle the Bundles coming from the Cloud and instruct them to be deployed on the on-board system of the UAVs. Specifically, by having the Client instance of the Plug and Play platform running on the CHAMELEON Gateway and the Worker of Plug and Play running UAV on-board, the Bundle can be transferred from the Gateway and be deployed on the on-board system of the UAV.

- Operation of UAVs:

  Flight instructions described through the HMIs of CHAMELEON will need to be communicated to the UAVs. This objective is being undertaken by the CHAMELEON Gateway through the messaging queue system running in the Gateway.

PLUG AND PLAY CLIENT

The Plug and Play Client is the part of the Plug and Play Platform that will be on the CHAMELEON Edge Gateway. It will deploy the bundles on the UAVs and will also work as a forwarder for data coming from the UAVs to the cloud. Figure 9 shows the functional elements of the Plug and Play Client.

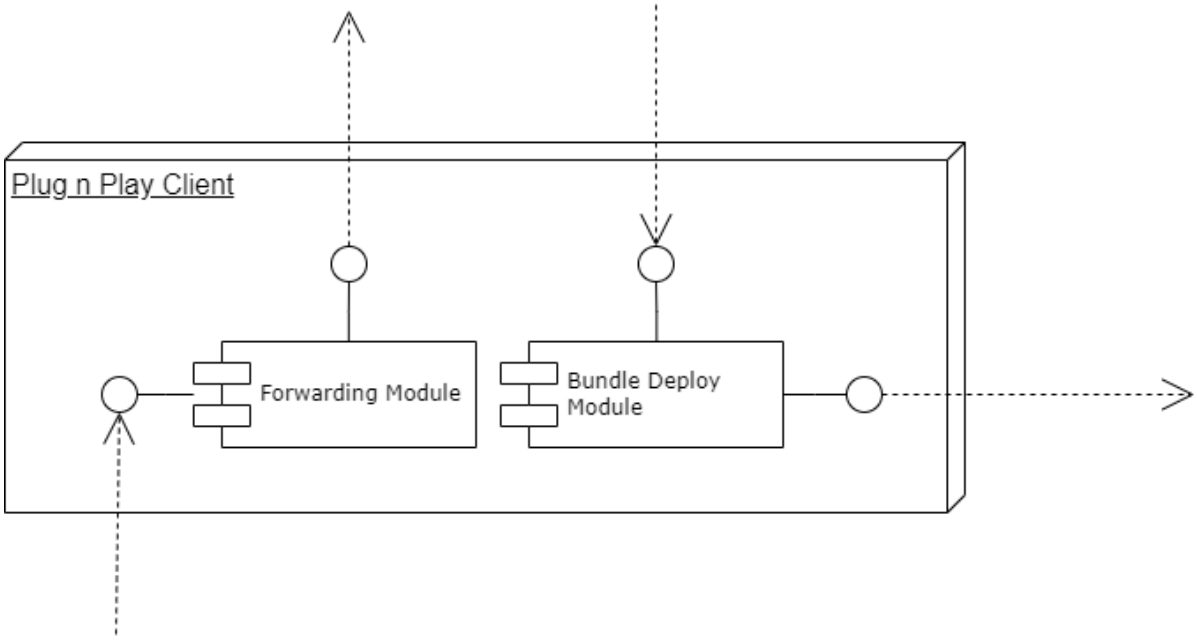


Figure 9: Plug-n-Play Client functional view

FORWARDING MODULE

The Forwarding Module of the Plug and Play Client will act as a forwarder for the cloud. Data coming from the UAVs will be pushed to the cloud through the Forwarding Module (e.g., data that require post-processing).

BUNDLE DEPLOY MODULE

The Bundle Deploy Module will deploy the bundles on the specified UAV. It will handle the requests coming from the Plug and Play Server and redirect them to the UAVs.

5.2.2 HUMAN MACHINE INTERFACE

Delair's user interface (Delair Flight Deck Pro) will be adapted under the scope of CHAMELEON to be able to support operating the CHAMELEON UAVs. This adaptation will allow for the same user interface to be used both between Delair and Acceligence UAVs.

Figure 10 shows the functional elements (modules) of the user interface.

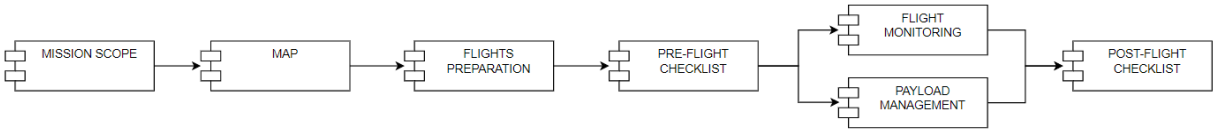


Figure 10: User interface functional view

MISSION SCOPE

Mission scope allows the user to create and import areas and lines of interest. The user can also create and define restricted areas instructing the UAV to avoid flying over areas out of scope.

MAP

The user can define maps and DEM needed for the mission. These maps are used in background in order to visualize over it all information of flight plan and also to monitor the flight (drone position, pictures projections...). DEM is used to check that the flight plan is safe. The HMI automatically checks if the altitude of each point of the flight plan does not collide with the ground. Online maps are available but also local files can be importing such as Geotiff, KML, Shapefiles, DTED.

FLIGHTS PREPARATION

The user can define all the flights needed to cover all the areas and lines defined in the mission scope.

For each flight, the user must choose the type of aircraft (e.g.: a CHAMELEON UAV) and the payload. The flight plan will be automatically created depending on the selecting settings. Warnings and errors can be displayed if the flight plan is considered not safe. The user can adjust the flight plan manually if need be.

A geocage, which is an area where the UAV cannot go outside of, can be defined in the flight preparation for safety reasons. If for any reason, the UAV goes outside of the geocage, the FTS (Flight Termination System) will be automatically triggered.

PRE-FLIGHT CHECKLIST

A dedicated pre-flight checklist is available for each type of UAV to ensure that the mandatory checks are performed before initiating the flight. Some checks are automated, such as GNSS reception, while others are manual, such as checking that the propellers are secured and not damaged.

FLIGHT MONITORING

The user can follow the UAV trajectory and other information, such as battery, altitude, and status and they can also modify the flight plan during the flight.

The HMI will be connected to the CHAMELEON Gateway to get results from the on-board running Bundles.

PAYLOAD MANAGEMENT

The user will be able to control the payload in flight. The commands depend on the payload used.

- Gimbal video EO /IR:
  - Manual control using joystick.
  - Zoom / Un-zoom.
  - Video tracking
  - Geo-tracking.
  - Mobile object tracking
  - Video switch between EO and IR camera
  - IR video colorization
  - Enable / disable video recording.
  - Enable / disable metadata display.
  - Enable / disable video projection on map.
  - Display video target position on map
  - Export coordinates from video
- LiDAR with 20MPix camera:
  - Enable / disable acquisition.
  - Enable / disable picture preview streaming.
  - Enable / disable picture/lidar projection display on map.
  - Enable / disable picture coverage display on map.
  - Enable / disable picture mapping display.
- Multi-Spectral sensor:
  - Enable / disable acquisition.
  - Enable / disable picture preview streaming.
  - Enable / disable picture projection display on map.
  - Enable / disable picture coverage display on map.
  - Enable / disable picture mapping display.

POST-FLIGHT CHECKLIST

After the UAV has landed, the drone operator will follow a Post-Flight Checklist in order to not miss any steps such as securing the FTS with the Remove Before Flight before powering off the UAV.

5.3 CHAMELEON PHYSICAL LAYER

This chapter outlines the functional view of the CHAMELEON Physical Layer components. These components are briefly explained here:

- **UAVs:** The drones and fixed wings designed for the project's purposes.
- **Ground Sensors:** Sensors that will send data to the UAVs.

5.3.1 UAVS

The remote piloted aircraft systems will adapt Delair's UAV in order to on-board all required equipment to cover pilot cases defined in T2.2. Delair UAV will be able to carry different payloads such as Gimbal video EO/IR, LiDAR, and Multi-Spectral sensors. An on-board processor will be added for AI processing in near real-time.

Figure 11 shows the functional elements (modules) of the Remote piloted aircraft systems.

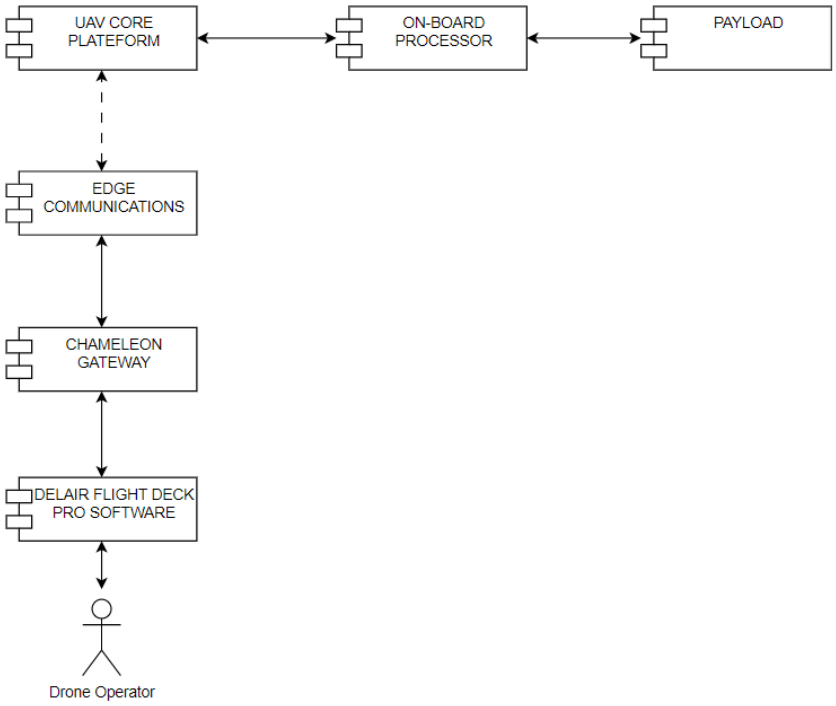


Figure 11: Remote piloted aircraft systems functional view

DELAIR FLIGHT DECK PRO (DFDPRO) SOFTWARE

DFDPro software, which is running on a Windows Laptop, will allow the drone operator to prepare and monitor the flight of the UAV. This software is described in T3.4.

EDGE COMMUNICATIONS

The edge communications allow for communication with the UAV. It integrates modems that allow to reach up to 50km of communication data link (Line of sight). Data (such as video), control and commands will be sent and received through this system.

UAV CORE PLATFORM

UAV Core Platform allow to manage communication with the edge communication and with the payload. It also integrates an autopilot which control autonomously the UAV and follow the flight plan prepared and sent by the drone operator. Payload data are saved on an onboard storage.

ON-BOARD PROCESSOR

The on-board processor (NVIDIA module) will allow to process payload data in near real-time in order to give high value information to the drone operator such as animals detections. It will be integrated into DT26E and CHAMELEON UAV.

PAYLOAD

Three payloads will be available on DT26E to cover pilot cases:

- Gimbal video EO / IR:
  - Full HD EO camera with x30 optical zoom (60° to 2.2°)
  - HD IR camera with x8 digital zoom (37° to 2.3°)
- LiDAR with 20MPix Camera and PPK module
- Multi-Spectral sensor (MicaSense)

PLUG AND PLAY WORKER

The Plug and Play Worker is the part of the Plug and Play Platform that runs on the UAV. It will get the bundle from the Plug and Play Client that runs on the Edge Gateway. The deployed bundle will run on the UAV and will be able to perform AI Processing, while being able to access the UAV's internal storage.

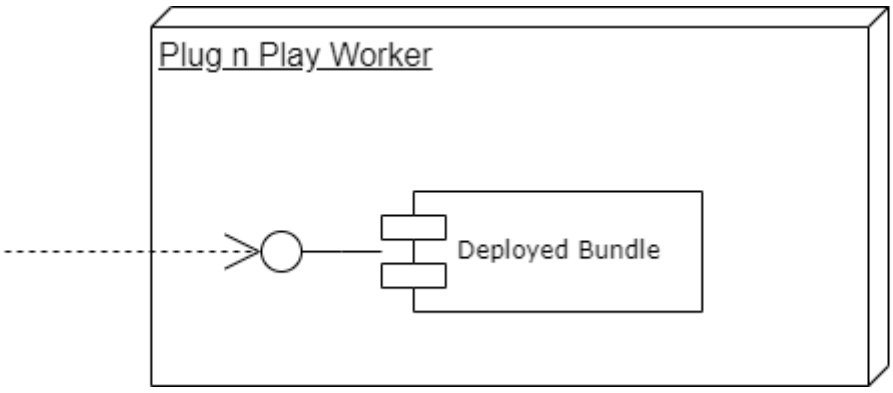


Figure 12: Plug-n-Play Worker functional view

CHAMELEON ON-BOARDING

The CHAMELEON On-Boarding is a process for creating and preparing structure dataset for bundles. The process will be done in on-board processor for creating visual results as orthophoto maps. On-boarding interacts with different sensors. The UAV on-board processor in a real time will process the maps from different sensors actual data. After processing the images into the structuralized map. CHAMELEON On-boarding will be able to either send structured data to the on-board deployed bundle or, depending on the requirements of each Bundle, to the CHAMELEON Gateway. The main objectives are to maximise the probability of positively identifying potential problems, including agriculture, water, or forest, across heterogeneous sources and networks, thus supporting in-time decisions. To maximise the effectiveness of multimodal monitoring/management.

OPEN PAYLOAD TOOLSET

The open payload toolset will be designed to host various sensors that have been identified through the pilot case surveys and feedback, allowing the UAV to perform a range of different tasks, from precision agriculture to selective spraying. Additionally, the toolset will include an onboard processor capable of executing demanding algorithms when necessary, making it an integral component of the Chameleon UAV and RPAS solutions. Furthermore, the toolset will be designed to allow end-users to adjust the appropriate payload through a simple and secure plug-and-play procedure, making it easy to switch out different sensors and other payloads as needed for specific mission requirements. This level of flexibility and ease-of-use will enable the end-users to have a single, flexible, and adaptable UAV that can be used for a range of different scenarios.

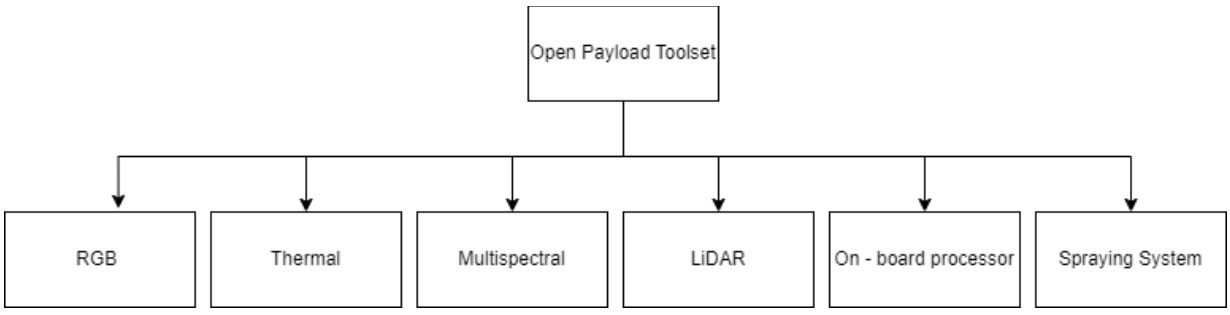


Figure 13: Available payloads for Toolset

5.3.2 GROUND SENSORS

To complement data streams captured through the UAVs' sensors (Lidar, RGB Camera etc.) and to aid the CHAMELEON Bundles in producing more accurate results, the overall system architecture of CHAMELEON is designed to support ground metrics coming from IoT capable Ground Sensors. Located in the Edge infrastructure, these Ground Sensors will provide valuable data streams to the Bundles that support them.

5.4 COMMUNICATIONS

In this chapter, the communications between the three main layers of CHAMELEON are further explained. As mentioned above, the CHAMELEON Gateway acts as a middleware between the CHAMELEON Cloud and the CHAMELEON Physical Layer. Therefore, this chapter presents the Gateway-Cloud and Gateway-Physical Layer communications. Communication between components ensures that data and connectivity procedures are seamless, fulfilling requirements Req-F02, Req-F03, Req-F04, Req-F-05, Req-NF01, an Req-NF02.

5.4.1 GATEWAY-CLOUD

The communication between the Gateway and the Cloud works both ways. The Cloud needs to communicate with the Gateway to deploy the bundles on the UAVs. The Plug and Play Server is responsible for starting this communication after a bundle has been selected for deployment. The Plug and Play Client of the Edge Gateway will receive the bundle and will push it on the UAV afterwards.

The communication from the Gateway to the Cloud can happen on two occasions, either during or after flight time. In both cases, the Forwarding Module of the Plug and Play Client handles the data transfer to the Cloud. In the first case, data can be sent during flight time to trigger an alert through the ADSS component in the cloud. In the second case, data can be sent to the cloud after flight for post-processing if a bundle requires it.

5.4.2 GATEWAY-PHYSICAL LAYER

The Gateway will host a Pub/Sub framework that will handle the communications between the Gateway and the Physical Layer, and between the different components of the Physical Layer. The Plug and Play Client will deploy the bundles on the Plug and Play Worker on the UAV. During the flight, the UAV will require data from the ground sensors. This data will go through the Pub/Sub framework located in the Gateway and will be pushed to the UAV. In addition, the UAV will utilize the Pub/Sub Framework to send data to the Cloud through the Gateway.

6 IMPLEMENTATION VIEW

This chapter documents the overall system architecture of CHAMELEON through the developer's perspective. Each building block of the architecture will be described in terms of behaviour and internal composition, aiming to provide a more detailed view on the way the individual parts will be integrated into the overall system architecture.

6.1 DRONE INNOVATION PLATFORM

The Drone Innovation Platform (DIP) consists of the front end of the CHAMELEON Cloud Platform. Therefore, it is protected with a common authentication mechanism for the several components it contains, i.e., the graphical interfaces through which users navigate and have access to the CHAMELEON functionalities. These include:

- The CHAMELEON Store with a set of bundles prepared to tackle specific problems within a given domain.
- Orders Manager for Consumers to create new orders, monitor their status and obtain the corresponding reports once they have been addressed. On the other hand, this allows Providers to consult the pending orders, solve them and produce and upload the final reports with the proposed strategies.
- UAV Manager for Providers to register and manage the UAVs they have available as well as the corresponding platforms.
- Bundle Executor that provides feedback to the user regarding the status of the bundles execution and displays the results to be included in the report.

The interactions between these elements are summarised in Figure 14.

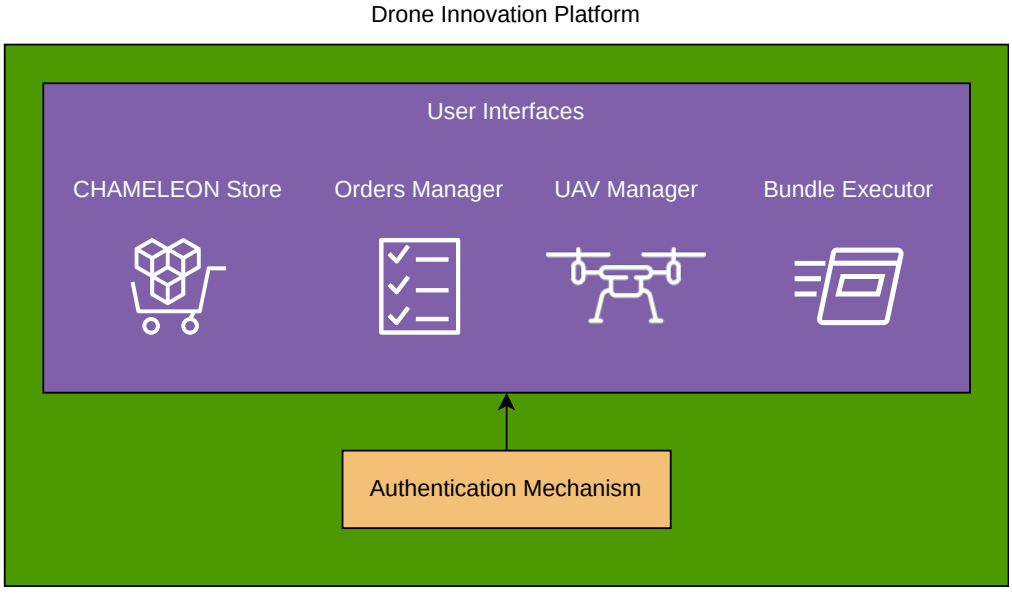


Figure 14: DIP Component Diagram

Since the platform is available to both non-technical (livestock unit owners) and technical end users (bundle developers), different flows can be identified throughout the DIP.

CONSUMER

From the perspective of the Customer, the flow is mainly related with the orders, as follows:

1. A bundle from the CHAMELEON Store is selected by the user according to their needs and what the bundle offers (Figure 15).

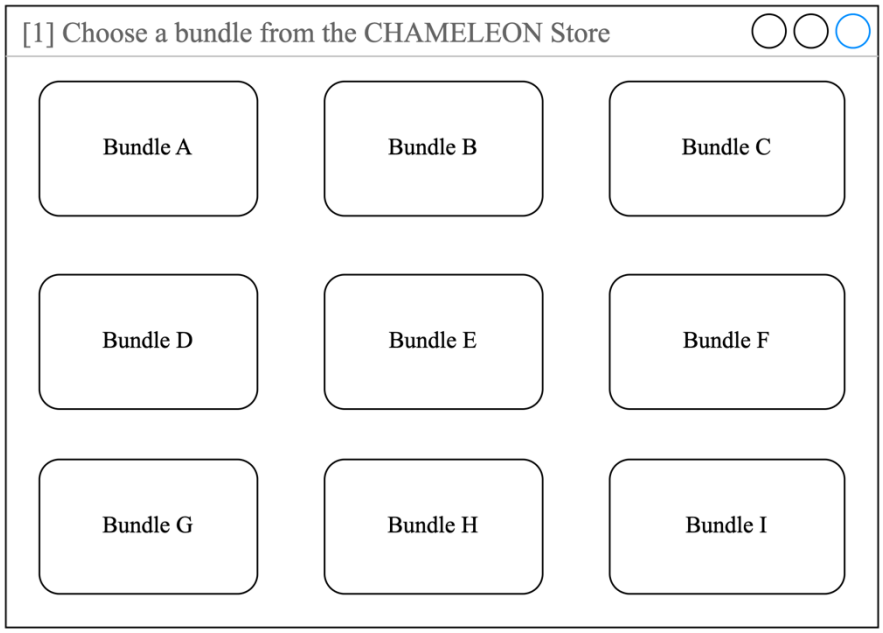


Figure 15: Choosing a bundle from the CHAMELEON Store

2. An order is created for the selected bundle, providing the necessary parameters (e.g., location, surface, needs), as shown in Figure 16.

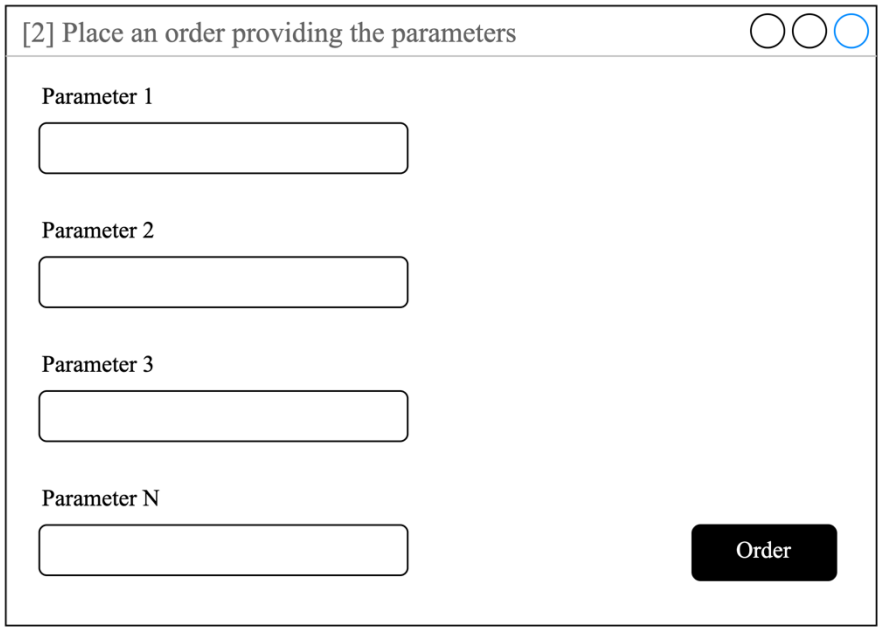


Figure 16: Creating a new order.

3. The user checks that the new order was added to their list, as portrayed in Figure 17.

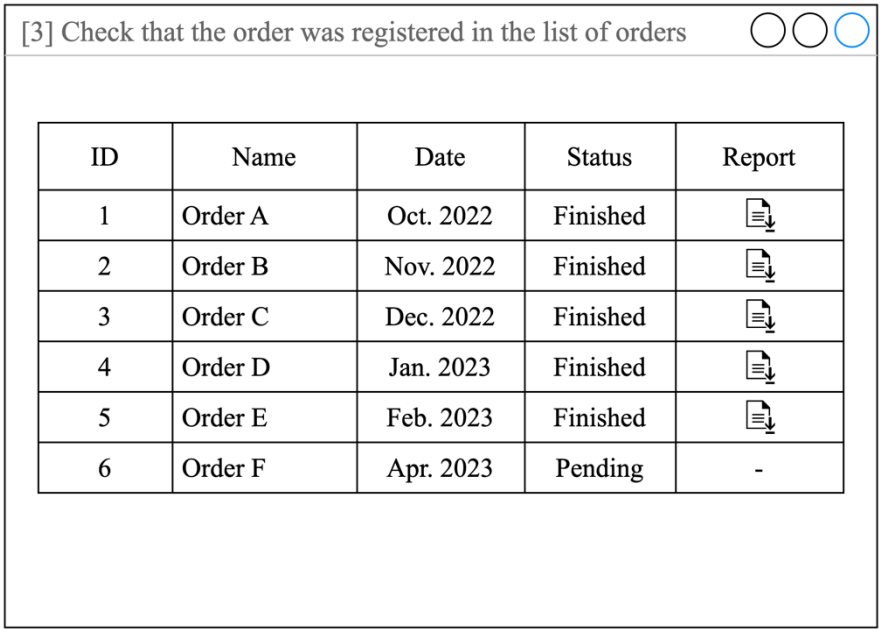


Figure 17: Checking that the new order was successfully created.

4. As soon as an order has been solved, the report can be previewed and downloaded, as depicted in Figure 18.

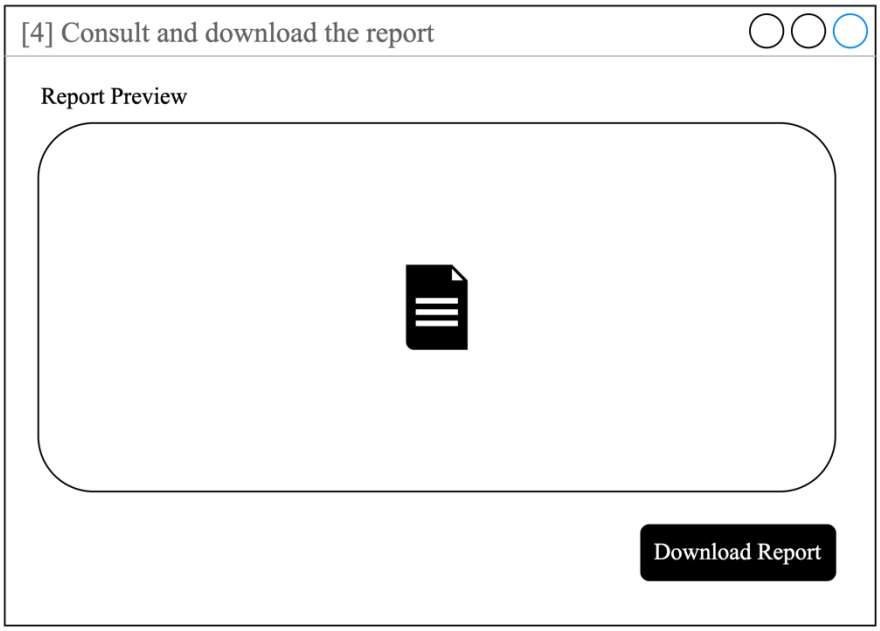


Figure 18: Consulting the report.

PROVIDER

From the standpoint of the Provider, the focus is on registering the available equipment as well as addressing open orders. The associated steps are listed below:

1. The user registers their UAVs and corresponding platforms providing the appropriate details, such as the available cameras and other equipment, as well as the location they operate at (Figure 19).

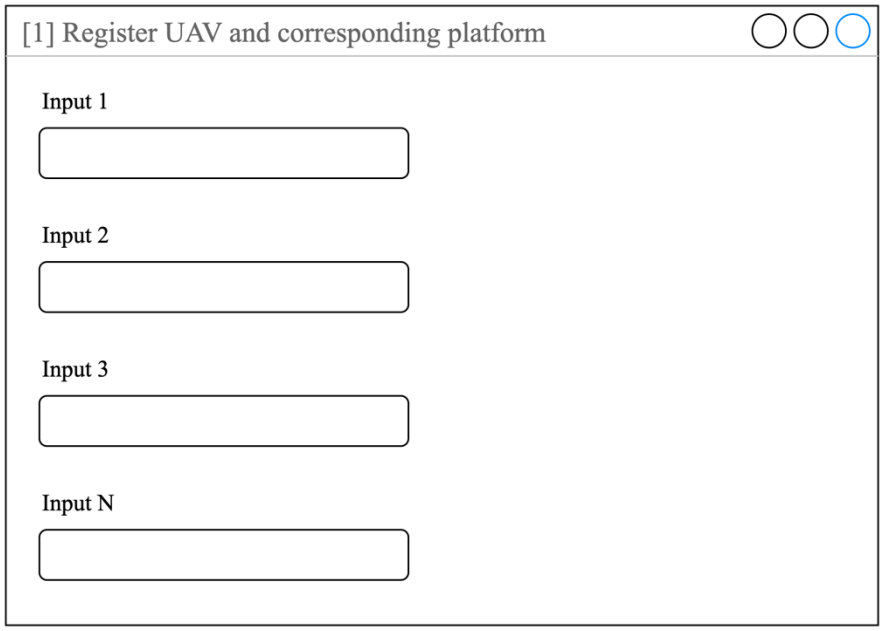


Figure 19: Registering UAVs and corresponding platforms.

2. In a similar page to the one available for the Consumers, the Provider chooses which order to address (Figure 20).

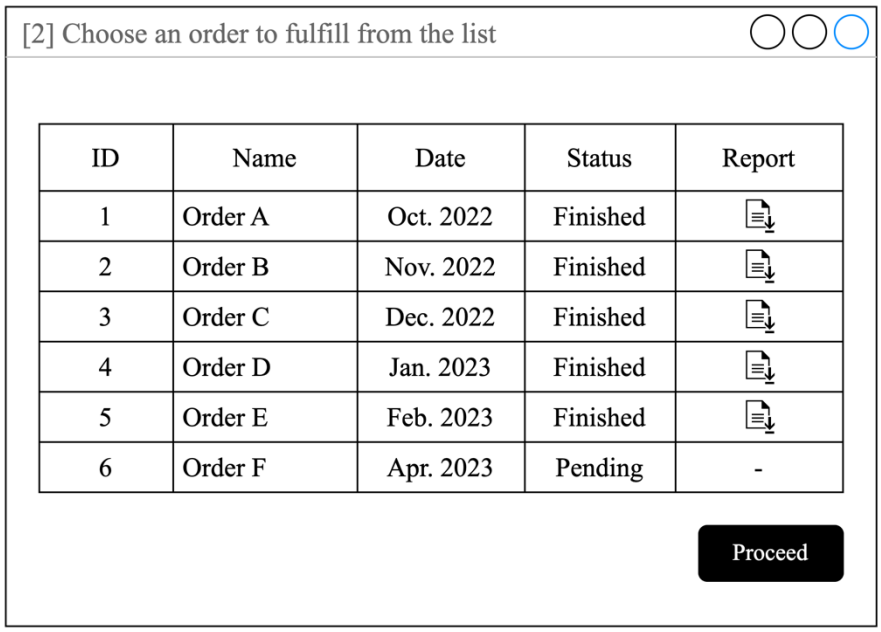


Figure 20: Choosing an order to address.

Additionally, the execution of the bundles is of vital importance for obtaining the results that are used in the production of the reports for the Consumers. The associated flow consists of the steps below:

1. The user provides the input required by the bundle, as shown in Figure 21. This may be images, ground control points (optional in case of drones) or field data coming from IoT sensors. The temporal window for the analysis or monitoring can also be an input.

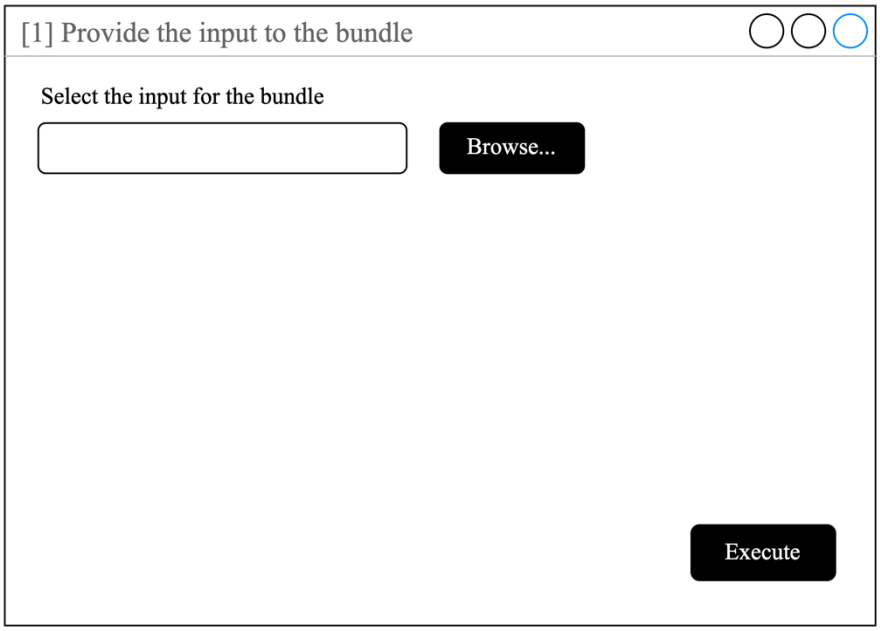


Figure 21: Providing the input to the bundle.

2. The bundle is executed (e.g., in a Docker) while some logs are displayed (Figure 22).

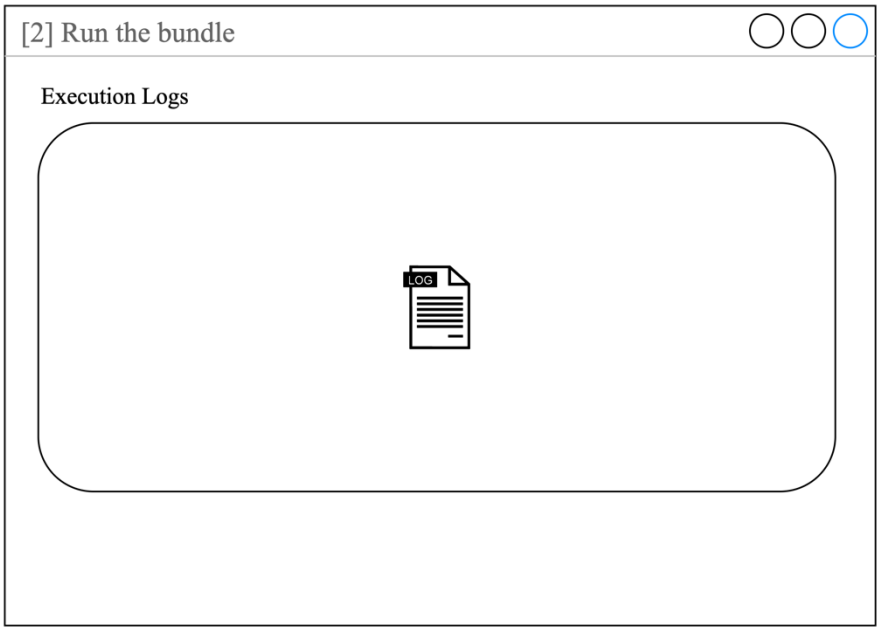


Figure 22: Executing the bundle.

3. As shown in Figure 23, the output of the bundle is used in the creation of a report with a clear, useful, and didactical solution. The report is then uploaded, i.e., made available to the Consumer.

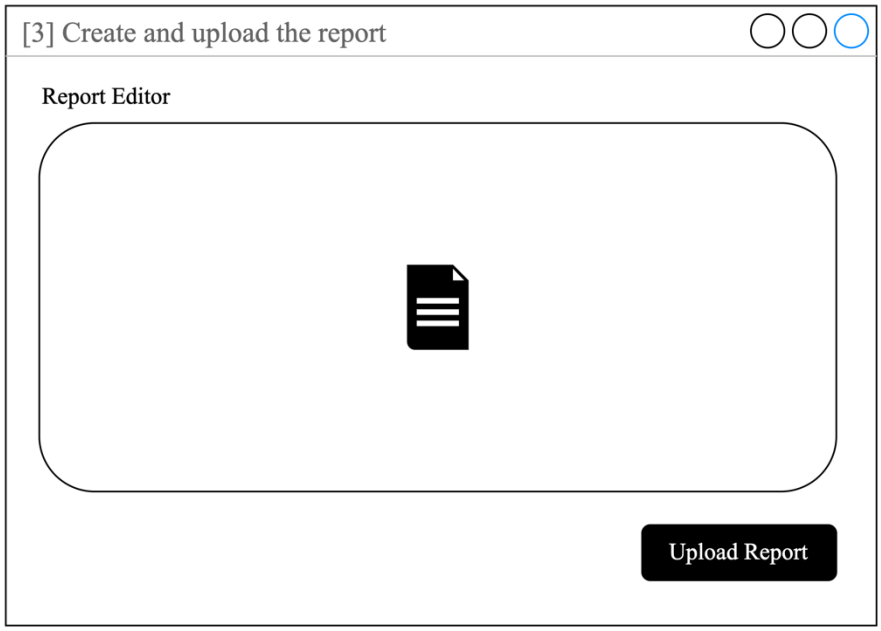


Figure 23: Creating and uploading the report.

6.2 CHAMELEON STORE

Chameleon store, located in the Cloud infrastructure of CHAMELEON, will host, present, and visualise the different bundles developed under the scope of the project.

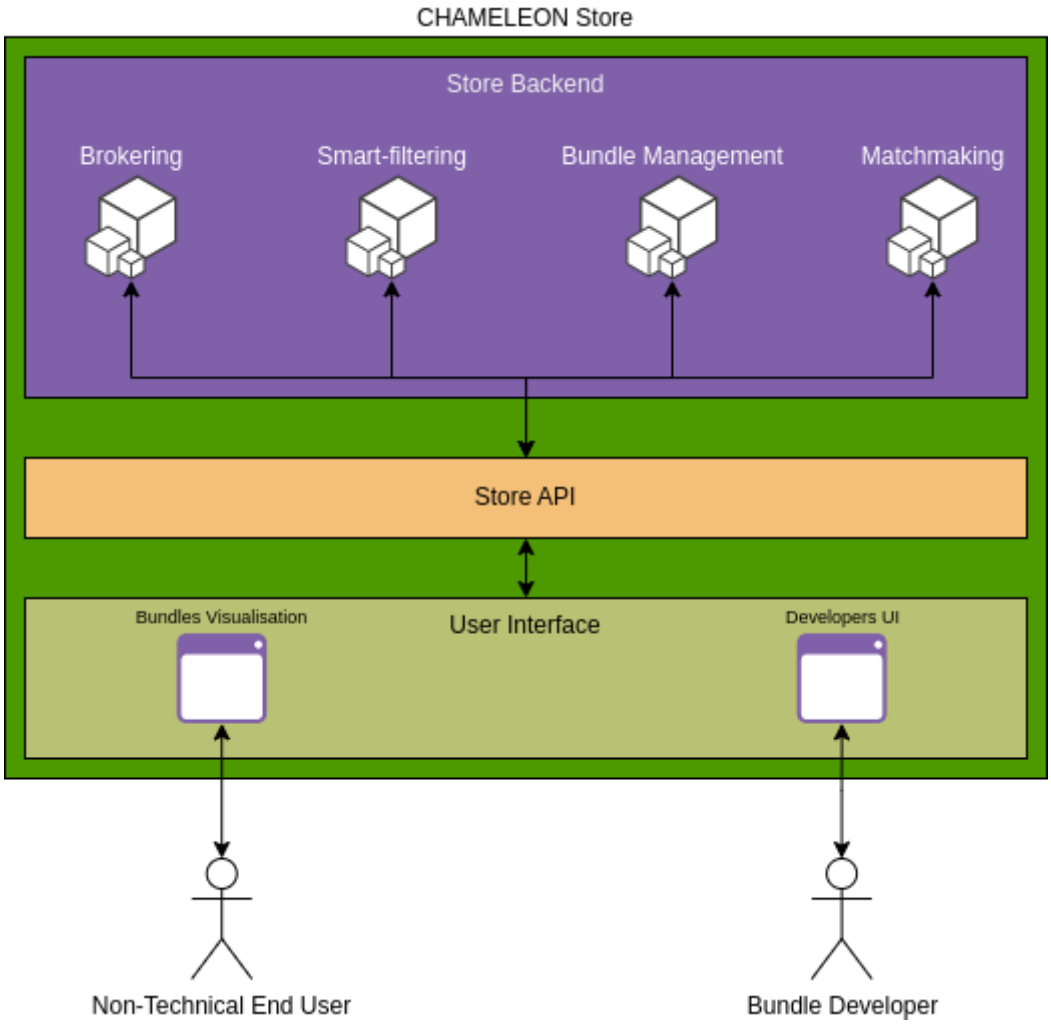


Figure 24: CHAMELEON Store Component Diagram

Available to both non-technical (livestock unit owners) and technical end users (bundle developers), the CHAMELEON Store as depicted in Figure 24, is designed to provide a suitable User Interface to carry out the corresponding operations:

- Bundle Upload

The CHAMELEON Store will provide the necessary User Interface to enable the Bundle developers to upload their developed solution to the Cloud infrastructure of CHAMELEON.

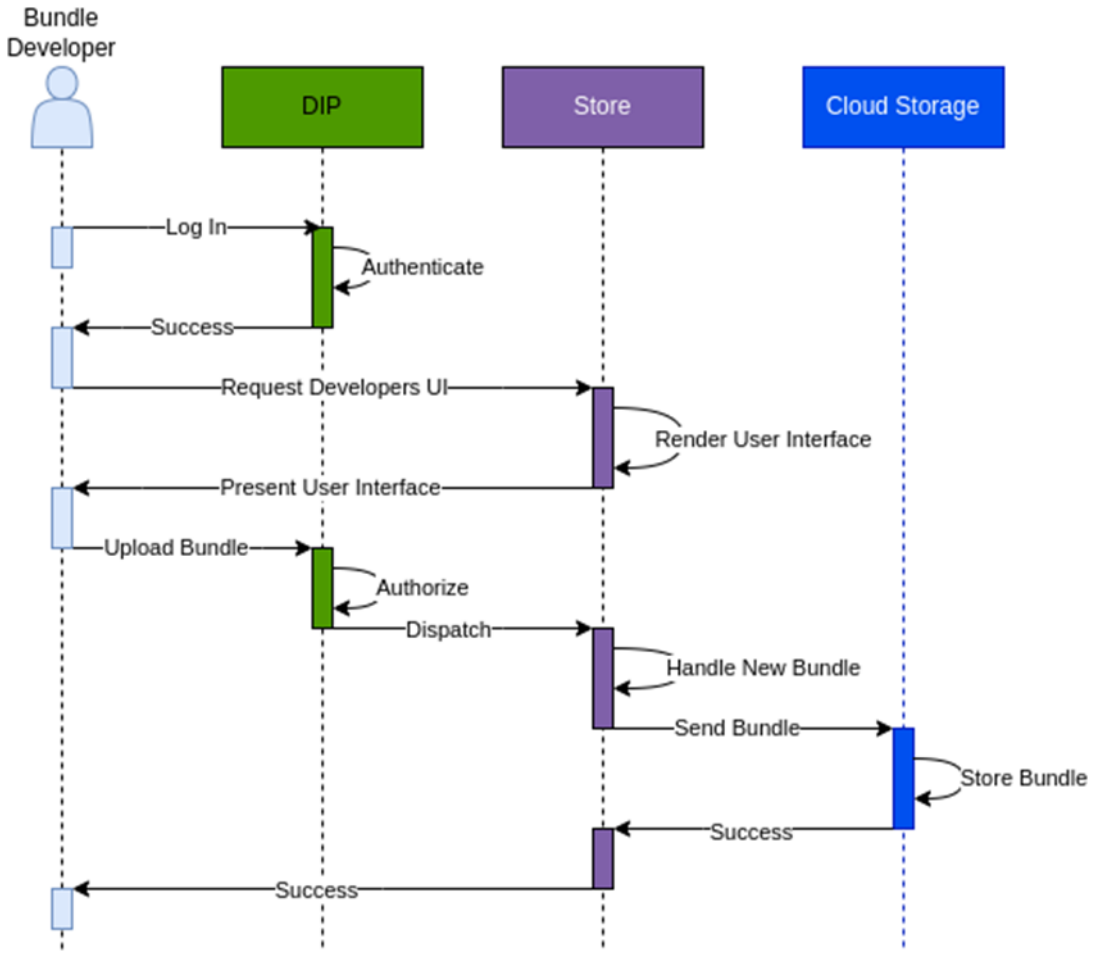


Figure 25: Bundle Upload Sequence Diagram

Bundle developers, authenticating through the Drone Innovation Platform (DIP) of CHAMELEON, can request access to the corresponding User Interface (UI). Through that interface they can upload the developed bundle after being authorised by the DIP. Store then handles the received bundle and parses it to a Cloud Storage endpoint to store and make it available for deployment. Ultimately, a success status message of the described sequence of events is presented to the Bundle developer.

- Bundle selection:

The CHAMELEON Store will offer a simple and easy to use User Interface for non-technical end users (e.g., Vineyard owners) by supporting a recommendations system to present each stakeholder with Bundles relative to their preferences.

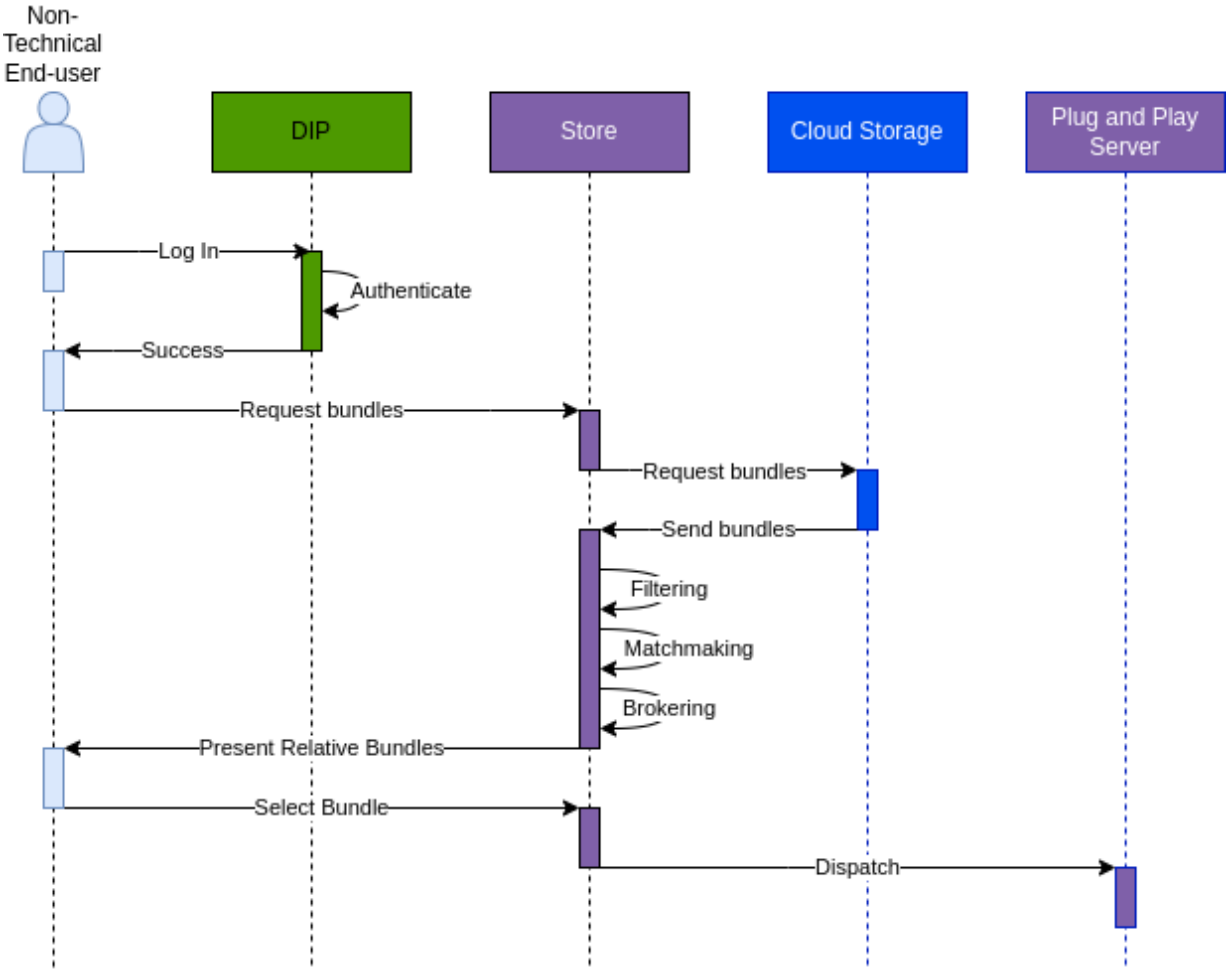


Figure 26: Bundle Selection Sequence Diagram

A non-technical stakeholder (e.g., Forest owner) after logging in through the DIP, can request the available bundles from the CHAMELEON Store. The CHAMELEON Store, after retrieving them from the Cloud storage, organises them through a filtering system to present bundles relative to the end-user that conducted the request. The end-user then selects the bundle that they are interested in, and this information is parsed to the Store. Finally, this request is further dispatched to the CHAMELEON Plug and Play Server to initiate the deployment operation.

The CHAMELEON Store, by supporting the operations depicted in Figure 25 and Figure 26 will provide a platform for end-users to interact with the whole CHAMELEON ecosystem.

6.3 BUNDLES FOR AGRICULTURAL AND RURAL AREAS

The agriculture and rural areas bundles will be containerized, which will facilitate their integration into the CHAMELEON platform. As an example, Figure 27 illustrates the bundle for crop growth and development monitoring, which is described as follows:

Starting from images acquired with RGB and multispectral cameras of the CHAMELEON UAV, frames are extracted and stored in the storage center in Chameleon. The crop growth and development monitoring bundle requests this information and sequentially proceeds to:

1. Process the images, in the cloud service provided by Chameleon ecosystem, where photogrammetry process will be implemented.

2. Determine different parameters at vine (plant) level, such as green canopy cover, volume occupied by the biomass, and different vegetation indices.
3. Merge both results.

Finally, the results are being sent to the Plug and Play Client.

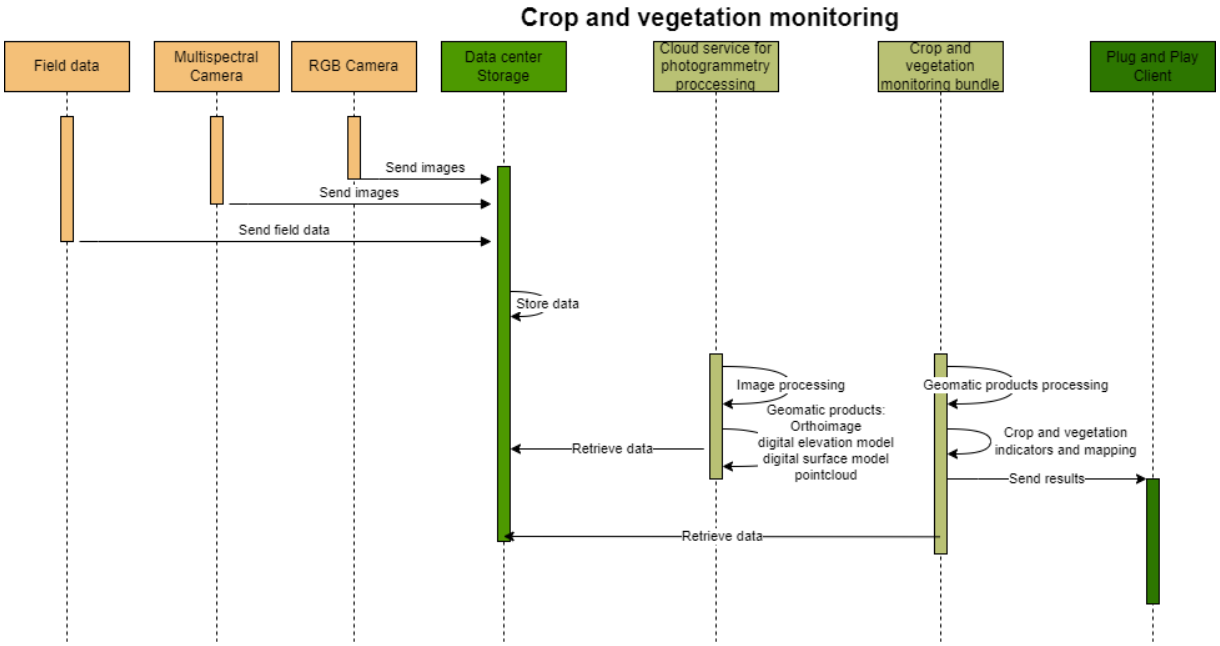


Figure 27: Crop growth and development monitoring bundle: sequence diagram

6.4 BUNDLES FOR LIVESTOCK MONITORING

Our approach to implementing the Animal detector module hinges on a combination of hardware and software resources.

- **Hardware:** We utilize the NVIDIA Jetson[2] platform, a powerful AI-capable edge device, for near real-time video processing.
- **Software:** We leverage pre-trained deep learning models (like YOLO [1]) for object detection. These models are integrated with our drone's control and GPS systems to effectively locate and track animals.

[2] https://elinux.org/Jetson_AGX_Xavier

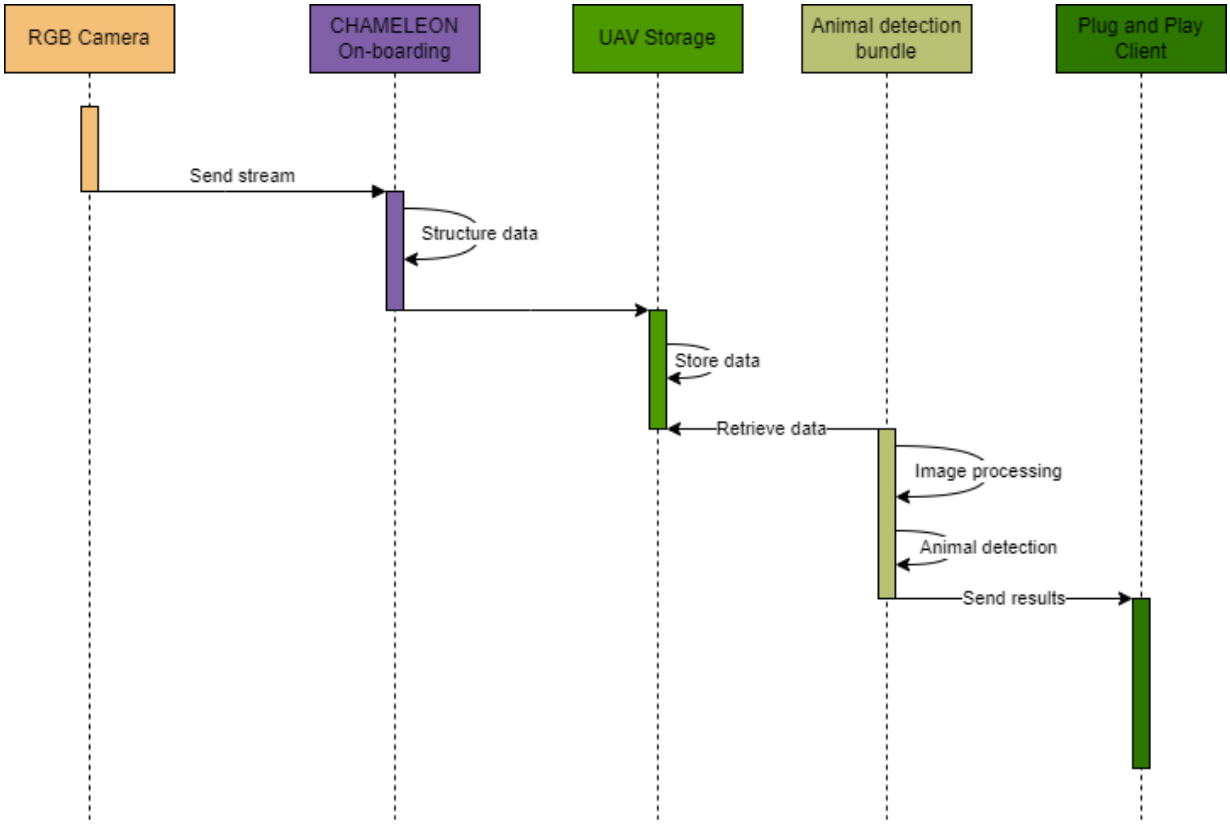


Figure 28: Animal detection bundle sequence diagram

In the context of the animal detector module within the CHAMELEON ecosystem, we have identified key components that contribute significantly to its functionality and integration. These include:

- **RGB Camera:** Captures real-time video footage during the drone's flight.
- **CHAMELEON On-boarding:** Ensures seamless integration of the animal detector module into the CHAMELEON ecosystem.
- **UAV Storage:** Stores a local copy of the processed data for redundancy.
- **Animal Detection Bundle:** Includes the Jetson device and the pre-trained object detection model. Processes the video feed and identifies animals.
- **Plug and Play Client:** Facilitates data communication between the animal detector module and other components of the CHAMELEON ecosystem.

The sequence diagram (Figure 28) would depict the key processes of the module:

- The **RGB Camera** captures video during the drone's flight and sends a continuous stream to the Animal Detection Bundle.
- The **Animal Detection Bundle** processes the video stream. This includes:
    - **Image Processing:** The raw video feed is processed frame by frame, preparing it for object detection.

    - **Animal Detection:** Using pre-trained models, animals within each frame are identified and classified.
- **The Animal Detection Bundle** sends the processed data (animal identifications, classifications, and locations) to the UAV Storage and the CHAMELEON On-boarding.
- The **CHAMELEON On-boarding** structures and formats the data for compatibility with the CHAMELEON ecosystem. This process repeats as new data is received.
- The **UAV Storage** stores a local copy of the processed data for redundancy. It also retrieves previously stored data when requested by the Animal Detection Bundle. This process repeats as new data is received.

Finally, the **Animal Detection Bundle** sends the detection results (animal identifications, classifications, and locations) to the Plug and Play Client for further communication with the rest of the CHAMELEON.

6.5 BUNDLES FOR FORESTRY

The forestry bundles will be containerized, which will facilitate their integration into the CHAMELEON platform. As an example, Figure 29 illustrates the bundle for detecting woody debris in rivers, which is described as follows:

Starting from the stream of the onboard RGB camera of the CHAMELEON UAV, frames are extracted and stored in the UAV Storage. The "Wood Debris in rivers" bundle requests this information and sequentially proceeds to:

1. Process the images.
2. Create a mask from the segmentation of the water areas.
3. Recognize the woody debris.
4. Merge both results.

Finally, the results are being sent to the Plug and Play Client.

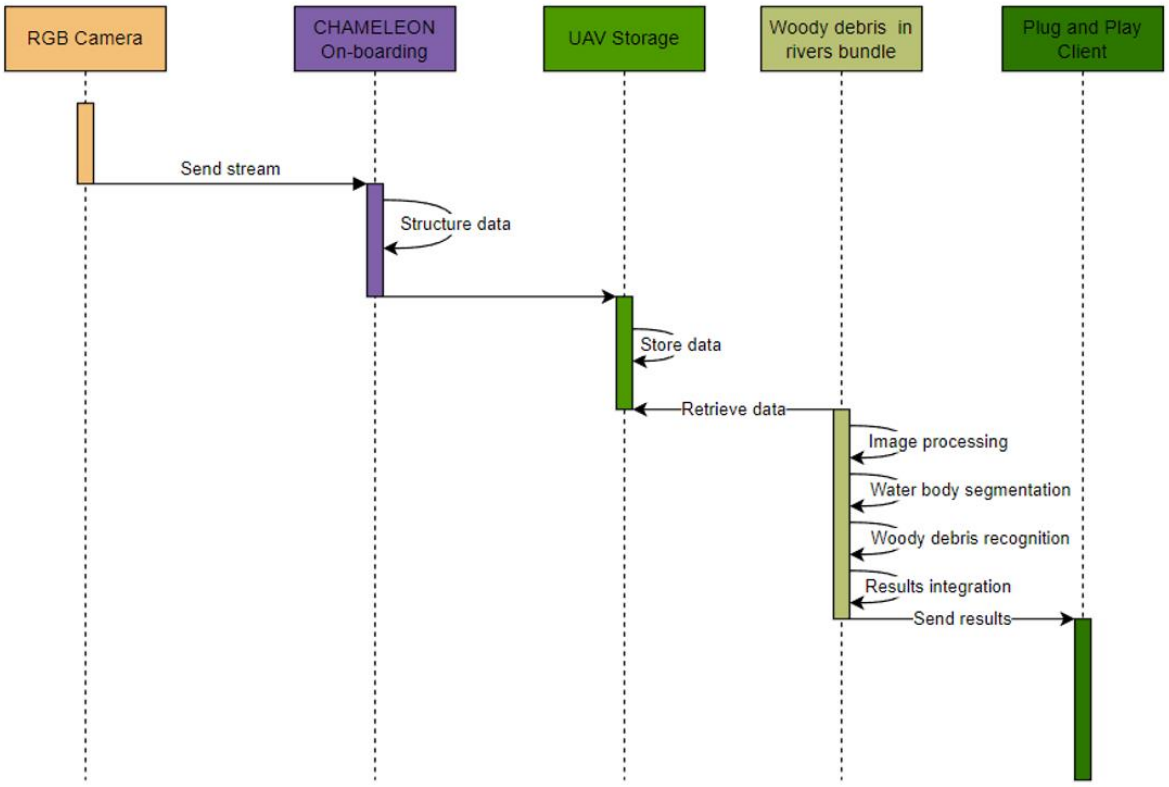


Figure 29: Wood debris bundle: sequence diagram

6.6 CHAMELEON APIS

The following sequence diagram represents the integration sequence of CHAMELEON's project, highlighting the communication pathways among the Bundle Developer, Drone Innovation Platform (DIP), Authentication & Authorization component, Store, Gateway, Plug and Play Platform Client (PPC), and Unmanned Aerial Vehicles (UAVs). It effectively visualizes the sequence of interactions based on the communication and it could be adjusted in the future accordingly. It should be emphasized that this is an initial design of the processes, outlining a chronological series of interactions, and the pathways of communication. However, the final communication protocols and the output format of each step will be clearly defined once the architecture and key requirements have been completed, so this estimate should not be considered as final, but an initial estimate.

Let's delve into the specifics:

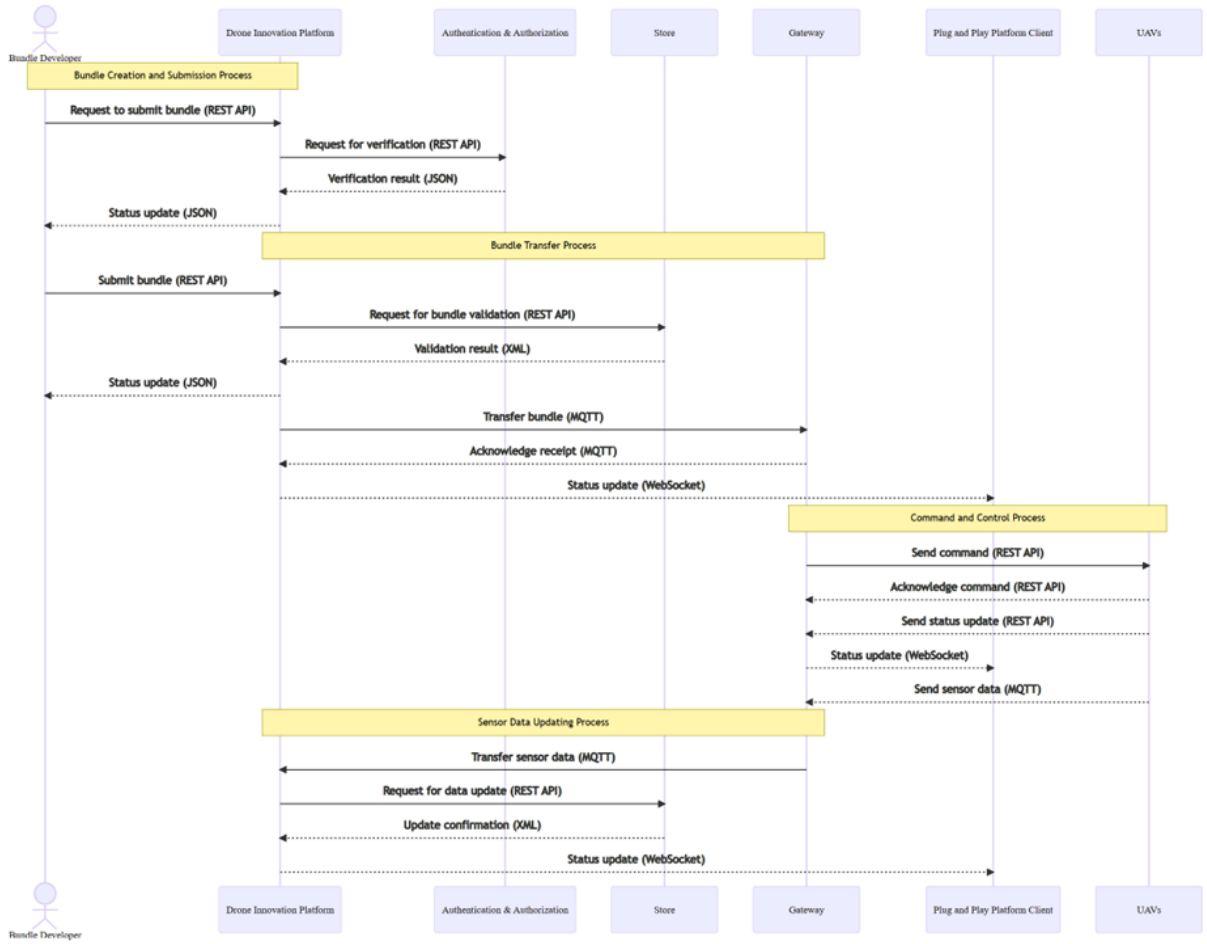


Figure 30: Integration Sequence Diagram based on Communication Pathways

Initially, we have clearly outlined the following four key processes which are fundamental to our system's integration design. These processes are instrumental in orchestrating the overall functionality and seamless integration of different components within our system:

1. **Bundle Creation and Submission Process:** The Command-and-Control Process represents the interactions that occur once a bundle is successfully transferred and acknowledged.

    a. **Request to Submit Bundle:** The sequence begins with the Bundle Developer initiating a request to submit a bundle to the Drone Innovation Platform, marking the start of the Bundle Creation and Submission Process.
    b. **Verification Request and Result:** Upon receiving the submission request, DIP forwards it to the Authentication & Authorization component to verify the Bundle Developer's credentials. The Authentication & Authorization component processes the request and returns a verification result to the Drone Innovation Platform.

c. **Status Update to Bundle Developer:** Based on the verification result, the DIP communicates a status update to the Bundle Developer. This status update informs the Bundle Developer whether the submission process can proceed.

2. **Bundle Transfer Process:** The Command-and-Control Process represents the interactions that occur once a bundle is successfully transferred and acknowledged.

    a. **Bundle Submission and Validation:** Once the bundle submission is authenticated and authorized, the Bundle Developer submits the actual bundle to the DIP, which then forwards this bundle to the Store for validation.
    b. **Status Update and Bundle Transfer:** The DIP communicates the validation status to the Bundle Developer. If the validation is successful, the DIP proceeds to transfer the bundle to the Gateway.
    c. **Acknowledgment of Receipt and Status Update:** The Gateway acknowledges receipt of the bundle, communicating this back to the DIP. Subsequently, the DIP updates the PPC regarding the status of the bundle transfer.

3. **Command and Control Process:** The Command-and-Control Process represents the interactions that occur once a bundle is successfully transferred and acknowledged.

    a. **Command Dispatch and Acknowledgment:** The Gateway dispatches a command to the UAVs, and the UAVs acknowledge this command. The UAVs also send a status update to the Gateway, indicating the state of the command execution.
    b. **Status Update and Sensor Data Transmission:** The Gateway communicates the command execution status to the PPC. Concurrently, the UAVs gather sensor data and transmit this data to the Gateway.

4. **Sensor Data Updating Process:** The Command-and-Control Process represents the interactions that occur once a bundle is successfully transferred and acknowledged.
    a. **Sensor Data Transfer and Update Request:** The Gateway transfers the sensor data to the DIP. The DIP, in turn, requests the Store to update its data using this new sensor data.
    b. **Update Confirmation and Final Status Update:** The Store confirms the successful data update to the DIP, which then relays this status to the PPC.

In essence, the sequence diagram outlines a chronological series of interactions among the entities, providing a clear understanding of the system's communication pathways, protocols, and operational flow. Each step in the sequence is critical to the overall process, from the bundle creation and submission to data update confirmation, and collectively, they form the backbone of the CHAMELEON integration system.

6.7 AGRICULTURAL DECISION SUPPORT SYSTEM

CHAMELEON ADSS (Agricultural Decision Support System), located in the Cloud infrastructure of CHAMELEON, will help visualize the different bundles developed under the scope of the project and enable the end users to make decisions.

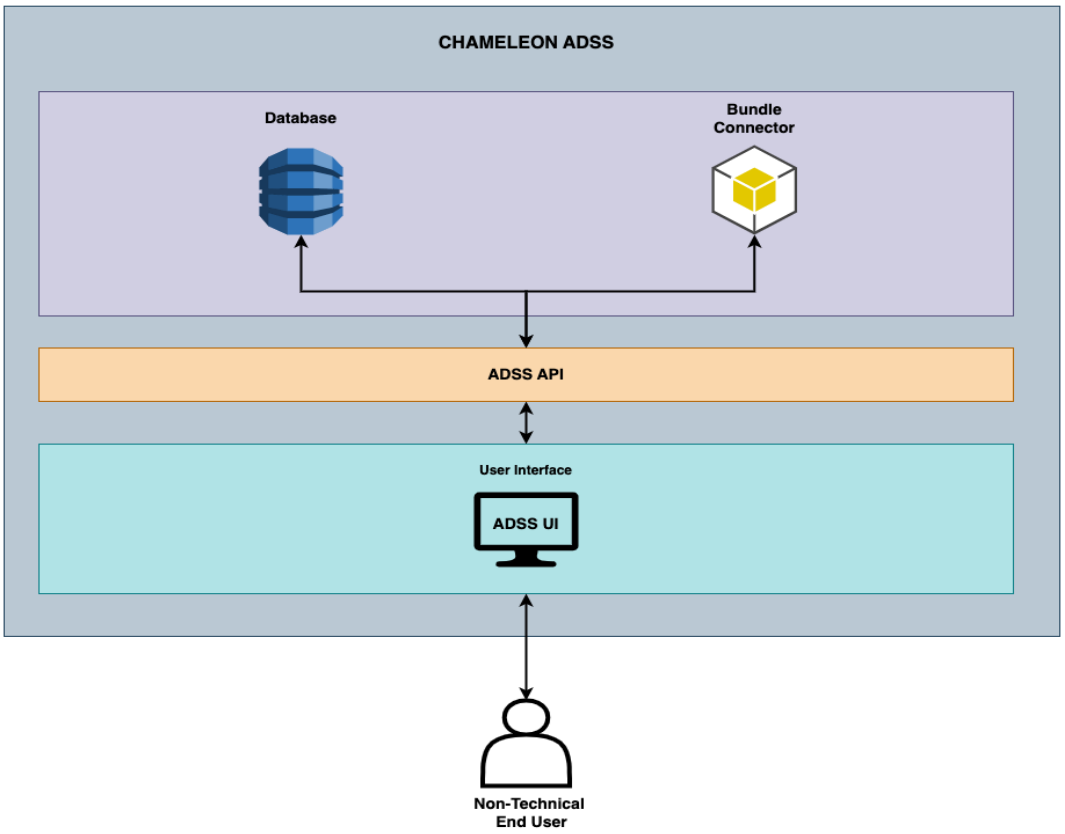


Figure 31: CHAMELEON ADSS Component Diagram

Focused on non-technical end users, the CHAMELEON ADSS is designed to provide a user-friendly graphical interface, presenting the outcome of the algorithms developed in the project's bundles and helping the farmers and livestock owners make decisions.

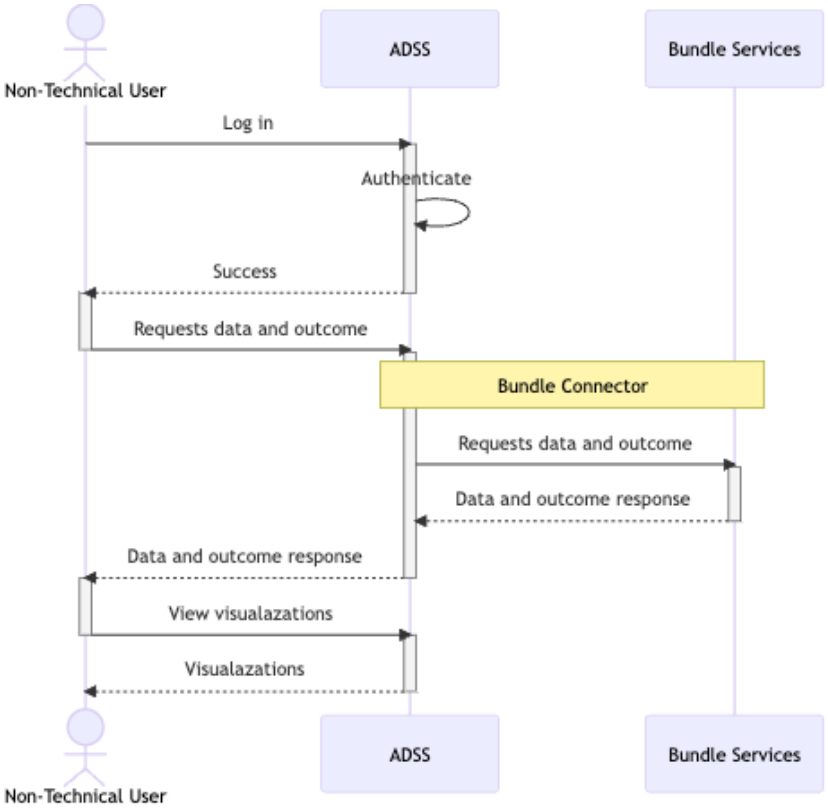


Figure 32: Visualization Sequence Diagram

A non-technical user after successfully authenticating, can request the available data and outcome from ADSS. The ADSS uses the bundle connector to connect to bundle services and request the end user's data and outcome. After the user receives the data and outcomes from the bundle can use ADSS to view the information.

6.8 PLUG AND PLAY PLATFORM

CHAMELEON Plug and Play Platform spans across the whole infrastructure of CHAMELEON by being composed of three main parts: the Plug and Play Server, Client, and Worker.

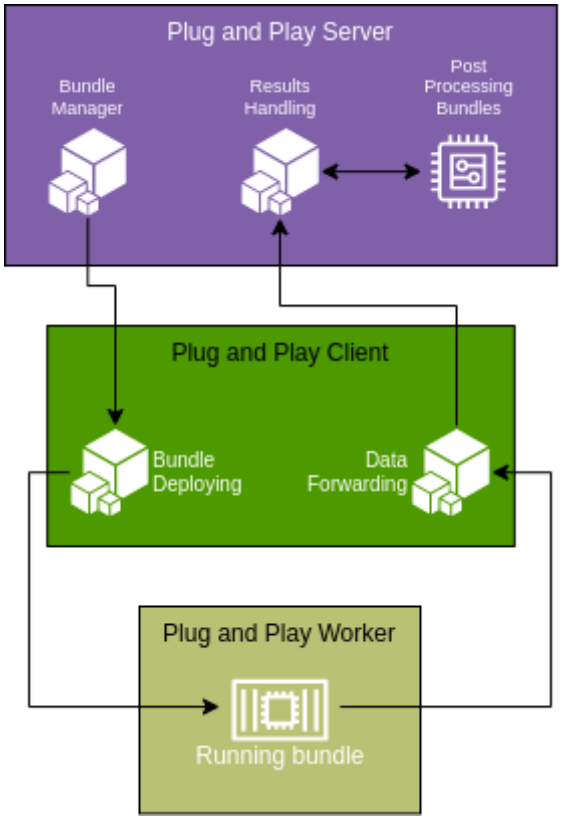


Figure 33: Plug and Play Platform Overall Diagram

- Server:

  The server instance of the CHAMELEON Plug and Play is located in the Cloud Infrastructure and is the main orchestration point of the whole component. It will contain the necessary functionality to support the following operations:

  - Dispatching the selected bundle to the Client instance:

    Once a bundle has been selected, the Server of Plug and Play platform shall disseminate it to the Client instance located at the Edge.

  - Handle incoming data from the Client:

    Running bundles produce results. These results, in order to be presented through the User Interface of CHAMELEON, must be handled by the Plug and Play Server.

  - Parse the incoming data to the running post-processing bundles:

    In some cases, data coming from the Edge to Cloud may require further processing before being presented to the User Interface of CHAMELEON. In such cases, Plug and Play Server shall send the received data for further processing before parsing them to be visualised.

- Client:

  The client of the CHAMELEON Plug and Play platform is located in the CHAMELEON Gateway and has two main objectives:

  - Data dissemination towards the Cloud:

    The results produced by the bundles running on-board on UAVs, in order to be presented to the interested end user, should be parsed to the Cloud infrastructure. Located to the Edge infrastructure and at the same time having connection to the Cloud, the Client instance of Plug and Play platform will be able to dispatch such Data to the Cloud infrastructure.

  - Handling and deploying the bundles coming from the Cloud infrastructure:

    Similar to the case of Data dissemination, due to the connection to both Cloud and Edge, the Client of Plug and Play platform will handle and deploy the bundles, coming from the Server instance to the Plug and Play worker.

- Worker:

The Plug and Play Worker is deployed on-board on the CHAMELEON UAV. By running on-board, the Worker can receive the bundles coming from the Client, to conduct their computations UAV internally.

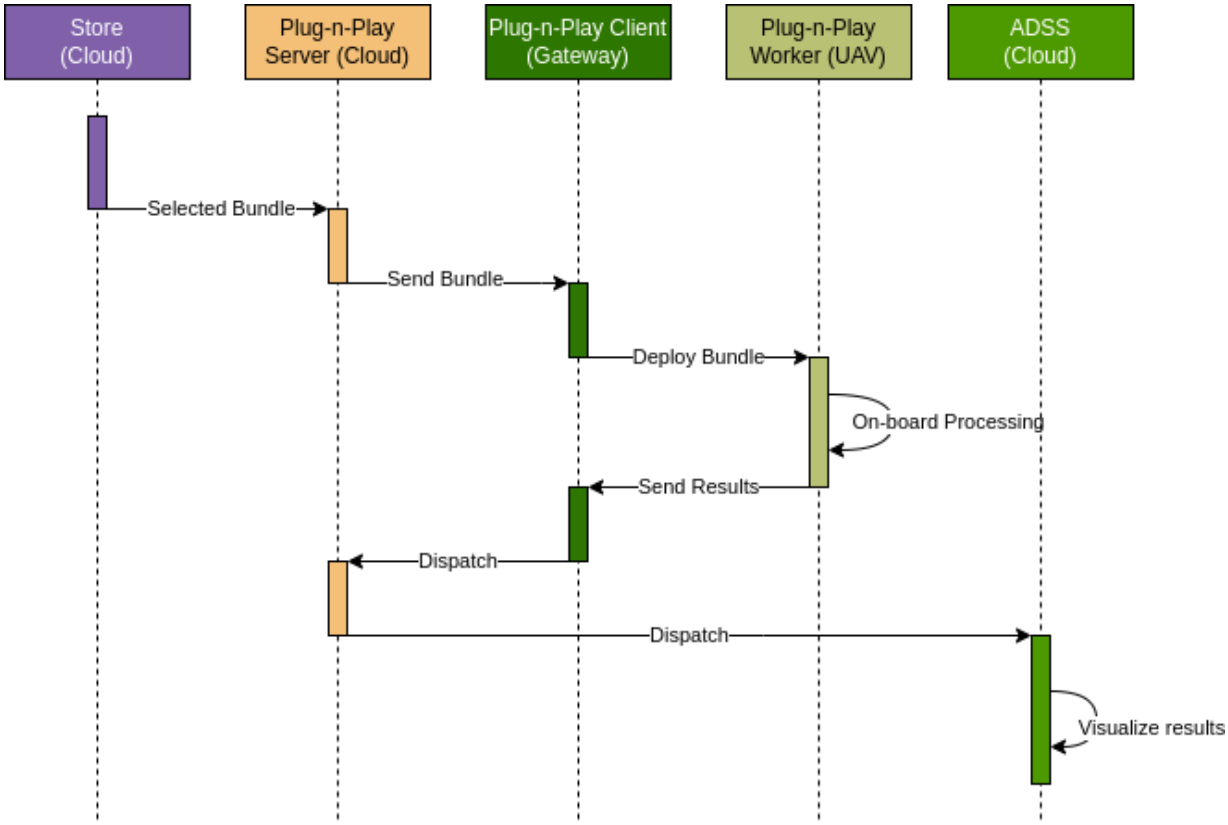


Figure 34: Plug and Play Platform On-Board Processing Sequence

The process of deploying a bundle is initiated through the CHAMELEON Store. The bundle selected is initially received by the Server instance of the Plug-n-Play platform located in the Cloud infrastructure of CHAMELEON. By handling the selected bundle, the Server of the Plug and Play platform can communicate it to the Client instance operating in the CHAMELEON Gateway. CHAMELEON Gateway, through established connection to the Physical layer of CHAMELEON, can further communicate the selected to be deployed to the on-board system of the UAV. Once the bundle has been successfully deployed, it can conduct its processing and produce the corresponding results. Conversely to the process of deploying the bundle, the results are returned to the CHAMELEON Gateway and are further pushed to the Cloud infrastructure, to be visualised by the responsible User Interface.

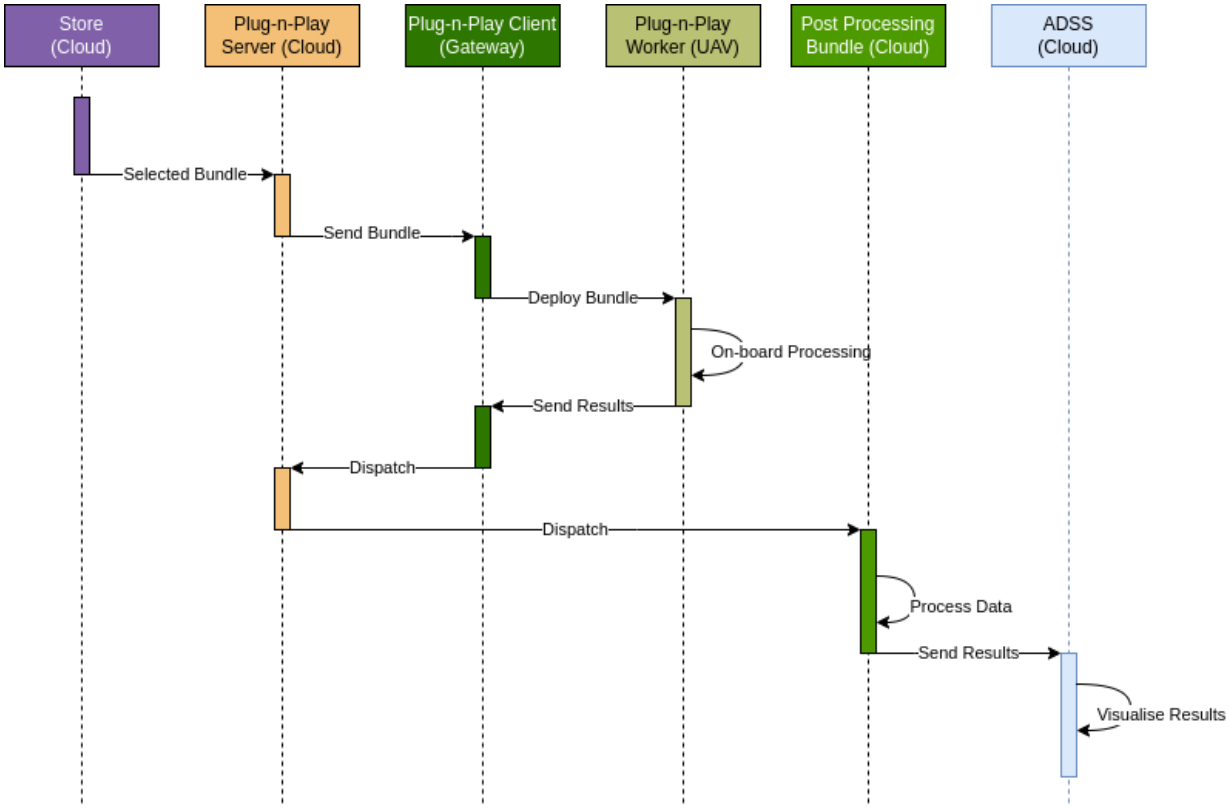


Figure 35: Plug and Play Platform Post-Processing Sequence

In the cases that a bundle requires results post-processing, a different approach is proposed to mitigate the hardware and software requirements necessary for such operation. The sequence is identical to the sequence presented in Figure 34: Plug and Play Platform On-Board Processing Sequence, however it deviates when the results are handled by the Cloud instance of the Plug-n-Play platform. In this case the results are not directly visualised since they require further processing. Thus, they are being pushed to the corresponding post-processing bundle running in the CHAMELEON Cloud. When the post-processing has been completed the results are visualised by the responsible User Interface of the Agricultural Decision Support System (ADSS).

6.9 CHAMELEON ON-BOARDING

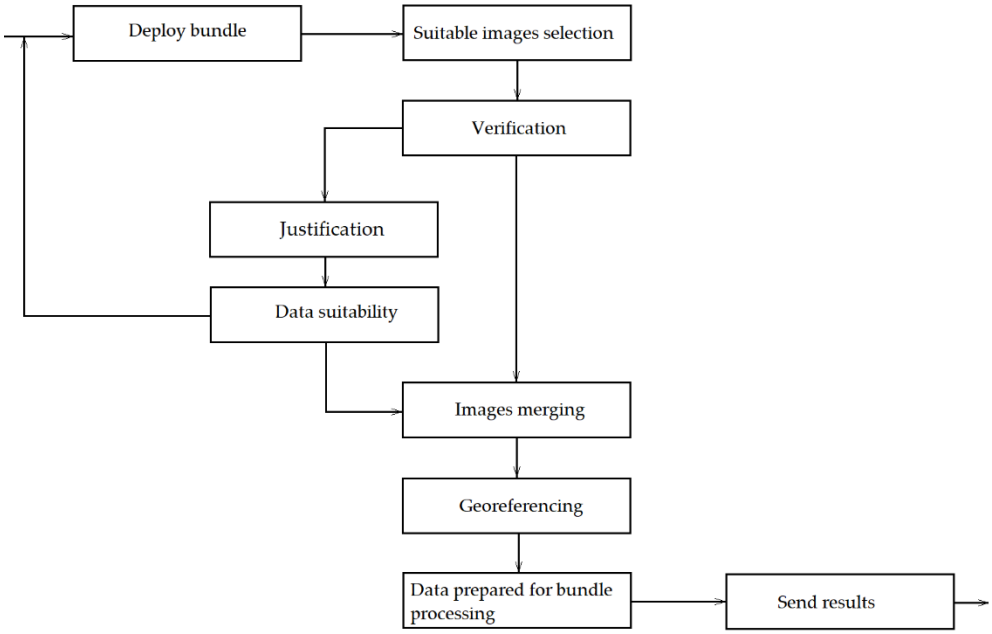


Figure 36: CHAMELEON On-boarding implementation view

The CHAMELEON On-Boarding is a process for images selection, images verification, images merging into orthomosaic, georeferencing, and preparation for the Bundle processing during the flight operation. The data from UAV sensor is sent into UAV Open Payload and then sent into UAV On-Board processing. The UAV On-Board processing is a set of operations. First of all, the verification of received images is performed. If the data turns out to be inappropriate, a data justification is made: errors and noises, unresolved seamlines and radiometric unbalanced color intensity, blank spot area of unknown error. If the error occurs the signal informs the drone operator to repeat the flight and re-collect previously unrepresentative data again. After repeated data collection or immediately after the first correct data collection, images are properly combined into an orthomosaic. The next step is orthomosaic georeferencing by using the geo-tagging (in-built GPS for navigation purposes). After this on-boarding process, the correct data is prepared for the execution of the respective bundles and transferred to the execution of further processes

6.10 UAVS

Remote piloted aircraft systems will be used to perform flights and retrieve all data needed for the end-user. The data acquired during flight will be processed in near real-time using AI algorithms to notify the user.

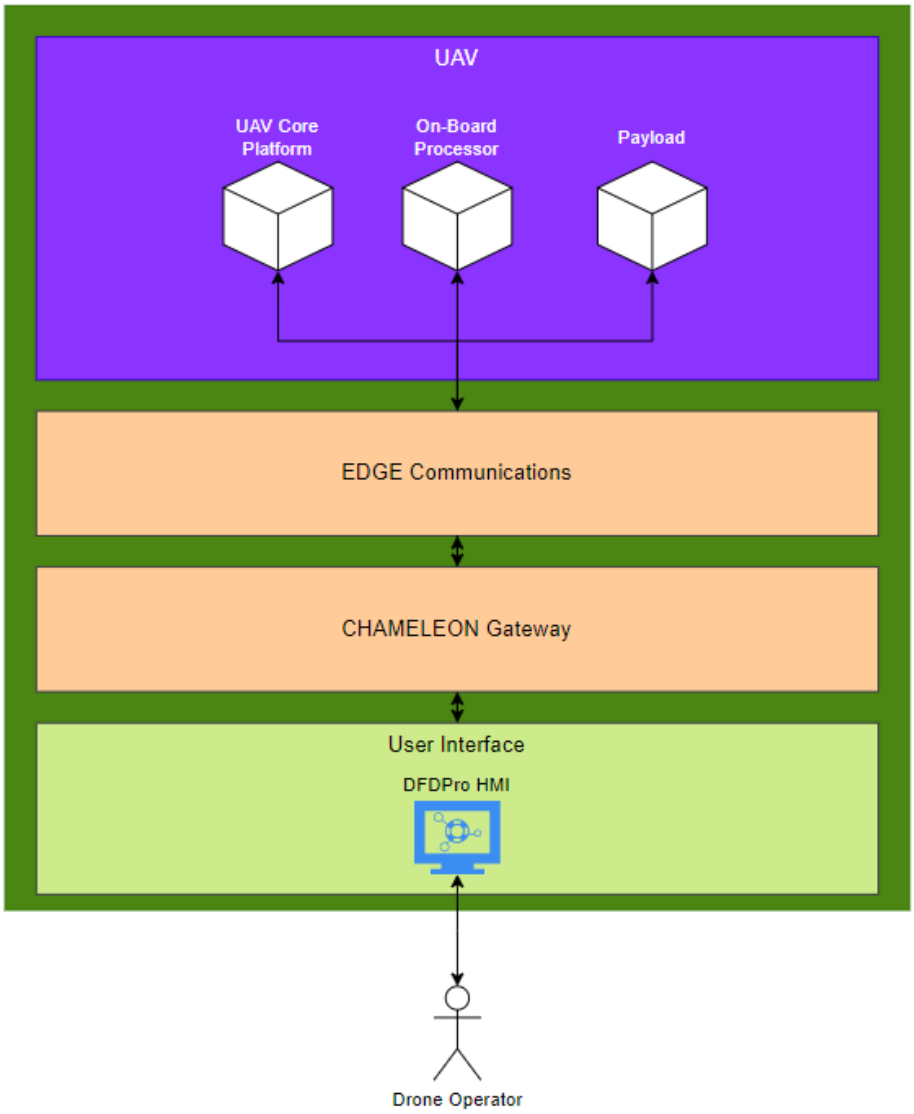


Figure 37: Remote piloted aircraft systems component diagram

The Delair Flight Deck Pro HMI will provide the necessary User Interface to enable the Drone Operator to manage flights and visualize data in real-time.

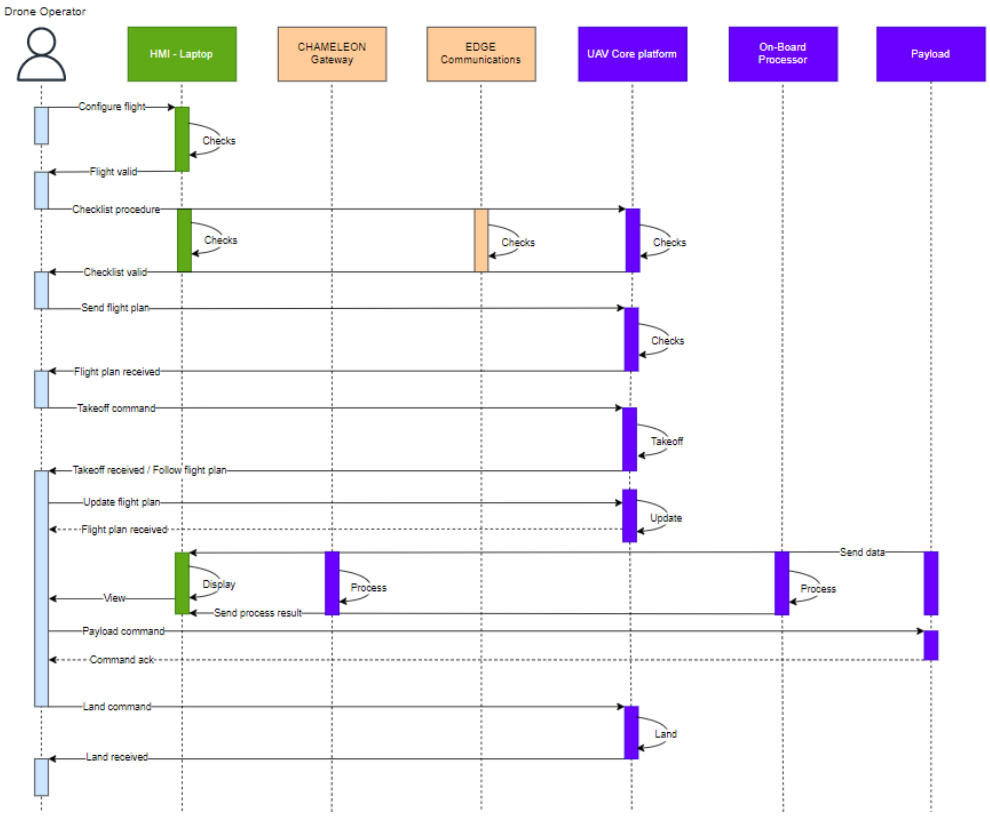


Figure 38: HMI / UAV communication diagram

Drone Operator configure the flight. The HMI (Delair Flight Deck Pro) checks the flight plan and displays it to the drone operator if it is valid. Then the drone operator can follow the checklist procedure. Each system does an auto-check. Once all checks are valid, the drone operator can send the flight plan. The UAV confirms the reception of the flight plan. The drone operator can send the take-off command. Once the UAV has taken off, it automatically follows the flight plan. The drone operator can update the flight plan at any moment. The payload sends the data (video, pictures) to the ground station. The data are processed in parallel by the On-Board Processor. The result is then sent to the HMI and displayed to the drone operator. It is also processed by the CHAMELEON Gateway and sent to the DIP. Once the drone has finished the flight plan or if the drone operator sends the land command, the UAV comes back and lands.

OPEN PAYLOAD TOOLSET

Through the Drone Innovation Platform (DIP), non-technical stakeholders will be able to easily choose a bundle to be deployed on their UAV. The Chameleon store will offer a filtering and recommendation system to guide end-users in selecting the most appropriate bundle and therefore the most suitable payload for their use case. This will be achieved by categorizing each sensor or hardware component, that will be available for mounting to the open payload toolset, into specific bundles based on their capabilities and the technical specifications. This

will ensure that the Chameleon ecosystem provides the end-users with the best match for their needs.

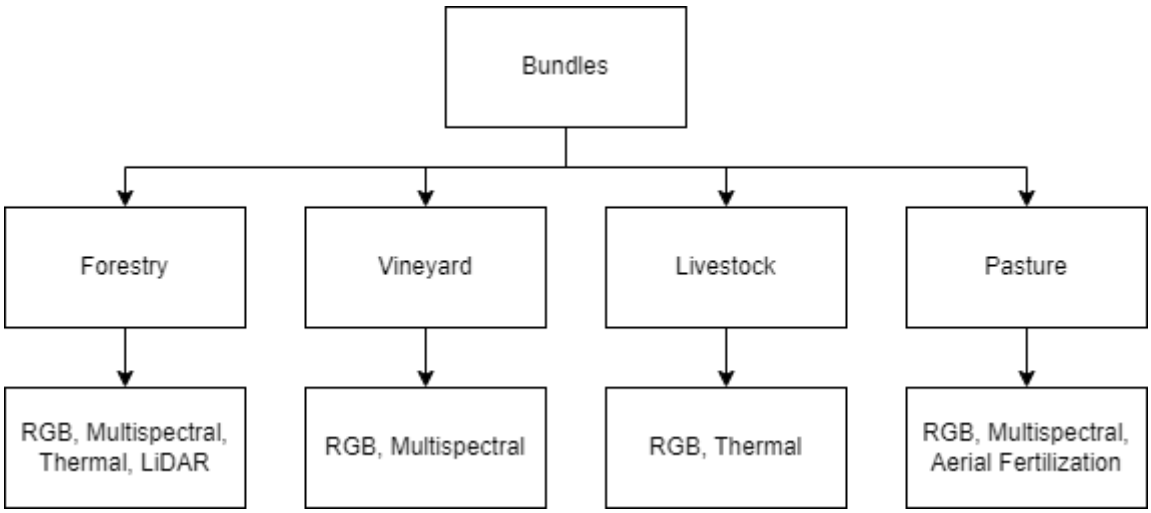


Figure 39: Categorization of payloads into different bundles

After the end-user has chosen a bundle, the request is dispatched to the CHAMELEON Plug and Play server, which initiates the deployment mode. This includes guiding the user with simple instructions and visuals to safely attach the payloads to the toolset, as well as providing the necessary information to operate the selected payload.

The open payload toolset will be an essential hardware component of the CHAMELEON UAV, providing a platform for hosting a wide range of payloads based on the needs of the end-user, such as a Vineyard Owner. The toolset will be designed with simplicity and ease-of-use in mind, featuring a secure plug-and-play procedure that even non-technical end-users can easily use to attach their selected payload to the hardware component.

6.11 HUMAN MACHINE INTERFACES

Delair's user interface (DFDPro) allows the drone operator to define, prepare and monitor UAV flights.

The user interface is defined as below:

- Mission scope definition:
  - Create or import areas and/or linears of interest.
  - Create or import No Fly Zones (areas where the UAV is not allowed to fly).

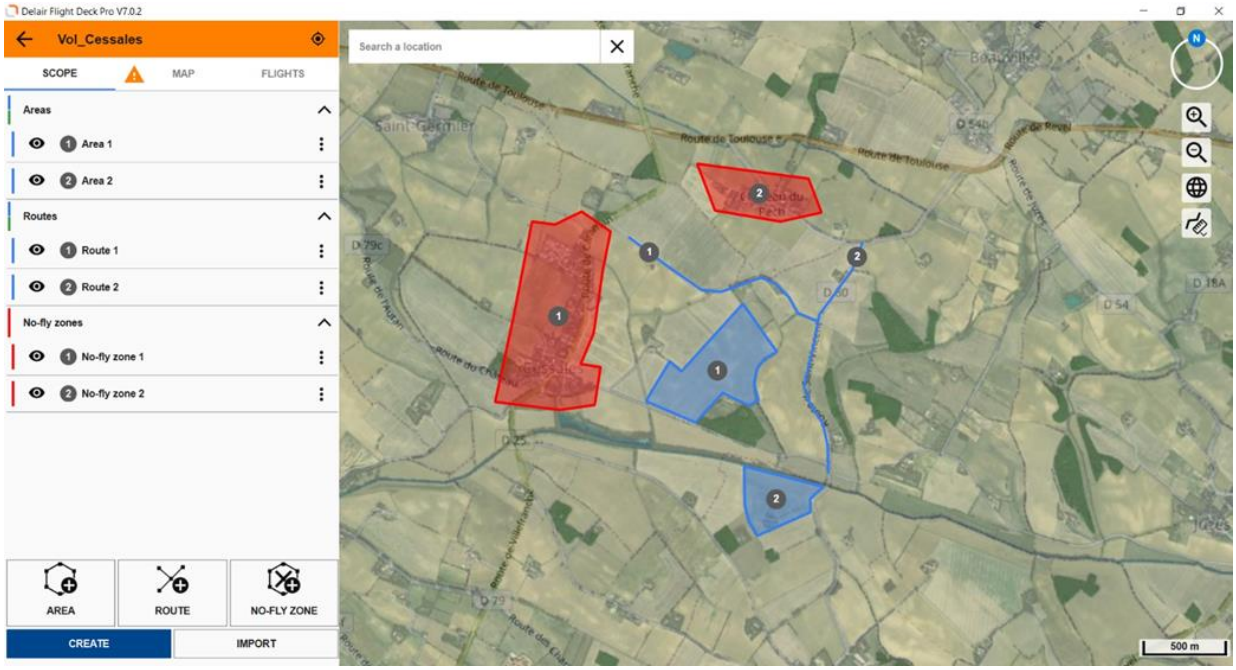


Figure 40: Mission scope definition

- Map definition:
  - Select and download background online maps for the defined scope.
  - Select and download DEM (Digital Elevation Model) for the defined scope.
  - Import custom DEM.
  - Import custom background maps.

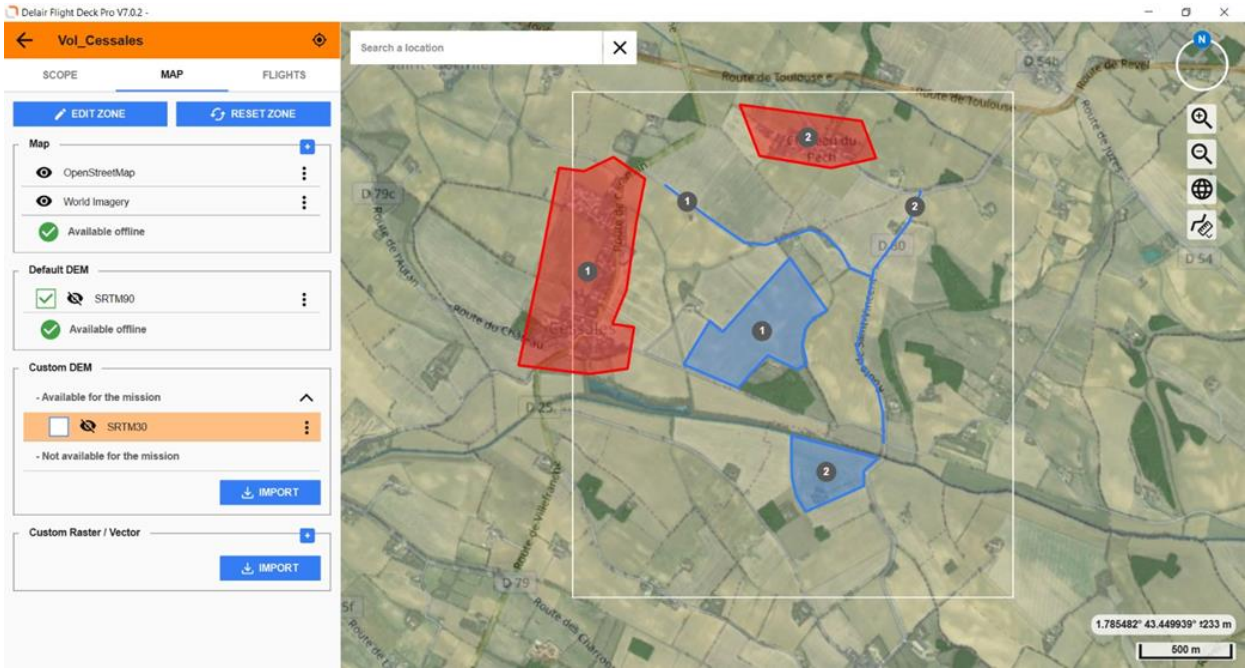


Figure 41: Map definition

- Flights preparation:

- Multiple flights can be prepared to cover all areas of interest defined in the mission scope.
- Defining some flight information (Name, Type, Pilot name…)
- Add a Geofence (circle or polygon)
- Selecting desired DEM for the flight
- Add and configure takeoff and landing location.
- Add and configure areas or linears defined in mission scope.
- Flight plan automatically generated from the inputs.
- Flight elevation displayed in a graph.

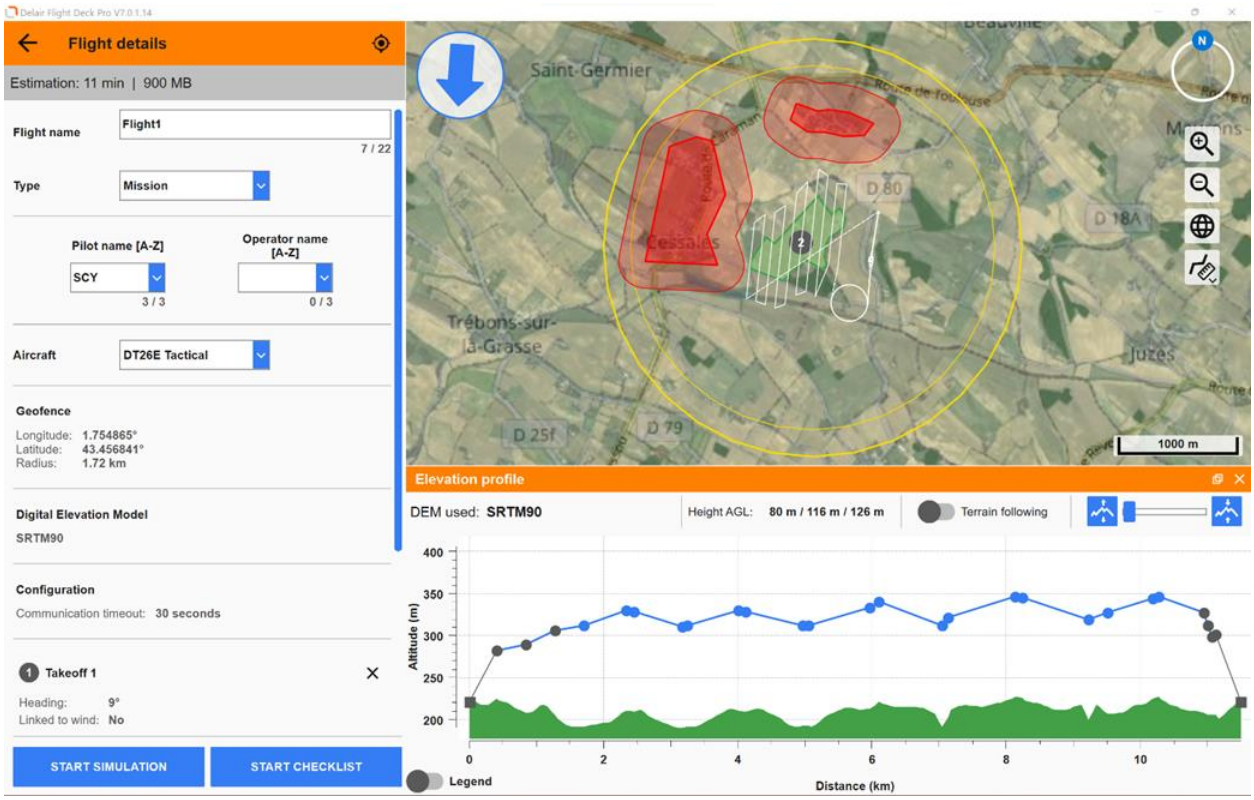


Figure 42: Flight preparation

- Pre-flight checklist:
  - List of all items to check (manually or automatically) to takeoff safe.

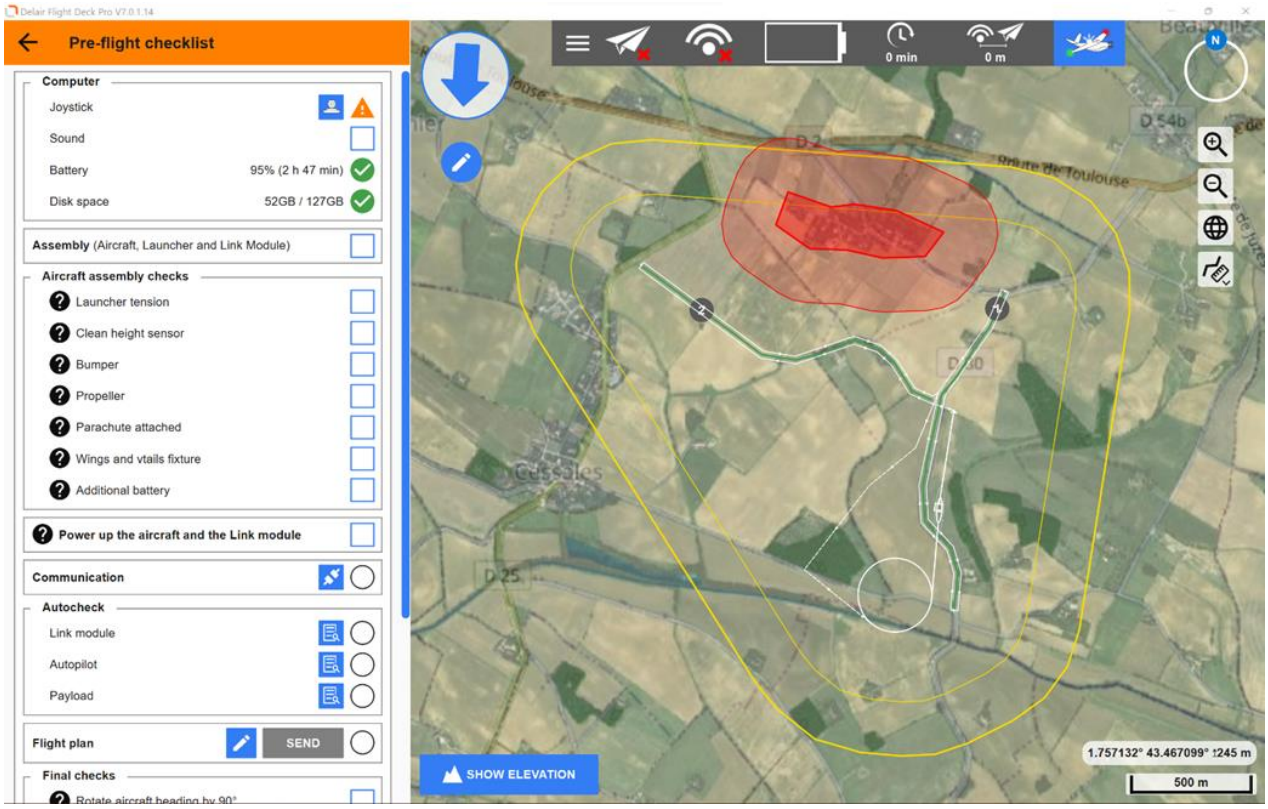


Figure 43: Pre-flight checklist

- Flight monitoring:
  - UAV trajectory and states
  - Commands to change UAV trajectory (mode Goto, Home, Land…)
  - Payload data in real time

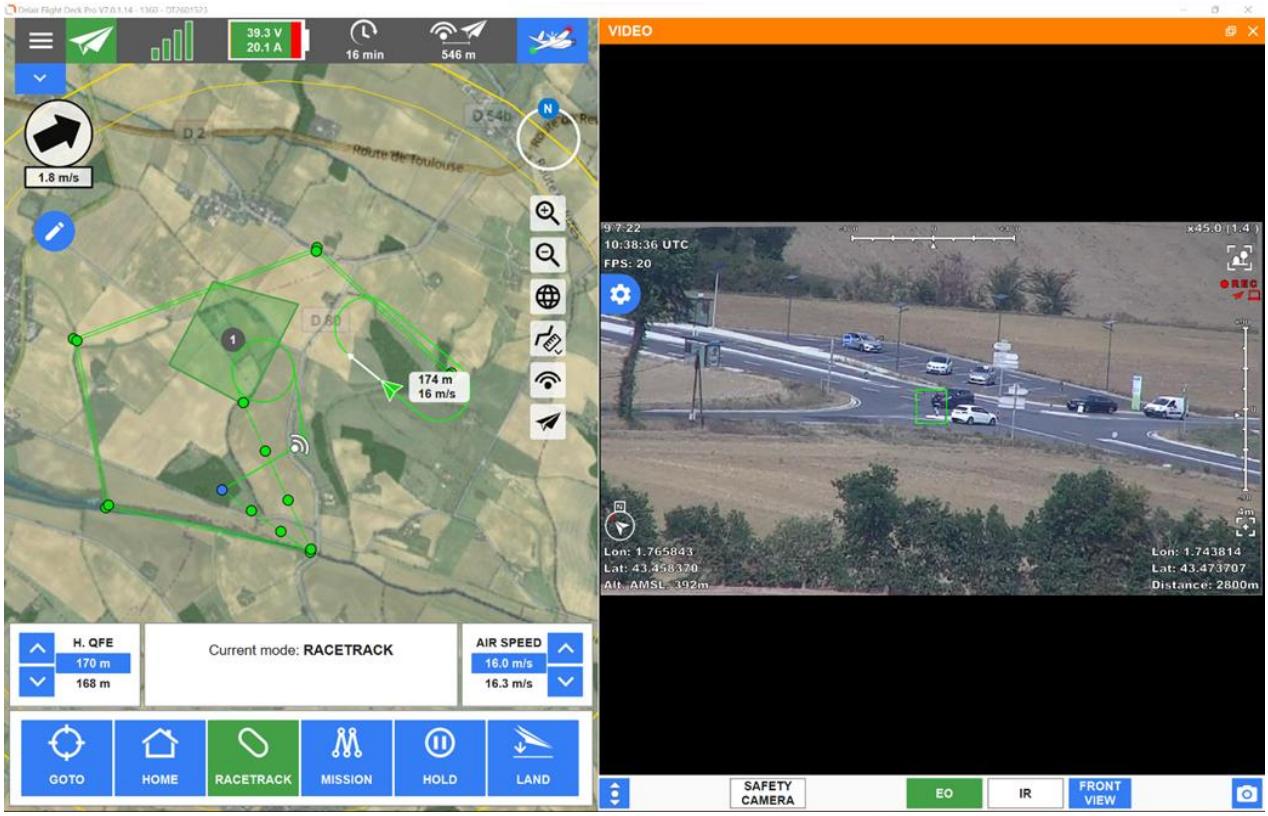


Figure 44: UAV in flight with video EO feedback

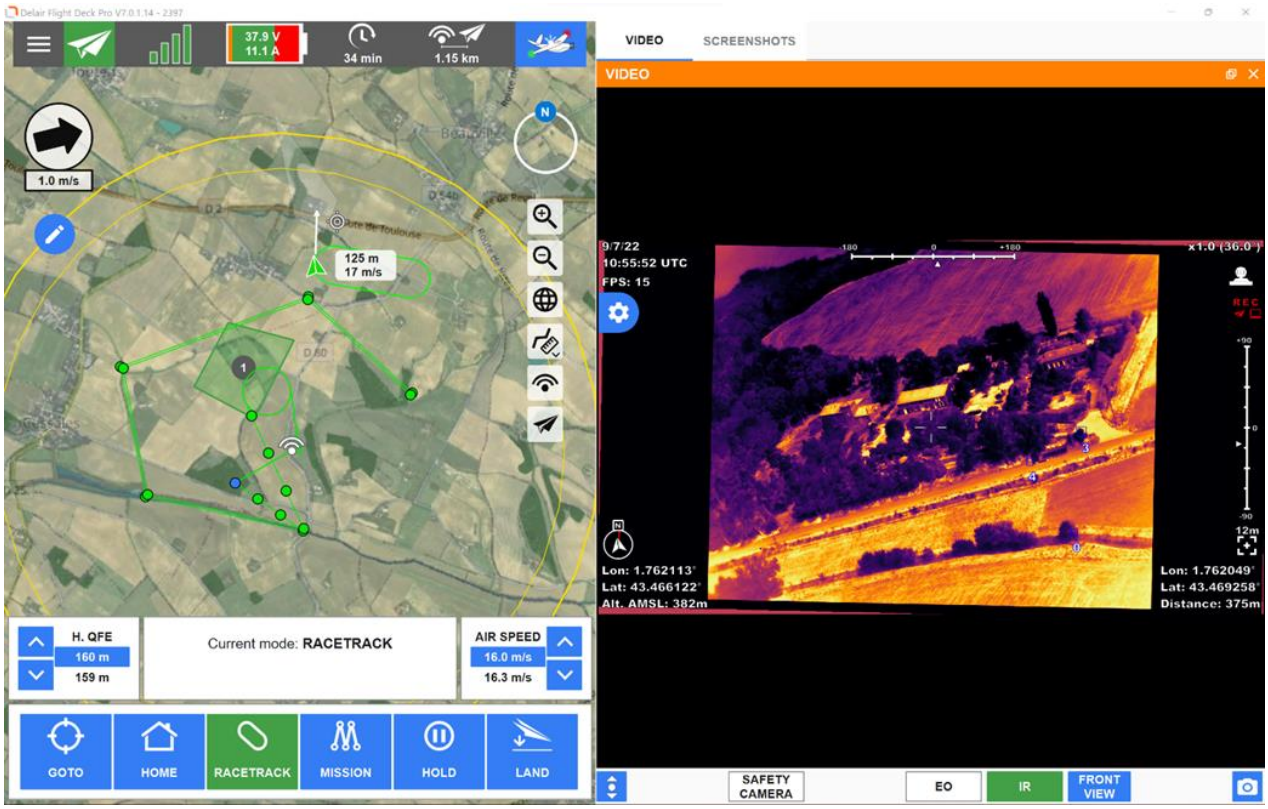


Figure 45: UAV in flight with video IR feedback

Figure 46: UAV in flight with camera picture coverage

Figure 47: UAV in flight with camera picture preview mapping

7 DEPLOYMENT VIEW

This chapter focuses on the requirements the CHAMELEON system has in order to be deployed. More specifically, the deployment view presents the following:

- **Software Deployment:** How software will be deployed (e.g., cloud microservices, containerized bundles etc.)
- **Hardware Requirements:** Hardware required for UAVs, Gateway, and Cloud platform.
- **Network Requirements:** Network connections required by the different layers of CHAMELEON.
- **High-level Deployment Diagram:** First version of the CHAMELEON Architecture

7.1 SOFTWARE DEPLOYMENT

In this section, the software deployment approach for the CHAMELEON System is presented. The components will be deployed as microservices that communicate with each other. Furthermore, the bundle deployment approach is explained.

7.1.1 CHAMELEON CLOUD MICROSERVICES

The Cloud components will be deployed as microservices in the Cloud infrastructure. Specifically, the DIP, the Store, the ADSS and the Plug and Play Server will be independent services, but they will communicate with each other. The DIP will be the main entry point for the stakeholders and bundle developers, therefore the Store and the ADSS frontend will be integrated in the DIP frontend in order to provide a seamless experience for the end-users. The Plug and Play Server will be a backend service that will communicate with the other microservices through an API.

7.1.2 CONTAINERIZED BUNDLES

The bundles that will run on the UAVs will be containerized using Kubernetes or another container orchestration system. This provides an easy way to deploy the bundles to any system, since they are inside a container that can run anywhere. Furthermore, the use of containers allows developers to package their bundles any way they want, and they can include third-party libraries and different versions, without the need to update or install packages.

7.1.3 CHAMELEON EDGE MICROSERVICES

The Edge components will also be deployed as microservices. On the Edge Gateway, there will be two backend microservices; the Plug and Play Client and the UAV/Sensors Communication. The Plug and Play Client will be a node of the container orchestrator used that will have knowledge of the UAVs present on the field and will be able to deploy the bundles. Lastly, the UAV/Sensors Communication microservice will contain a message broker that will be used for communication between the different parts of the Physical Layer, namely the UAVs and the Ground Sensors.

7.2 NETWORK REQUIREMENTS

The Edge Gateway must have an internet connection to communicate with the CHAMELEON Cloud. This is required because the bundles need to be deployed on the UAVs, through the Edge Gateway. Furthermore, the bundles send data back to the Cloud, so the network capacity should be enough to support these actions. On the other hand, the Gateway will communicate with the UAVs through a local network via a 3G/4G or Wi-Fi connection.

7.3 HIGH LEVEL DEPLOYMENT DIAGRAM

Figure 48 shows the initial version of the CHAMELEON Architecture. The figure shows the three main layers of CHAMELEON, namely the CHAMELEON Cloud, the CHAMELEON Edge, and the CHAMELEON Physical Layer. The edge communications are also shown. In the CHAMELEON Cloud (top rectangle) the DIP, the Store, the Plug and Play Server, and the ADSS are deployed. The CHAMELEON Edge is split into two parts: the Edge Gateway, and a device that will have the HMIs deployed. In the Edge Gateway, the Client part of the Plug and Play Platform and the UAV/Sensors Communication components are deployed. Likewise, the CHAMELEON Physical Layer is split into two parts: the UAVs, and the Ground Sensors. Both the UAVs and the Ground Sensors are hardware components deployed on the field. The UAVs will have the UAV core platform, the Plug and Play Worker, and the Open Payload Toolset deployed.

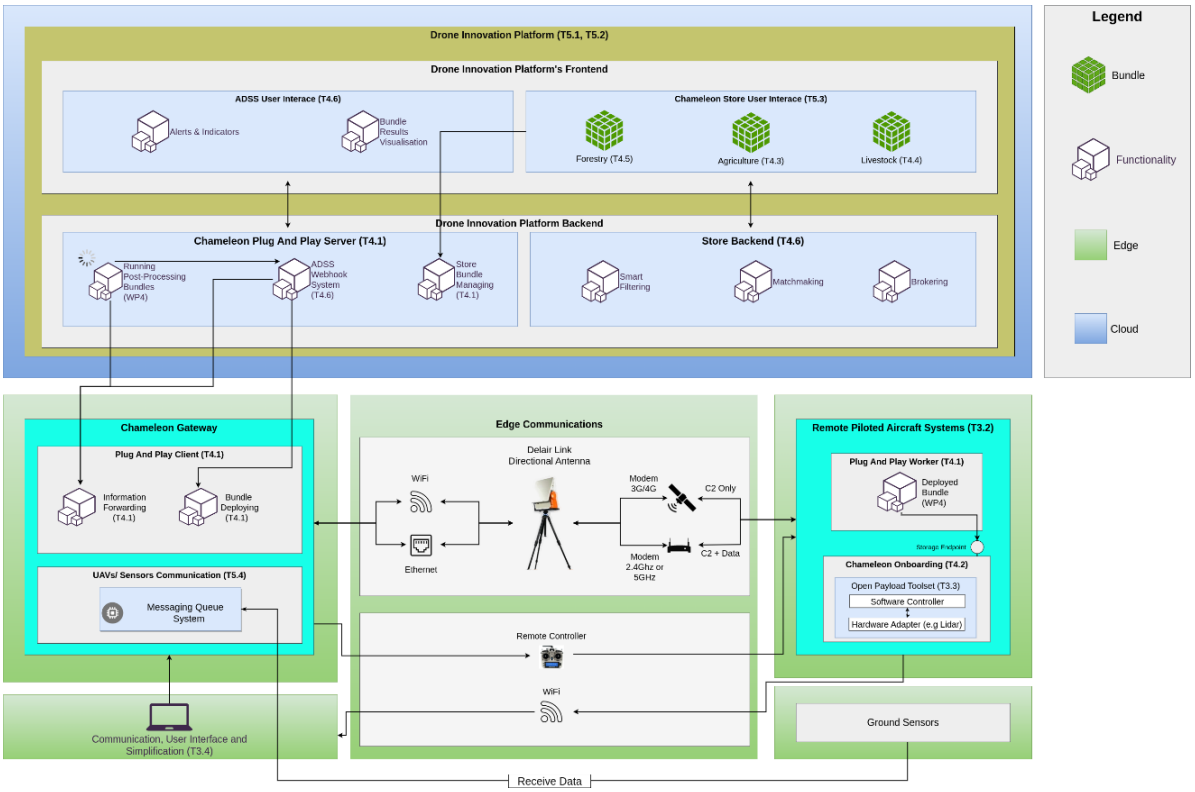


Figure 48: CHAMELEON Architecture

7.4 INTEGRATION WITH CHAMELEON DRONE ECOSYSTEM

For the purposes of the CHAMELEON project, the following initial high-level integration diagram that primarily depicts the major components and the communication method is presented. It also captures the user interfaces and the primary interaction scenarios.

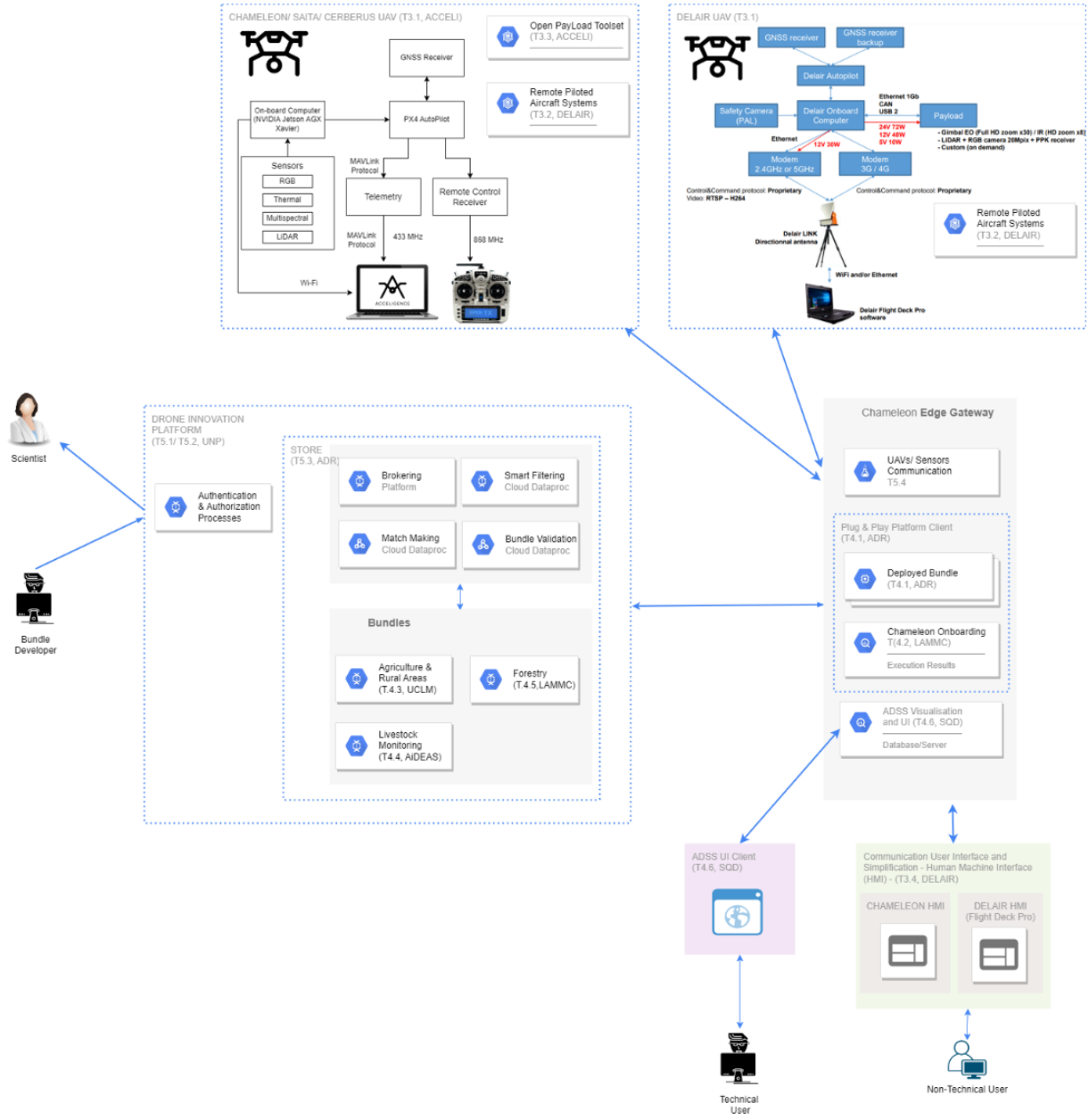


Figure 49: High-level integration diagram

The integration diagram for the project begins with the DIP, located on the lower left of the diagram. This platform serves as the entry point for bundle developers to submit their bundles and for scientists to receive data relevant to their interests. Each component of the diagram is annotated with the associated project task and the responsible parties.

The DIP is equipped with essential authentication and authorization processes to ensure secure and controlled access. DIP serves as the central hub for all interactions related to the development and submission of bundle applications. This platform is designed to facilitate interactions between different users, namely bundle developers and scientists. It is structured

in a way that allows developers to submit their applications and scientists to access the data they need seamlessly. The DIP's authentication and authorization processes ensure that access to the platform is restricted only to verified users, thus maintaining the platform's security and integrity. DIP includes the 'Store', an essential component that performs numerous functions. The Store is a dedicated component that handles multiple critical tasks. These tasks include hosting the bundles submitted by developers to ensure they remain available for deployment. The Store also provides brokering, smart filtering, and matchmaking services, which help managing updates and allow users to access specific data from other users' bundles. All these processes are conducted on our cloud server, which offers robust performance and reliability. Brokering, smart filtering, and matchmaking services are crucial for managing updates and facilitating access to specific data within the existing bundles.

The next stage in our diagram is the Chameleon Gateway Server. The Chameleon Gateway Server is an essential link between the cloud server and the drones. This server, deployed in the field, receives the bundles from the cloud server and initiates the Chameleon Onboarding process. The server houses the User Interface (UI) for Agriculture. This is a field-deployed computing device that receives bundles from the cloud server.

The Gateway Server also connects to the Unmanned Aerial Vehicles (UAVs). For these UAVs, we utilize the 'Open Payload Toolset' and 'Remoted Aircraft Systems', both of which are integral parts of our project. These systems enable the UAVs to carry out specific tasks associated with their mission. Once the bundles are deployed onto the drones, they collect and communicate the sensor data and flight results back to the Gateway Server.

Finally, the UAVs are where the bundles will be deployed, and they are responsible for feeding back sensor data and flight results to the Gateway Server. The UAVs serve as the endpoint in the data and control flow. They receive the bundles from the Gateway Server, deploy them, and relay the sensor data and flight results back to the server. This continuous two-way communication ensures smooth data transfer and facilitates swift response to emerging issues or changes in the operational environment. This constant communication loop ensures reliability in data flow and enables rapid response to any emerging issues.

This integration diagram, thus, presents a comprehensive and detailed picture of the project's structure, revealing the data flow, interdependencies, and interactions among the different components. It also highlights the key tasks associated with each component and the entities responsible for them. The diagram is instrumental in ensuring that all parties have a clear understanding of the project's operational mechanics and their individual roles and responsibilities.

8 COMPONENT INTERACTION VIEW

This chapter documents communication scenarios in the CHAMELEON ecosystem. Main architectural elements of CHAMELEON, as well as interactions between different components of CHAMELEON are documented through several end-to-end operations, namely: Bundle Upload, Bundle Selection and Deployment, Bundle On-Board Processing and Bundle Post-Processing. Scenarios depicted in this chapter are work in progress and their final version will be included in the final version of the overall CHAMELEON system architecture described in D2.4 CHAMELEON Technical Requirement and System Architecture v2.

The sequence of uploading a developed Bundle for the CHAMELEON Platform to make it available for deployment is one of the key operations of CHAMELEON.

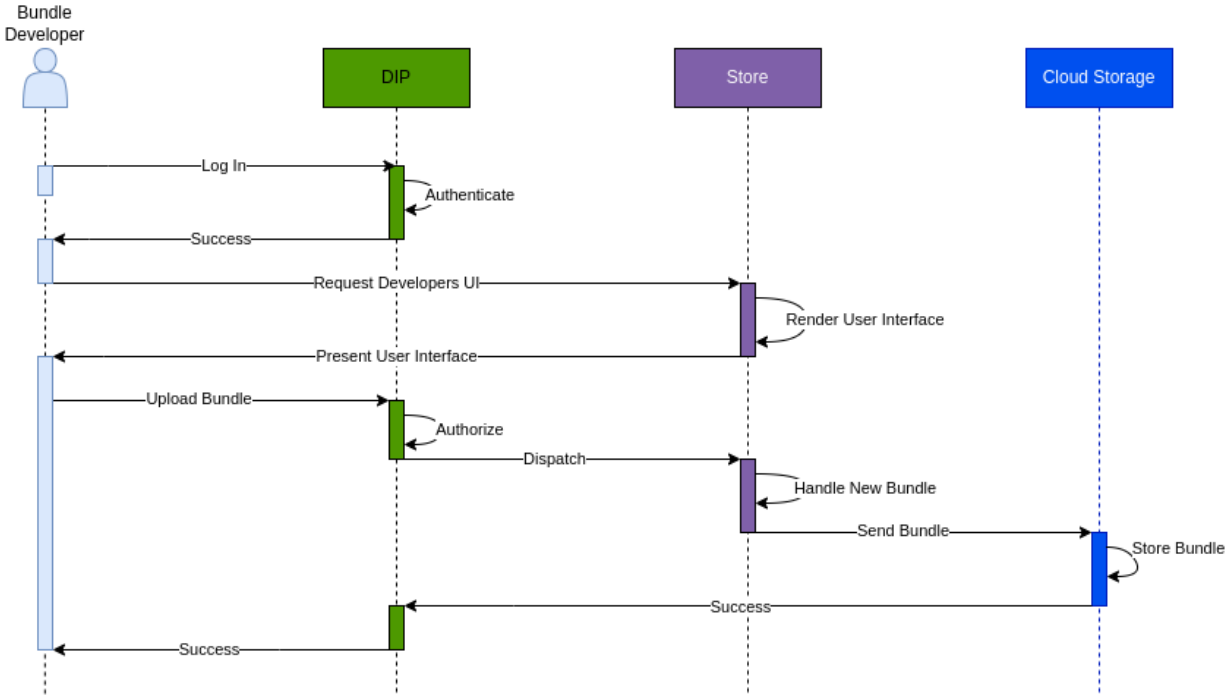


Figure 50: Bundle Upload Sequence Diagram

The operation of uploading a Bundle to the CHAMELEON Platform is the process that will allow the Bundle to be available for deployment. The process is initiated by the Bundle Developer. Since uploading content to the CHAMELEON Platform is an operation that requires both successful authentication and authorisation, DIP firstly undertakes these operations. Once the Bundle Developer has authenticated their identity and gained the relevant authorisation privileges, they can request the corresponding User Interface from the CHAMELEON Store. CHAMELEON Store then renders the User Interface designed specifically for such action. The Bundle Developer, through that Interface, uploads the developed bundle to the CHAMELEON Platform. The CHAMELEON Store handles and saves the Bundle to a dedicated Cloud Storage endpoint to make it accessible for the rest of the CHAMELEON Cloud infrastructure. Finally, a success message is being sent to the Bundle Developer to inform about the status of the whole sequence.

Bundles located in the Cloud Platform of CHAMELEON shall be available for the end-users to deploy them. The following sequence diagram depicts the operation of deploying a CHAMELEON Bundle.

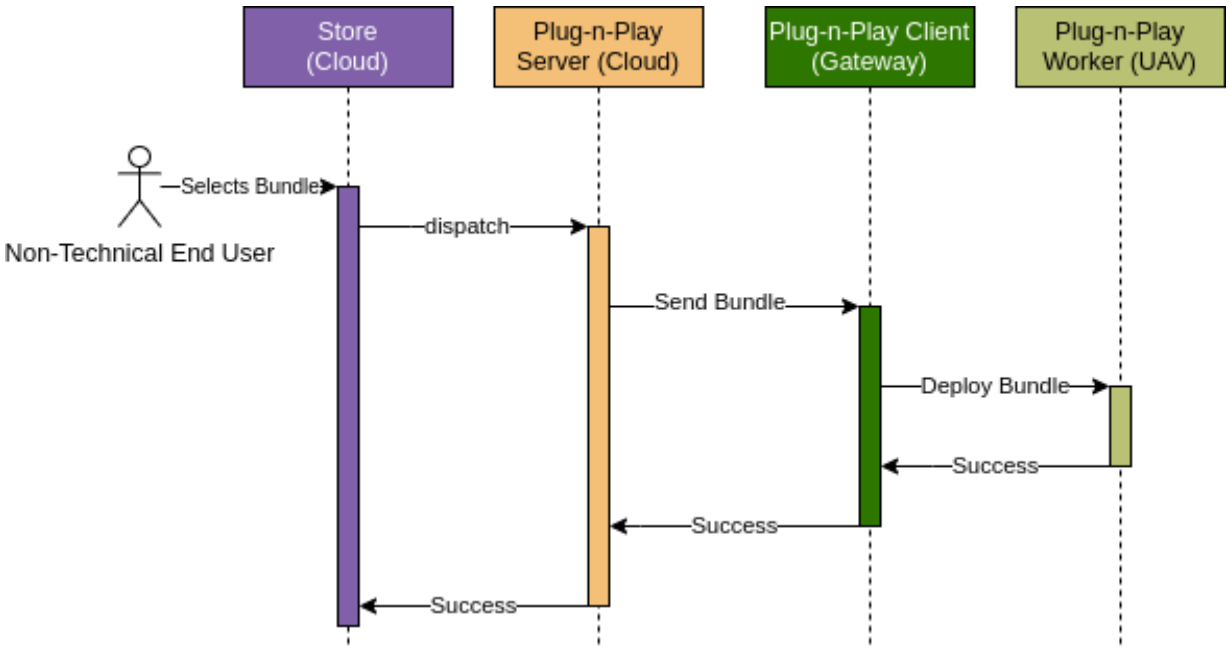


Figure 51: Bundle Selection and Deployment Sequence Diagram

In this scenario the sequence is initiated by a non-technical end user (e.g., Vineyard or Forest Owner). The CHAMELEON Store receives the instruction, described through the corresponding User Interface, to deploy a specific bundle. The details of the deployment are dispatched to the Server of Plug and Play Platform, located in the Cloud Infrastructure of CHAMELEON. By having connection with its Client instance, the Plug and Play Server communicates the Bundle and the relevant deployment instructions to the Plug and Play Client located in the CHAMELEON Gateway, in the Edge. The Bundle is then being sent to the Worker of Plug and Play platform, located on-board the UAV in the Physical Layer of CHAMELEON, before the success status of the whole sequence being communicated back to end-user.

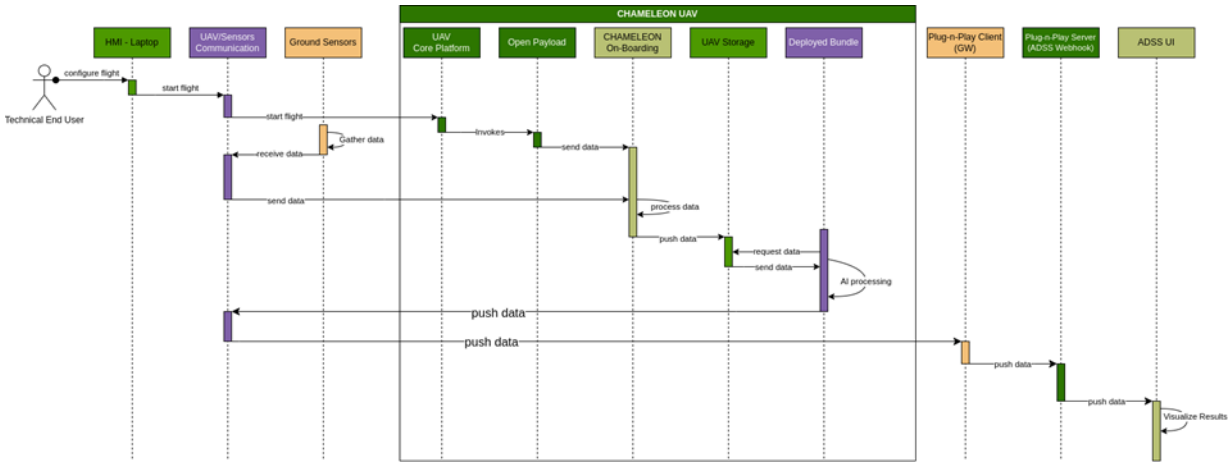


Figure 52: On-Board Processing Sequence Diagram

Once a CHAMELEON Bundle has been successfully deployed to the Worker instance of the Plug and Play Platform a Technical End User (e.g., Drone Operator) can describe the flight route of UAV through the corresponding User Interface. The UAV, by having its flight route defined, can initiate the flight and thus the information gathering through the hardware sensor attached to it (e.g., Lidar, RGB Camera). Information gathered through its sensor, coming in the form of stream of data, is being handled by the CHAMELEON On-boarding with the goal to structure the information to be exploited by the running Bundle. After that operation, the structured data are being saved in a storage endpoint UAV internally. The running Bundle, by having access to that endpoint, can retrieve this information and apply its constructed logic. When the Bundle has finished its computation, the results are being sent back to the UAV/Sensors Communication component, located in the CHAMELEON Gateway. The CHAMELEON Gateway, by having connection to the Cloud Infrastructure of CHAMELEON can forward the results to the corresponding component responsible for visualising the specific results.

In some cases, further time and resource consuming computations are required by the Bundles. Such computations, that require multiple hours, even days in some cases, will take place in the Cloud Infrastructure of CHAMELEON.

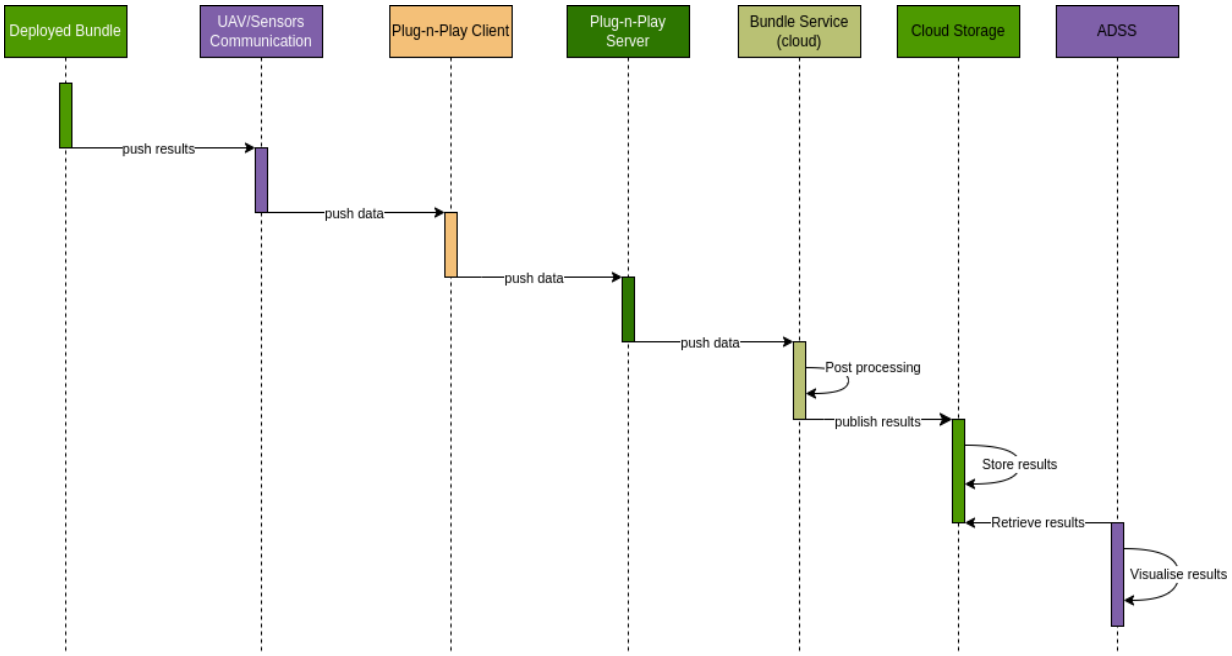


Figure 53: Post-Processing Sequence Diagram

In those cases, the same sequence for deploying the bundle is being followed as depicted in Figure 52. Deviation occurs when the results are produced by the running Bundle. In these cases, instead of directly visualising the results, they are being parsed, as input, to a Bundle that functions in the Cloud Infrastructure of CHAMELEON. After the Post-processing has finished the output is finally stored in a Cloud storage unit. Finally, ADSS retrieves the results and visualises them.

9 CONCLUSIONS

The main components and interactions of CHAMELEON as well as the overall system architecture have been described under the context of D2.2: Technical Requirements and System Architecture v1. This document utilized the outputs of T2.1 “Stakeholders Use Cases, Requirements And Workshop” and deliverable D2.1 “CHAMELEON conceptualisation, and use cases definition” as a base to design the CHAMELEON Platform’s architecture functional and non-functional requirements. Those requirements were used as a foundation for this first iteration of the architecture’s design.

This document will be used as a reference as the development and integration of the CHAMELEON ecosystem proceeds, since the conceptual view provided insights on the key entities and activities of CHAMELEON. The implementation view documented the implementation approach of each of the CHAMELEON solutions. The functional view defined the internal structure of each component. The Use Case view outlined the key sequences of the overall system and finally the deployment view presented the overall system architecture and requirements.

10 REFERENCES

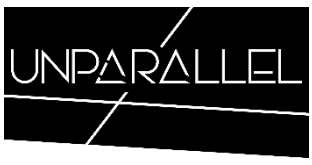
[1] Terven, J., & Cordova-Esparza, D. (2023). A Comprehensive Review of YOLO: From YOLOv1 to YOLOv8 and Beyond. arXiv preprint arXiv:2304.00501.

[2] Kruchten, Philippe (1995, November). Architectural Blueprints — The “4+1” View Model of Software Architecture. IEEE Software 12 (6), pp. 42-50.

[3] Nick Rozanski and Eoin Woods. 2012. Software Systems Architecture: Working with Stakeholders Using Viewpoints and Perspectives, second edition. Pearson Education Inc.

A Holistic Approach to Sustainable, Digital EU Agriculture, Forestry, Livestock and Rural Development based on Reconfigurable Aerial Enablers and Edge Artificial Intelligence-on-Demand Systems

The Members of the CHAMELEON Consortium:



Contact:

Project Coordinator: **Pantelis Velanas** Acceligence Ltd.	pvelanas@accelligence.tech

Disclaimer

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Executive Agency. Neither the European Union nor the European Research Executive Agency can be held responsible for them.