

WHITE PAPER

GMV Prodigy[®]

Code:	GMV-PRODIGI-WHP-001
Version:	1.0
Date:	April 20, 2026
Internal code:	GMV 22766/26 V1/26

DOCUMENT STATUS SHEET

Version	Date	Pages	Changes
1.0	April 20, 2026	38	First version

TABLE OF CONTENTS

1. INTRODUCTION	6
1.1. PURPOSE	6
1.2. SCOPE	6
1.3. DEFINITIONS AND ACRONYMS	6
1.3.1. Definitions	6
1.3.2. Acronyms	6
2. REFERENCES	8
2.1. APPLICABLE DOCUMENTS	8
2.2. REFERENCE DOCUMENTS	8
3. KEY ASPECTS OF GMV PRODIGI ®	9
4. MAIN COMPONENTS	11
5. PROCESSING PLATFORM	12
5.1. SCOPE	12
5.2. KEY FEATURES.....	12
5.2.1. Easy integration of processors	12
5.2.2. Flexible definition of processing chains	13
5.2.3. Priority Management	15
5.2.4. Configurable data ingestion	15
5.2.5. Unattended data-driven processing	16
5.2.6. Exhaustive processing monitoring	17
5.2.7. Adaptable data dissemination	20
5.2.8. Rolling or permanent data archiving	20
5.2.9. Mission-agnostic data cataloguing	21
5.2.10. Scalability	22
5.2.11. Interoperability.....	22
5.3. CONFIGURABILITY AND ADAPTABILITY	23
6. CUSTOMER SERVICES	24
6.1. SCOPE	24
6.2. KEY FEATURES FOR END USERS	24
6.2.1. Product catalogue browsing	24
6.2.2. On-demand processing	25
6.2.3. On-demand acquisition	26
6.2.4. Subscriptions.....	27
6.2.5. Services request	27
6.2.6. Federation hub	28
6.2.7. Shopping Cart	28
6.2.8. Specification of dissemination methods.....	28
6.2.9. Payments, billing and invoices	28
6.2.10. Monitoring orders status	29
6.2.11. Orders history	29
6.2.12. Special users	29
6.3. KEY FEATURES FOR OPERATORS.....	29
6.3.1. Manual approvals.....	29

6.3.2.	Monitoring orders	30
6.3.3.	User management	30
6.3.4.	Management of Ground Stations	31
6.3.5.	Restricted areas definition	31
6.3.6.	Enabling / disabling products in the catalogue.....	31
6.3.7.	Prices and discounts	32
6.4.	CONFIGURABILITY AND ADAPTABILITY	33
7.	DEPLOYMENT	34
7.1.	DEPLOYMENT SCENARIOS	34
7.2.	MULTI-SITE DEPLOYMENT AND DISASTER RECOVERY	34
7.3.	DEPLOYMENT ARCHITECTURE	35
7.4.	DEPLOYMENT TOOLS	36
8.	TECHNOLOGY STACK	37

LIST OF TABLES AND FIGURES

Table 1-1 Definitions.....	6
Table 1-2 Acronyms.....	6
Table 2-1 Applicable Documents.....	8
Table 2-2 Reference Documents.....	8
Table 5-1: Prodigy Catalogue Metadata.....	21
Figure 3-1: <i>Prodigi</i> drivers.....	9
Figure 3-2 Key Aspects of <i>Prodigi</i>	10
Figure 4-1: Prodigy Main Components	11
Figure 5-1 Workflow definition in <i>Prodigi</i>	15
Figure 5-2 Task definition in <i>Prodigi</i>	15
Figure 5-3: Ingestion configuration.....	16
Figure 5-4: Orders status	18
Figure 5-5: Workflow execution monitoring	19
Figure 5-6: Task execution monitoring	19
Figure 5-7: <i>Prodigi</i> gives access to the processor logs.....	19
Figure 5-8: <i>Prodigi</i> Dashboard	20
Figure 5-9: Prodigy offers a Swagger interface to its API.....	23
Figure 6-1 Browse catalogue of existing products	25
Figure 6-2 Catalogue filter parameters.....	25
Figure 6-3 On-demand processing	26
Figure 6-4 Search new acquisitions.....	26
Figure 6-5 Create subscription	27
Figure 6-6 Service request	28
Figure 6-7 Orders status	29
Figure 6-8 Monitoring dashboard	30
Figure 6-9 User details.....	30
Figure 6-10 List of available ground stations.....	31
Figure 6-11 Restricted areas.....	31
Figure 6-12 Catalogue Management Page, Area List.....	32
Figure 6-13 Prices list	32
Figure 6-14 Discounts list.....	33
Figure 7-1: Deployment scenarios of <i>Prodigi</i>	34
Figure 7-2: Typical deployment over infrastructure.....	35

1. INTRODUCTION

This document presents the key capabilities of the **GMV Prodigy®** product, GMV's solution for payload data ground segments.

1.1. PURPOSE

This document is intended to give a detailed description of the product capabilities and available components to potential customers or users.

1.2. SCOPE

The description covers the different capabilities of the **GMV Prodigy®** product and its components, including optional components.

1.3. DEFINITIONS AND ACRONYMS

1.3.1. DEFINITIONS

Concepts and terms used in this document and needing a definition are included in the following table:

Table 1-1 Definitions

Concept / Term	Definition

1.3.2. ACRONYMS

Acronyms used in this document and needing a definition are included in the following table:

Table 1-2 Acronyms

Acronym	Definition
AMQ	Active Message Queue
API	Application Programming Interface
AWS	Amazon Web Services
CADU	Channel Access Data Unit
CFI	Customer Furnished Item
COTS	Commercial Off-The-Shelf
CPU	Central Processing Unit
DEM	Digital Elevation Model
FTP	File Transfer Protocol
FTPS	File Transfer Protocol Secure
GDAL	Geospatial Data Abstraction Library
GIPP	Ground Image Processing Parameters
GIS	Geographic Information System
GPU	Graphics Processing Unit
LTS	Long-Term Support
LWIR	Long-Wave Infrared
MIME	Multipurpose Internet Mail Extensions
MWIR	Mid-Wave Infrared
OGC	Open Geospatial Consortium
PAN	Panchromatic

Acronym	Definition
PDGS	Payload Data Ground Segment
RAM	Random Access Memory
REST	Representational State Transfer
SSO	Single Sign-On
STAC	SpatioTemporal Asset Catalog
URI	Uniform Resource Identifier
UTC	Coordinated Universal Time
UUID	Universally Unique Identifier
WKT	Well-Known Text

2. REFERENCES

2.1. APPLICABLE DOCUMENTS

The following documents, of the exact issue shown, form part of this document to the extent specified herein. Applicable documents are those referenced in the Contract or approved by the Approval Authority. They are referenced in this document in the form [AD.x]:

Table 2-1 Applicable Documents

Ref.	Title	Code	Version	Date
[AD.1]				
[AD.2]				
[AD.3]				
[AD.4]				

2.2. REFERENCE DOCUMENTS

The following documents, although not part of this document, amplify or clarify its contents. Reference documents are those not applicable and referenced within this document. They are referenced in this document in the form [RD.x]:

Table 2-2 Reference Documents

Ref.	Title	Code	Version	Date
[RD.1]				
[RD.2]				
[RD.3]				

3. KEY ASPECTS OF **GMV Prodigy**[®]

GMV Prodigy[®] encompasses a suite of products aimed at being a full, end-to-end, modular, complete solution for Payload Data Processing in Ground Segments. The different **Prodigi** components can be deployed independently, in scenarios where part of the features is not required. **They comprise a set of highly customizable Production Ordering, Data Ingestion, Processing, Dissemination, Archival, Cataloguing, Monitoring and Billing services that can be used as-is or tailored for the customer needs.**

GMV is qualified as **Processing Services Provider for the Copernicus Space Components** with **Prodigi**, integrating processors used in the Copernicus constellation (Sentinel-1, Sentinel-2 and Sentinel-3 Land). It is the solution selected by the **MBRSC** (Mohammed Bin Rashid Space Centre in Dubai) for the high-resolution **MBZ-Sat** optical mission to process and make available L0, L1A, L1B, L2 and L3 products. It is also the solution selected for an optical **constellation mission** that will provide L0 and L1A products to the **Polish MoD**. It is therefore **ready for operations** with engineering adjustments or adaptations for the specific concept of operations and interfaces of the mission.

Prodigi follows a set of drivers related to the performance, to the ease-of-use, to reliance on state-of-the-art technologies and to be agnostic of the mission through adaptability.



Figure 3-1: Prodigy drivers

Those drivers directly define key aspects of the product:

Multi-mission, multi-satellite, multi-level Prodigi is suitable as a multi-mission PDGS solution for any type of Earth observation mission, capable of generating any product level.	Data-driven and on-demand Full automation of processing tasks is achieved via automatic monitoring of input channels and rule-based workflow instantiation. Prodigi supports systematic unattended processing, on-demand processing (generation of products from inputs already existing, requested by users) and reprocessing campaigns.
Processor independence Prodigi allows the easy integration of any processor or quality control step as an external CFI by using containerization technologies. Container orchestration allows the best utilization of resources.	High Availability and Reliability Introduced as design key drivers from the start.

Scalability The system is natively designed to be deployed in a cloud environment following a decoupled micro-services architecture, providing transparent and automatic horizontal scalability up to hundreds of processing nodes according to system load and processing requirements, able to cope with high demands. It can run operationally a PDGS from 1 to any number of infrastructure available nodes.	Cloud native Prodigi has been designed to be deployed in a cloud environment but does not depend on any specific cloud service. It has been tested in AWS and OVHcloud. The design and technology choices make Prodigi also independent of the underlying hardware or VM structure. This flexibility allows tailored configuration in bare-metal environments for those use-cases where processing requirements are very demanding, and virtualization is not an option.
Adaptability Given the heterogeneous nature of the mission processing needs, Prodigi is highly customizable minimizing its modifications, allowing mission specific logic, using different product and imagery formats or interfaces to be easily connected to the system through de-facto protocol standards.	Security Access to Prodigi functions is controlled using its identity system. This authentication and authorization service, based on OpenID Connect SSO access tokens, provides granular role-based control to system resources, enabling administrators the possibility to assign permissions and revoke access to users, roles or groups.

Figure 3-2 Key Aspects of *Prodigi*

4. MAIN COMPONENTS

Prodigi is made of two main components, that can be deployed together to cover the full end-to-end needs of a PDGS, or separately in other scenarios: the **Prodigi Processing Platform** and the **Prodigi Customer Services**.

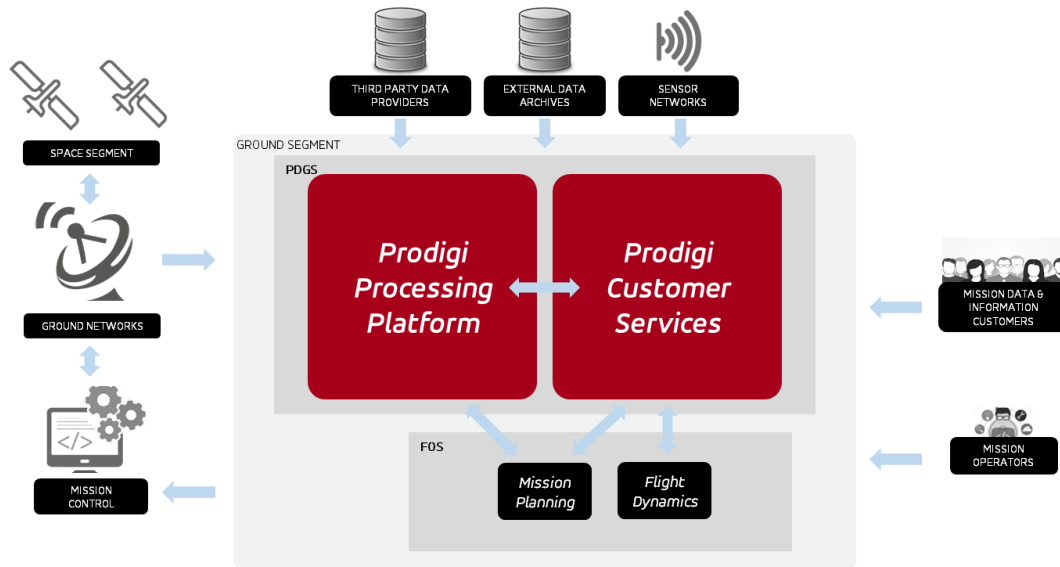
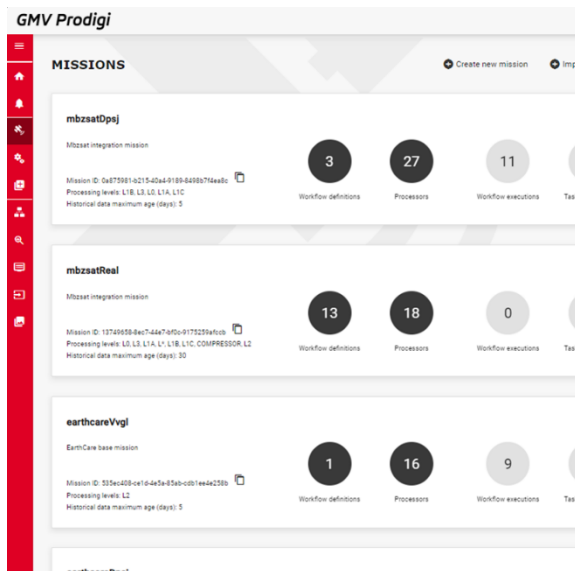


Figure 4-1: Prodigi Main Components

5. PROCESSING PLATFORM

5.1. SCOPE



The **Processing Platform** receives the **input data** to be processed (typically raw data from the antenna) and the **auxiliary data** (e.g. DEM, GIPP) needed for its processing. It is responsible for generating mission **products** of any processing level, ensuring completeness and under specific timeliness constraints, in an automated **data-driven** way.

The Processing Platform includes essential functions of the processing system: ingestion, archive, catalogue and dissemination.

5.2. KEY FEATURES

The key features that the **Prodigi Processing Platform** brings to the PDGS are the following:

5.2.1. EASY INTEGRATION OF PROCESSORS

The **Prodigi Processing Platform** can integrate the algorithms that perform generation of higher-level products from acquired data. Several processors or processor versions can be made available to the platform, at any time. This integration is based on few requirements:

- Processors must be containerised in runnable docker images (using 'docker run' commands). They are considered as black boxes by the platform (can be obfuscated, even). Only knowledge about how to run the processor will be required.
- Information about the resource requirements of the processor (RAM, CPU, etc) to carry out their activities within the required performance parameters, is needed.
- Processors must operate on file-based storage, provided by the platform. The structure of inputs, auxiliary files and outputs must be known so that the platform can provide it to the processor. Selection policies for auxiliary data (e.g. validity period of the auxiliary data must cover the validity period of the input) must be known. Understanding the data dependencies between processors will help defining processing chains.
- If there are policies to comply with when selecting input data (e.g. an auxiliary file must have a validity period covering the validity period of the main input), they must be known.
- Processing parameters will be made available to the processor through arguments, environment variables or configuration files (e.g. Job Orders or any other format), especially to specify paths to the inputs, outputs, etc.
- Processors must generate logs, either in specific log files or through the standard output, describing the progress and the status, the problems found, and any other information useful to troubleshoot problems when executing the processor.
- Processors must exit with known exit codes and their meaning (success or failure) must be declared.

- Documentation about how to chain the execution of processors, and specifications of the products, must be available.

Processors of several sources (developed by GMV or third parties) have been successfully integrated in the **Prodigi Processing Platform**: Copernicus processors for Sentinel-1 L1, Sentinel-2 L0 and L1C; Sentinel-3 (land) L0, L1 and L2; EarthCARE L2; MBZ-Sat L0, L1A/B/C, L2, L3; Polish MoD optical constellation L0, L1A; added-value fusion processors such as crop type classification, methane detection, vineyards status...

This approach ensures the capability to integrate custom workflow components, written in any programming language (Java, Python, C++), using any library (GDAL, GeoPandas) to fulfil the needs of the processing.

5.2.2. FLEXIBLE DEFINITION OF PROCESSING CHAINS

Multiple data-driven processing chains (workflows) can be defined, making use of the different processors available for the different steps (tasks) in the chain, considering data dependencies between them. Processing chains can be reconfigured at any moment without having to stop the platform, and its configuration can be exported and imported into other **Prodigi** instances or missions in the same instance.

For each task in the workflow, it is possible to define every aspect required for the execution:

- The processor to use (docker image), together with its default working directory and the exit codes considered successful.
- Command to run in the docker image, arguments and environment variables.
- Resources required for the execution (RAM, CPU/GPU, storage).
- Input data: their types, and the location where they are expected by the processor.
- Auxiliary data: their types, selection policies and the location where they are expected by the processor.
- Output data: the location where the processor will generate them, whether the data is archived, disseminated, or is stored locally in the node storage to speed up I/O operations (rather than being stored in shared storage). It is also possible to indicate whether an output has to be catalogued, archived or if it can be disseminated.
- Retry approach (no retry, or automatic retry).

Data types, mentioned above for the description of the input, auxiliary and output data of a processing step play an important role in the **Prodigi Processing Platform**. Data ingested is classified into data types, and when the data type matches the type of input defined for a task, the task is triggered.

Affinity-based execution policies allow specific processing steps to be bound to defined hardware profiles. This is particularly relevant when GPU-intensive algorithms require dedicated computational resources. This is achieved by identifying at definition time the processing steps that require GPU, and labelling nodes with GPU at Kubernetes cluster level.

For output data it is possible to indicate whether it will be archived, disseminated or will use local storage in the processing node. This is especially useful in case of disk-intensive processors that will benefit from using local storage rather than shared storage (the default). In addition, this setting will create an affinity for other processing steps in the same workflow to use the same processing node, getting access to the data previously generated in the local storage and therefore speeding up the processing time. In scenarios where I/O operations on storage are low, shared storage can be used, lowering the cost of the infrastructure.

An additional way of increasing the performance, especially when addressing large auxiliary data such as DEMs, is to make a copy in each node and access it locally. This can be done at installation time and later configure **Prodigi** to manage this scenario.

Prodigi is responsible for hiding all this storage complexity from the processors, providing them with access to Kubernetes persistent volumes mounted as local filesystem. From the processor perspective, it is as if local storage is accessed. This means that the policy can change (using shared storage instead of local storage, for instance) just by configuration in **Prodigi**, without any change to the processor code.

To define the processing chains, product and pipeline specifications and documentation must be available.

Typical processing chains define how to process RAW data as is downlinked from the satellite up to L0, while other processing chains define how to further process the L0 products to a higher level according to the production requests.

This capability allows, for example, to define processing chains capable of generating the following data levels for optical instruments¹:

- **Level 0:** Raw telemetry data intended for long-term archiving, calibration, and reprocessing if needed. This product will be stored in the archiving system upon generation for future use, if needed.
- **Level 1:** Provides radiometric correction.
- **Level 2:** Includes both radiometric and geometric corrections.
- **Level 3:** Delivers ortho-images (ortho-rectified Level 2 images based on a digital elevation model) and panchromatic sharpened images (colour images combining PAN and MS data, with the resolution of PAN images).
- **Other levels:** the Processing Platform is also able to launch processors for the highest product levels (Level 4, level 5 and upper)

The flexibility offered by **Prodigi** enables the configuration of multiple processing chains, for instance allowing the fusion of data from different sensor images acquired and processed through separate chains for each sensor (e.g. PAN + MS with PAN+LWIR).

Prodigi includes a graphical workflow design tool that allows users to define and configure processing chains. Through an intuitive interface, users can create processing steps, define dependencies, and configure execution parameters. A guided wizard supports the creation of new processing steps, simplifying the integration of new processors or algorithms.

The following figures show examples of the graphical representations associated to a processing workflow.

¹ Based on ESA definitions for product levels:

- "RAW" or "Level-0 candidate": telemetry packets straight from the satellite, unprocessed sensor counts, no corrections applied.
- Level 0A: Reformatted raw instrument data with Earth location and ancillary information; closest to instrument's direct measurement but organized for processing.
- Level 0B: Decompressed and error-corrected (validated) radiation data in the sensor system.
- Level 1A: Sensor data reconstructed, time ordered, geolocated, with calibration data attached but not applied.
- Level 1B: Radiometrically corrected and calibrated sensor data, but not yet atmospherically corrected (top-of-atmosphere).
- Level 2A: Atmospherically corrected surface reflectance data (bottom-of-atmosphere).
- Level 2B: Map projected data.
- Level 2C: Orthorectified, Analysis-ready data.
- Level 3+: Value-added products based on Geo-AI analysis.



Figure 5-1 Workflow definition in *Prodigi*

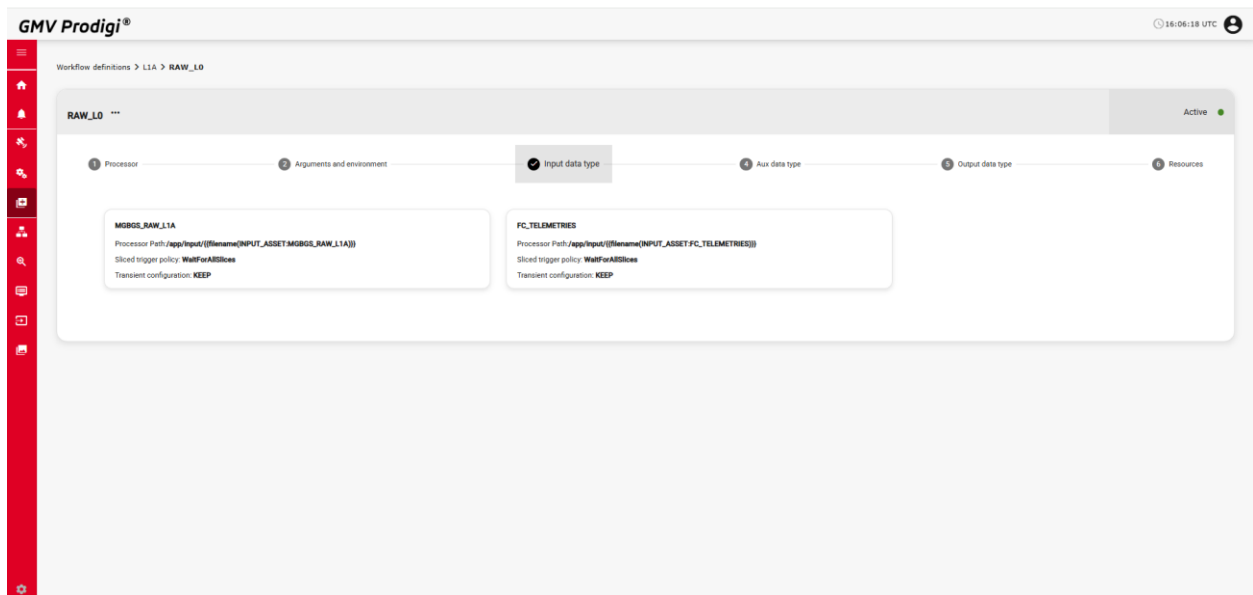


Figure 5-2 Task definition in *Prodigi*

5.2.3. PRIORITY MANAGEMENT

Prodigi allows the execution of enqueued processing tasks sorted according to user-selected priority levels that operators can modify according to the needs.

The **Prodigi** prioritization policy is not preemptive, meaning that requests that are being executed are not interrupted even if requests with more priority happen to arrive. Priorities are managed at request level, meaning that all workflow and task executions involved inherit such priority when competing for resources at Kubernetes level.

5.2.4. CONFIGURABLE DATA INGESTION

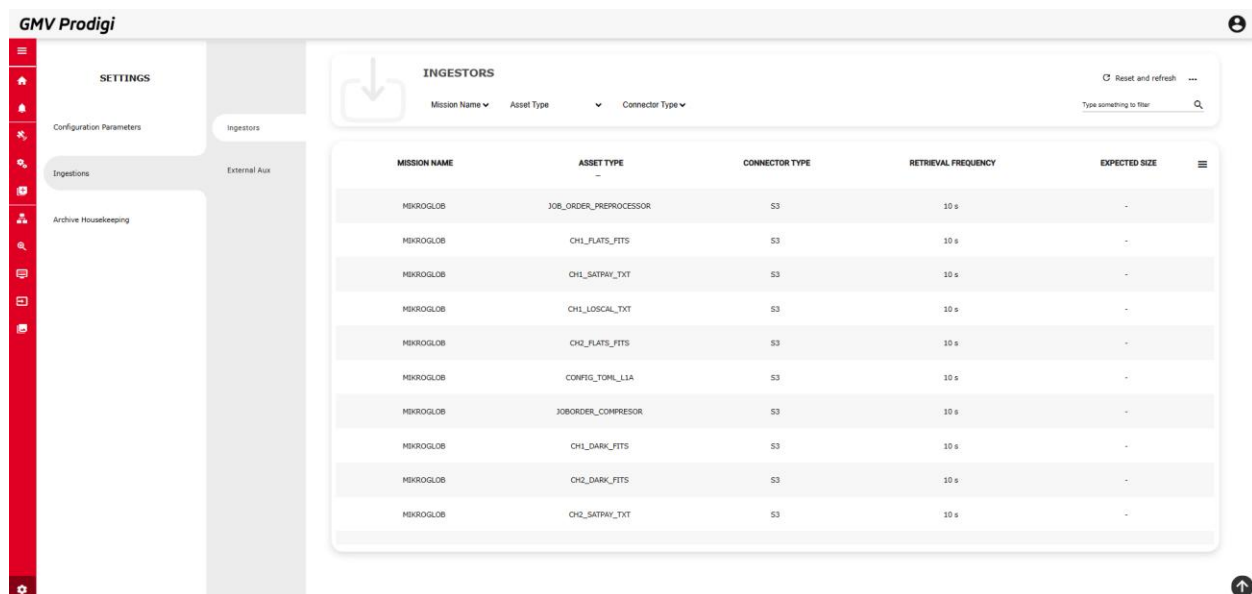
The Prodigy Processing Platform integrates with the satellite communication subsystem to receive downloaded data packages, or with data providers and the Cal/Val subsystem to retrieve auxiliary data.

Availability of new data is automatically detected in configurable exchange locations, triggering the ingestion and associated processing.

Operators can define ingestion details such as origin, polling frequency, credentials, etc. **S3** and **FTP** protocols are currently supported for the discovery and transfer of data. Our solution provides data ingestion interfaces responsible for:

- Polling data sources with configurable frequency.
- Discovery data in the source compliant with configurable name patterns.
- Transference of data using S3 or FTPS protocols.
- Classification of the data using configurable data types. This allows **Prodigi** to match new data with data needs and trigger processing workflows.
- Removal of data in the source to avoid re-discovery and perform housekeeping.

Other ingestion interfaces can be covered by adapting the platform to ingest data from specific endpoints using other protocols for discovery and transfer, including external archives from other missions.



MISSION NAME	ASSET TYPE	CONNECTOR TYPE	RETRIEVAL FREQUENCY	EXPECTED SIZE
MKROGLOB	JOB_ORDER_PROCESSOR	S3	10 s	-
MKROGLOB	CH1_FLATS_FITS	S3	10 s	-
MKROGLOB	CH2_SATRAY_TXT	S3	10 s	-
MKROGLOB	CH1_LOSCAL_TXT	S3	10 s	-
MKROGLOB	CH2_FLATS_FITS	S3	10 s	-
MKROGLOB	CONFIG_TORH_L1A	S3	10 s	-
MKROGLOB	JOBORDER_COMPRESSOR	S3	10 s	-
MKROGLOB	CH1_DARK_FITS	S3	10 s	-
MKROGLOB	CH2_DARK_FITS	S3	10 s	-
MKROGLOB	CH2_SATRAY_TXT	S3	10 s	-

Figure 5-3: Ingestion configuration

5.2.5. UNATTENDED DATA-DRIVEN PROCESSING

Processing activities are initiated based on two kinds of activation:

- **Data-driven activation:** upon successful detection, transfer and classification of input data. When the source of the data is a ground station, typically the associated kind of processing consists of two stages:
 - CADU to L0: a dedicated workflow decrypts, reconstructs and decompresses the data for later processing.
 - L0 to target level: dedicated workflow to obtain the target processing level.

The orchestration and triggering of each processing chain are governed by information received in advance from e.g. the mission planning system, which specifies the requested product type (target processing level), processing parameters, dissemination parameters, and the priority level.

In nominal operation, the Processing Platform receives the user request and awaits the corresponding acquisition data from the satellite pass. Upon reception, the system correlates the

incoming data package with the original request, providing full visibility of acquisition status and readiness for processing.

Once the acquisition corresponding to a request is obtained from the downlinked data, the platform launches the appropriate workflow using the user-defined parameters and based on matching data types between the ingested data and the definition of the processing chain. The processing model is therefore inherently data-driven, with workflow instantiation triggered by the combination of validated request information and confirmed data availability.

- **Order-driven activation (on-demand):** upon reception of a processing order (on-demand processing). The order must include all the information needed for the processing: archived product to process, target processing level, processing parameters and dissemination parameters.

In a similar way as before, the processing platform will select the appropriate workflow to trigger based on the target processing level. Processing parameters will be passed on to the processors and dissemination parameters will be used to distribute the products generated.

For each triggering, a new instance of the workflow is created, based on the workflow definition available at the time. This means that if a workflow definition changes, new instances of the workflow will incorporate the changes.

Each workflow instance groups a set of task instances (the processing steps during execution), that are triggered when their input data is available. Each task instance is fully traceable, with state transitions recorded, timestamped, and associated with the produced outputs and processing context, including logs, inputs, auxiliary data used, and outputs. This way the calibration and processing context used to generate any product is recorded.

Auxiliary data is assumed to be present in the system when tasks that need them are triggered. Otherwise, the tasks will fail. Auxiliary data is ingested as soon as it is available and archived. The auxiliary data to be used in each case is selected by means of a selection policy that normally relates to the main input (e.g. latest auxiliary data available, covering a period that covers that of the input, etc).

Once a task finishes correctly, the workflow is analysed to trigger those tasks with data dependencies with the one just finished.

It is possible to navigate from the task execution level to workflow execution level and also to order level, if applicable. This provides operators of the PDGS an order reference that can be used to trace activity at system level.

Error handling features such as automatic retry of tasks (providing the delay between retries and the maximum number of retries) and timeout are also provided. When a task fails (typically because the processor exists with an error code), the retry configuration is used. If a task runs for a period longer than the timeout, it is automatically cancelled.

Executions can also be manually cancelled, retried or resumed. A task that failed can be manually launched by the operator and, in case of success, the workflow can continue. All those actions involve state changes and, as such, are recorded.

5.2.6. EXHAUSTIVE PROCESSING MONITORING

The **Prodigi Processing Platform** provides operators with all necessary tools to monitor and control the execution of processings:

- **Order Status:** Users can monitor the lifecycle of each order (Imaging or Processing), from ingestion through processing to final product generation and delivery.

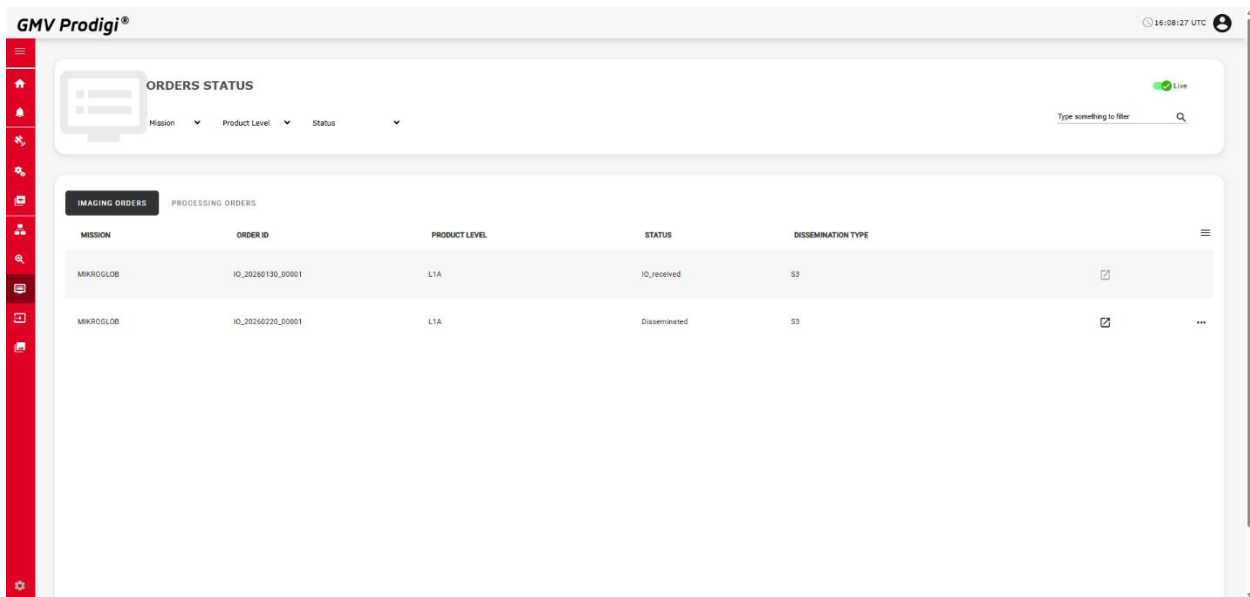


Figure 5-4: Orders status

- Status of the ingestions, as defined by the mission timeline. Ingestion activities, associated to the Imaging Orders, are planned and their progress is reported.
- Status of the processing chains. Triggered according to the Imaging or Processing Orders received, the processing status of the chains is continuously monitored. Imaging Orders are associated to planned acquisitions, while Processing Orders are triggered by users using catalogued data.

Users can track the progress of processing chains dynamically, including the status of each step.

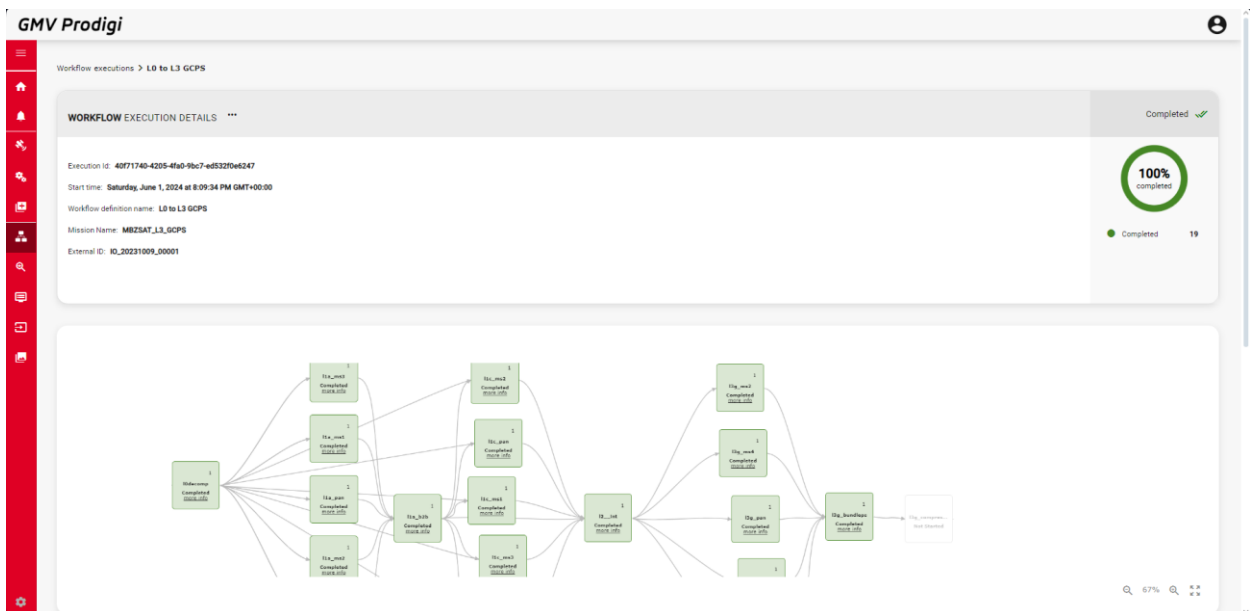


Figure 5-5: Workflow execution monitoring

- Status of the steps inside the chains. The fine-grained status of each step inside a chain can be accessed by operators including the processor progress.

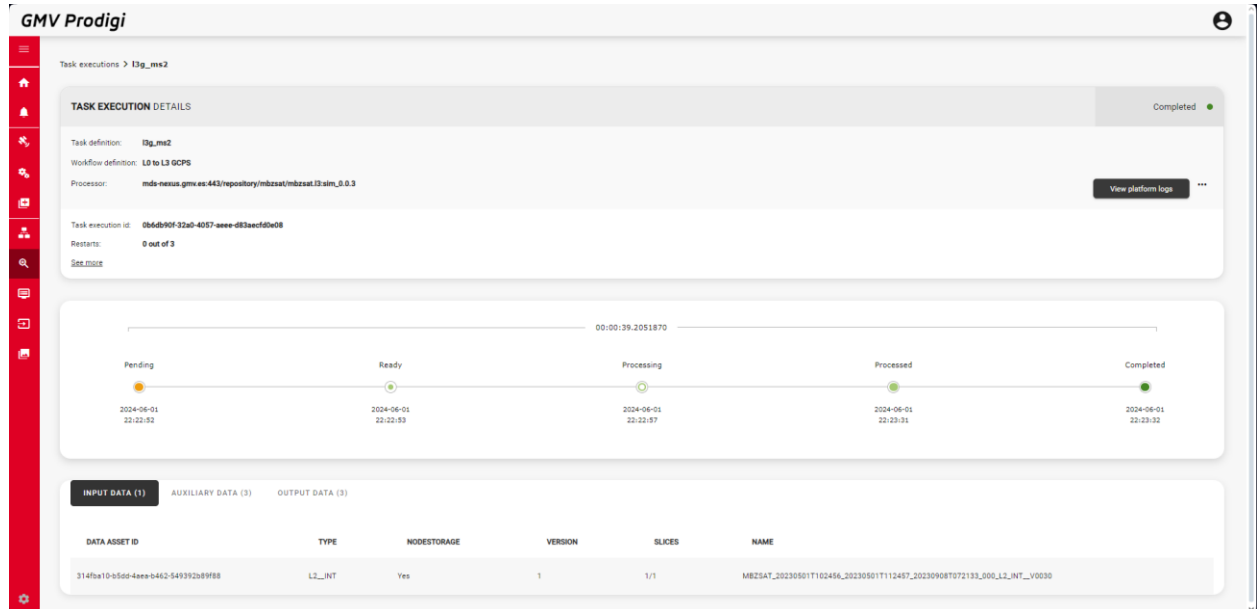


Figure 5-6: Task execution monitoring

- Logs of the processors invoked inside processing steps. Either live (while the processor is running) or after finalization, the processor logs are available for analysis.

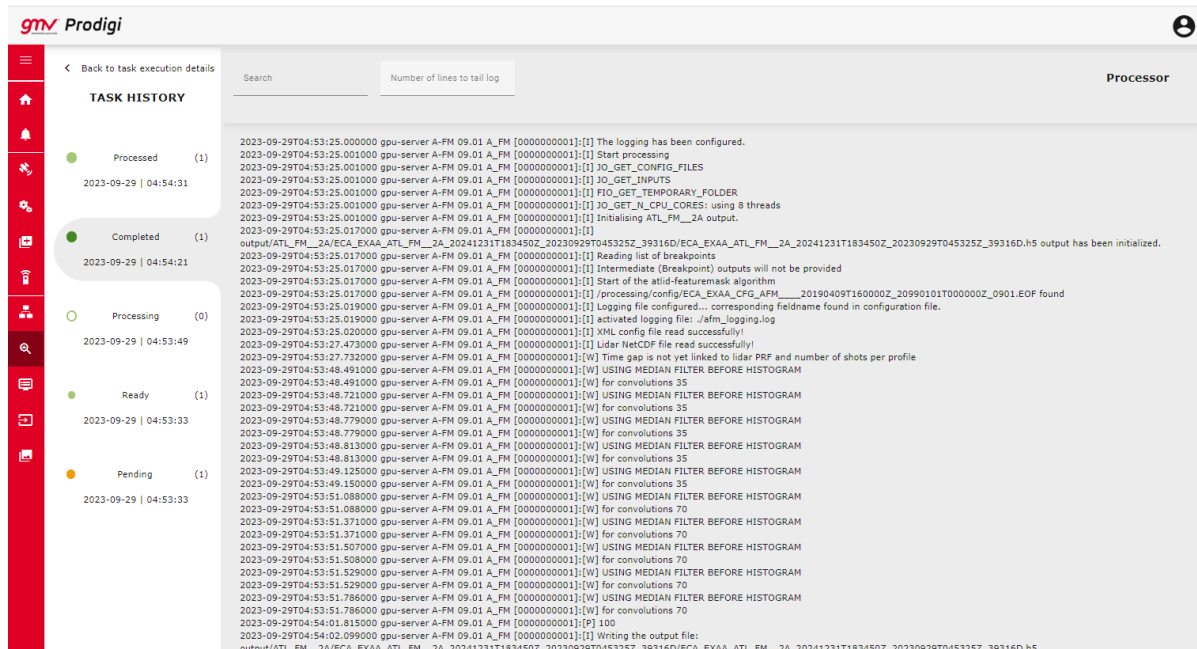


Figure 5-7: Prodigy gives access to the processor logs

- Status of the disseminations. Disseminations, triggered by orders, are monitored and any problem is reported.
- **Prodigi** also provides a **dashboard** where operators get a consolidated, real-time overview about the platform activity, including processing activities, order status, and production planning. It

enables operators to quickly assess system health, future foreseen activity and overall processing performance.

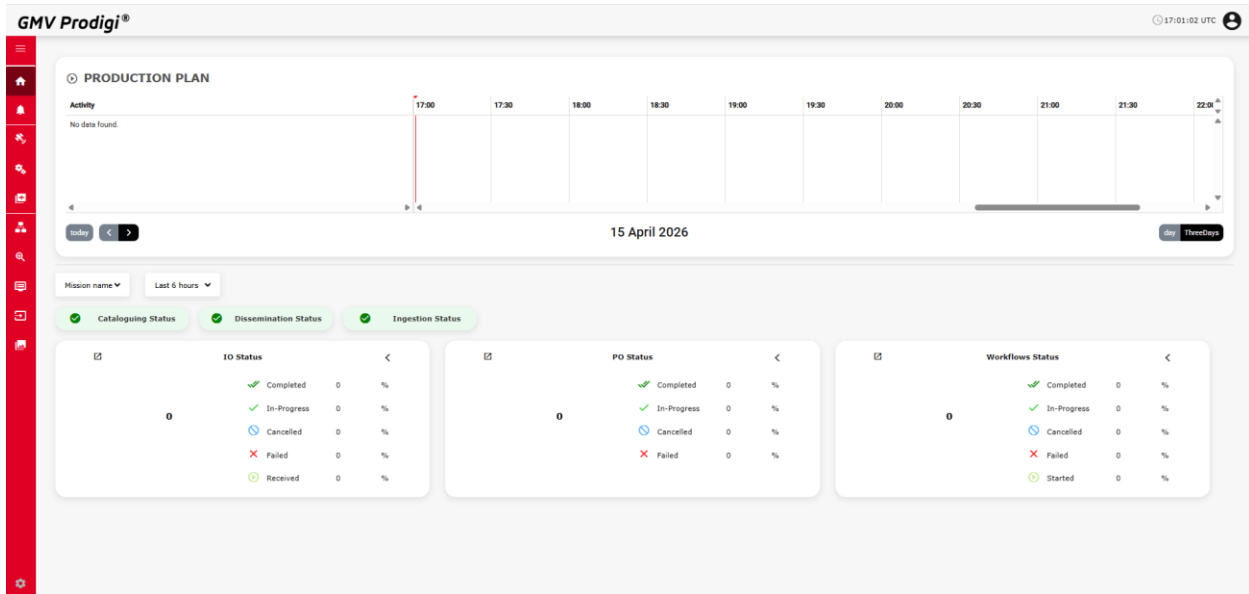


Figure 5-8: *Prodigi* Dashboard

Operators can control the execution of the processing chains at several levels:

- Tasks can be activated and deactivated at definition level, so they are not triggered when the workflow is executed.
- Task executions can be manually cancelled and retried.
- Workflow executions can be manually cancelled and resumed, meaning that any failed task is restarted and the execution continues.
- Orders can be manually restarted from the beginning and resumed.

5.2.7. ADAPTABLE DATA DISSEMINATION

By configuration, it is possible to define which outputs will be disseminated as a result of the execution of a processing chain. The kind of dissemination is part of the request triggering the processing. Currently two dissemination types are supported:

- **Data push to an S3 bucket:** the platform makes a copy of the output product in an S3 bucket, isolated from the rest of the storage for security reasons. Once the product is transferred, a temporary, self-signed download link is created.
- **Transfer to an external FTPS server.**

Other dissemination interfaces can be covered by adapting the platform to the required protocol.

5.2.8. ROLLING OR PERMANENT DATA ARCHIVING

It is possible to configure which outputs generated by the processors are archived. Ingested data is also archived by default. Archive data is stored in an **S3 block storage**.

Configurable housekeeping policies are available, allowing users to keep the archive size under control. Housekeeping policies are defined by data type and age (number of days the data has been stored in the archive) according to mission needs and storage capacity, thus providing a rolling archive for selected data types. When data of a certain type reaches the maximum retention time, it is removed from the archive, freeing up space in the storage space.

This way, a mission can keep in the archive e.g. L0 products (products with a minimum transformation, just deciphered and decompressed), for the life of the mission, while other higher-level products can be kept for a while, then removed and generated again if needed. This is typically a good option when using cloud resources since the cost of regenerating a product is normally lower than the cost of storing it for a long time.

The housekeeping configuration can be changed at any time, adapting the amount of data stored to the mission needs.

5.2.9. MISSION-AGNOSTIC DATA CATALOGUING

Archived generated outputs can be catalogued in a metadata database accessible through **STAC** and **OData endpoints** supporting **complex queries** including geospatial coordinates for integration with GIS-enabled applications. This way the catalogue can be integrated as metadata provider to external aggregators.

The catalogue serves as a link between the Processing Platform and the Customer Services, allowing customers to smoothly discover and browse the catalogue of archived products by means of **efficient indexing** (tests done with metadata from Sentinel 1 and Sentinel 2 up to more than 25 million products).

The metadata model in the catalogue is **mission agnostic** and **extensible**: from a minimal part common to any product (type, footprint, etc) it is possible to define specific attributes as required by the use case, per product type (e.g. specific attributes for optical or radar products, or compliance with metadata standards).

The catalogue can also store **quicklooks** and **thumbnails**, providing a preview of the data available in the archive.

The following table shows the list of metadata stored in the internal **Prodigi** Catalogue.

Table 5-1: Prodigi Catalogue Metadata

Metadata	Type	Description
Product ID	UUID	Unique product ID generated by the catalogue in ingestion
File Name	String	Data file name
Product Name	String	Name of the product
Product Type Name	String	Identification of the product type and mode (e.g. L0_MS1)
Product Version	String	Version number of the product
Content Type	String	MIME type of the product
Content Length	Unsigned Big Integer	Actual size in bytes (B) of the downloadable file
Origin Date	Timestamp	Timestamp recording when the entire downlinked raw data for the downlink session containing the data take was available at the acquisition interface (e.g. the ground station). Expressed in UTC in the format "YYYY-MM-DDThh:mm:ss.sssZ"
Publication Date	Timestamp	Timestamp recording when the data file has been published in the catalogue. Expressed in UTC in the format "YYYY-MM-DDThh:mm:ss.sssZ"
Last Modification Date	Timestamp	Timestamp recording when happened the last modification of the data for this product in the catalogue. Expressed in UTC in the format "YYYY-MM-DDThh:mm:ss.sssZ"
Last Modification Entity	String	Reference to the entity (subsystem, human...) that performed the last modification of the data of the product in the catalogue
Eviction Date	Timestamp	Timestamp recording when the data file will be removed from the rolling archive. Should be provided if a rolling policy is in place. Expressed in UTC in the format "YYYY-MM-DDThh:mm:ss.sssZ"
Content Start Date	Timestamp	Sensing start timestamp. Expressed in UTC in the format "YYYY-MM-DDThh:mm:ss.sssZ"
Content End Date	Timestamp	Sensing stop timestamp. Expressed in UTC in the format "YYYY-MM-DDThh:mm:ss.sssZ"

Metadata	Type	Description
Content Is Available	boolean	Is the product physically available? Useful in case the catalogue publishes 'potentially existing' products, or the product has been removed from the origin but we still want to keep the metadata in the Catalogue
Content Location	String	Means to locate the product (e.g. an URI)
Production Type	String	Type of production (e.g. 'systematic', 'on_demand_default', 'on_demand_non_default')
Footprint	WKT (Well-Known Text)	Footprint of the product
List of checksums, containing:	see below	List of checksums for the product file
Algorithm	String	Name of the checksum algorithm
Value	String	Checksum value
List of attributes, containing:	See below	List of attributes for the product
Attribute Name	String	Name of the attribute
Attribute Type	String	Type of the attribute: string, integer, double, boolean, timestamp, binary
Attribute Value	According to the type	Value of the attribute

A common, fixed part is populated whatever the product format or nature, while the variable part (attributes) can be extended to cover the mission needs and be different for each product type. For instance, the relationship between the product and the acquisition request, traceability information or scientific metadata needed find the desired products, are examples of variable metadata.

5.2.10.SCALABILITY

The platform can **easily scale** by adding more resources as needed to adjust the processing, storage and cataloguing capabilities according to the global processing requirements. These new resources are integrated seamlessly, without service interruptions. The Processing Platform supports both types of scalabilities:

- **Horizontal Scalability:** The number of processing nodes can be increased, allowing for parallel execution of processing tasks.
- **Vertical Scalability:** Resources within a processing node can be increased to run multiple tasks simultaneously. For example, RAM and CPU capacity can be increased to support additional processing tasks in the same node.

5.2.11.INTEROPERABILITY

An extensive REST API is available using a **Swagger** interface is available with access to the Processing Platform functions. From configuration to execution, this API can be used to integrate the platform with other systems.

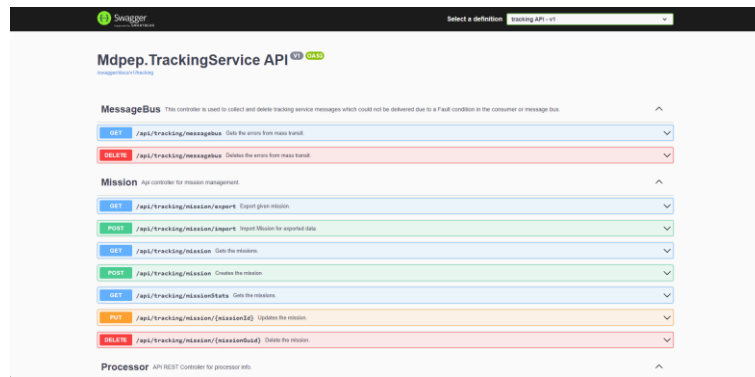


Figure 5-9: Prodigy offers a Swagger interface to its API

In addition, an **OGC API – Processes** implementation is available for triggering the execution of processing chains and retrieving the status. This is especially useful in a federation scenario where the ground segment is free to decide how to fulfil the requests (e.g. by delivering data from the archive or triggering new acquisitions).

Other useful interfaces are:

- **Downlink Plans:** files containing information about the foreseen production activities, they specify how the data for each user request can be extracted from the RAW data, and what should be done with it once extracted (target processing level, processing parameters, dissemination parameters).
- **Processing Orders:** files containing information about on-demand processing activities, they specify the archived data to be used as input and what is required to do with it (target processing level, processing parameters, dissemination parameters).
- **Health Status:** REST API providing summarised information about the health of the processing platform.
- **Logs:** All logs generated by the processing platform are collected and optionally forwarded to external systems.
- **Processing Status:** message sent to message queues informing about the processing status for each request.

5.3. CONFIGURABILITY AND ADAPTABILITY

Just by configuration, the **Prodigy Processing Platform** can be adapted to a wide range of missions and needs. There are some areas, however, where further adaptations might be needed such as:

- **Ingestion interfaces:** in case a new protocol for detection, transfer or selection of input and auxiliary data is required.
- **Cataloguing:** to adapt the metadata extraction to the data format and the required searches.
- **Dissemination interfaces:** in case a different protocol to transfer the outputs is needed.
- **Other external interfaces:** for integration e.g. with other monitoring tools.

6. CUSTOMER SERVICES

The **Prodigi Customer Services** offers an **entry point for the mission**, providing users with user-friendly, intuitive and easy access to satellite imagery services.

6.1. SCOPE

Since the **Customer Services** represent the external image of the mission, we provide a highly available and secure portal designed to **scale** with your business—from a single satellite to a full constellation and multiple missions, supporting the production of **satellite imagery** or acting as a **hub** interfacing **federations** of ground segments and providing **fusion** products.

It provides final users with end-to-end functionality such as:

- Order products from the Catalogue, where users can browse, purchase and download archived products.
- On-demand processing from Catalogue, where users can request to generate new data products from existing archived source products.
- Order new Acquisitions, where users can request new observations directly to the satellites and generate new data products from them.
- Subscribe to new acquisitions over an area of interest.
- Request value-added services or high-level requests that are transformed into lower-level requests to federated ground segments.

The **Customer Services** is also a portal for the **operators**, who are in charge of managing user access, approve orders, etc. according to the operational concept of the system.

6.2. KEY FEATURES FOR END USERS

The key features that make the **Prodigi Customer Services** especially useful for your end users are the following:

6.2.1. PRODUCT CATALOGUE BROWSING

Users can **browse and search** the catalogue of existing products generated by the Processing Platform typically by systematic processing. The customer can draw an area of interest (AoI) and search the catalogue for products intersecting that area and complying with search parameters such as temporal range, cloud coverage or tilt angle (for optical instruments).

The AoI can be defined by the user over a **map view** (satellite or political) by means of polygons, rectangles or points. Together with the other search parameters, the catalogue will be filtered so that users can explore a subset of the available products, getting their **footprints** and details, including thumbnails and **quicklooks**. Selecting a product will show the footprint in the map.

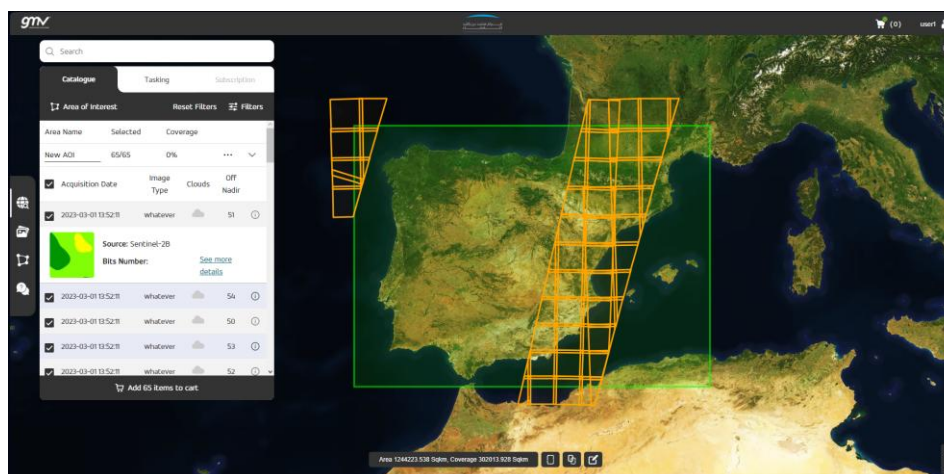


Figure 6-1 Browse catalogue of existing products

The filters available to search the catalogue will depend on the mission and the product type.

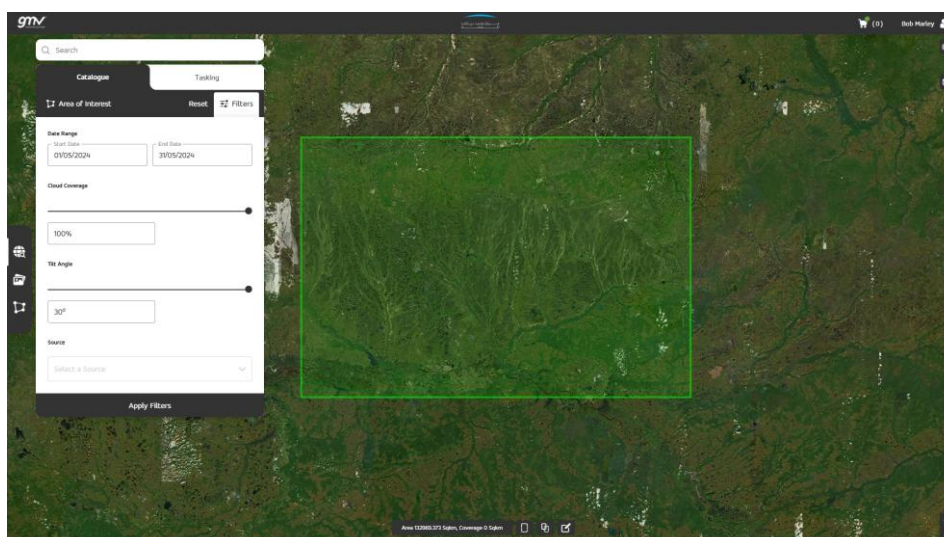


Figure 6-2 Catalogue filter parameters

In case no further processing of the product available in the catalogue is needed, a **direct dissemination** of the product can be ordered.

Note: The map view relies on the Geocoding and Static Tiles APIs from Mapbox.

6.2.2. ON-DEMAND PROCESSING

Products in the catalogue can be **further processed** to higher-level products (e.g. from L0 to L2). Customers can order this kind of processing, on demand, by selecting the source product from the catalogue and specifying the processing parameters they need, according to the capabilities of the processors integrated into the processing platform.

In a typical scenario, where the catalogue is populated with low-level products (e.g. L0 or L1A), usually a processing parameter is the target level expected for the output (e.g. L1C, L2 or L3). Other processing parameters may include the output format, the processing mode, etc.

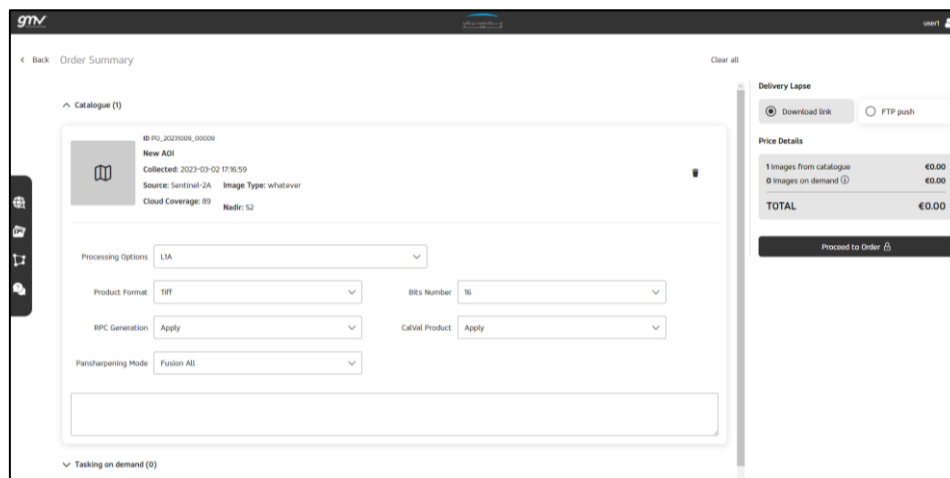


Figure 6-3 On-demand processing

6.2.3. ON-DEMAND ACQUISITION

Similar to browsing the catalogue, Customers can browse **acquisition opportunities** available for a certain AoI and filtering parameters (such as tilt angle, imaging mode, etc.). The selected acquisitions can be added to the shopping cart, defining also processing parameters for it (that will be applied when the acquisition is received on ground).

A **weather forecast** is provided for the obtained opportunities, together with a **clouds map**, to give users a better understanding of the foreseen conditions of the acquisition and select the best opportunity. Selecting an opportunity will show its footprint on the map.

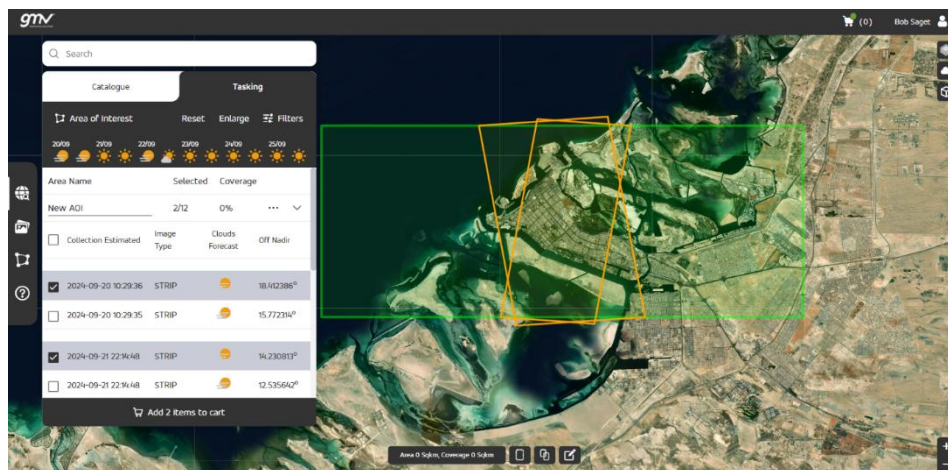


Figure 6-4 Search new acquisitions

The Customer Services allows users to request combined acquisitions from all available constellations (e.g., PAN+MWIR, PAN+LWIR, PAN+MS), thereby simplifying the inclusion of all required imagery within a single request.

Note: This feature requires an interface with a Flight Dynamics System (e.g. **FocusSuite®**, the GMV suite for full lifecycle flight dynamics operations for any mission) that performs opportunity calculations based on the orbit and visibility computations.

Note: The weather forecast and clouds layer features make use of the "5 Day/ 3 Hour Forecast" and "Weather Maps 2.0 3-hour step" APIs from OpenWeatherMap.

6.2.4. SUBSCRIPTIONS

Users can create **subscriptions**. Subscriptions enable the **automatic scheduling of acquisitions** over a specified area of interest within a configurable period and at a defined frequency and filter parameters as for the case of tasking a new acquisition.

The system will use the filter parameters defined in the subscription to periodically perform a search for tasking opportunities, selecting the best one and creating an imaging order including the processing parameters. All this happens automatically.

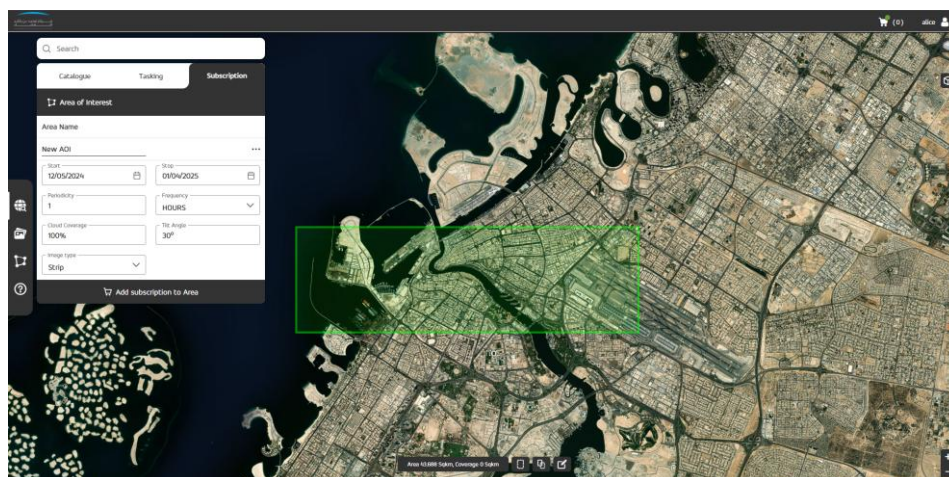


Figure 6-5 Create subscription

6.2.5. SERVICES REQUEST

Similar to browsing a catalogue of products, in some scenarios, the user has access to a catalogue of **services**, high-level requests normally involving some **data fusion** where the source is not a single product. The complexity of the request is hidden from the user and orchestrated to provide a smooth experience.

Typical services involve time-series processing and/or data from different satellites, and the result is usually a high-level product such as crop type classification, methane detection, vineyards status...

Services are normally associated with a study area (a geographical area where the service is available), whose footprint is highlighted in the map view. Areas of interest with an intersection with the study area will be able to request the associated service.

When a service is selected and added to the shopping cart, it is possible to define details specific for that service. Since services usually involve time-series processing, start and end time of the processing, spatial resolution or temporal resolution are typical parameters.

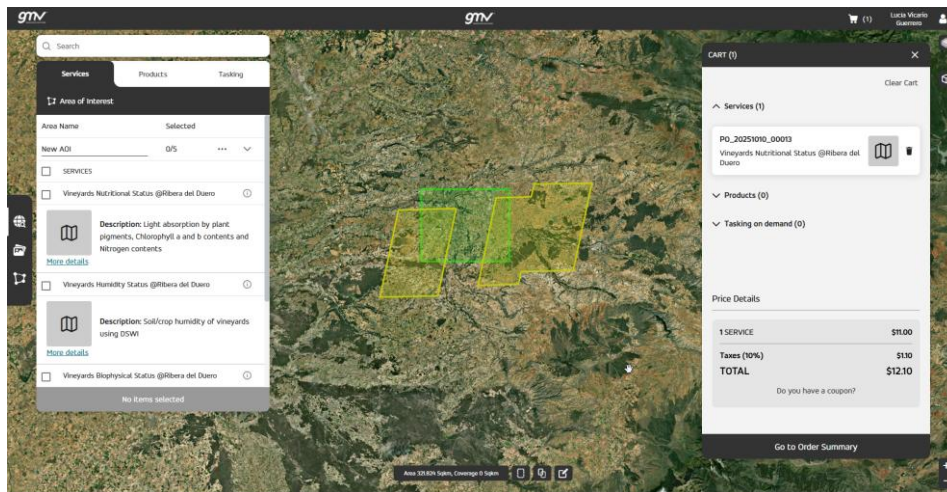


Figure 6-6 Service request

6.2.6. FEDERATION HUB

The capability of hiding the complexity of a service to the user can be further exploited using the Customer Services as a **federation hub** involving different mission data providers or whole ground segments. High-level customer requests are transformed into mission-level requests involving catalogue searches and/or acquisitions taskings that are delegated to specific missions and **orchestrated** to provide the result.

The level of integration of the federated providers may vary, from giving access to their **catalogues and mission planning systems**, to a simple **OGC Processes API** endpoint to their ground segment, delegating the decision whether to answer the request from archived data or new acquisitions.

6.2.7. SHOPPING CART

Any order, (direct dissemination, on-demand processing or acquisition, subscription or service request) can be added to a **shopping cart** that will provide the individual prices and total amount to be paid for it (if applicable to the business model case). If no payments are required, the shopping cart can still be used to visualize and manage all items a Customer is requesting in one go. It is possible to add and remove items from the cart at any time.

6.2.8. SPECIFICATION OF DISSEMINATION METHODS

In addition to the details for each request, the user can detail the dissemination method for all the items being ordered. Following the capabilities of the Processing Platform, two dissemination methods are available:

- **Download link** from an S3 bucket, so that when the order is fulfilled the user is able to directly download the result.
- **FTPS push** to a server managed by the user, where the results are transferred when they are available.

6.2.9. PAYMENTS, BILLING AND INVOICES

In case the business model implies payments, this can be done through the Customer Services using a **payment gateway**. The final price will depend on the total price of the order, to which **discounts** and **coupons** can be applied.

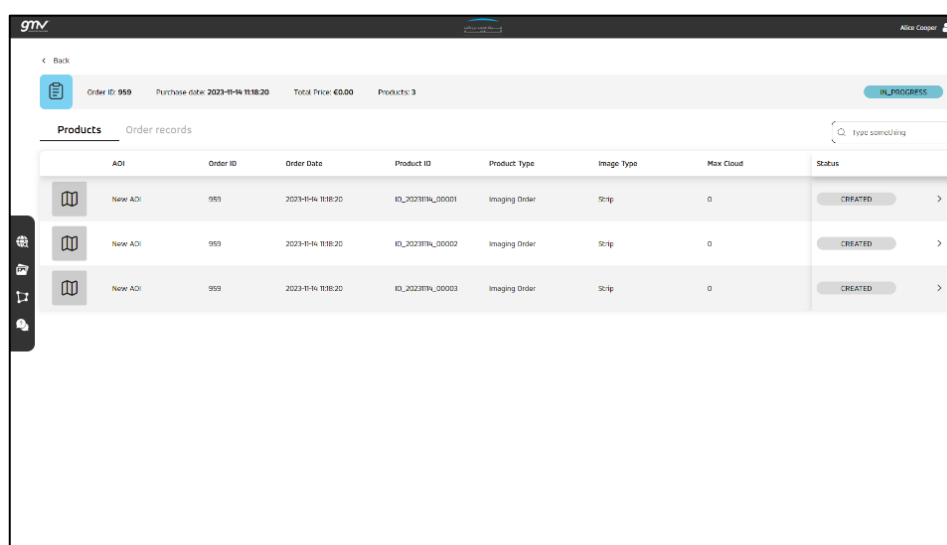
Payment results can be tracked and invoices can be issued if needed.

6.2.10.MONITORING ORDERS STATUS

Users can monitor the **status** of orders. The possible statuses differ depending on the order nature. For instance, an imaging order will include information about the mission planning phase of the order, while a processing order will only report about the processing status. This information is collected by the Customer Services from the rest of the components that make up the ground segment.

It is possible to dig in the **details** of an order, to check the status of the individual items of the order (processing or imaging requests). Each one will have its own status since a request can be already finished while others are still in progress.

For each request, the details about acquisition and/or processing parameters can be reviewed.



The screenshot shows a web interface for monitoring orders. At the top, it displays 'Order ID: 959', 'Purchase date: 2023-01-04 11:18:20', 'Total Price: €0.00', and 'Products: 3'. Below this, there's a 'Products' section with a table of order records. The table has columns: AOI, Order ID, Order Date, Product ID, Product Type, Image Type, Max Cloud, and Status. Three rows are visible, all with 'New AOI' and 'Imaging Order' status, and a 'CREATED' status button next to each.

AOI	Order ID	Order Date	Product ID	Product Type	Image Type	Max Cloud	Status
New AOI	959	2023-01-04 11:18:20	ID_20230104_00001	Imaging Order	Strip	0	CREATED
New AOI	959	2023-01-04 11:18:20	ID_20230104_00002	Imaging Order	Strip	0	CREATED
New AOI	959	2023-01-04 11:18:20	ID_20230104_00003	Imaging Order	Strip	0	CREATED

Figure 6-7 Orders status

Note: the Customer Services sends requests and receives request statuses from other elements of the ground segment, such as the mission planning system (e.g. **GMV Flexplan**®, a real general planning and scheduling system) and the processing platform.

6.2.11.ORDERS HISTORY

Users can access all the orders placed by her/him to retrieve details about previously submitted requests. Users can access orders she/he has requested.

6.2.12.SPECIAL USERS

It is possible to define **special user roles** with access to advanced capabilities, such as tasking or subscriptions. A typical case is to provide **Cal/Val users** with additional acquisition parameters when requesting a new image for calibration purposes, or access to products in the catalogue of certain level (e.g. L0).

6.3. KEY FEATURES FOR OPERATORS

The Customer Service provides a set of special functionalities for operators, allowing them to manage the behavior of the User Services. These functionalities include:

6.3.1. MANUAL APPROVALS

Sometimes the concept of operations involve manual activity from the operators for certain workflows, for instance:

- After a user submits a registration request, the operator can **approve the registration** for access. Additionally, the administrator can configure users' privileges by limiting, modifying, or revoking their rights.
- Some users may require **manual approval of their orders**, depending on their rights. Operators are therefore activated when this is required.

Operators are provided with a list of actions pending to do.

6.3.2. MONITORING ORDERS

Operators have visibility to all user requests and can monitor their progress individually or in a summarized dashboard. Details about each order, their payment and acquisition/processing status and the products inside, are available.

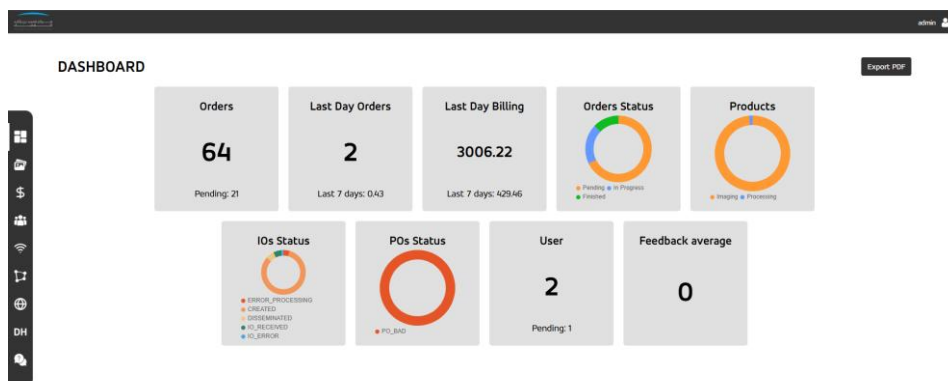
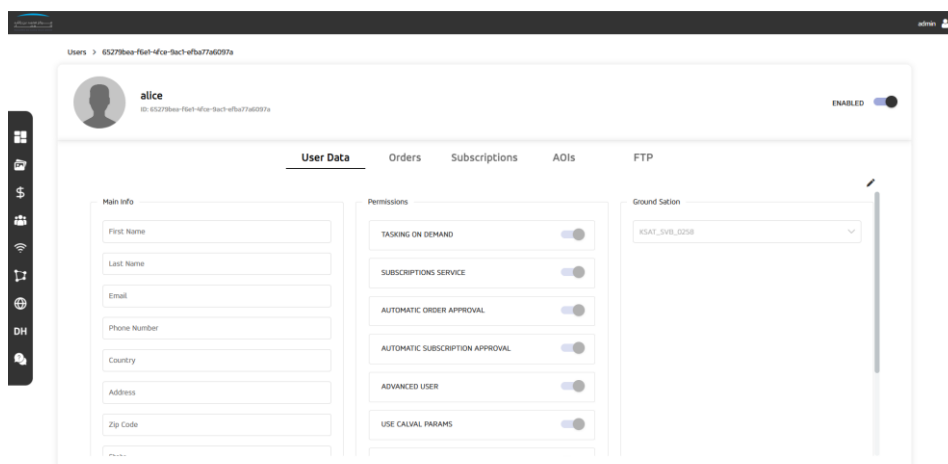


Figure 6-8 Monitoring dashboard

6.3.3. USER MANAGEMENT

Apart from approving their registration, operators can **define details** for users such as whether they are enabled or disabled, and what are their **limits and permissions**: whether they can order taskings or subscriptions, whether they can access Cal/Val features, maximum number of orders or scenes in an order, maximum priority for the orders the user can create, etc. All this is governed by the operational concept of the use case.

It is also possible, for a user, to check the **historic** of orders that have been requested, the **restricted areas** that are applicable etc.



The screenshot shows the "User Data" form for a user named "alice". The form is divided into several sections:

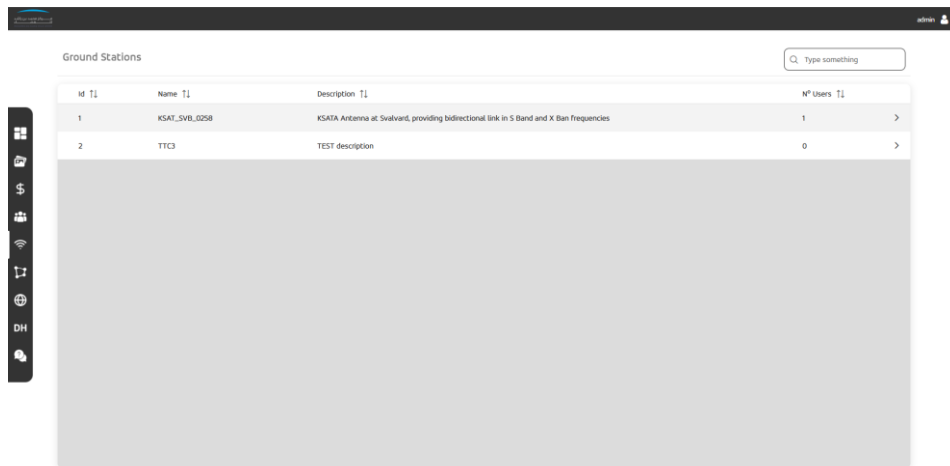
- Main Info:** Fields for First Name, Last Name, Email, Phone Number, Country, Address, and Zip Code.
- Permissions:** A list of permissions with toggle switches: TASKING ON DEMAND, SUBSCRIPTIONS SERVICE, AUTOMATIC ORDER APPROVAL, AUTOMATIC SUBSCRIPTION APPROVAL, ADVANCED USER, and USE CAVAL PARAMS.
- Ground Station:** A dropdown menu currently set to "KSAT_0108_0208".

The form is titled "User Data" and includes tabs for "Orders", "Subscriptions", "AOIs", and "FTP". The user's status is shown as "ENABLED" with a toggle switch.

Figure 6-9 User details

6.3.4. MANAGEMENT OF GROUND STATIONS

Operators can define the **list of ground stations** available and assign them to users. That way, tasking requests can include the preferred ground station for the user making the request.



Id	Name	Description	N° Users
1	KSAT_SVB_0258	KSATA Antenna at Svalbard, providing bidirectional link in S Band and X Band frequencies	1
2	TTCS	TEST description	0

Figure 6-10 List of available ground stations

6.3.5. RESTRICTED AREAS DEFINITION

Restricted areas are specific locations on which no user (or certain users) will be able to obtain products intersecting with them, either from the catalogue or from new acquisitions, normally for security reasons.

Restricted areas can be added, edited, deleted and assigned to users.

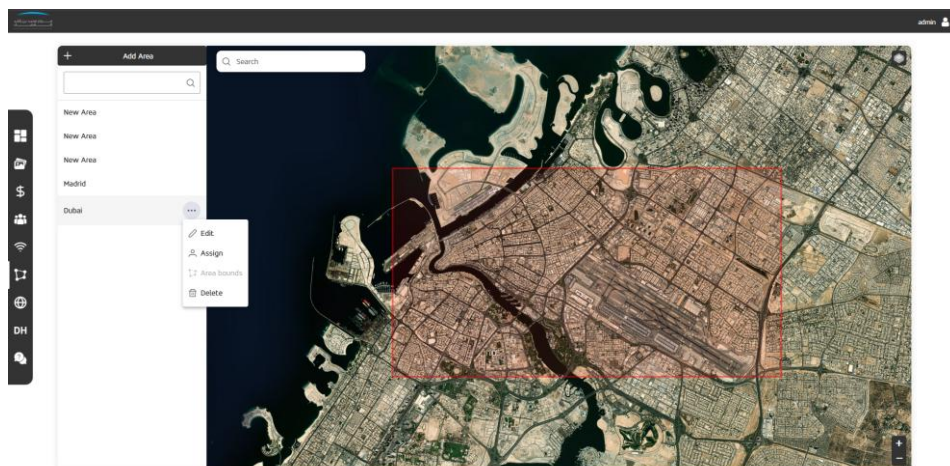


Figure 6-11 Restricted areas

6.3.6. ENABLING / DISABLING PRODUCTS IN THE CATALOGUE

Operators can **enable or disable the visibility** of products in the catalogue for users. The visibility status of the products in the catalogue can be checked and modified according to the needs.

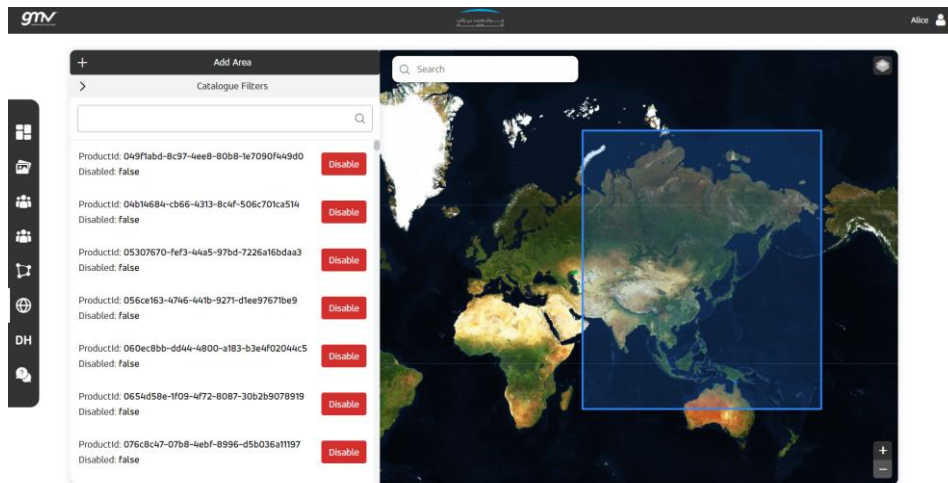


Figure 6-12 Catalogue Management Page, Area List

6.3.7. PRICES AND DISCOUNTS

Operators can define **one or several prices lists**. Each list defines the price for a product type and can be defined **per unit or per area size**. Prices lists are assigned to users so **different users can have a different price list**.

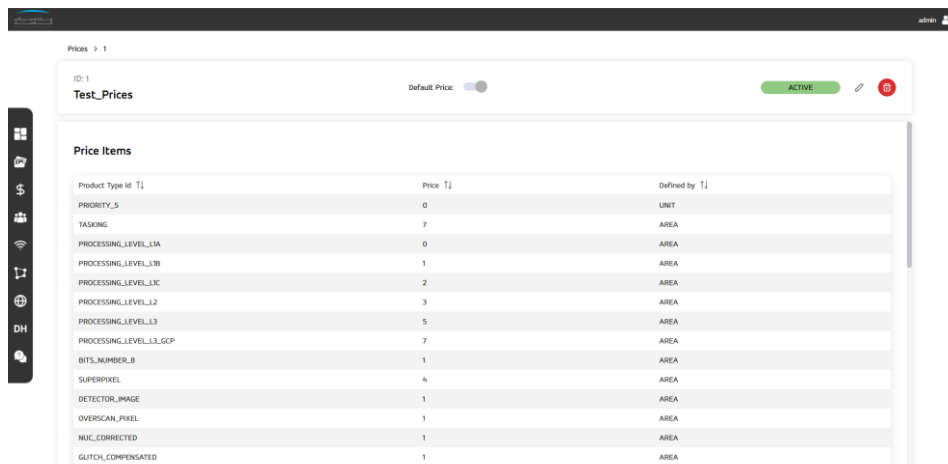
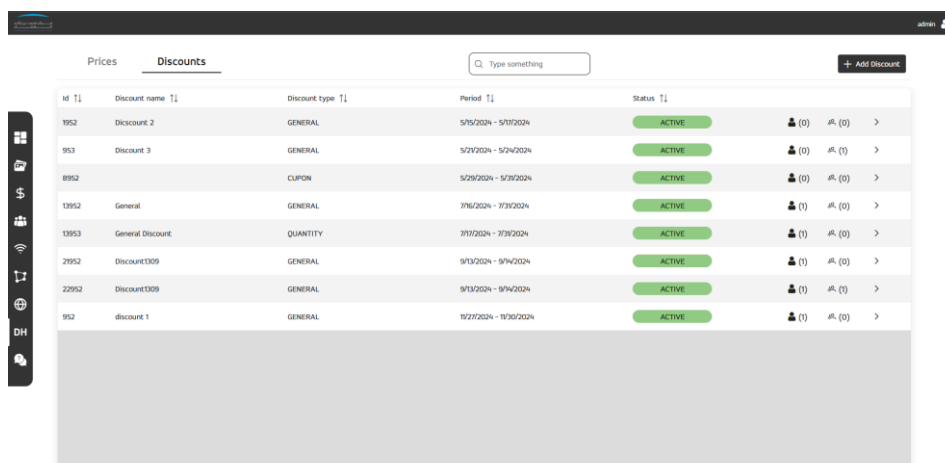


Figure 6-13 Prices list

Discounts and coupons can also be defined and assigned to users. A discount can be applicable for a **period of time** and to a **list of product types**. They can be general discounts or apply for a quantity of orders.



ID	Discount name	Discount type	Period	Status	
1952	Discount 2	GENERAL	5/15/2024 - 5/17/2024	ACTIVE	👤 (0) 📄 (0) >
953	Discount 3	GENERAL	5/21/2024 - 5/24/2024	ACTIVE	👤 (0) 📄 (0) >
8952	CLIPON	CLIPON	5/29/2024 - 5/30/2024	ACTIVE	👤 (0) 📄 (0) >
13952	General	GENERAL	7/16/2024 - 7/31/2024	ACTIVE	👤 (0) 📄 (0) >
13953	General Discount	QUANTITY	7/17/2024 - 7/31/2024	ACTIVE	👤 (0) 📄 (0) >
21952	Discount109	GENERAL	9/13/2024 - 9/14/2024	ACTIVE	👤 (0) 📄 (0) >
22952	Discount109	GENERAL	9/13/2024 - 9/14/2024	ACTIVE	👤 (0) 📄 (0) >
952	discount 1	GENERAL	10/27/2024 - 10/30/2024	ACTIVE	👤 (0) 📄 (0) >

Figure 6-14 Discounts list

6.4. CONFIGURABILITY AND ADAPTABILITY

The adaptation of the **Prodigi Customer Services** to a specific case involves:

- **Aesthetic adaptations**, such as logos, colour palette, etc. to comply with the brand image.
- Adaptations to the **business model**: paid/free service, refunds, payment methods.
- **Operational concept**: manual or automatic approval of customer operations.
- **User profiles**: adaptation to define functional limits of users in the portal (e.g. some users can request new acquisitions, others cannot).
- **Search parameters**, which depend on the type of products in the catalogue and the granularity to be provided to users when making a search.
- **Processing parameters**, which depend on the products and capabilities portfolio being offered to customers.

Since all those aspects have significant impacts, the adaptations normally involve the development of custom extensions.

7. DEPLOYMENT

Prodigi can be deployed in real hardware, virtual machines, or in the cloud, using standard cloud services if needed. Hardware resources are abstracted through containers orchestration (Docker and Kubernetes), making migration to different platforms easy.

7.1. DEPLOYMENT SCENARIOS

Prodigi can be deployed in several configurations to adapt to your concept of operations:

- Dual use with separate processing platforms to restrict access and achieve data separation.
- Hybrid deployment (cloud & on-premises) to comply with security constraints.
- Testbench for processors development to smooth the pass to operations of algorithms.

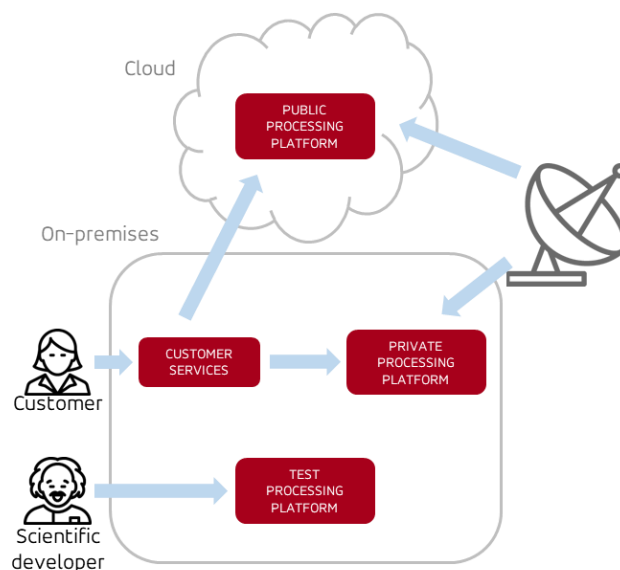


Figure 7-1: Deployment scenarios of Prodigi

7.2. MULTI-SITE DEPLOYMENT AND DISASTER RECOVERY

Prodigi supports deployments in separate data centers, or sites, with different configurations depending on the scenario:

- **Prime-Backup**: all the data is received and processed by one site. Enough data to proceed with operations in case of disaster is sent from the prime site to the backup site. In case of site switch, the backup site can take over operations, and the roles are inverted.
- **Active-Active**: both sites are simultaneously operational. Load balancing is achieved, e.g. associating each site to the nearest ground station, in case there is more than one, or using other mechanisms depending on the case. By performing all processing steps – from raw data to final product generation – in the same site, latency due to cross-site data transmission is avoided.

In both cases, **inter-site data synchronisation** is needed, but limited to essential datasets required for operational continuity. Typically, only the following elements are replicated between data centres:

- RAW/L0 input datasets.
- Ancillary data for processing.
- Operational metadata necessary for traceability and lifecycle continuity.

Intermediate artefacts, temporary files, and transient workflow data remain local to the originating site. This approach significantly reduces storage and network replication load while maintaining full disaster recovery capability.

In failover scenarios, activities are reorganised configuration changes. Because raw inputs and required operational metadata are synchronized, the alternate site can resume or continue processing activities without loss of context. Failover and failback procedures are controlled and deterministic, preventing duplicate product generation and ensuring lifecycle consistency.

Depending on the scenario and the sizing of the resources, the alternate site may operate in a degraded but controlled mode, prioritizing critical processing requests and ensuring continuity of essential services. Though full nominal capacity may not be immediately sustained, the system is designed to maintain operational availability until the failed site is restored.

A multi-site deployment usually does not affect the PDGS alone. **Prodigi** can be adapted to comply with the overall multi-site deployment architecture, connectivity details and recovery policies that apply to the ground segment.

7.3. DEPLOYMENT ARCHITECTURE

The deployment of **Prodigi** in each site is based on a containerized architecture orchestrated using Kubernetes, enabling modular deployment, infrastructure abstraction, and controlled scaling across the hardware resources.

The following figure illustrates the deployment of one site. Two instance pools are shown. The first, the Control Plane, hosts all services that support **Prodigi** operations. The second, the Processor Workers pool, is responsible for executing the various algorithms that generate the requested products. This pool can scale according to the mission's processing demands.

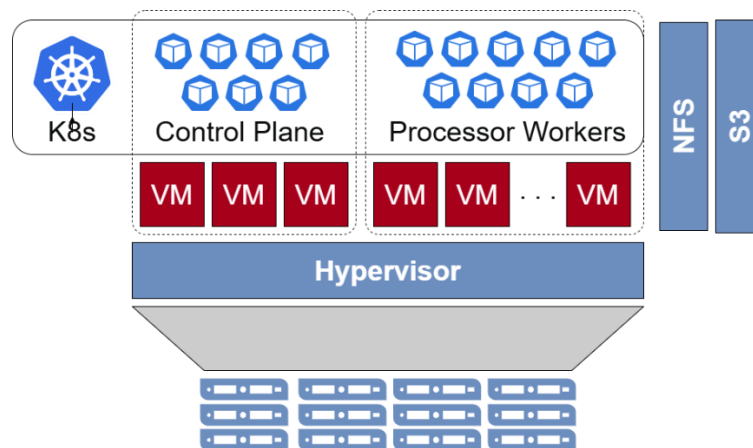


Figure 7-2: Typical deployment over infrastructure

The figure is assuming Kubernetes runs over a set of virtual machines in a typical on-premises or public cloud environment. **Prodigi** has been successfully deployed in vanilla Kubernetes over virtual machines or real hardware, Rancher distributions and managed Kubernetes clusters in cloud providers.

High availability within each site is achieved through container-level redundancy and Kubernetes-native resilience mechanisms. Other capabilities provided e.g. by a cloud provider (separation in different availability zones, etc) can add up to the high availability concept. Service components can be replicated across multiple nodes, ensuring that node-level failures do not interrupt processing operations. Automatic restart policies and health checks maintain service continuity without manual intervention.

The architecture supports fault isolation at multiple levels, including container, node, and site. Failures affecting a specific processing workflow or hardware resource do not propagate to unrelated workflows or to other sites.

7.4. DEPLOYMENT TOOLS

System installation and upgrade is streamlined through the use of containerized deployment and infrastructure-as-code principles. **Prodigi** is deployed using Kubernetes and Helm charts, enabling:

- Rapid and reproducible deployment of the full system stack.
- Consistent configuration across environments.
- Baselining of the deployment configuration.
- Compatibility with CI/CD approaches for the full ground segment.
- Downtime reduction in case of maintenance or disaster recovery scenarios, avoiding complex manual procedures.

8. TECHNOLOGY STACK

The technology stack used by **Prodigi** is continuously under review, using LTS versions whenever possible and applying security updates whenever necessary. A full orientation to cloud-compatible technologies while allowing multiple deployment scenarios is reflected in our COTS list:

- Ubuntu is the standard distribution for all our images.
- Kubernetes provides the resource orchestration layer.
- PostgreSQL and PostGIS provides us with persistence and geographical capabilities to search data.
- RabbitMQ and compatibility with AMQ for communication of our services.
- Keycloak and Kong to secure the access.
- HayStac to provide STAC interface to our catalogue.
- FullCalendar (licensed, new license required only if customer edits code) to improve aspects of our UI when representing schedules.
- Vector to collect and distribute logs.
- Mapbox (licensed, pay per use) to provide geocoding and tiles to our maps.
- OpenWeather (licensed, pay per use) to allow weather forecasting, clouds representation and filtering.
- Helm to install **Prodigi** anywhere.

END OF DOCUMENT