This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101082230.

Validation strategy and results of the demonstration for Coverage Service

Grant Agreement number	101082230
Call identifier	HORIZON-CL4-2022-SPACE-01
Project Acronym	DOMINO-E
Project title	Earth Observation Multi-mission federation layer
Funding Scheme	Horizon Europe Innovation Action (IA)
Project Starting date	01/11/2022
Project Duration	36 months
Project Coordinator	Airbus Defence and Space
Deliverable related number and full name	D27 Demonstration of the CS
Delivery Date	17/10/2025
Issue	**V1.0**
Document produced by	Marie Devant - Capgemini
Document verified by WP Leader	Stéphane Derrien - Capgemini
Document authorised by Project Coordinator	Airbus Defence and Space
Dissemination Level	SEN

* Please indicate the dissemination level using one of the following codes:
PU = Public,
SEN = Confidential, only for members of the consortium (including the Commission Services).

Abstract

This document presents the validation strategy and the results that has been set up to validate the Coverage Service (WP3000). It specifies the environments in which tests have been performed, several test scenarios and their inputs as well as the strategy of comparison of each algorithmic solution.
The document is consolidated with the results of the demonstrations.

List of participants

Participant No	Participant organisation name	Country
1	Capgemini Technology Services	FR

Applicable documents

Ref. / Document Title	Ref	Date
Domino-E Description of Work	Version 1	29/09/2022
Domino-E Grant Agreement 101082230	Ares(2022)7032529	11/10/2022
Domino-E Consortium Agreement	VF	14/11/2022
Domino-X Architecture Document	DOMX-DD-ADST-1001804669 V2	06/07/2023

Reference documents

Ref. / Document Title	Ref	Ext. Ref	Date
E2E Scenario Report	RD1		
DOMINO-E Mathematical problems to be solved for coverage management	RD2	D3.3 version 1.1	31/01/2024
DOMINO-E Algorithms for cover management	RD3	D3.4 version 1.1	15/03/2024
DOMINO-E Use Cases Definition	RD4	D2.2 version 3.0	16/01/2024
DOMINO-E Instances of scenarios, scenario scheduler and analysis of tests for coverage management algorithms	RD5	D3.5 version 2.0	24/01/2025

Document Change Record

Issue	Date	Page / paragraph affected
V1.0	08/05/2024	Document creation
V1.1	03/06/2024	Add Occitania / France coverage test scenario

Table of Contents

List of Tables

List of Figures

1 INTRODUCTION

This document aims at describing the validation strategy that has been set up to test the Coverage Service (WP3000).

This validation of the Coverage Service seeks to achieve several objectives.
First, the validation phase tests all the available functionalities of the Coverage Service:

- Creation, update, save and follow up of a User Request,
- Launch of coverage dispatch and reassessment algorithms,
- Creation and update of Programming Request for Sovereign and External Missions,
- Launch of completion algorithms for each User Request
- Launch Feasibility analysis for each Missions.

Secondly, the validation phase compares the performances of the different algorithmic solutions in various contexts.

1.1 Structure of the document

The document is organised as follows:

- this chapter gives an overview of the document and a reminder of the objectives of the Coverage Service application,
- chapter 2 explains the vocabulary used,
- chapter 3 specifies the environment in which tests have been performed,
- chapter 4 describes all the KPI that have been used to assert the effectiveness of the algorithmic solutions,
- chapter 5 lists all the test scenarios that have been performed to test Coverage Service functionalities,
- chapter 6 describes all the test scenarios that have been performed for algorithmic comparison,
- chapter 7 delivers an analysis of the result as well as improvements aiming at an operational usage of the Coverage Service
- chapter 8 contains optional content the analysis relies on.

1.2 Coverage Service – Reminder

The Coverage Service is an application which aims at optimizing the time of coverage for a large area. It provides an optimized split of an area amongst a set of available Mission Chains.

As described in precedent deliverables (RD4), four Use Cases have been developed to test different dispatch solutions:

- Use Case 1a: The User Request area is manually dispatched before being submitted. It will be used as a comparison point for other Use Cases.
- Use Case 1b: The User Request area is automatically dispatched at the submission **only**, between all the Mission Chains available.
- Use Case 1c: This Use Case is an extension of the Use Case 1b. The dispatch at submission is performed, as well as a periodic re-dispatch, considering all the still activated User

Requests. It can change the initial dispatch to match the updated inputs of the processing.

- Use Case 2: No initial dispatch is performed for Use Case 2, only a periodic dispatch considering all the activated User Requests. This dispatch algorithm focuses on the next access available for a given area, by opposition to the Use Case 1, where an estimation of the completion date of the request is done.

The algorithms are described in RD3.

2 Definition and Acronyms

Definition and acronyms can be found in the Domino-X Glossary included in the [RD01].

2.1 Acronyms

The following table lists some additional acronyms used in this document:

Acronym	Meaning
AOI	Area of Interest
DB	Database
KPI	Key Performance Indicator
MC	Mission Chain
UAS	User Access Service
UR	User Request

Table 1 : Acronyms

2.2 Terms

The following table lists some additional terminology used in this document, and its context:

Term	Meaning	Context
Mission Chain	Part of the Ground Segment in charge of the planning of requests	For each constellation of satellite, a Mission Chain is deployed to manage the planning of all satellites of the constellation.
PROPhET	An algorithm, developed by ADS, to estimate the coverage completion time of a Programming Request	This algorithm has been used to estimate the time of completion for Programming Request that has not been completed at the end of tests.
ProgR	Programming Request	A User Request will be transformed into a list of ProgR during dispatch (Mono-Mission). Those ProgR will be deposit with Mission Chain to be planned.
Sovereign / External	Sovereign/External Mission Chain	A Sovereign Mission Chain is a Mission Chain from which a larger amount of information is available to make the dispatch more precise.
UAS Simulator	Simulator of the User Access Service	The UAS is the interface through which client can submit User Request
UR	User Request	Request of planning at a Federation level (Multi-Mission)
Workload	The workload of a Mission Chain is the list of ACTIVATED Programming Requests that already have an impact on the planning.	During dispatch or coverage completion assessment, the algorithms need to know the load of each Mission Chains to evaluate their impact on the completion of the User Request they are dispatching.

Table 2 : Terminology

3 ENVIRONMENT OF TESTS

In this chapter, we will describe the different environments in which tests have been performed.
To achieve all the objectives described during the introduction, two environments of tests have been set up.

3.1 Realistic approach

As a first approach, tests have been performed in a realistic environment, composed of:

- The Coverage Service component,
- The UAS Simulator,
- Two realistic Mission Chains filled with test data,
- An adapter for the interfaces between Coverage Service implementing DOMINO-E interfaces and MC implementing operational interfaces.

Having the response of two real Mission Chains implementation allows a more accurate comparison of the different algorithmic solutions.

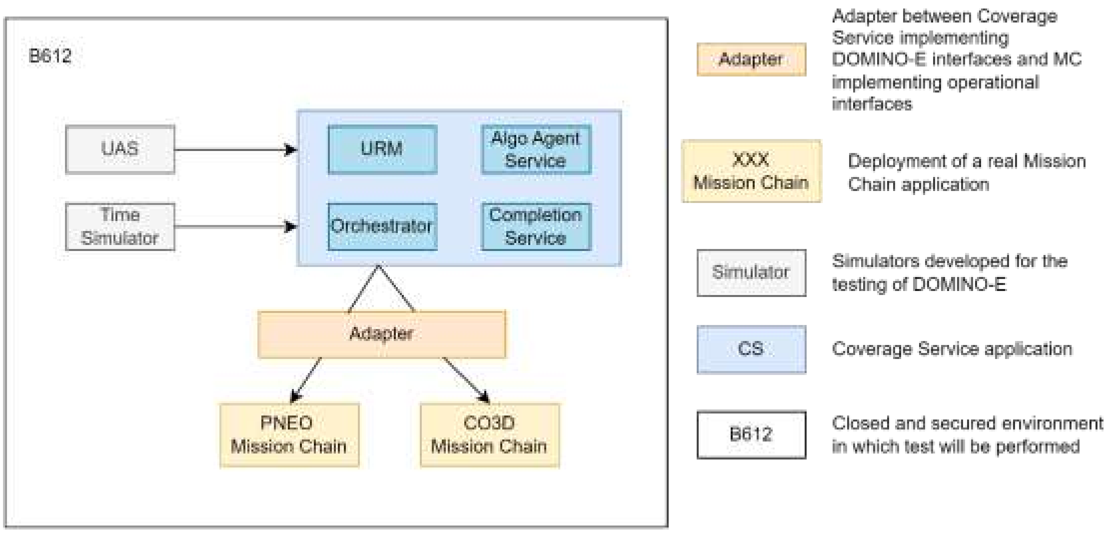


Figure 1 : Schema of the realistic testing environment

3.2 Demonstration approach

The real Mission Chains application cannot be deployed in a cloud environment for confidentiality reasons.
Thus, once tests would have been performed a first time in the realistic environment, response interfaces have been saved to be used in mocked Mission Chains in the demonstration environment: the tests bed described in WP 6000.

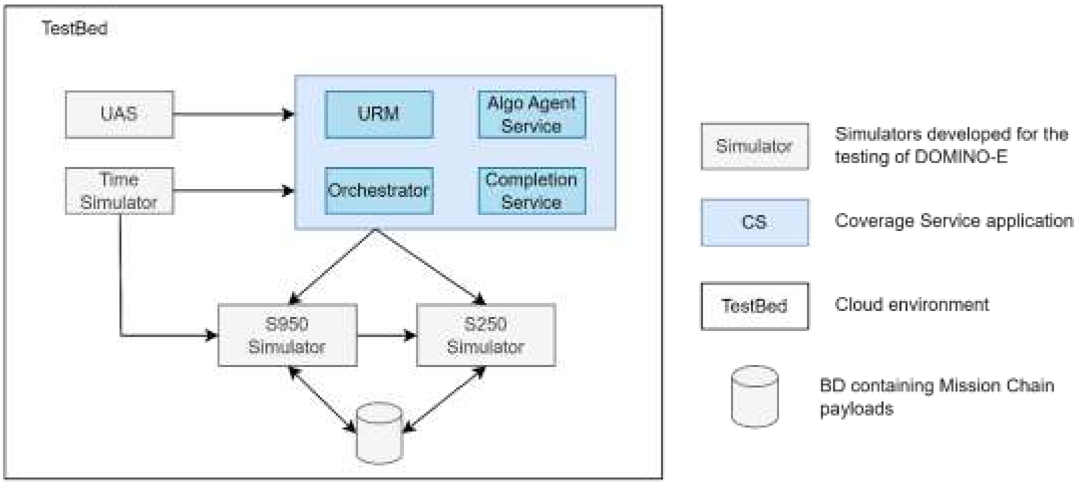


Figure 2 : Schema of the demonstration testing environment

3.3 Long-term simulation approach

In the previous approach, the mission chain simulators return precomputed execution results, that would have been extracted from a realistic environment, with the real mission chains. Using the real mission chains imposes a constraint on the horizon of test: running plan computation with two deployed mission chains is not possible during multiple months because of resources need.
As some specific test on large area needs a horizon of several weeks, another mode of simulation has been implemented on the mission chain simulator. We used probabilistic computation to simulate the planning of the mission chains.

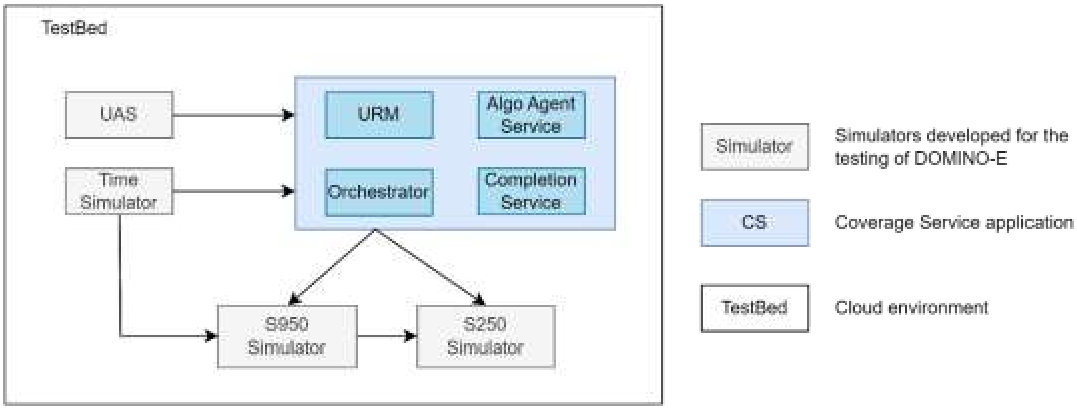


Figure 3 : Schema of the long-time testing environment

4 METHOD OF COMPARISON OF ALGORITHMIC SOLUTIONS

To compare the algorithmic solutions of dispatch, we defined a list of indicators that have been retrieved during tests to be compared. Those KPIs are described here after and numbered to be referenced in further descriptions.

4.1 Description of KPIs

4.1.1 KPI#1: Time of completion of the User Request

The Time of completion of the User Request corresponds to the time a User Request takes to reach 90% and then 100% of coverage.
The Completion of the User Request is also retrieved at the end of tests if it didn't reach 100%.

4.1.2 KPI#2: Area Waste

The Area Waste is, for one User Request, the value (in percentage) of area that will be either programmed more than once or programmed while not contained in the original AOI of the User Request.

4.1.3 KPI#2bis: Area Waste with Mission Chains planning optimization

The Area Waste can be mitigated with several Mission Chains mechanisms that optimize the planning of the requested area: when an area is requested more than once, Mission Chains detect the case and plan the area only once.
Only complex Mission Chains will have this functionality, we therefore decided to keep both KPI#2 and KPI#2bis in the analysis of our results.

4.1.4 KPI#3: External calls

The number and the volume of necessary emitted calls are retrieved per interface and per receiver. Those external calls are the creation, cancellation and retrieval of Programming Requests amongst each Mission Chains.

4.1.5 KPI#4: Time of computation

The Time of computation is the time needed by the algorithmic solutions to perform the expected computation. A mean of all the algorithms will be computed.

4.1.6 KPI#5: Resource usage

The Resource usage (CPU and memory) for each micro-service of the Coverage Service is retrieved.

4.1.7 KPI#6: Load balancing on Mission Chains

The Load Balancing helps qualifying the neutrality of the algorithmic solutions in relation to Mission Chain choices. We want to determine if the algorithmic solutions favours one or another of the Mission Chain during the dispatch choice.

4.2 KPIs retrieval

4.2.1 KPI#1: Time of completion of the User Request

The completion of a User Request is computed by the coverage service and the time of completion is retrieved by tests.

4.2.2 KPI#2 + KPI#2bis: Area Waste

The Area Waste is computed at the end of each test by retrieving all the submitted Programming Request and intersecting them with the User Request AOI.

4.2.3 KPI#3 + KPI#5: External calls and resource usage

The number of external calls and the resource usage is retrieved thanks to Prometheus monitoring.

4.2.4 KPI#4: Time of computation

The Time of computation is retrieved by the tests, based on emitted logs. The time retrieved during the test is the complete duration of the process of dispatch or coverage assessment. In the demonstration environment, the algorithmic computation time represents between 98% or 99% of the complete duration of the call.

4.2.5 KPI#6: Load balancing on Mission Chains

This KPI takes the form of a ratio between area submitted on MISSION2 and area submitted on MISSION1. Those areas will be retrieved through logs emitted by the application.

4.3 KPIs exploitation

4.3.1 KPI#1: Time of completion of the User Request

The value of coverage Completion is retrieved at the end of the test and will be used for comparison.

4.3.2 KPI#2 + KPI#2bis: Area Waste

We decided to split the Area Waste KPI on smaller and more understandable sub-KPIs.
The first one is the **"Outside Area Waste"**, which is the area planned whereas they were not contained inside the initial User Request AOI.
Another sub-KPI would be the area that are taken more than once inside the User Request initial AOI: the **"Intra Area Waste"**.

The latest sub-KPI is a derivative of the previous one: it is the area taken by more than one Mission Chain. Some Mission Chains can identify if an area has been submitted more than once and therefore plan it only once during plan computation. The MISSION2 and MISSION1 Mission Chains are both able to do so. Therefore, the last sub-KPI we use is the area that is taken by more than one Mission Chain, suppressing from the last KPI the area submitted on the same Mission Chain: the **"Inter Missions Area Waste".**
Finally, the #KPI2 displayed in the summary table of each scenario is the addition of the "Outside Area Waste" and the "Intra Area Waste". The #KPI2bis is the addition of the "Outside Area Waste" and the "Inter-Missions Area Waste".

4.3.3 KPI#3 External calls

The number of calls for each type of actions are retrieved for each Mission. It is displayed as a table, and the addition of all those calls is displayed in the summary table.

4.3.4 KPI#4: Time of computation

We computed an average of computation time for each type of algorithmic actions:

- [UC1] Dispatch: Complete time of call to the *findOptimizedSubdivision* algorithmic method (including the input/output mapping and algorithmic call)
- [UC1] Unitary Coverage Completion: Mean on a Mission of all the time of call to the *callCompletionAssessmentOnSys* algorithmic method (including the input/output mapping and algorithmic call)
- [UC1] Global Coverage Completion Estimation: Time of Coverage Completion Estimation on all the ACTIVATED Programming Requests. It includes the time of retrieval of the ACTIVATED Programming Requests in the database, the workload retrieval time, the call to *callCompletionAssessmentOnSys* on each Programming Request and the save of the results. Time of call to *callCompletionAssessmentOnSys* represents in average 99,9% of the complete time of calculation.
- [UC1][UC2] Redispatch: Time of Redispatch on all ACTIVATED User Requests. It includes the time of retrieval of the ACTIVATED User Requests, the time of retrieval of the Workload, the time of call to the algorithmic method (*updateDispatch* or *reAssessOptimizedSolution*), updates according to the results (cancel of creation of Programming Requests amongst Mission Chains), sent of the contact-needs to SCRMS. For UC2, the time of call to *updateDispatch* represents between 96% to 99% of the global time. For UC1, the time of call to *reAssessOptimizedSolution* represents 99,9% of the global time

In the summary table of each scenario, the Global Coverage Completion Estimation computation and Redispatch are displayed when Use Case allows it.

4.3.5 KPI#5: Resource usage

That information is retrieved on the demonstration environment and displayed under a graphic format.

4.3.6 KPI#6: Load balancing on Mission Chains

To try to quantify this KPI we decided to retrieve for each dispatch call the number of activation and cancellation on each Mission. Those values can only be analysed considering the capacity of the satellite (the acquisition speed) as well as the number of satellites used for the simulation.
Therefore, we computed a "perfectly balanced" ratio according to the satellite's characteristics.
For MISSION1, the *numberOfSatellite*=4 and *acquisitionSpeed*=4.3km²/s
For MISSION2, the *numberOfSatellite*=2 and *acquisitionSpeed*=13km²/s

$$ratio_{balanced} = \frac{acquisitionSpeed_{MISSION2} * numberOfSatellite_{MISSION2}}{acquisitionSpeed_{MISSION1} * numberOfSatellite_{MISSION1}} = \frac{26}{17{,}2} = 1.5$$

We then computed, for each dispatch, the ratio of the MISSION1 planned area and the MISSION2 planned area. If this ratio is close to the $ratio_{balanced}$ then we can consider the dispatch as balanced.
This ratio must be considered with caution, as the value must be mitigated with other parameters that can influence the planning and prediction (weather forecast, orbit pass, conflict with other mission manoeuvres…) and that are not considered in the $ratio_{balanced}\cdots$

5 TEST SCENARIOS FOR BASIC FUNCTIONALITIES TESTING

5.1 Creation and visualisation of User Requests

5.1.1 Goal and expected result.

The goal of this test is to create a User Request and visualize it with the UAS simulator to check the core functionalities of the Coverage Service.

5.1.2 Inputs

- The User Request to submit
- Initialized Mission Chains simulators containing the splitting of the User Request by real Mission Chains

5.1.3 Major Steps

- Through the UAS Simulator, create a User Request
    - [UC1.a] with linked ProgRs
    - [UC1.b.c / UC2] or without
- Visualize the User Request on the UAS Simulator to check that it has been correctly saved with the split returned by the Mission Chains.

5.1.4 Results

It is possible to create a User Request via form or file import. The maps of the left part of the screen allows the display of the Area Of Interest to be acquired. On the right side of the screen, all the acquisitions parameters can be filled, for each Mission.

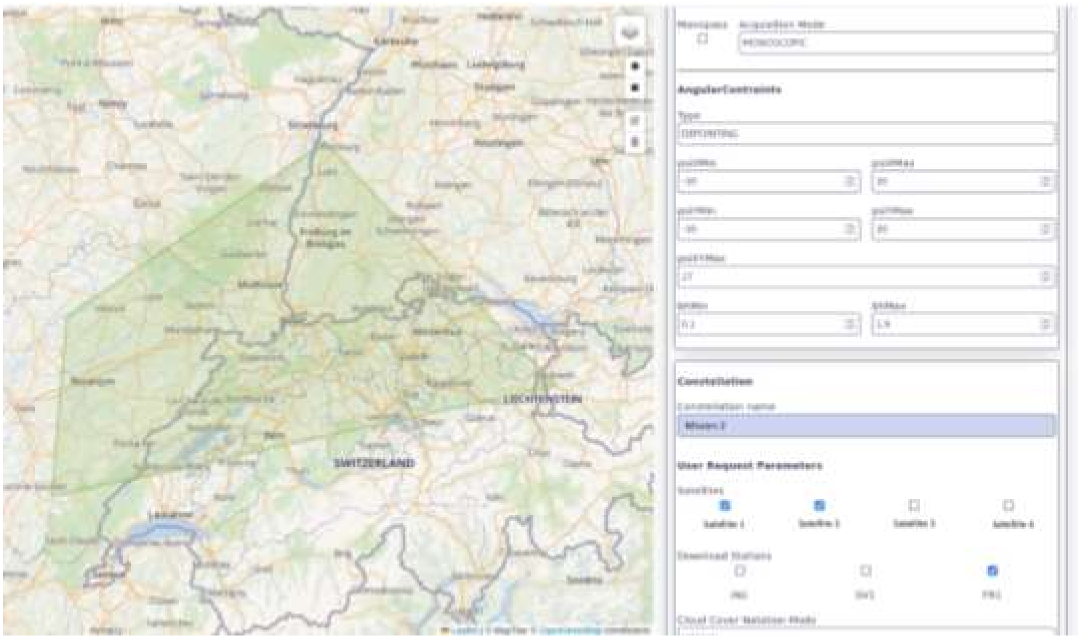


Figure 4 : Creation of User Request

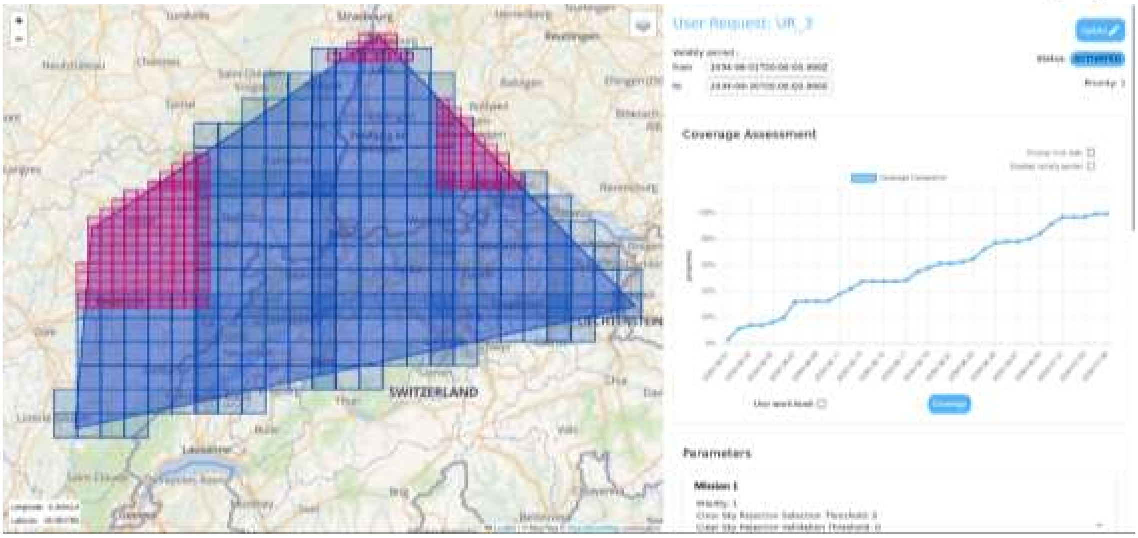


Figure 5 : Creation of User Request result

Once the User Request is submitted, it is dispatched among the Mission Chains. In the upper example, the area has been dispatched between MISSION1 Programming Requests (in red) and MISSION2 Programming Requests (in blue). Those areas are then split into smaller area to be acquired by the Mission Chains, the Acquisitions. A Programming Request is therefore composed of several Acquisitions, that are more easily plannable by the Mission Chain.
On the right-most part of the screen, several parameters of the User Request are displayed: submission parameters, list of Programming Request for each Mission, list of Acquisitions for each Programming Request, Coverage Completion charts…
In the case of Use Case 2, only the real coverage completion is displayed. For the Use Case 1, the real coverage is displayed as well as an estimation of future completion.

5.2 Feasibility analysis

5.2.1 Goal and expected result.

A feasibility analysis is used to retrieve the splitting information of each Mission Chain. It can help the user to find the most pertinent User Request submissions.
The Coverage Service should be able to launch an intrinsic feasibility analysis on each Mission to help the operator during deposit phase. While requesting an analysis on the UAS Simulator, the Coverage Service should call all the Sovereign Missions with the User Request in creation and display the result on the UAS Simulator.

5.2.2 Inputs

- The User Request to analyse
- The Mission Chains simulator containing the analysis result of a real Mission Chain

5.2.3 Major Steps

- Through the UAS Simulator, fill the needed fields for a User Request
- Manually launch an analysis on this User Request and visualize the result

5.2.4 Results

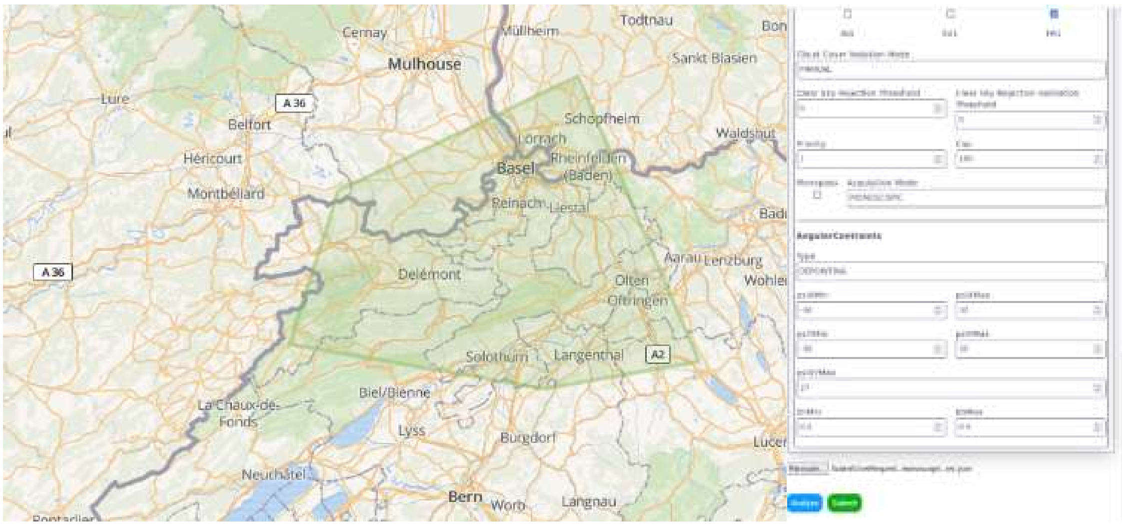


Figure 6 : Analysis Input

In the upper schema, the form has been filled with all the parameters useful for a User Request creation. The result of this analysis is displayed here after:

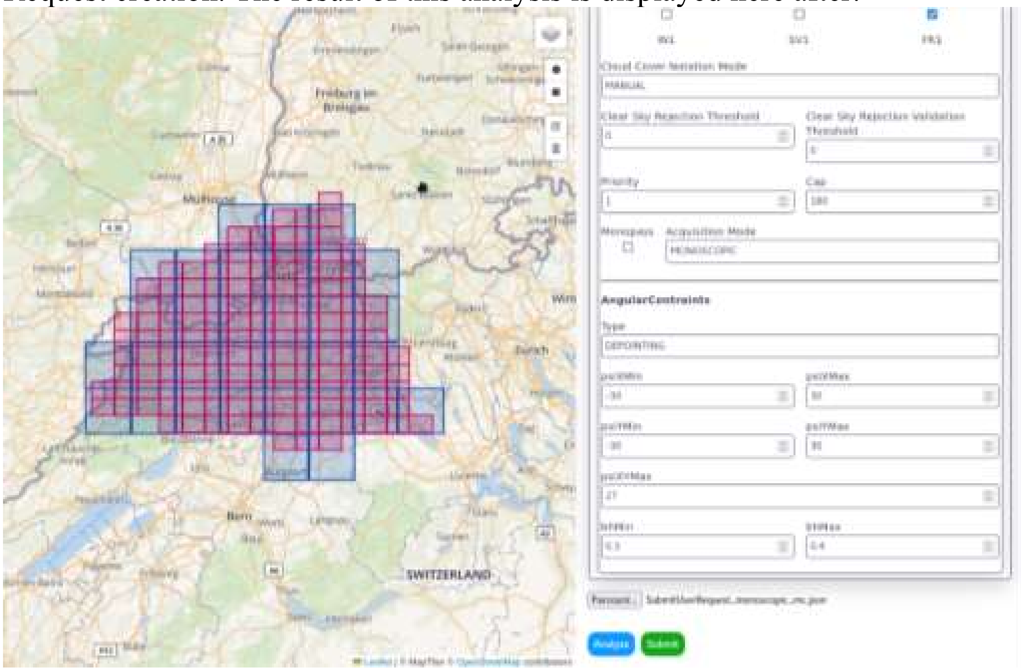


Figure 7 : Analysis result

The blue grid corresponds to the MISSION2 split while the red grid corresponds to the MISSION1 split. This view can be filtered on the answer for one or another of the Missions to avoid the interface clustering, as shown in the next screenshots.

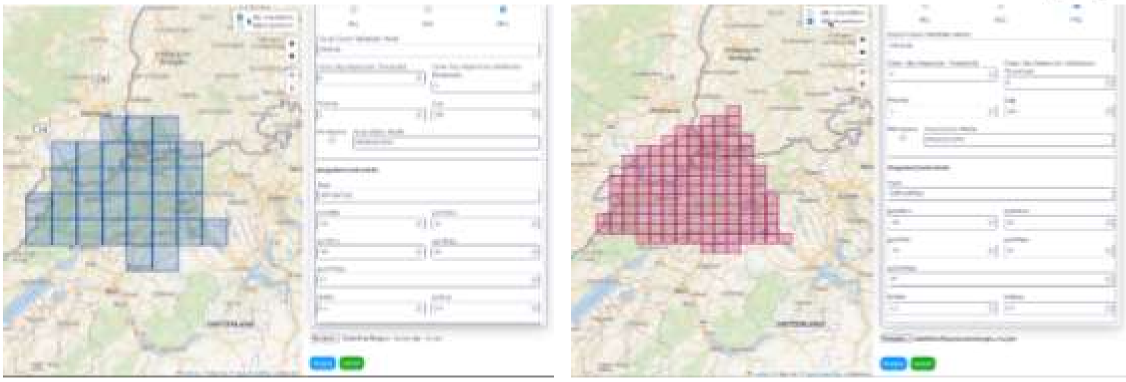


Figure 8 : Analyse - view per Mission

In a more advanced version of this functionality, the Mission Chains will be able to provide the access opportunity in addition to the split. The information of access opportunity can be used by the user to decide on which Mission Chains User Requests can be submitted on. We can also imagine calling the coverage estimation algorithms to provide advancement for each Mission Chains to facilitate the decision-making process.

5.3 Update of a User Request

5.3.1 Goal and expected result.

The Coverage Service should be able to manage an update of the User Request and propagate this modification to all the linked ProgRs on each Mission Chains.

5.3.2 Inputs

- A Coverage Service with an already submitted User Request

5.3.3 Major Steps

- Update the priority of the User Request through the UAS Simulator
- Visualize the User Request on the UAS Simulator to check the update

5.3.4 Results

The User Request is submitted as shown in the following screen:

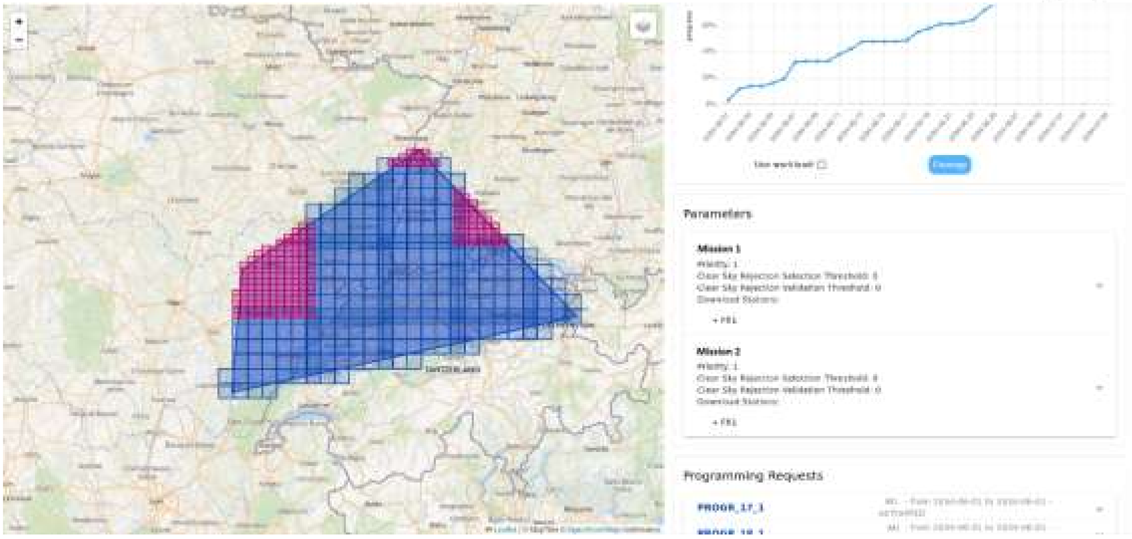


Figure 9 : Update initial state

Only a few parameters can be modified on the User Request and the Mission parameters linked. We change the MISSION2 mission priority and clear sky rejection threshold. Those two parameters have an impact on the planning estimation and therefore we expect the update to have an impact on the dispatch. The priority is reduced, and the planning constraints are stronger, we expect that the next dispatch will disadvantage the MISSION2 Mission Chain.

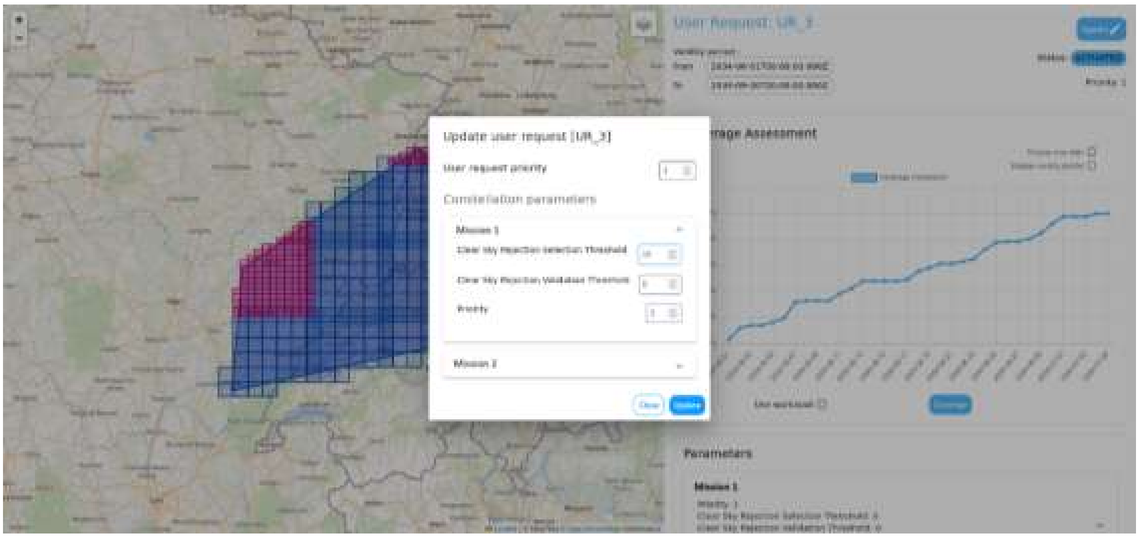


Figure 10 : Update Processing

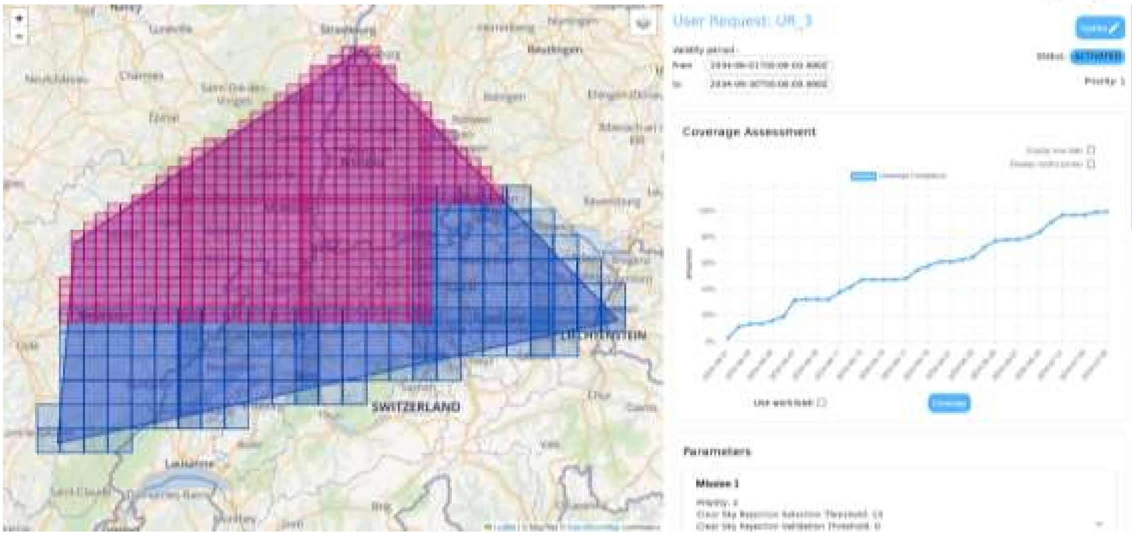


Figure 11 : Update final state

After the update, some of the MISSION2 Programming Requests have been cancelled to be replaced with MISSION1 Programming Requests.

5.4 Manual Follow up request

5.4.1 Goal and expected result.

The follow-up functionality allows the Coverage Service to retrieve the up-to-date progression of the planning of the Programming Request on each Mission. The Coverage Service should be able to launch a follow up of the User Requests on any ProgR linked to it.

5.4.2 Inputs

- An already submitted User Request
- Mission Chains simulators initialized with some COMPLETED acquisitions

5.4.3 Major Steps

- Launch a follow-up on the already submitted User Request
- Check that the COMPLETED acquisitions are updated

5.4.4 Results

The following User Request has been submitted and dispatched on the MISSION2 and MISSION1 Mission Chains. The MISSION1 Mission Chains has planned some of the acquisitions and they will return as COMPLETED when the follow up will be called.

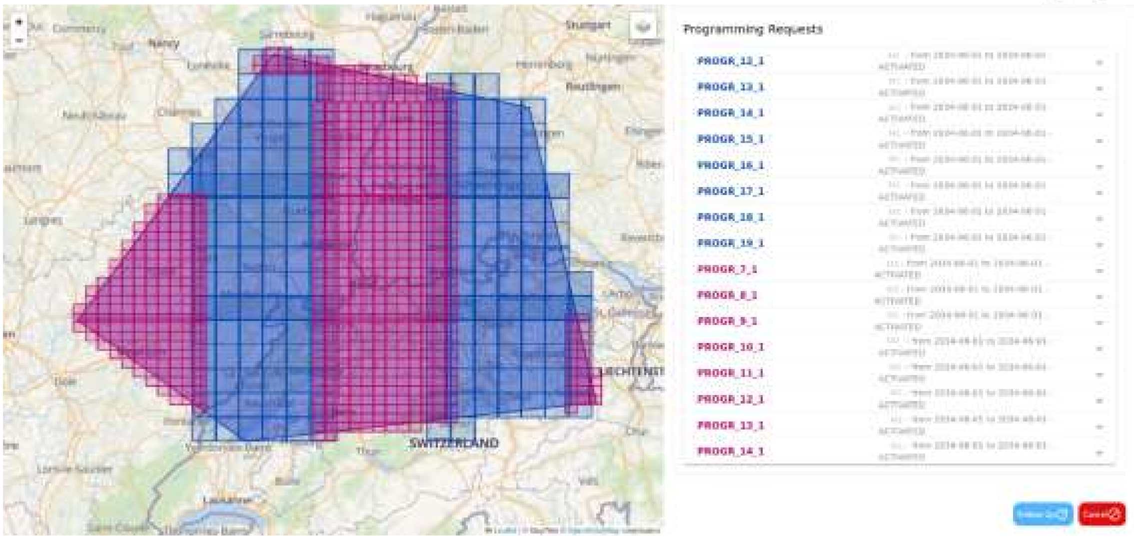


Figure 12 : Follow Up initial state

The COMPLETED acquisitions are displayed in a darker colour on the UAS Simulator. It also updates the real coverage completion on the charts on the right. The next Use Case will be to update the estimation of Coverage Completion once the planning information are updated.

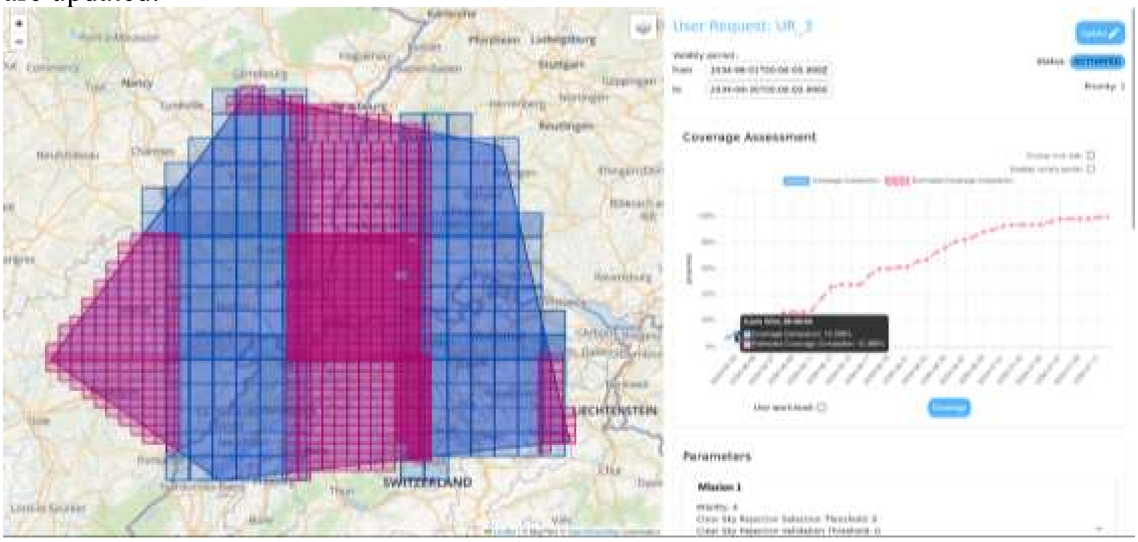


Figure 13 : Follow Up Final State

5.5 Coverage Completion Computation

5.5.1 Goal and expected result.

The Coverage Completion Computation is available only on Use Case 1. It computes the estimation of Coverage Completion for the User Request, according to the up-to-date information.
The goal of this test is to check if the manual launch of computation of the coverage completion works as expected.

5.5.2 Inputs

- An already submitted User Request with some acquisitions already COMPLETED

5.5.3 Major Steps

- Manually launch a coverage completion
- Visualize the User Request on the UAS Simulator to check the result of the completion coverage computation

5.5.4 Results

In the following screenshot, we see an already submitted User Request and its ProgRs. The coverage estimation charts forecast the completion of the User Request on the 30/06/2034.

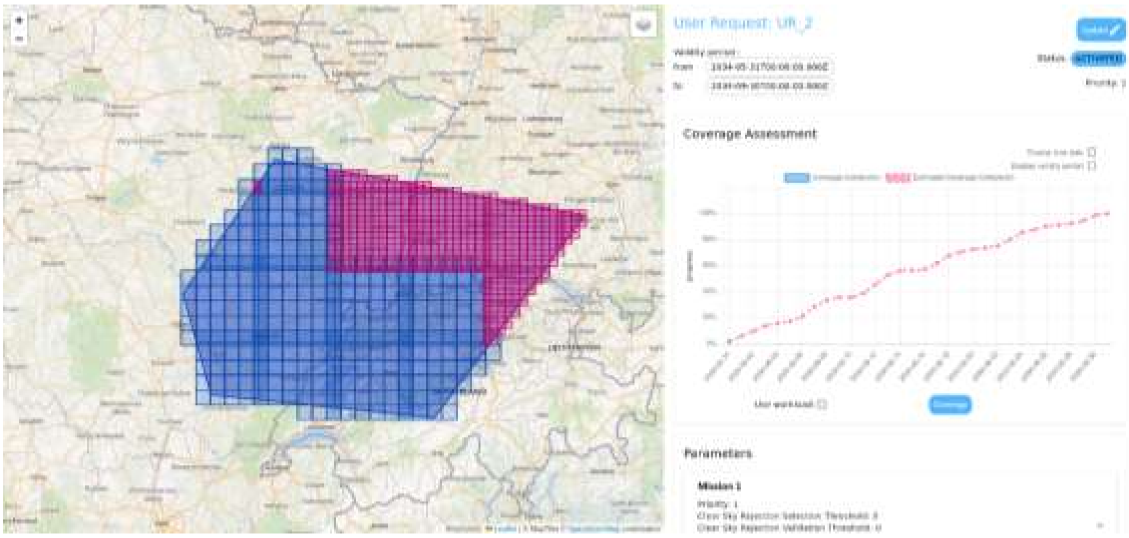


Figure 14 : Manual Coverage Completion - Initial state

In a second time, after a day of planning, some of the acquisitions of the ProgR are completed. We then launch a follow up on this User Request. The coverage completion charts are updated with a “real” coverage completion. The estimation remains the one of the last coverage completion computations.

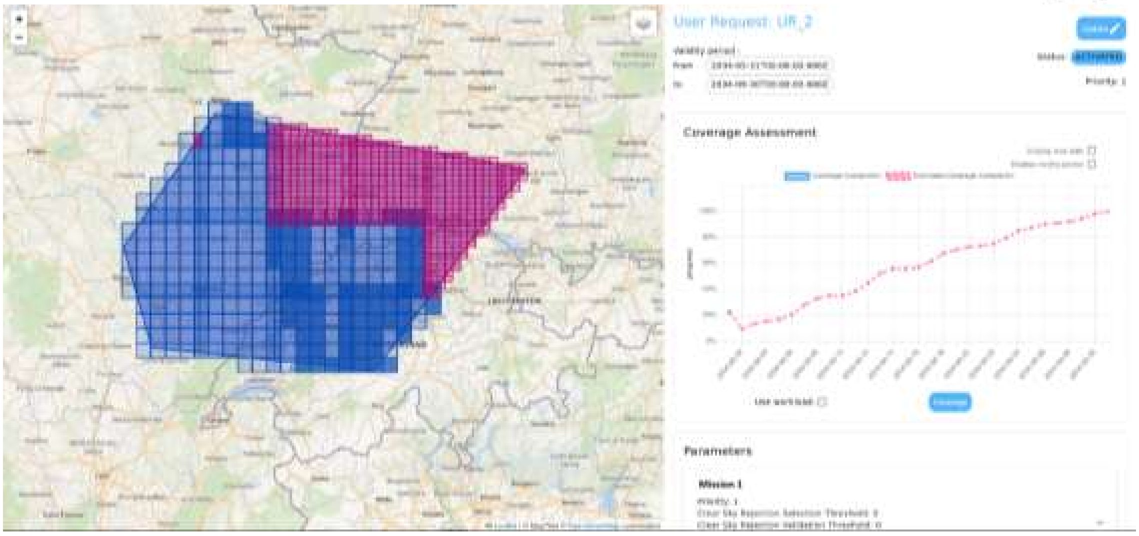


Figure 15 : Manual Coverage Completion after Follow-up

After a call to the coverage completion estimation service, the estimated curve is updated to consider the latest "real" coverage completion computed thanks to the follow up information. We then see this curve starting to the real coverage completion point, on the 01/06/2034.

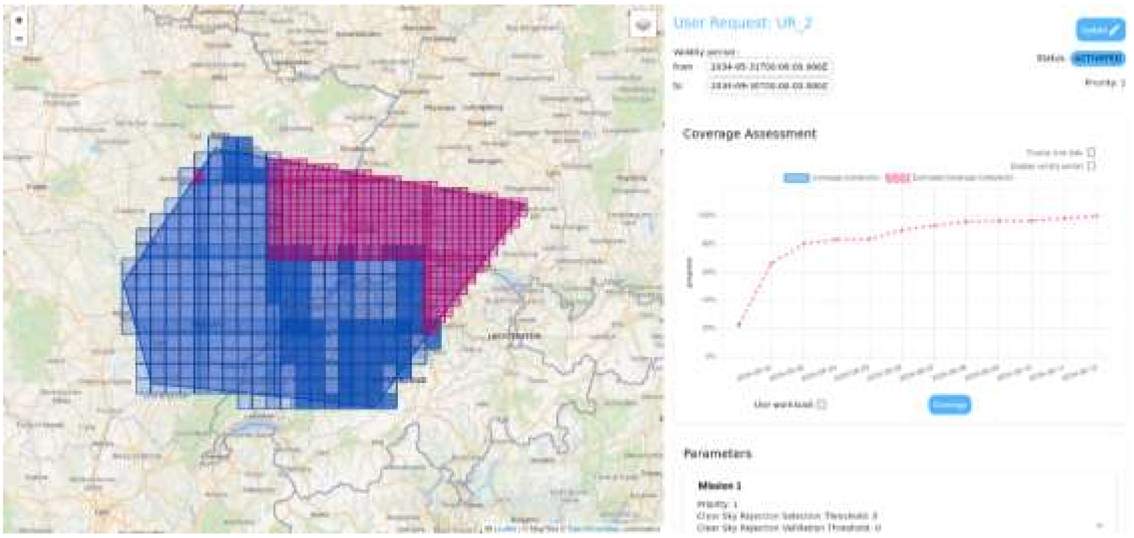


Figure 16 : Manual Coverage Completion - Final State

6 TEST SCENARIOS FOR ALGORITHMIC COMPARISON

The following chapter describes several scenarios that have been run with all the algorithmic Use Cases and allows the comparison of the result in different context.

6.1 Occitania coverage

6.1.1 Goal and expected result

The goal of this test is to check the completion progression on a unique User Request, with all sensors available (M2_1, M2_2, M1_1, M1_2, M1_3, M1_4) and on an empty Mission Chain context. To achieve it, one mission plan computation per day is launched on each Mission for 10 days.
At the end of the test, we will be able to compare all KPIs including the time of completion between the different algorithmic solutions.

6.1.2 Inputs

- The orderbook: User Request covering Occitania region of France
- The MISSION2 Mission Chain context:
  - Not loaded / empty
  - 1 plan per day
- The MISSION1 Mission Chain context:
  - Not loaded / empty
  - 1 plan per day

6.1.3 Major Steps

By default, all those steps are executed in every UC. Some specific steps are specified with [UC*].

- Create the User Request
  - [UC1.a] Deposit the UR and a ProgR corresponding to the half of the AOI on M2_1 and M2_2 and the rest on M1_1, M1_2, M1_3 and M1_4.
  - [UC1.b.c / UC2] Deposit the UR without linked ProgR.
- [UC2] Shift time to execute the first dispatch call (planned at 23:00 each day)
- Every day:
  - Shift time to execute the first MISSION2 Mission Chain plan computation
  - Shift time to execute the first MISSION1 Mission Chain plan computation
  - Set to COMPLETED all the ProgR planned by the MISSION2/MISSION1 Mission Chain
  - Shift time to execute the next Follow up activities (planned at 17:00 each day)
  - [UC1] Shift time to execute the next Coverage Completion Computation activities (planned at 20:00 each day)
  - [UC1c/UC2] Shift time to execute the next Redispatch activity (planned at 23:00 each day)

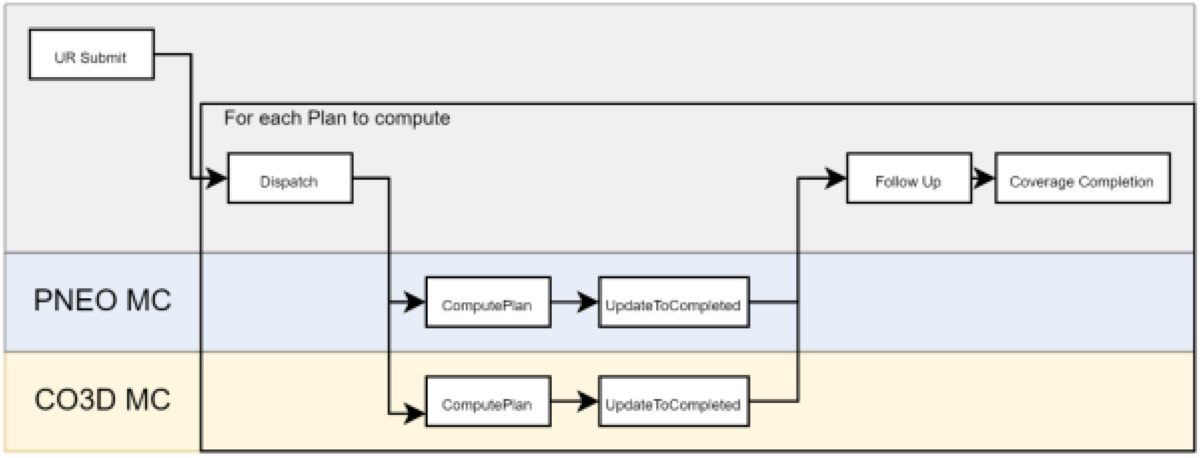


Figure 17 : Chronology of Test Scenario for single User Request

6.1.4 Results

6.1.4.1 KPI extraction

	UC1a	UC1b	UC1c	UC2
KPI#1 *Coverage Completion*	49.08%	57,26%	54,63%	52,89%
KPI#2 *Area Waste*	21,3%	50,2%	56,29%	21,23%
KPI#2bis *Area Waste with MC optimization*	11,26%	22.0%	29,05%	16,7%
KPI#3 *External calls*	85	264	327	5191
KPI#4 - Global Coverage Completion Estimation *Computation time*	75.9s	162.56s	150.02s	None
KPI#4 – Global Redispatch *Computation time*	None	None	37.37s	44.44s
KPI#6 *Load Balancing*	0.2	1.2	1.7	1.3

Table 3 : KPIs extraction for Occitania

6.1.4.2 Observations & Explications

6.1.4.2.1 KPI#1 Scenario progression

6.1.4.2.1.1 Use Case 1a

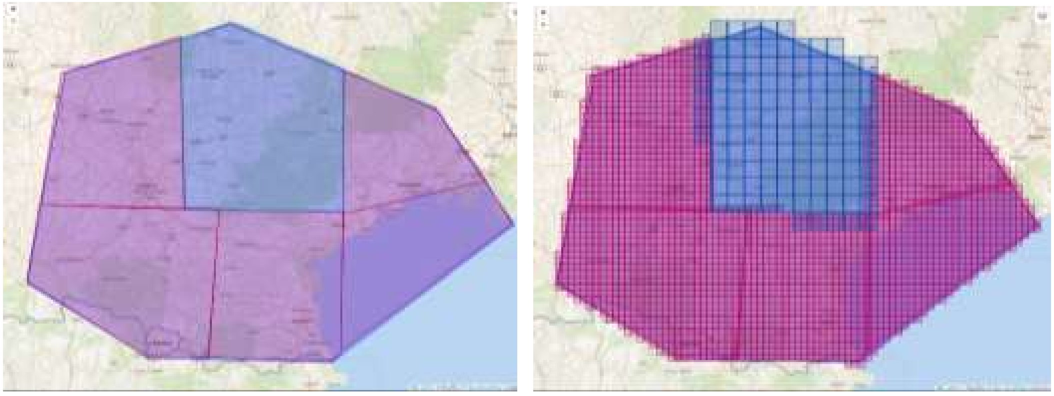


Figure 18 : Dispatch of the Occitania Area by Use Case 1a

The red subareas have been submitted on the MISSION1 Mission Chain whereas the blue subareas have been submitted on the MISSION2 Mission Chain.
On the right-most screenshot, the internal split of the Mission Chains of the Programming requests into acquisitions have been represented. In the left-most, only the Programming Requests are displayed.
At the end of the 10th mission plan, the planned areas are displayed with a darker colour on the following schema:

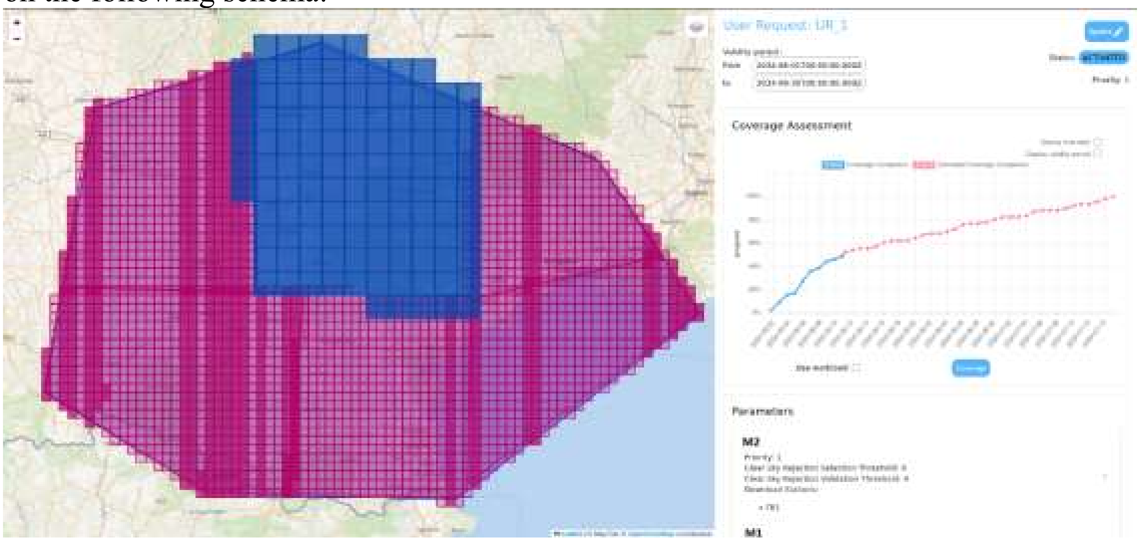


Figure 19 : Completion of the Occitania after the 10th plan for Use Case 1a

The rightmost schema represents the completion of the User Request. At the end of the test, **49,08%** of the initial AOI has been planned. The red dotted line is an estimation of the coverage completion until the end of the coverage. This functionality is only available with the ADS algorithmic solution.

6.1.4.2.1.2 Use Case 1b

The following schema is the dispatch of the Occitania Area by the algorithms of the Use Case 1b and 1c:

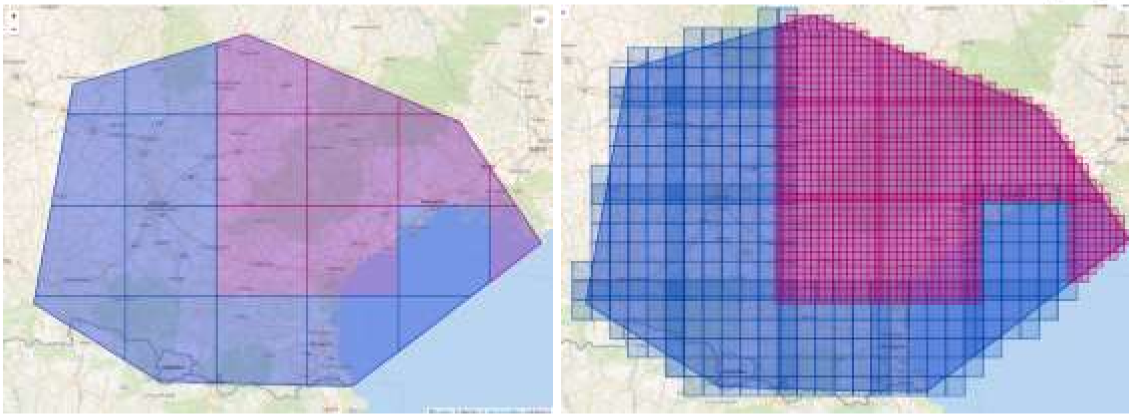


Figure 20 : Dispatch of the Occitania Area by Use Case 1b

The following schema is the final status of the coverage at the end of the test.

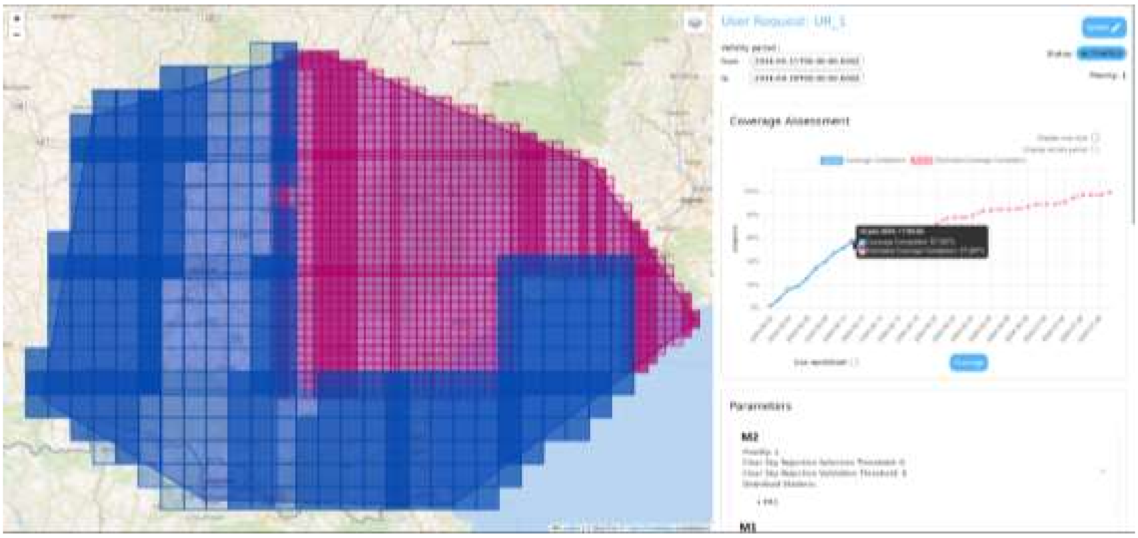


Figure 21 : Completion of the Occitania after the 10th plan for Use Case 1b

The coverage at the end of the 10 days is **57,26%,** which is higher than the first manual dispatch of Use Case 1a.
We see that the PNEO project is on the bridge to be completed faster than the CO3D one. On the UC1b, the dispatch is definitive. The Use Cases with redispatch option (UC1c and UC2) represent a possibility to reallocate part of the CO3D project on PNEO will be completed.

6.1.4.2.1.3 Use Case 1c

Without surprise, the initial dispatch is the same as in Use Case 1b. On the other hand, the final status at the end of the test is different, due to the periodic redispatches.

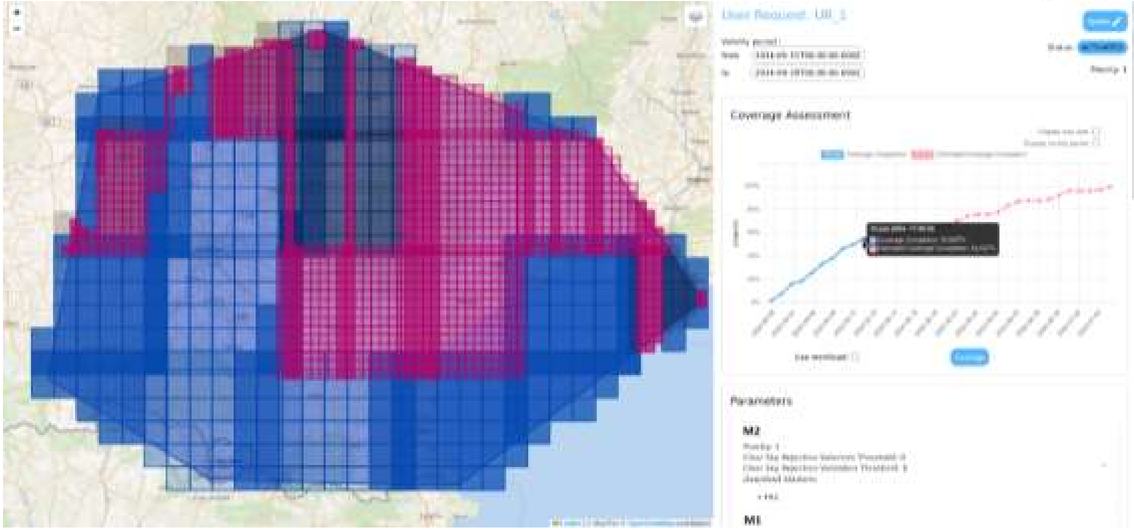


Figure 22 : Completion of the Occitania after the 10th plan for Use Case 1c

The black areas correspond to the Programming Request that have been cancelled during a dispatch but that cannot be simply erased because a part of this request was already planned. At the end of the 10th day, the coverage completion is **54,63%.**
As it is complex to understand the progression of the dispatch with only the representation of the final state, the complete progress of the test can be found in Annexe 8.2.1.

We can also see that the coverage completion after the 10th plan is greater in the Use Case 1b and Use Case 1c. However, the estimation of complete coverage is planned on the 04/07/2034 for Use Case 1c instead of 08/07/2034 for Use Case 1b. The Use Case 1c adapts to the real planning progression to reduce the time to 100% of coverage.

6.1.4.2.1.4 Use Case 2

The shape of the Programming Requests dispatched by the Use Case 2 follows the Mission Chain splitting, it splits the request into bands of a certain number of cells instead of squares, as in Use Case 1.

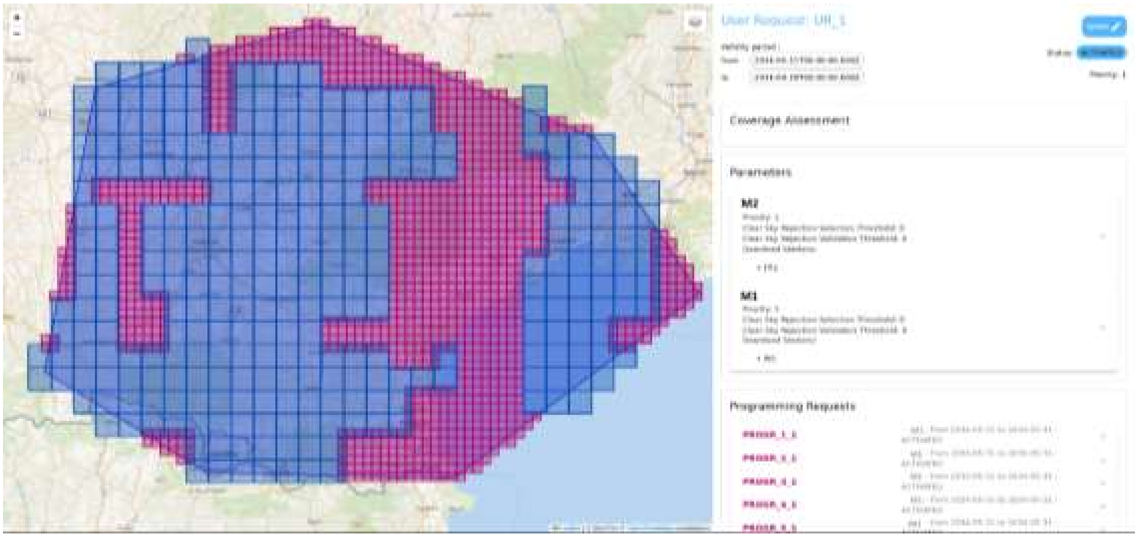


Figure 23 : Dispatch of the Occitania Area by Use Case 2

The completion estimation is not available with the Use Case 2, in the following figure the coverage at the end of 10 days is displayed. Following the same legend than for Use

Case 1c, the darker area is the one that have been planned by Mission Chains and black area are the ones cancelled by a redispatch.

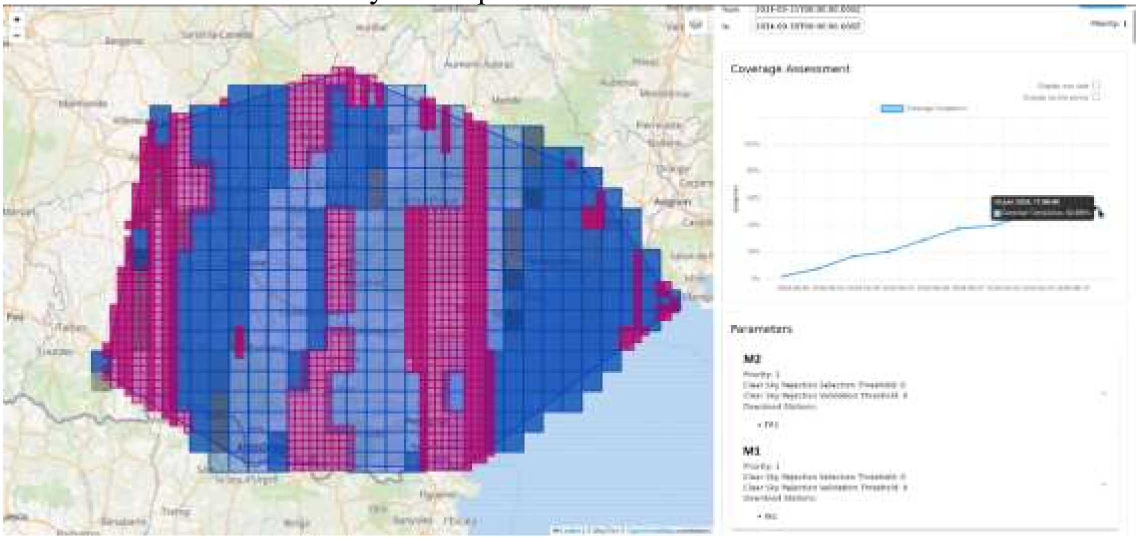


Figure 24 : Completion of the Occitania after the 10th plan for Use Case 2

At the end of the 10 days of test, the coverage completion is **52,89%.**

6.1.4.2.2 KPI#2 Area Waste

In the next sub-section, we will geographically display the representation of all the sub-KPIs described in the section 4.3.2.

6.1.4.2.2.1 Use Case 1a

The “Outside Area Waste” at the end of the test is **5,84%**.

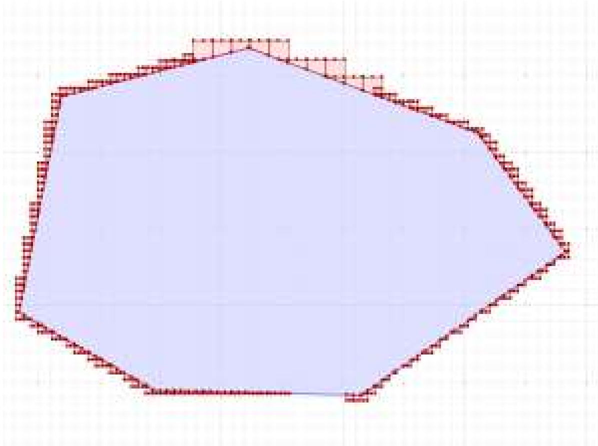


Figure 25 : Outside Area Waste Occitania Use Case 1a

The “Intra Area Waste” at the end of the test is **15,46%**.

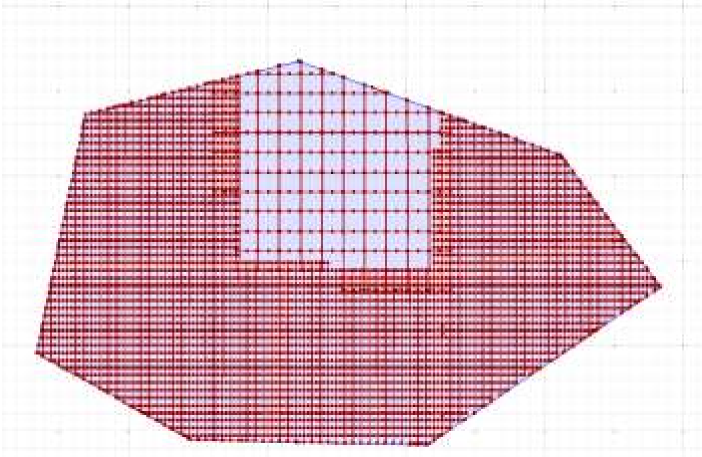


Figure 26 : Intra Area Waste Occitania Use Case 1a

The “Inter-Mission Area Waste” at the end of the test is **5,42%**.

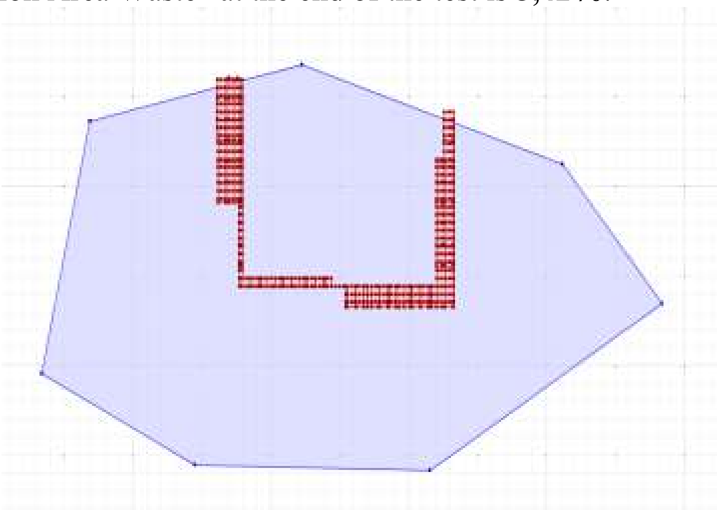


Figure 27 : Inter-Missions Area Waste Occitania Use Case 1a

6.1.4.2.2.2 Use Case 1b

The following schema gives an overview of this “Outside Area Waste” for the Occitania Area and the Use Case 1b. It represents **11.1**% of the global Area.

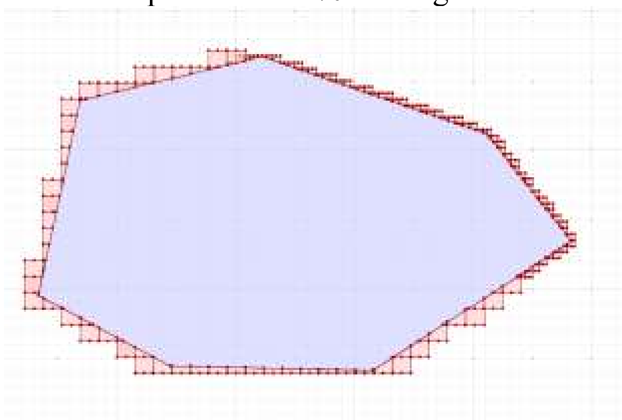


Figure 28 : Outside Area Waste Occitania Use Case 1b

In the following schema, we see, in red, the area taken at the same time by MISSION1 and MISSION2, but also the area taken more than once by the same Mission Chain: the “Intra Area Waste”.

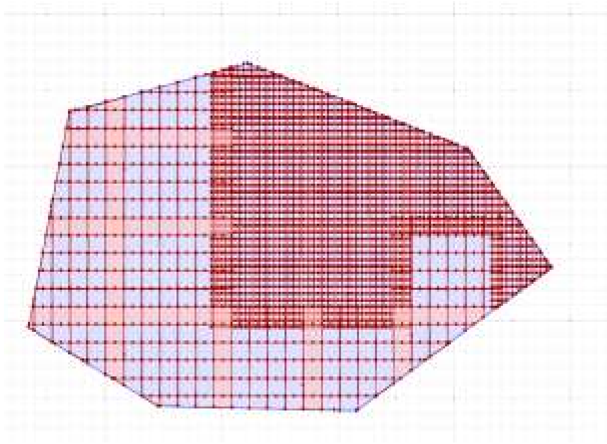


Figure 29 : Intra Area Waste Occitania Use Case 1b

The thin lines, creating a grid on the area, are the margin taken by the Mission Chain around each acquisition during the submission: this area waste is inevitable as it ensures that the complete area will be taken by the satellite without border hazards.
Overall, this area waste represents **39.1**% of the global User Request Area.

The next schema displays the case "Inter Missions Area Waste" for the Occitania Area on Use Case 1b:

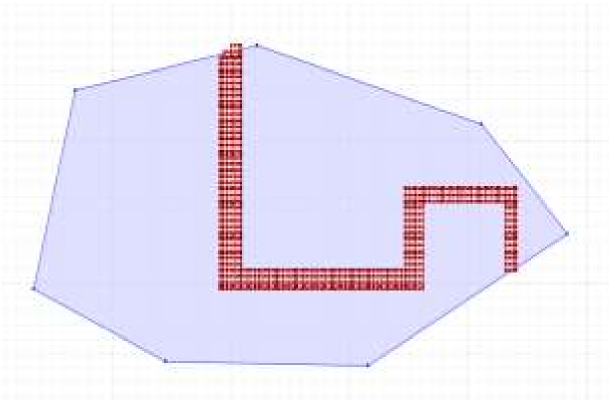


Figure 30 : Inter-Missions Area Waste Occitania Use Case 1b

This area waste represents **10.9**% of the global Area.

6.1.4.2.2.3 Use Case 1c

The equivalent analysis for Use Case 1c is displayed here after. The values have been retrieved at the end of the test.

The "Outside Area Waste" at the end of the test is **12,16%**.

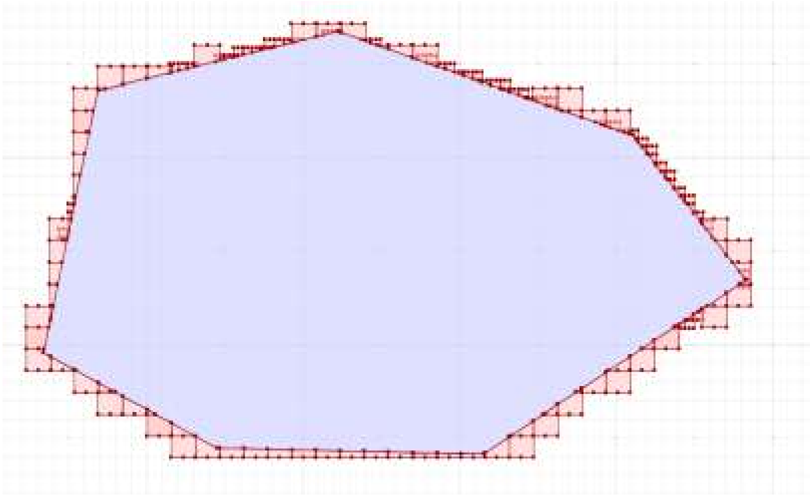


Figure 31 : Outside Area Waste Occitania Use Case 1c

The “Intra Area Waste” at the end of the test is **44,13%**.

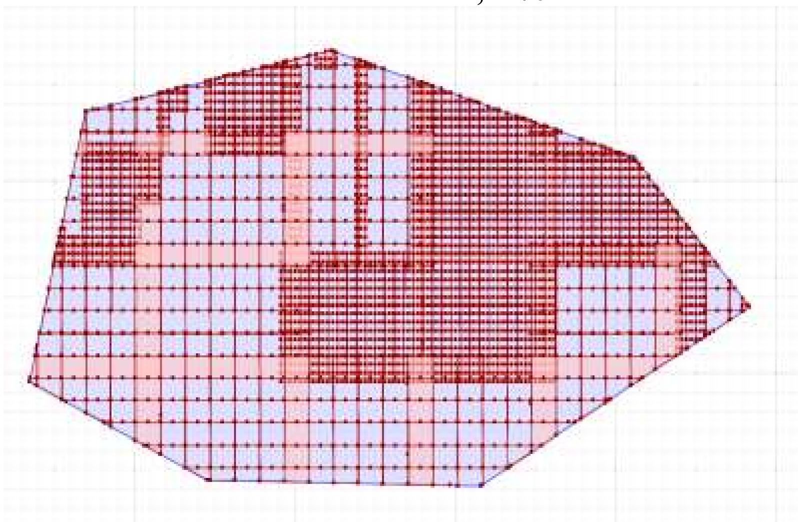


Figure 32 : Intra Area Waste Occitania Use Case 1c

The “Inter-Mission Area Waste” at the end of the test is **16,89%**.

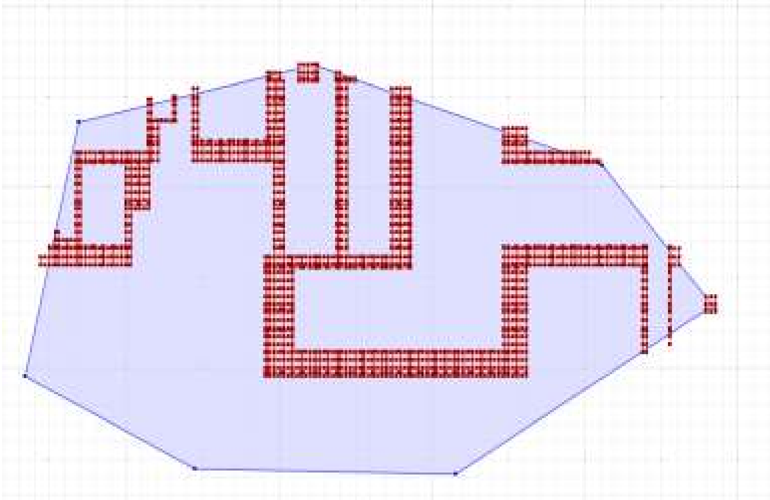


Figure 33 : : Inter-Missions Area Waste Occitania Use Case 1c

6.1.4.2.2.4 Use Case 2

The equivalent analysis for Use Case 2 is displayed here after. The values have been retrieved at the end of the test.

The “Outside Area Waste” at the end of the test is **8,93%**.

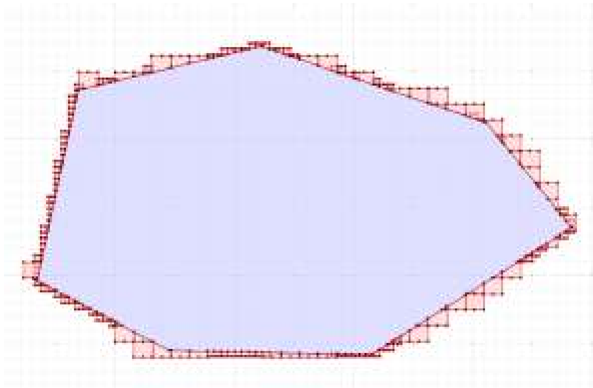


Figure 34 : Outside Area Waste Occitania Use Case 2

The “Intra Area Waste” at the end of the test is **12,30%**.

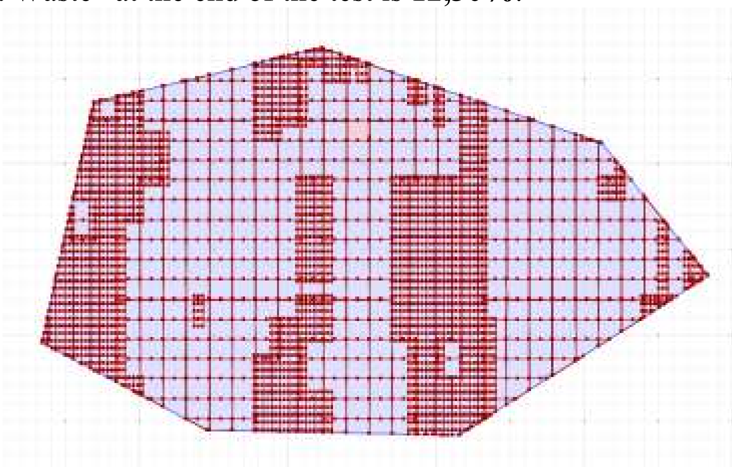


Figure 35 : Intra Area Waste Occitania Use Case 2

The “Inter-Mission Area Waste” at the end of the test is **7,77%**.

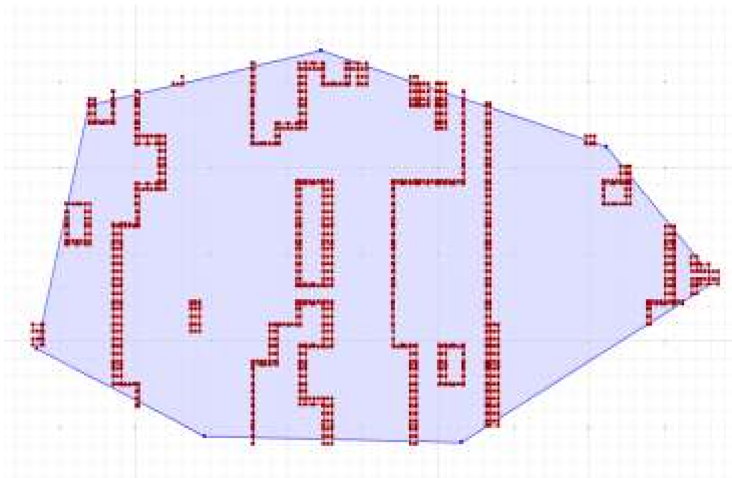


Figure 36 : Inter-Missions Area Waste Occitania Use Case 2

6.1.4.2.2.5 Conclusion

The Use Case 1 split is not completely aligned with the Mission Chains splitting; it therefore generates an important area waste, which is reduced when we take in consideration the capacity of the Mission Chains to optimise their own planning. At the opposite, the Use Case 2 takes as an input the splitting of the Mission Chains, the area Waste is therefore reduced thanks to this implementation solution.

6.1.4.2.3 KPI#3 Mission Chains solicitation

To dive further into the details of the KPI#3 External calls, we retrieved the number of calls realized of each type on each Mission Chains.

6.1.4.2.3.1 Use Case 1a

	CREATE	CANCEL	FOLLOW-UP	WORKLOAD	ALL
MISSION1	5	0	50	11	**66**
MISSION2	1	0	10	8	**19**

Table 4 : Mission Chain solicitations Occitania UC1a

In this Use Case, the subdivision is manually performed. We decided, arbitrary, to dispatch on 6 areas distributed over the two Mission Chains. No redispatch is performed, there are no CANCEL actions and the only CREATE we witness are the one of the submissions.
One FOLLOW-UP is performed for each Programming Request once a day and the WORKLOAD is retrieved before each Coverage Assessment.
We can see that only 8 workloads are retrieved on MISSION2 as the only Programming Request submitted on this Mission is COMPLETED during the 8th plan (1 workload for submission and 7 for the coverage reassessment of the 7 first days). An improvement of the Coverage Service could be to also suspend the follow-up for the COMPLETED Programming Request: we can see there are still 10 follow-ups whereas only 7 were needed.

6.1.4.2.3.2 Use Case 1b

	CREATE	CANCEL	FOLLOW-UP	WORKLOAD	ALL
MISSION1	10	0	100	11	**121**
MISSION2	12	0	120	11	**143**

Table 5 : Mission Chain solicitations Occitania UC1b

The first dispatch is equally distributed between the two mission chains. A FOLLOW-UP is performed once a day before the Coverage Reassessment computation. The WORKLOAD is retrieved once at the initial DISPATCH and once per Coverage Reassessment computation.

6.1.4.2.3.3 Use Case 1c

	CREATE	CANCEL	FOLLOW-UP	WORKLOAD	ALL
MISSION1	13	5	105	21	**144**
MISSION2	17	3	142	21	**183**

Table 6 : Mission Chain solicitations Occitania UC1c

In the Use Case 1c chronology, the WORKLOAD is retrieved once on the initial Dispatch, once per Coverage Reassessment computation and once per Redispatch.
Some Programming Requests are cancelled on a Mission Chain to be planned on the other. Therefore, the CANCEL actions on a Mission Chain are found as CREATE on the other Mission Chain.

6.1.4.2.3.4 Use Case 2

	CREATE	CANCEL	FOLLOW-UP	WORKLOAD	ALL
MISSION1	870	650	2502	10	**4032**
MISSION2	261	194	694	10	**1159**

Table 7 : Mission Chain solicitations Occitania UC2

The subareas dispatched by the Use Case 2 are smaller than the subareas of the Use Case 1. Therefore, a significative number of requests are sent to the Mission Chains, whether during submission or later redispatches.

6.1.4.2.4 KPI#4 Time computation

6.1.4.2.4.1 Use Case 1a

	Mean (in seconds)
Global Coverage Completion Estimation computation	75.98067971
MISSION2 – Coverage Completion Estimation computation	1.78137573
MISSION1 – Coverage Completion Estimation computation	14.93525905

Table 8 : Time computation Occitania UC1a

During a Global Coverage Completion Estimation computation, a Completion Coverage is computed for each Programming Request. It contains the time used to retrieve the workload amongst the Mission Chain and the time of computation of coverage completion estimation for all Programming Requests.
In the two last lines is an average of the time of computation for a unique Programming Request for each Mission. The time of computation for the Programming Request of MISSION2 is lower than the time for MISSION1 because of the number of acquisitions contained in each Programming Request: the MISSION1 acquisition size is smaller than the MISSION2 ones, and therefore the MISSION1 Programming Request contains more acquisitions than a MISSION2, even if the size of the Programming Requests is the same.
The computation time of Use Case 1 depends on the number of acquisitions contained in the Programming Requests, more than the size of the AOI of the Programming Request directly.

6.1.4.2.4.2 Use Case 1b

	Mean (in seconds)
Global Coverage Completion Estimation computation	162.565174
MISSION2 – Coverage Completion Estimation computation	1.8559842
MISSION1 – Coverage Completion Estimation computation	14.302826
Dispatch	69.5852106

Table 9 : Time computation Occitania Uc1b

The dispatch time is the time used to call the Dispatch algorithms.

There is a unique dispatch launched at the submission of the User Request. On Use Case 1a, the Global Coverage Completion Estimation computation is two times faster than the Global Coverage Completion Estimation computation on the Use Case 1b. The time of computation is proportional to the number of Programming Requests more than the size of the User Request.

6.1.4.2.4.3 Use Case 1c

	Mean (in seconds)
Global Coverage Completion Estimation computation	150.0198374
MISSION2 – Coverage Completion Estimation computation	1.891682989
MISSION1 – Coverage Completion Estimation computation	14.37830345
Dispatch	70.65073778
Redispatch	37.37623108

Table 10 : Time computation Occitania Uc1c

We can see from the upper table that the time of computation of the initial dispatch is higher than the average time of computation of the redispatches. On each iteration, the area to redispatch is smaller as some acquisitions are planned. The mean computation for a redispatch is therefore lower than the initial dispatch time. In addition, the initial dispatch is starting from a blank page, when the redispatches are taking the last state of dispatch as an input: the time of computation is then reduced.

6.1.4.2.4.4 Use Case 2

	Mean (in seconds)
Redispatch	44.4402085

Table 11 : Time computation Occitania Uc2

The configuration of the Use Case 2 allows us to define the number of “destroy and repair” actions that will be performed by the ONERA algorithms during the dispatch computation. For equity purpose, this parameter has been tuned to approximatively match the ADS algorithms time of computation.
To be noted: on the Testbed, the time of submission of the Programming Request is negligible, as the Mission Chain simulators are used. On the realistic environment, this submission is longer and increase proportionally to the area size of the Programming Request.

6.1.4.2.5 KPI#5 Resources usage

For Use Case 1, we decided to deploy 6 replications of the ads-algo-agent containing the algorithmic solution able to perform Dispatch, Redispatch and Coverage Reassessment computation.
For Use Case 2, as the only call performed is the Redispatch call, that cannot be parallelized, only one replication of the onera-algo-agent is deployed.
The following charts will allow us to analyse the pertinence of those deployment choices as well as to detect performances problems.

6.1.4.2.5.1 Use Case 1a

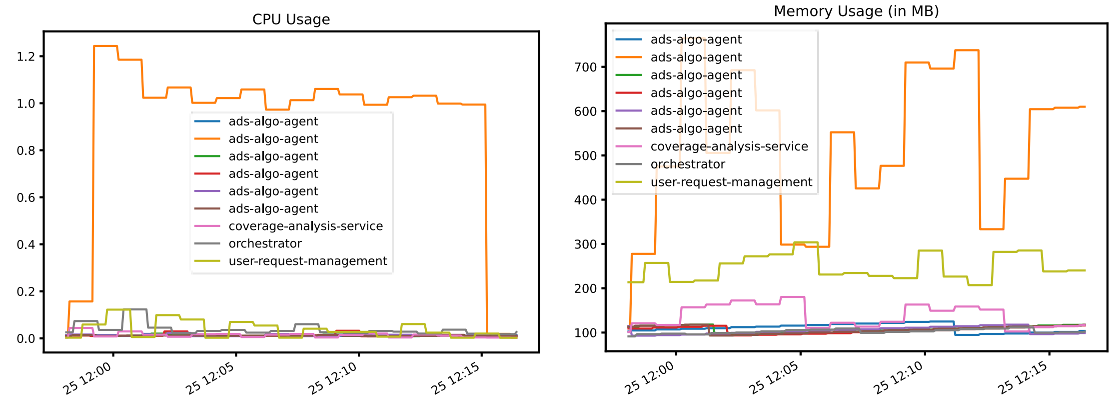


Figure 37 : Resources usage Occitania for Use Case 1a

All the CPUs and memory peaks we witness on a replication of the ads-algo-agent represents the launch of the Coverage Reassessment computation. As we see, only one agent is fully used: the only computation that are performed are the periodic call to the Coverage Reassessment method which are not performed in parallel.

6.1.4.2.5.2 Use Case 1b

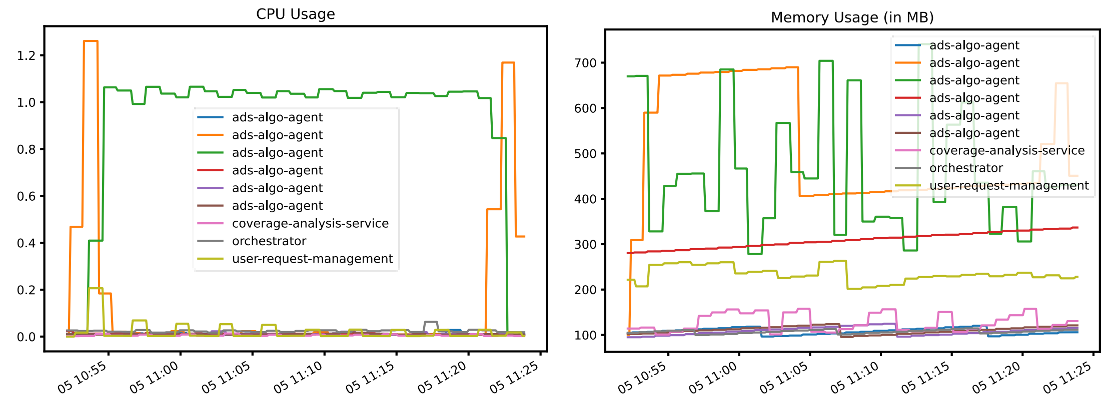


Figure 38 : Resources usage Occitania for Use Case 1b

The first important peak of resource consumption on the orange ads-algo-agent represents the first and unique Dispatch performed in the Use Case 1b, at the submission of the User Request. All the others are the periodic calls of the Coverage Reassessment. Here, the question of the replication of ads-algo-agent is still debatable: two replicas are used, even if the time of parallelisation is short.

6.1.4.2.5.3 Use Case 1c

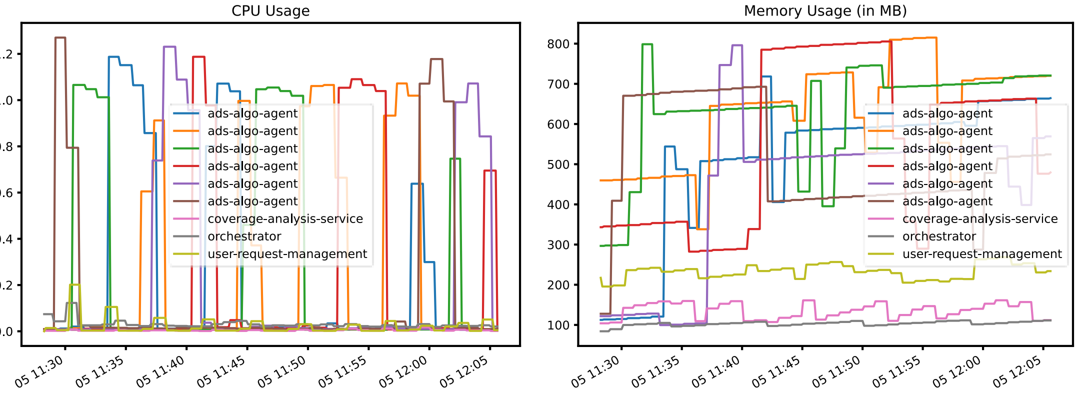


Figure 39 : Resources usage Occitania for Use Case 1c

The Use Case 1c chronology, a dispatch computation is performed at the User Request submission and Coverage Reassessment and Redispatch are periodically performed (once day).
In this Use Case, all the ads-algo-agent are used to perform all the calls needed but the overlapping time of computation is still short; further studies can be performed to define the number of replications of the ads-algo-agent that is optimal.

6.1.4.2.5.4 Use Case 2

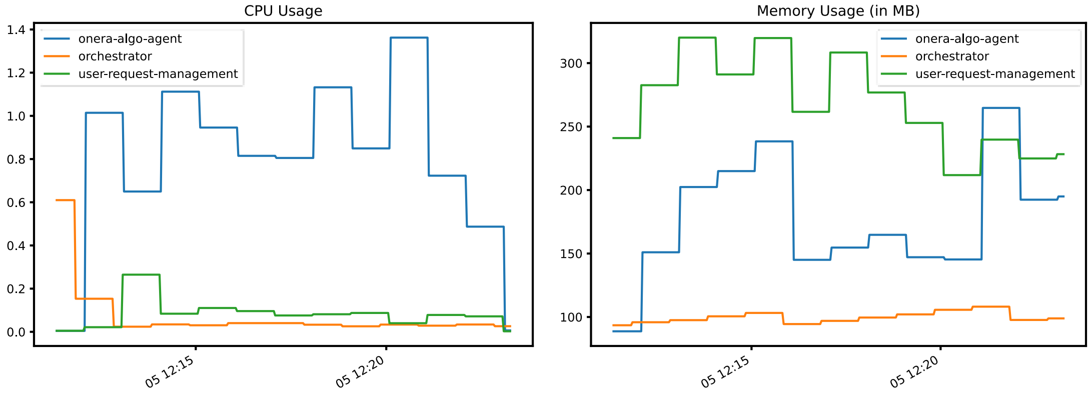


Figure 40 : Resources usage Occitania for Use Case 2

On the Use Case 2, only three components are deployed: the user-request-management, the orchestrator and the onera-algo-agent. The coverage-analysis-service, responsible of the launch of the Coverage Reassessment (only available on the ADS algorithmic solution), and the ads-algo-agent, containing the ADS algorithmic solution, are not deployed.
Comparatively to the ads-algo-agent, the onera-algo-agent consumes less memory (550 Mb in mean, compared to 177Mb). The CPU Usage is equivalent for the two algo agents.

6.1.4.2.6 KPI#6 Load balancing

6.1.4.2.6.1 Use Case 1a

	MISSION1		MISSION2		Ratio
	Activation	Cancel	Activation	Cancel	
Initial Dispatch	5	0	1	0	0.2

Table 12 : Load balancing Occitania Uc1a

This manual dispatch obviously favours MISSION1.

6.1.4.2.6.2 Use Case 1b

	MISSION1		MISSION2		Ratio
	Activation	Cancel	Activation	Cancel	
Initial Dispatch	10	0	12	0	1.2

Table 13 : Load balancing Occitania Uc1b

On this first dispatch, only the creations of Programming Requests are sent to the Mission Chains. There is no cancellation as there is no redispatch activated.

6.1.4.2.6.3 Use Case 1c

	MISSION1		MISSION2		Ratio
	Activation	Cancel	Activation	Cancel	
Initial Dispatch	10	0	12	0	1.2
Redispatch 1	1	1	1	1	1.2
Redispatch 2	0	0	0	0	1.2
Redispatch 3	1	2	2	1	1.7
Redispatch 4	0	1	1	0	2.1
Redispatch 5	1	0	0	1	1.7
Redispatch 6	0	0	0	0	1.7
Redispatch 7	0	0	0	0	1.7
Redispatch 8	0	1	1	0	2.1
Redispatch 9	0	0	0	0	2.1
Redispatch 10	0	0	0	0	2.1

Table 14 : Load balancing Occitania Uc1c

We see on the Redispatch number 1, 3, 4, 5 and 8 invert the planning between the MISSION1 and MISSION2 missions. The iterations number 2, 6, 7, 9 and 10 estimate that the inversion of planning is not necessary and doesn't do any change.
The mean of the ratio of the dispatch is 1.7, relatively close to the balanced ratio.

6.1.4.2.6.4 Use Case 2

	MISSION1		MISSION2		
	Activation	Cancel	Activation	Cancel	
Redispatch 1	274	0	66	0	1.2
Redispatch 2	90	105	29	30	1.3
Redispatch 3	83	66	24	22	1.2
Redispatch 4	111	116	34	33	1.3
Redispatch 5	100	114	28	30	1.3
Redispatch 6	58	76	21	24	1.3
Redispatch 7	39	48	16	18	1.3
Redispatch 8	43	33	15	13	1.3
Redispatch 9	41	45	13	9	1.4
Redispatch 10	31	47	15	15	1.5

Table 15 : Load balancing Occitania Uc2

Contrary to the Use Case 1c, each dispatch generates changes of planning.
To compute the ratio for Use Case 2, we considered that the Programming Request size for MISSION2 is 5 times bigger than the MISSION1 ones.
The mean ratio is 1.31, relatively close to the balanced ratio of 1.5.

6.2 Palma coverage

6.2.1 Goal and expected result

The goal of this test is to check the completion of a small User Request to 100%, with all sensors available (M2_1, M2_2, M1_1, M1_2, M1_3, M1_4) and on an empty Mission Chain context. To achieve it, one mission plan computation per day is launched on each Mission for 10 days.
We expect this area to be covered within the horizon of 10 days but will however allow us to witness the behaviour of the application on a small area.

6.2.2 Inputs

- The orderbook: User Request covering Palma Island
- The MISSION2 Mission Chain context:
    - Not loaded / empty
    - 1 plan per day
- The MISSION1 Mission Chain context:
    - Not loaded / empty
    - 1 plan per day

6.2.3 Major Steps

This test follows the same Chronology as the 6.1**Error! Reference source not found.**, only the initial orderbook is different during the test initialisation.

6.2.4 Results

6.2.4.1 KPI extraction

	UC1a	UC1b	UC1c	UC2
KPI#1 *Coverage Completion*	100%	100%	100%	100%
KPI#2 *Area Waste*	80,96%	121.67%	121.67%	60,40%
KPI#2bis *Area Waste with MC optimization*	80,96%	90,83%	90,83%	55,47%
KPI#3 *External calls*	11	30	43	184
KPI#4 - Global Coverage Completion Estimation *Computation time*	1.88s	4.62s	7.23s	None
KPI#4 – Global Redispatch	None	None	22.85s	15.15s

Computation time				
KPI#6 *Load Balancing*	4	3	1.12	3.52

Table 16 : KPIs extraction for Palma

6.2.4.2 Observations

6.2.4.2.1 KPI#1 Scenario progression

6.2.4.2.1.1 Use Case 1a

Figure 41 : Dispatch of the Palma Area by Use Case 1a

In this Use Case, we arbitrary decided to manually dispatch the entire area on a unique Mission Chain.

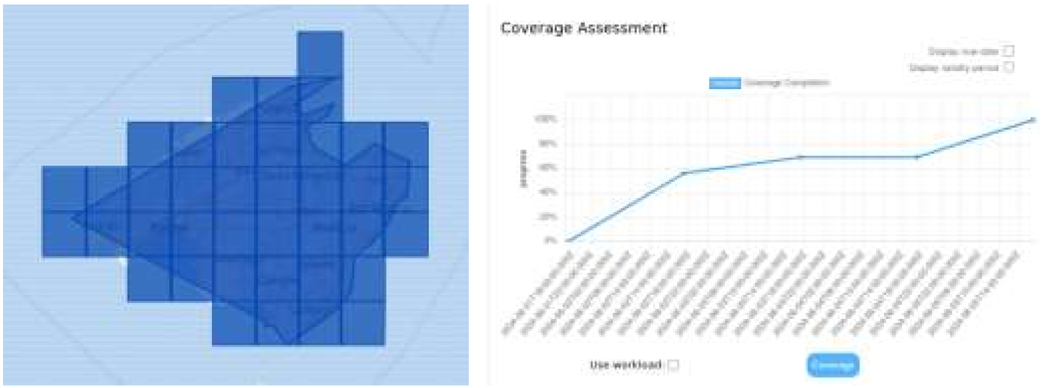


Figure 42 : Completion of Palma after the fifth plan for Use Case 1a

After five plans computation on MISSION2 satellites, the area is completely covered by mission chains.

6.2.4.2.1.2 Use Case 1b and 1c

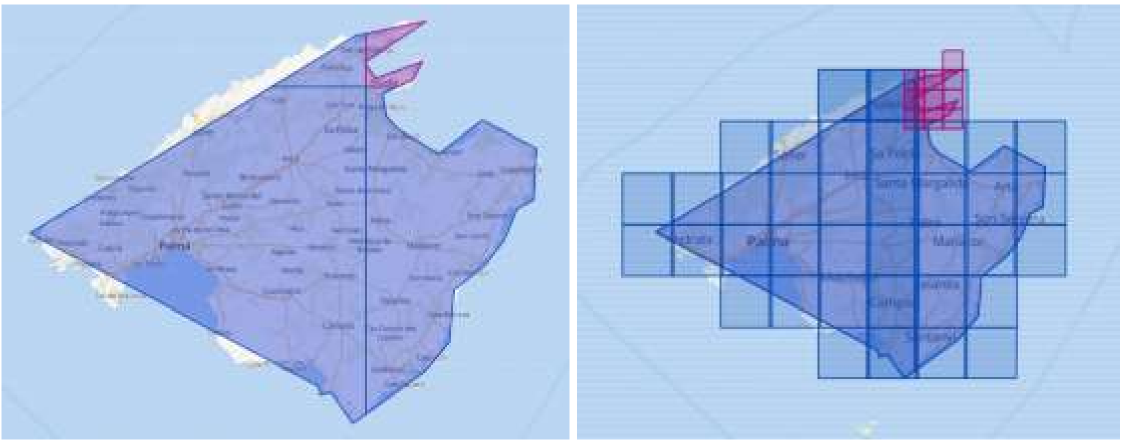


Figure 43 : Dispatch of the Palma Area by Use Case 1b and 1c

Figure 44 : Completion of Palma after the fifth plan for Use Case 1b and 1c

The completion of the area by the Use Case 1b and 1c is as fast as the Use Case 1a. After 5 days of plans, the area is completely covered by mission chains. The area initially dispatched on MISSION1 is also planned on MISSION2 because of the fixed split desynchronisation explained in 8.2.1.1. The Use Case 1b and 1c will therefore generate more area waste than the Use Case 1a on this specific case.
The progression of the Use Case 1c is described in Annexes 9.3.1. We can witness the temporary redispatch of a MISSION2 area on MISSION1, before re-dispatching it on MISSION2 again before the end of the test.

6.2.4.2.1.3 Use Case 2

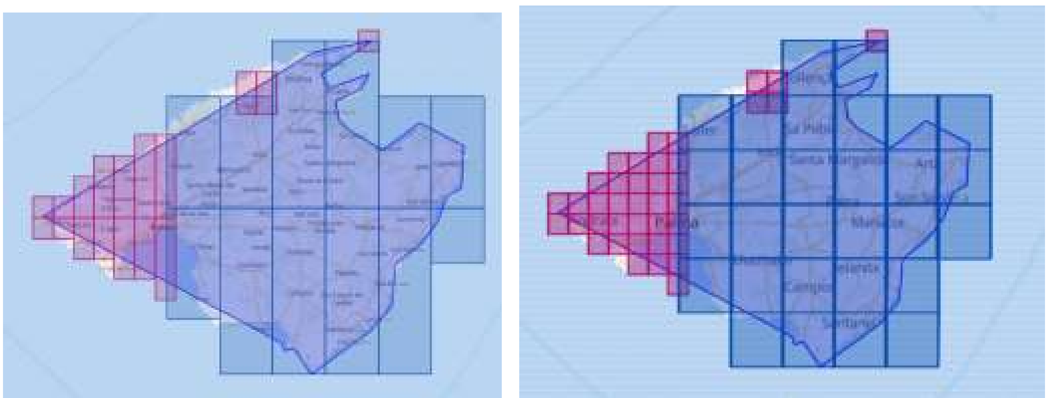


Figure 45 : Dispatch of the Palma Area by Use Case 2

Figure 46 : Completion of Palma after the fifth plan for Use Case 2

We can see that the complete coverage completion is also reached after five plans computation on MISSION2 and MISSION1 satellites, even if the dispatch is a little different than the one chosen by Use Case 1.
The progression of the Use Case 2 is described in Annexes 9.3.2.

6.2.4.2.2 KPI#2 Area Waste

6.2.4.2.2.1 Use Case 1a

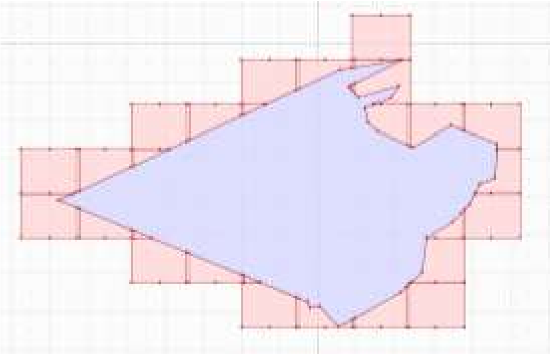


Figure 47 : Outside Area Waste Palma Use Case 1a

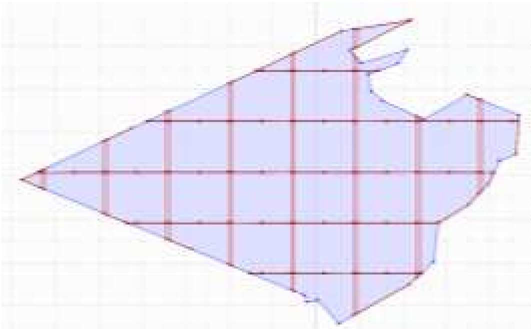


Figure 48 : Intra Area Waste Palma Use Case 1a

As the area have been submitted on a unique Mission Chain, the intra Area Waste is only composed of the overlapping borders of the acquisitions, which represents **6,18%** of the initial Area.
The Inter Mission Area Waste is empty, as there is no conflict between multiple Mission.
The Outside Area Waste represents **74,78%** of the initial Area.

6.2.4.2.2.2 Use Case 1b and Use Case 1c

The “Outside Area Waste” at the end of the test is **84,31%**.

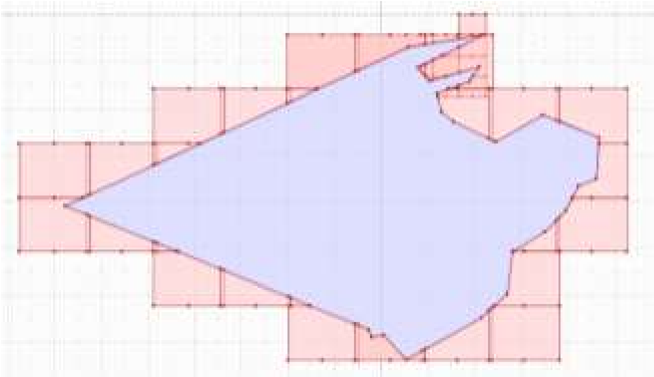


Figure 49 : Outside Area Waste Palma Use Case 1b and 1c

The "Intra Area Waste" at the end of the test is **37,36%**.

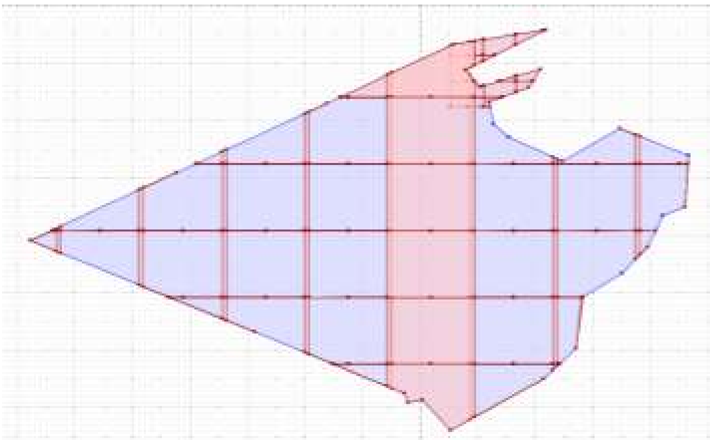


Figure 50: Intra Area Waste Palma Use Case 1b and 1c

The problem of overlapping is clear in this figure: in the Figure 43, we see the area on the MISSION2 (blue) is separated into three different Programming Requests; it creates a vertical and an horizontal overlapping once the Mission Chain map the area on its own world grid.

The "Inter-Mission Area Waste" at the end of the test is **6,52%**.

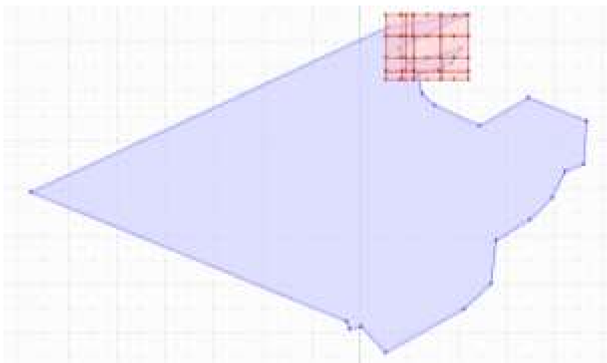


Figure 51 : Inter-Missions Area Waste Palma Use Case 1b and 1c

On this small area, the result for the area waste is the same for the Use Case 1b and Use Case 1c, whereas the Use Case 1c has redispatched once the Programming Request covering the Northeast of the Area.

6.2.4.2.2.3 Use Case 2

The “Outside Area Waste” at the end of the test is **51.09%**.

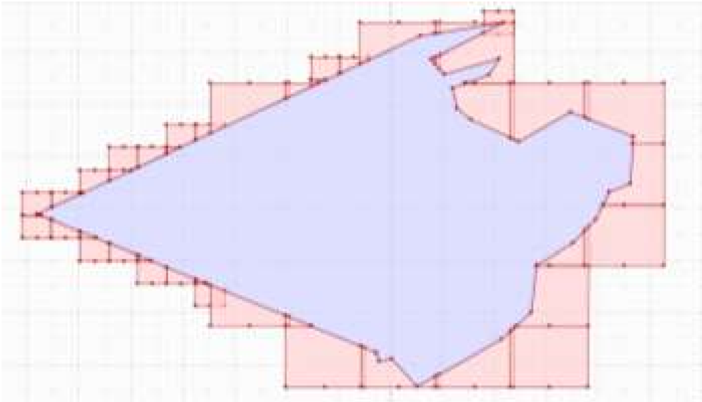


Figure 52 : Outside Area Waste Palma Use Case 2

The “Intra Area Waste” at the end of the test is **9,31%**.

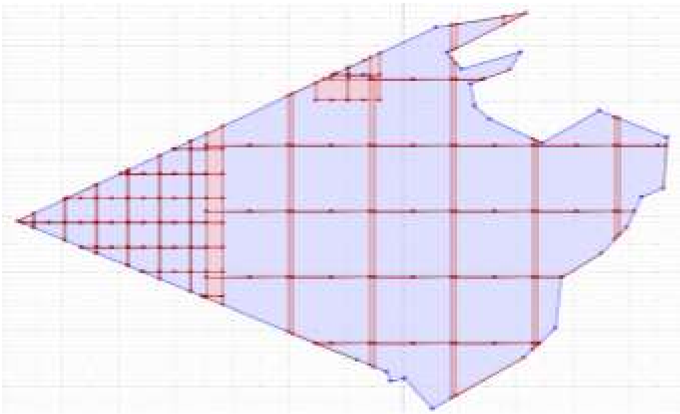


Figure 53: Intra Area Waste Palma Use Case 2

The “Inter-Mission Area Waste” at the end of the test is **4,38%**.

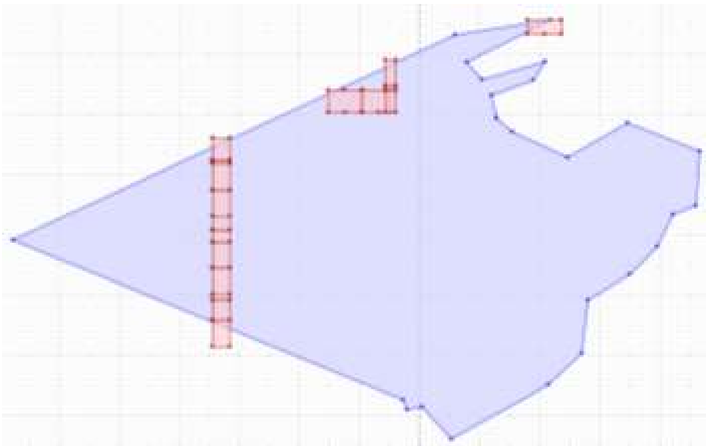


Figure 54 : Inter-Missions Area Waste Palma Use Case 2

6.2.4.2.2.4 Conclusion

The smaller the area, the biggest the Area Waste.

6.2.4.2.3 KPI#3 Mission Chains solicitation

6.2.4.2.3.1 Use Case 1a

	CREATE	CANCEL	FOLLOW-UP	WORKLOAD	ALL
MISSION2	1	0	5	5	**11**

Table 17 : Mission Chain solicitations Palma UC1a

The manual dispatch of this small area has been made only on a unique Mission, no interaction on MISSION1 is detected.

6.2.4.2.3.2 Use Case 1b

	CREATE	CANCEL	FOLLOW-UP	WORKLOAD	ALL
MISSION2	3	0	15	5	**23**
MISSION1	1	0	5	1	**7**

Table 18 : Mission Chain solicitations Palma UC1b

6.2.4.2.3.3 Use Case 1c

	CREATE	CANCEL	FOLLOW-UP	WORKLOAD	ALL
MISSION2	4	1	13	8	**26**
MISSION1	2	1	7	7	**17**

Table 19 : Mission Chain solicitations Palma UC1c

An area has been temporarily dispatched from MISSION2 to MISSION1, it generates another CREATE on MISSION1 and a CANCEL MISSION2. It is then redispatch on MISSION2, generating a CANCEL on MISSION1 and another CREATE on MISSION2.

6.2.4.2.3.4 Use Case 2

	CREATE	CANCEL	FOLLOW-UP	WORKLOAD	ALL
MISSION2	12	0	12	5	**77**
MISSION1	17	0	85	5	**107**

Table 20 : Mission Chain solicitations Palma UC2

The initial dispatch is not challenged by the 4 other iterations of dispatch; no CANCEL actions are performed.

6.2.4.2.4 KPI#4 Time computation

6.2.4.2.4.1 Use Case 1a

	Mean (in seconds)
Global Coverage Completion Estimation computation	1.88553574
MISSION2 – Coverage Completion Estimation computation	1.85701761
MISSION1 – Coverage Completion Estimation computation	0

Table 21 : Time computation Palma Uc1a

No Programming Requests have been submitted on MISSION1.

6.2.4.2.4.2 Use Case 1b

	Mean (in seconds)

Global Coverage Completion Estimation computation	4.62084288
MISSION2 – Coverage Completion Estimation computation	1.81064412
MISSION1 – Coverage Completion Estimation computation	0
Dispatch	46.5196405

Table 22 : Time computation Palma Uc1b

The MISSION1 Programming Request is COMPLETED on the first plan. Therefore, no Coverage Completion Estimation computation is launched for MISSION1 for this scenario on this Use Case.
Compared to the scenario of Occitania, the time of computation for the Global Coverage Completion Estimation computation is obviously reduced (148s to 4s); it is proportional to the number of Programming Requests contained inside of the User Request. The Dispatch time is also reduced, to a lesser extent (68s to 46s).

6.2.4.2.4.3 Use Case 1c

	Mean (in seconds)
Global Coverage Completion Estimation computation	7.23705448
MISSION2 – Coverage Completion Estimation computation	1.70257433
MISSION1 – Coverage Completion Estimation computation	13.7540252
Dispatch	45.960215
Redispatch	22.8559933

Table 23 : Time computation Palma Uc1c

The same observations can be made for Use Case 1c than Use Case 1b: reduced time for Global Coverage Completion Estimation computation and Dispatch.
The Redispatch time is also reduced (35s to 22s).

6.2.4.2.4.4 Use Case 2

	Mean (in seconds)
Redispatch	15.1582866

Table 24 : Time computation Palma Uc2

The time of redispatch computation is reduced compared to the Occitania scenario.

6.2.4.2.5 KPI#5 Resource Usage

6.2.4.2.5.1 Use Case 1a

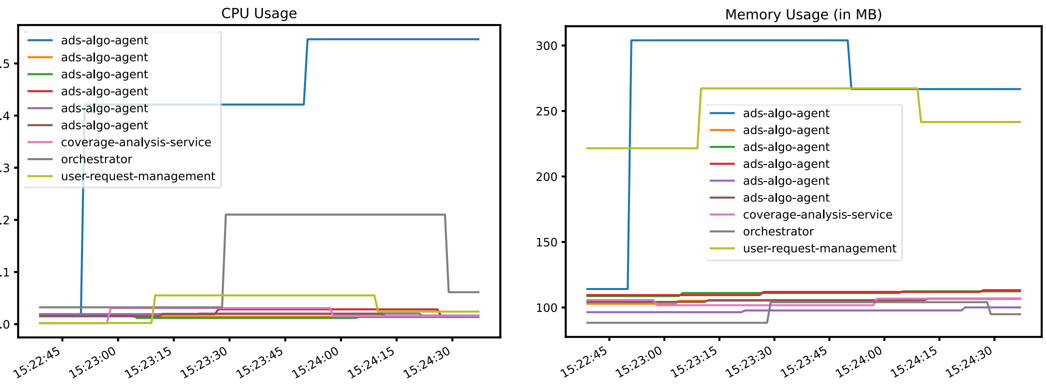


Figure 55 : Resources usage Palma for Use Case 1a

6.2.4.2.5.2 Use Case 1b

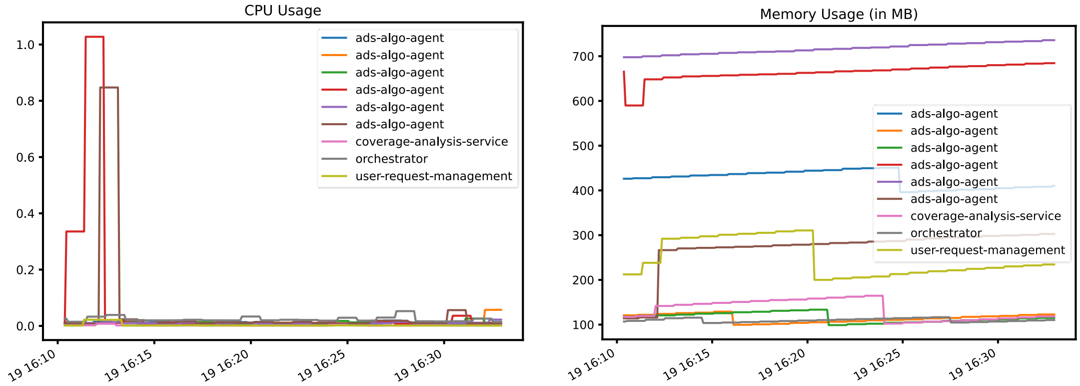


Figure 56 : Resources usage Palma for Use Case 1b

6.2.4.2.5.3 Use Case 1c

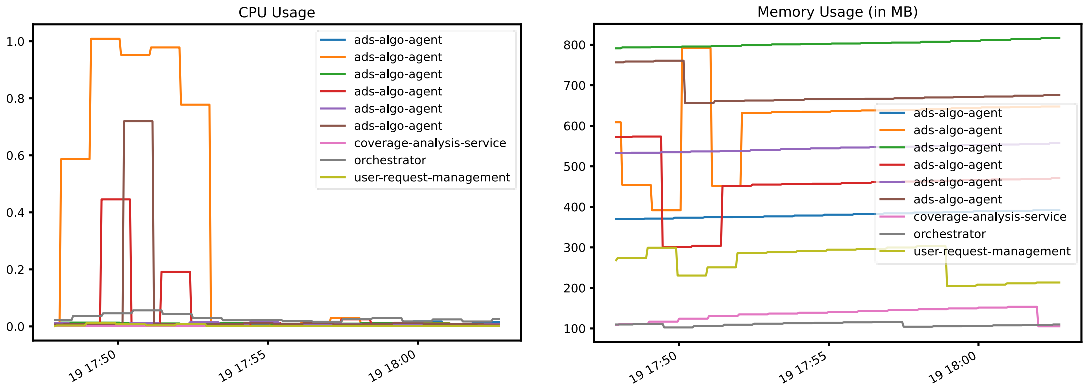


Figure 57 : Resources usage Palma for Use Case 1c

Compared to the Occitania scenario, where all the agent replications were used for Use Case 1c, here only three of them are called, mainly for the beginning of the call, corresponding to the dispatches having a non-null impact.

6.2.4.2.5.4 Use Case 2

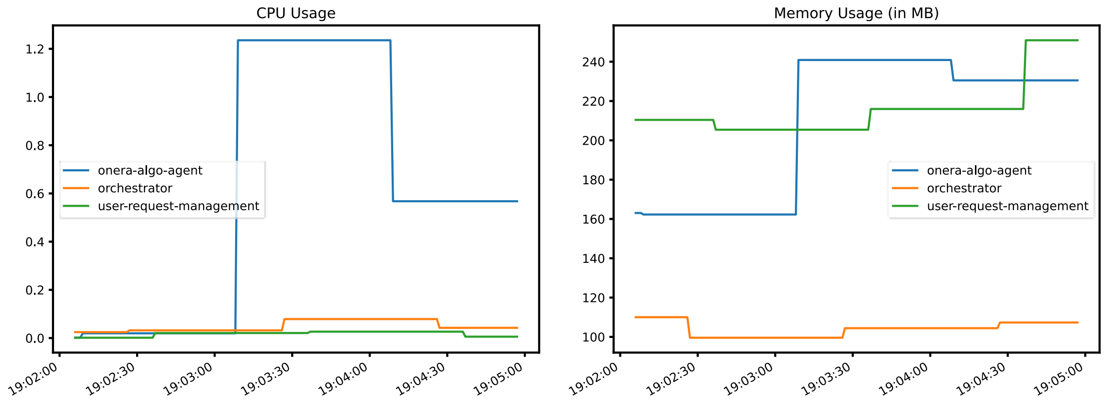


Figure 58 : Resources usage Palma for Use Case 2

6.2.4.2.5.5 Conclusion

The replication of agents is not mandatory for small areas. The algorithmic solutions as well as the global application seems to consume a volume of resources proportional to the size of the area to treat and to the number of calls to be performed.

6.2.4.2.6 KPI#6 Load balancing

6.2.4.2.6.1 *Use Case 1a*

	MISSION1		MISSION2		Ratio
	Activation	Cancel	Activation	Cancel	
Initial Dispatch	0	0	4	0	4

Table 25 : Load balancing Palma Uc1a

This manual dispatch obviously favours MISSION2.

6.2.4.2.6.2 *Use Case 1b*

	MISSION1		MISSION2		Ratio
	Activation	Cancel	Activation	Cancel	
Initial Dispatch	1	0	3	0	3

Table 26 : Load balancing Palma Uc1b

6.2.4.2.6.3 *Use Case 1c*

	MISSION1		MISSION2		
	Activation	Cancel	Activation	Cancel	
Initial Dispatch	1	0	3	0	3
Redispatch 1	1	0	0	1	1
Redispatch 2	1	0	0	1	0.3
Redispatch 3	0	0	0	0	0.3
Redispatch 4	0	1	1	0	1

Table 27 : Load balancing Palma Uc1c

In average, the ratio is 1.12 but with a very large range of possibilities.

6.2.4.2.6.4 *Use Case 2*

	MISSION1		MISSION2		Ratio
	Activation	Cancel	Activation	Cancel	
Redispatch 1	17	0	12	0	3,52
Redispatch 2	0	0	0	0	3,52
Redispatch 3	0	0	0	0	3,52
Redispatch 4	0	0	0	0	3,52
Redispatch 5	0	0	0	0	3,52

Table 28 : Load balancing Palma Uc2

The Use Case 2 never changed of the initial dispatch found.
The ratio is 3.52 average: on small areas, the dispatch choices are largely motivated by the accesses on few next orbits for Use Case 2, the “balanced” ratio we calculated doesn’t take this aspect into account and therefore the average is far from the balanced ratio we computed.

6.3 France coverage

6.3.1 Goal and expected result

The goal of this test is to check the completion progression on a unique very large User Request, with all sensors available (M2_1, M2_2, M1_1, M1_2, M1_3, M1_4) and on an empty Mission Chain context. To achieve it, one mission plan computation per day is launched on each Mission for 8 days. For performance and test time reason, we reduced the horizon of the test from 10 to 8 days, compared to other scenarios.
We don't expect such a large zone to be covered in 8 days, but we will be able to compare the different KPIs defined earlier for any algorithmic solution.
We also judged the scenario UC1a, with a manual dispatch, not essential, as the only thing we want to assess with this scenario is the robustness with regards to a bigger zone. We therefore didn't perform the dispatch of the France in UC1a mode.

6.3.2 Inputs

- The orderbook: User Request covering France country
- The MISSION2 Mission Chain context:
    - Not loaded / empty
    - 1 plan per day
- The MISSION1 Mission Chain context:
    - Not loaded / empty
    - 1 plan per day

6.3.3 Major Steps

This test follows the same Chronology as the 6.1, only the initial orderbook is different during the test initialisation.

6.3.4 Results

6.3.4.1 KPI extraction

	UC1b	UC1c	UC2
KPI#1 *Coverage Completion*	17,29%	16,93%	17,42%
KPI#2 *Area Waste*	41,27%	41,9%	14,6%
KPI#2bis *Area Waste with MC optimization*	12,11%	15,9%	9,5%
KPI#3 *External calls*	1467	1564	33 502
KPI#4 - Global Coverage Completion Estimation *Computation time*	384.9s	449.8s	None
KPI#4 – Global Redispatch *Computation time*	None	136.3s	285.7s

KPI#6 *Load Balancing*	1.2	1.1	1.4

Table 29 : KPIs extraction for France

6.3.4.2 Observations

6.3.4.2.1 KPI#1 Scenario progression

6.3.4.2.1.1 Use Case 1b

Here is the initial dispatch of France with UC1b.

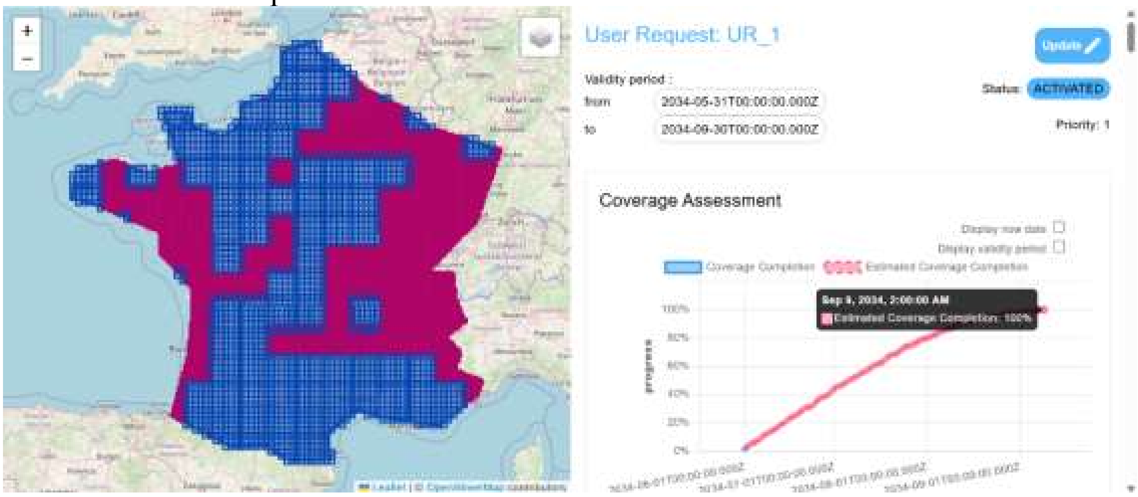


Figure 59 : Dispatch of the France country by Use Case 1b

The status at the end of the test is:

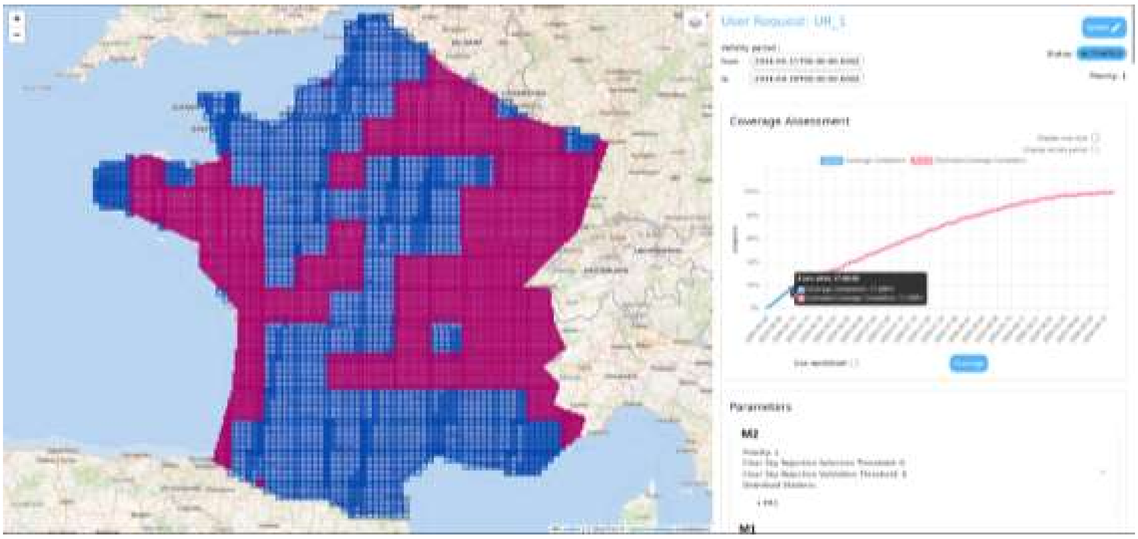


Figure 60 : Completion of the France after the 8th plan for Use Case 1b

The coverage at the end of the 8 days is **17.29%.**

6.3.4.2.1.2 Use Case 1c

The initial dispatch is the same than with UC1b, but the final result is different.

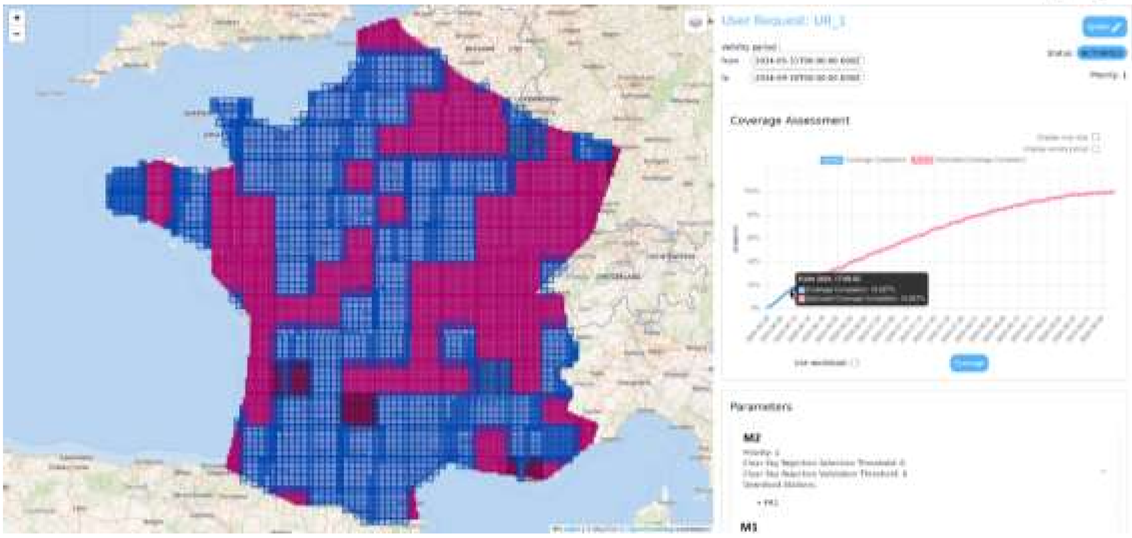


Figure 61 : Completion of the France after the 8th plan for Use Case 1c

The coverage at the end of the 8 days is **16.93%.**

6.3.4.2.1.3 Use Case 2

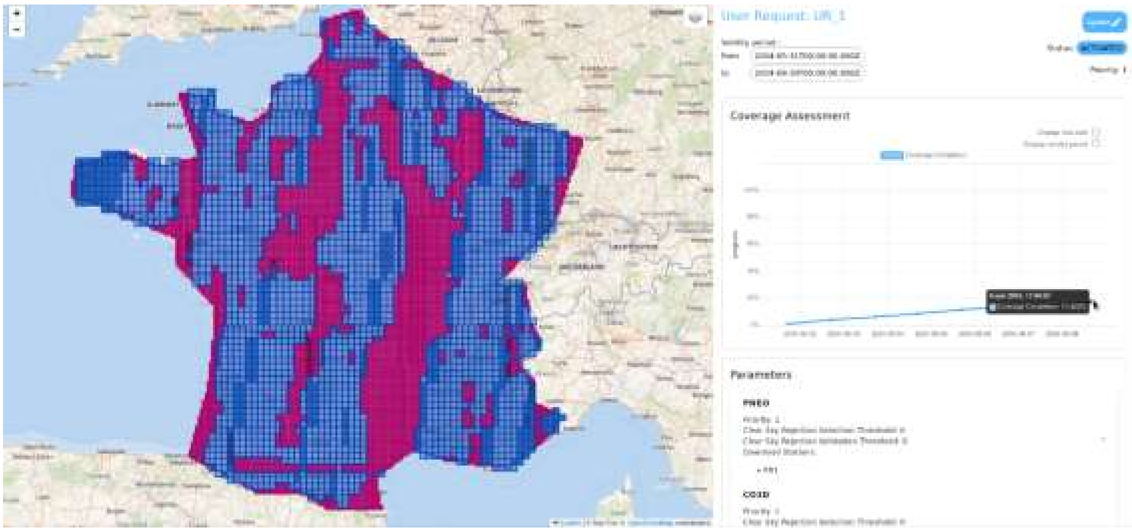


Figure 62 : Completion of the France after the 8th plan for Use Case 2

The coverage at the end of the 8 days is **17,42%.**

6.3.4.2.2 KPI#2 Area Waste

6.3.4.2.2.1 Use Case 1b

The following schema gives an overview of this "Outside Area Waste" for the France Area and the Use Case 1b. It represents **4,8**% of the global Area.

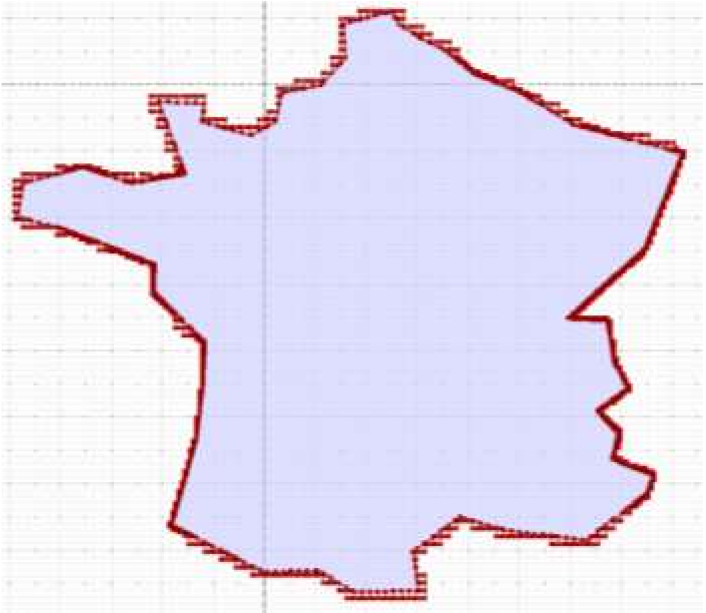


Figure 63 : Outside Area Waste France Use Case 1b

In the following schema, a zoom on the northwest of the France is performed to distinguish the Intra Area Waste.

Figure 64 : Intra Area Waste France Use Case 1b

This area waste represents **36,47**% of the global User Request Area.

The next schema displays the case "Inter Missions Area Waste" for the France Country on Use Case 1b:

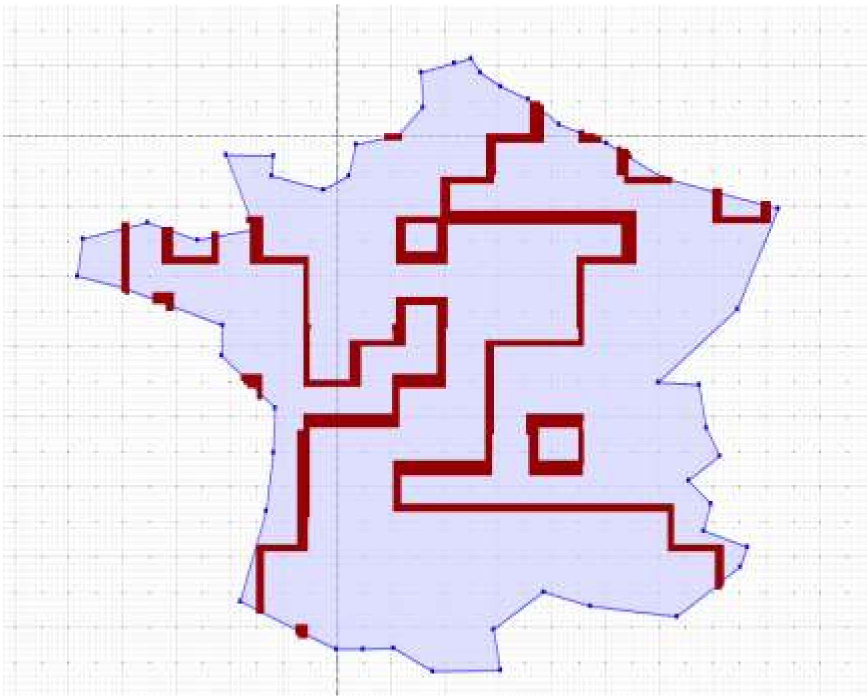


Figure 65 : Inter-Missions Area Waste France Use Case 1b

This area waste represents **7,31**% of the global Area.

It is interesting to note that the Intra and Inter-Mission percentage remain stable between small and wide area (Occitania and France), while the outside Area Waste is reduced when computed on wide areas.

6.3.4.2.2.2 Use Case 1c

The following schema gives an overview of this “Outside Area Waste” for the France Area and the Use Case 1c. It represents **4,8**% of the global Area.

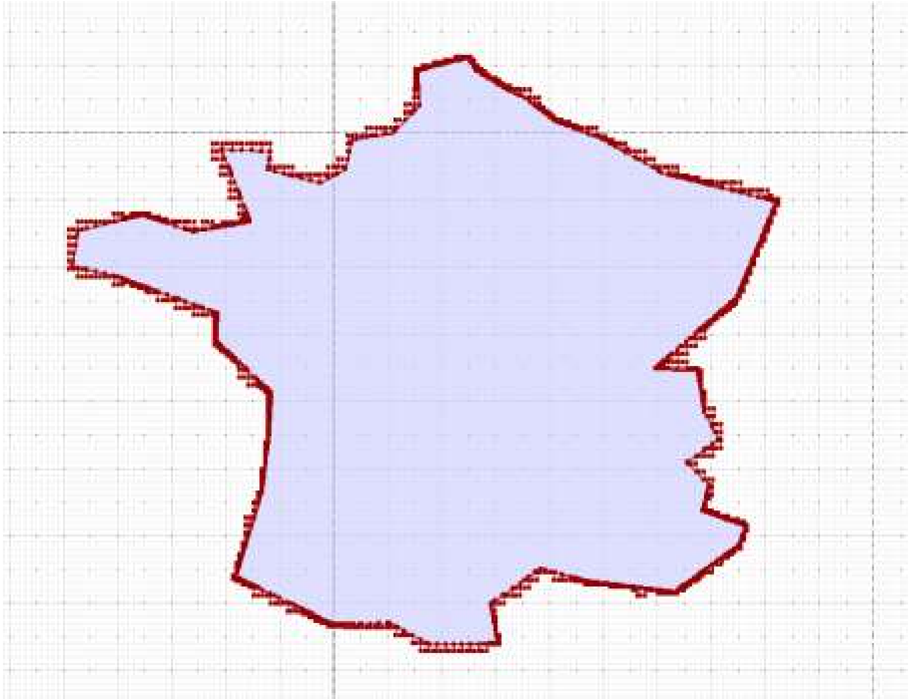


Figure 66 : Outside Area Waste France Use Case 1c

In the following schema, a zoom on the northwest of the France is performed to distinguish the Intra Area Waste.

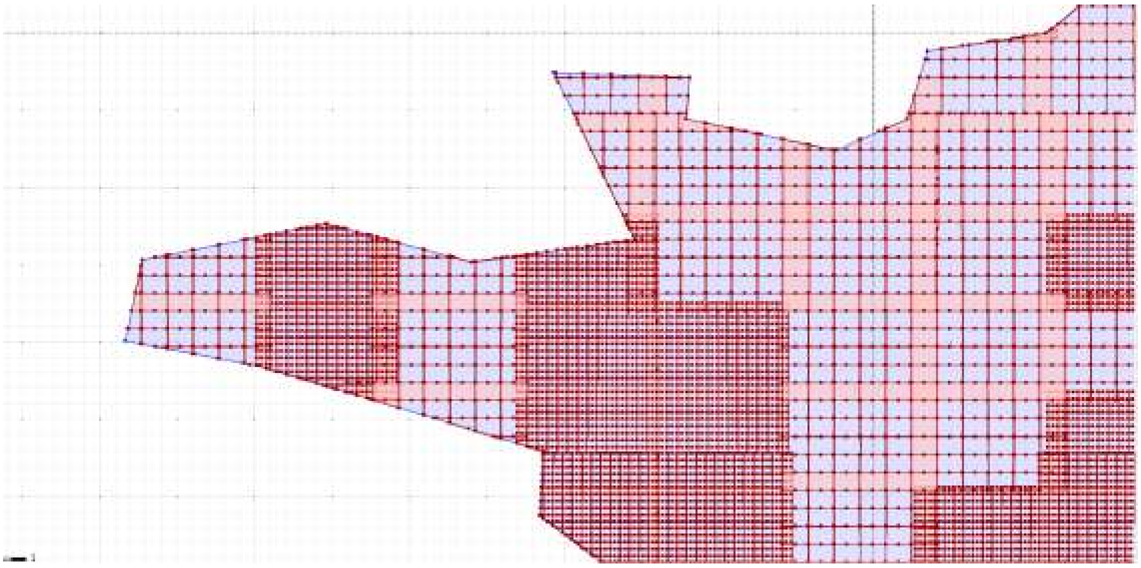


Figure 67 : Intra Area Waste France Use Case 1c

This area waste represents **37.1**% of the global User Request Area.

The next schema displays the case “Inter Missions Area Waste” for the France Country on Use Case 1b:

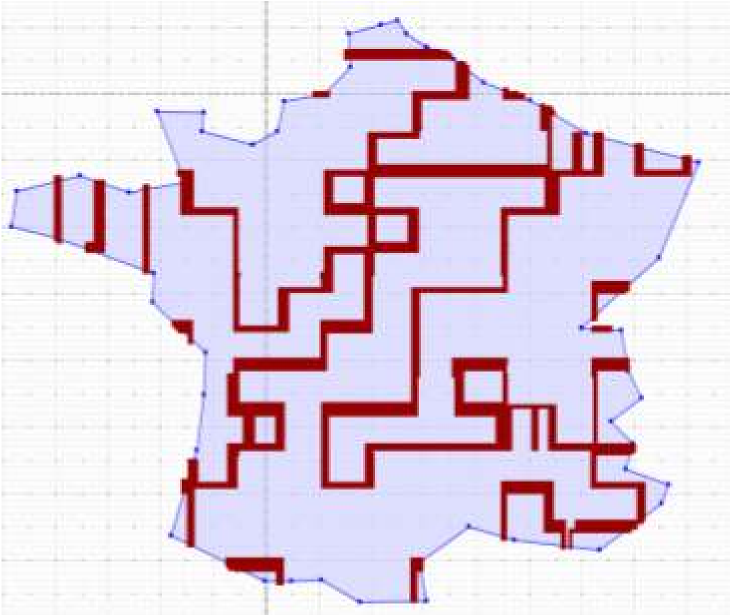


Figure 68 : Inter-Missions Area Waste France Use Case 1c

This area waste represents **11,1**% of the global Area.

6.3.4.2.2.3 Use Case 2

The following schema gives an overview of this "Outside Area Waste" for the France Area and the Use Case 1c. It represents **3.5**% of the global Area.

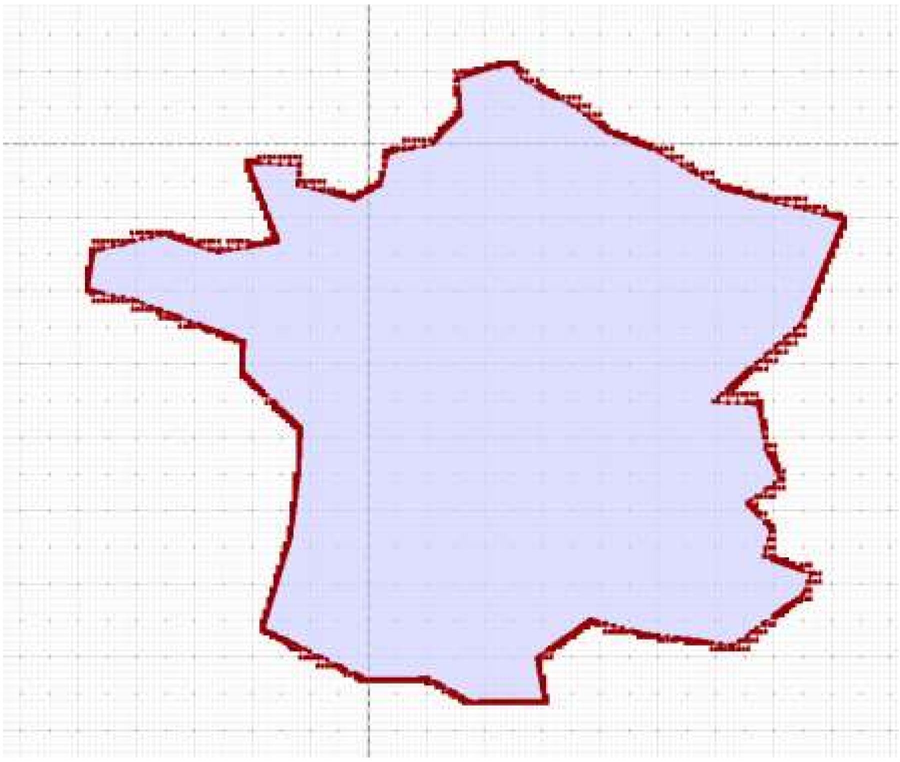


Figure 69 : Outside Area Waste France Use Case 2

In the following schema, a zoom on the northwest of the France is performed to distinguish the Intra Area Waste.

Figure 70 : Intra Area Waste France Use Case 2

This area waste represents **11.1**% of the global User Request Area.

The next schema displays the case "Inter Missions Area Waste" for the France Country on Use Case 1b:

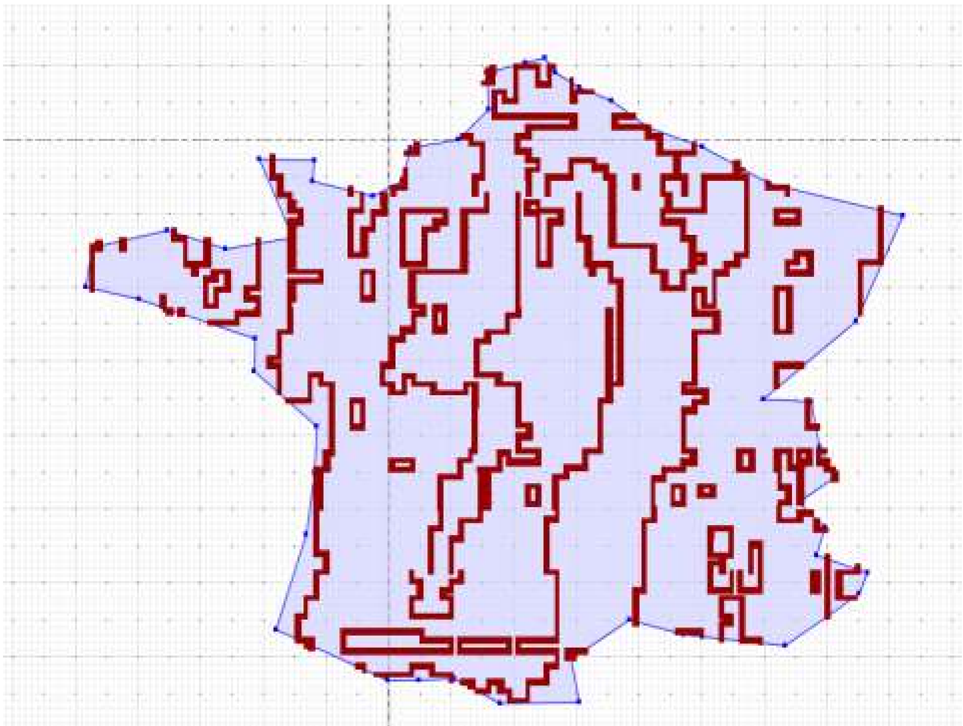


Figure 71 : Inter-Missions Area Waste France Use Case 2

This area waste represents **6,0**% of the global Area.

6.3.4.2.3 KPI#3 Mission Chains solicitation

6.3.4.2.3.1 Use Case 1b

	CREATE	CANCEL	FOLLOW-UP	WORKLOAD	ALL
MISSION1	74	0	592	9	**675**
MISSION2	87	0	696	9	**792**

Table 30 : Mission Chain solicitations France UC1b

6.3.4.2.3.2 Use Case 1c

	CREATE	CANCEL	FOLLOW-UP	WORKLOAD	ALL
MISSION1	89	12	633	17	**751**
MISSION2	99	15	682	17	**813**

Table 31 : Mission Chain solicitations France UC1c

6.3.4.2.3.3 Use Case 2

	CREATE	CANCEL	FOLLOW-UP	WORKLOAD	ALL
MISSION1	6243	4388	15184	8	**25823**
MISSION2	1955	1424	4292	8	**7679**

Table 32 : Mission Chain solicitations France UC2

6.3.4.2.4 KPI#4 Time Computation

6.3.4.2.4.1 Use Case 1b

	Mean (in seconds)
Global Coverage Completion Estimation computation	384.959915
MISSION2 – Coverage Completion Estimation computation	3.06300645
MISSION1 – Coverage Completion Estimation computation	27.3738092
Dispatch	231.218578

Table 33 : Time computation France Uc1b

6.3.4.2.4.2 Use Case 1c

	Mean (in seconds)
Global Coverage Completion Estimation computation	449.8389106
MISSION2 – Coverage Completion Estimation computation	3.135120593
MISSION1 – Coverage Completion Estimation computation	30.05797287
Dispatch	380.2871108
Redispatch	136.3416253

Table 34 : Time computation France Uc1c

6.3.4.2.4.3 Use Case 2

	Mean (in seconds)
Redispatch	285.777964

Table 35 : Time computation France Uc2

In the original configuration of ONERA Algo Agent, with maxNbIterationsAfterGreedy= 5000, the duration of a single dispatch was over 13 000s ~ 3h30min (Time of algorithmic dispatch + time of submission over the Mission Chains). As it is hardly acceptable to have such duration of tests, and for the sake of fairness, we decided to reduce the number of iterations until we reach the amount of time needed in the worst case by the dispatch of UC1, which is around 1000 seconds. It corresponds to a maxNbIterationsAfterGreedy=300.

6.3.4.2.4.4 Summary

The ADS dispatch is based on the result of the Coverage Assessment. The computation time of the dispatch is therefore reduced, compared to the ONERA one. However, the ONERA time of computation is configurable by limiting the number of iterations or directly the CPU time allocated to the algorithmic call of redispatch.
We also observe that the ONERA time of dispatch is very linear, compared to the ADS one, which is more random: there are several iterations of ADS redispatch which produce no changes, compared to the ONERA solution where a lot of changes are requested on each dispatch.

6.3.4.2.5 KPI#5 Resource Usage

For performance reason during France scenario, the CPU and Memory Usage have been retrieved through Grafana dashboard directly instead of generated in the reports. It explains the difference of design between the following screenshots of this scenario and others.

6.3.4.2.5.1 Use Case 1b

Figure 72 : Resources usage France for Use Case 1b – CPU Usage

Figure 73 : Resources usage France for Use Case 1b – Memory Usage

Here, the algo-agent replication proves useful, as a 4 algo agent replicas are used at the same time.

6.3.4.2.5.2 Use Case 1c

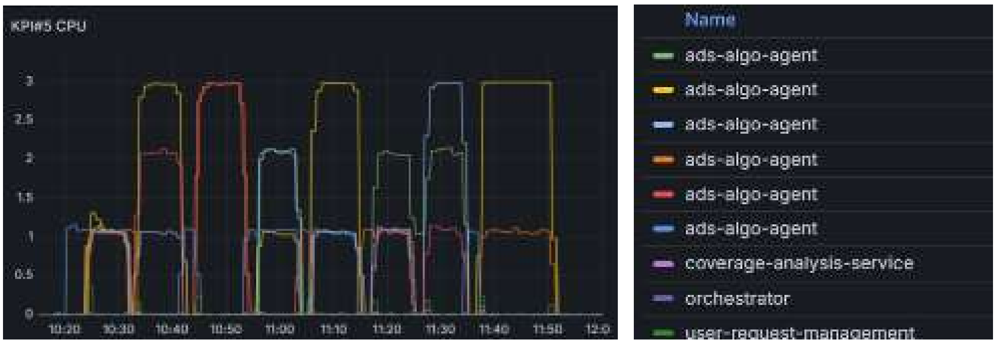


Figure 74 : Resources usage France for Use Case 1c – CPU Usage

Figure 75 : Resources usage France for Use Case 1c – Memory Usage

6.3.4.2.5.3 Use Case 2

Figure 76 : Resources usage France for Use Case 2 – CPU Usage

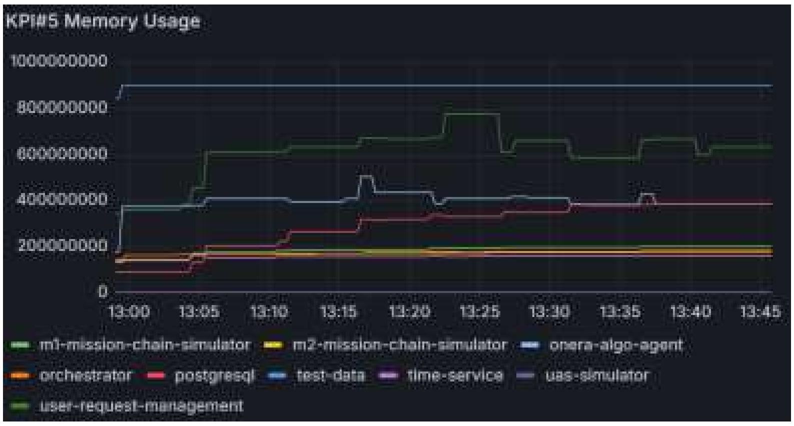


Figure 77 : Resources usage France for Use Case 2 – Memory Usage

6.3.4.2.6 KPI#6 Load balancing

6.3.4.2.6.1 Use Case 1b

	MISSION1		MISSION2		Ratio
	Activation	Cancel	Activation	Cancel	
Initial Dispatch	74	0	87	0	1.2

Table 36 : Load balancing France Uc1b

6.3.4.2.6.2 Use Case 1c

	MISSION1		MISSION2		Ratio
	Activation	Cancel	Activation	Cancel	
Initial Dispatch	74	0	87	0	1.2
Redispatch 1	1	0	0	1	1.1
Redispatch 2	5	1	1	5	1.0
Redispatch 3	3	3	3	3	1.0
Redispatch 4	2	0	0	2	1.0
Redispatch 5	1	5	5	1	1.1
Redispatch 6	2	1	1	2	1.1
Redispatch 7	1	2	2	1	1.1

Table 37 : Load balancing France Uc1c

6.3.4.2.6.3 Use Case 2

	MISSION1		MISSION2		Ratio
	Activation	Cancel	Activation	Cancel	
Redispatch 1	1965	0	537	0	1.4
Redispatch 2	876	832	252	257	1.3
Redispatch 3	682	793	235	234	1.4
Redispatch 4	582	610	193	199	1.4
Redispatch 5	592	567	197	192	1.4
Redispatch 6	545	589	194	206	1.4
Redispatch 7	482	492	173	160	1.4
Redispatch 8	519	505	174	176	1.4

6.4 Occitania Coverage for loaded Missions

6.4.1 Goal and expected result

The goal of this test is to check the completion progression on a unique User Request, with all sensors available (M2_1, M2_2, M1_1, M1_2, M1_3, M1_4) and on a loaded Mission Chain context. To achieve it, one mission plan computation per day will be launched on each Mission for 10 days.
We expect the coverage completion time to be better with a re-dispatch than without.

6.4.2 Inputs

- The orderbook: User Request covering Occitania region
- The MISSION2 Mission Chain context:
  - Loaded context: Programming Requests are progressively submitted on Mission Chains directly to reach 57 Programming Requests on MISSION 2
  - 1 plan per day
- The MISSION1 Mission Chain context:
  - Loaded context: Programming Requests are progressively submitted on Mission Chains directly to reach 38 Programming Requests on MISSION 1
  - 1 plan per day

Visually, the repartition of the load is as followed:

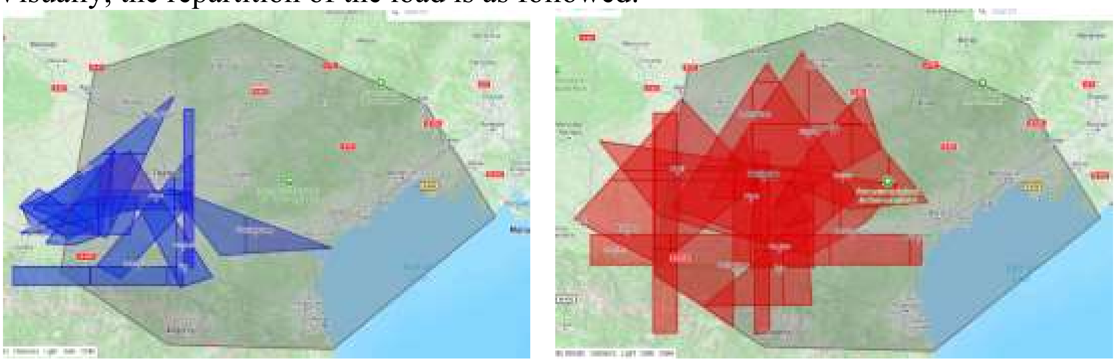


Figure 78 : Load repartition over the Mission Chains (blue for MISSION1, red for MISSION2)

6.4.3 Major Steps

This test follows the same Chronology as the Occitania scenario 6.1, only the initial orderbook and the initial Workload of the Mission is different during the test initialisation.

6.4.4 Results

6.4.4.1 KPI extraction

	UC1a	UC1b	UC1c	UC2
KPI#1 *Coverage Completion*	1,70%	32,6%	31,64%	26,46%

KPI#2 *Area Waste*	21,3%	50,16%	50,78%	20,08%
KPI#2bis *Area Waste with MC optimization*	11,25%	21,99%	22,31%	15,43%
KPI#3 *External calls*	88	265	330	6082
KPI#4 - Global Coverage Completion Estimation *Computation time*	18,30s	35.44s	34.64s	None
KPI#4 – Global Redispatch *Computation time*	None	None	62.38s	51.85s
KPI#6 *Load Balancing*	0.2	1.2	1.35	1.5

Table 38 : KPIs extraction for Occitania Loaded

6.4.4.2 Observations

6.4.4.2.1 KPI#1 Scenario progression

6.4.4.2.1.1 Use Case 1a

The initial dispatch is like the nominal Occitania scenario. What differs is the estimation completion and the final completion result, as some load is already processed by Mission Chains.

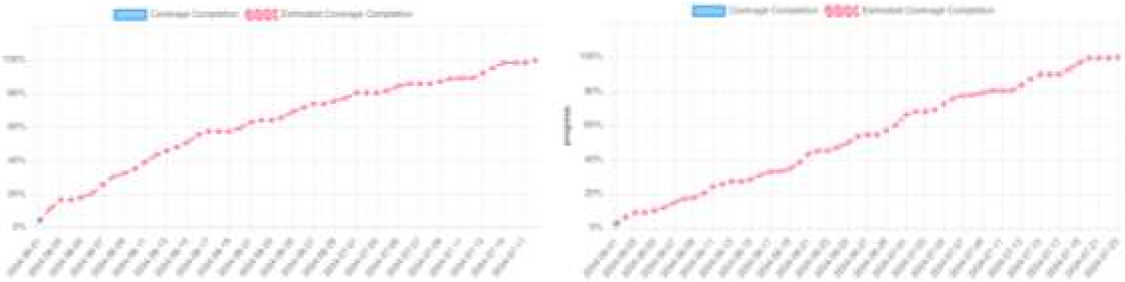


Figure 79 : Completion estimation of the Occitania with (on the right) or without (on the left) workload

We can see that the complete coverage completion is estimated later when the Mission Chains are loaded (17/07 without workload and 23/07 with the initial workload of 10 Programming Requests per Mission Chain).

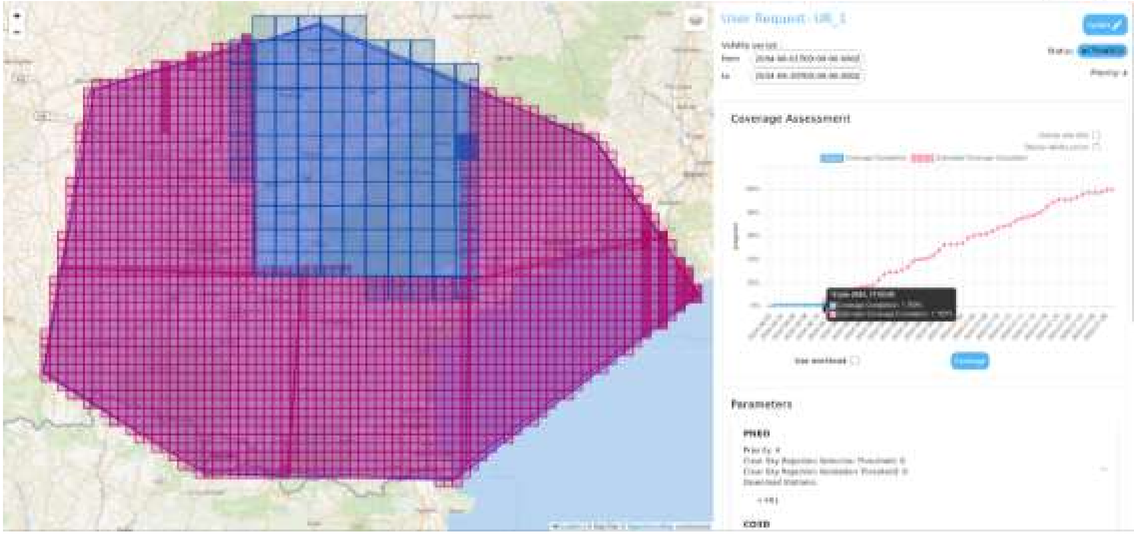


Figure 80 : Completion of Occitania with workload after the tenth plan for Use Case 1a

The global coverage of the User Request at the end of the test is **1,70**%, compared to the 49,1% without workload.
It is particularly small, as the workload of the Mission Chains take all the possible acquisitions.

6.4.4.2.1.2 Use Case 1b

The dispatch at submission doesn't change with the initial load of the Mission Chains.

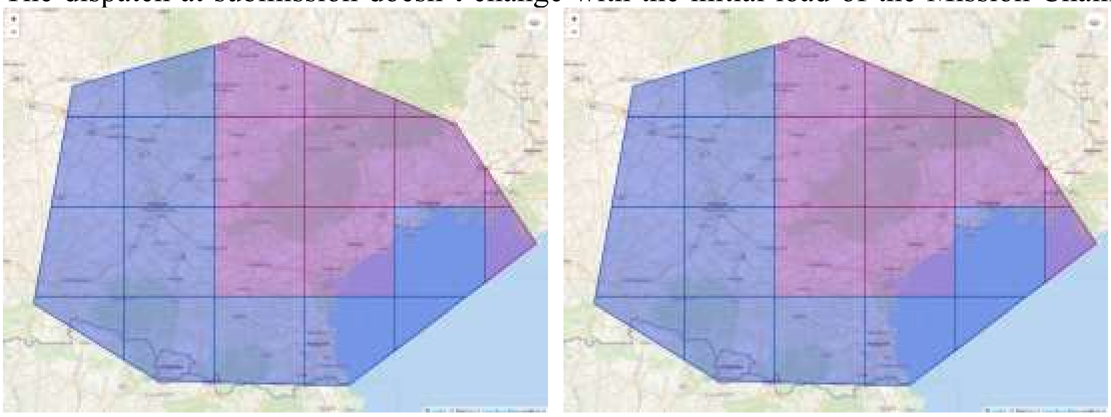


Figure 81 : Initial Dispatch for Occitania with (on the right) and without (on the left) workload

However, the estimated date of completion slightly changes, from 05/07/2034 to 06/07/2034.

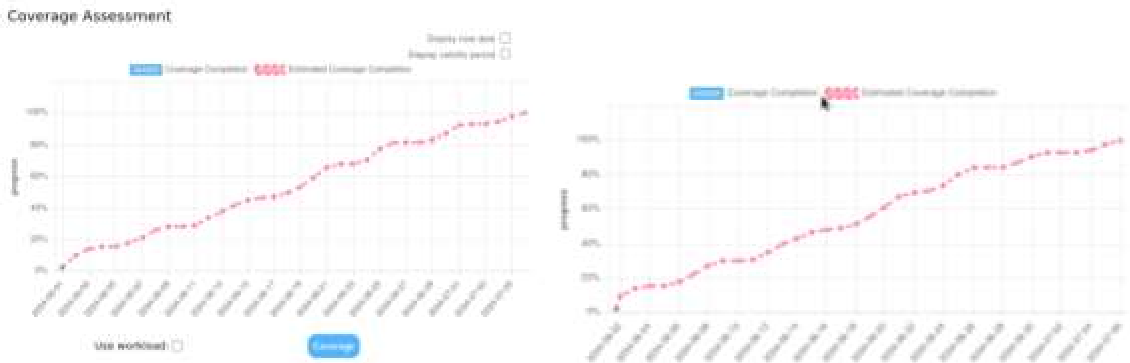


Figure 82 : Initial Completion estimation for Occitania with (on the right) and without (on the left) workload

The coverage estimation completion shifts during the run of the test, as the workload is getting heavier, to finally reach the 18/07/2034.

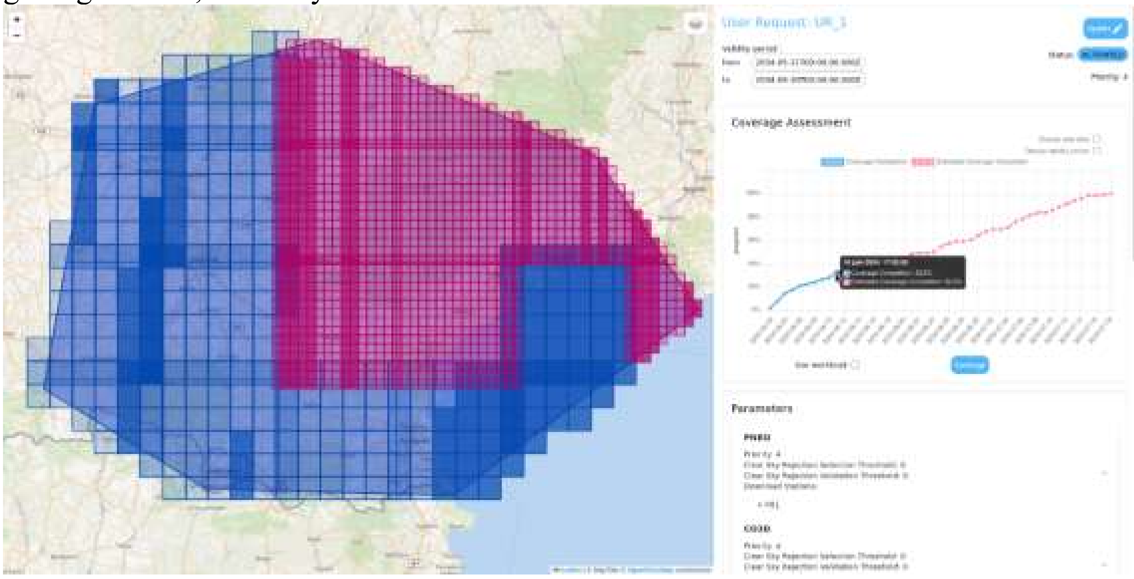


Figure 83 : Completion of Occitania with workload after the tenth plan for Use Case 1b

The global coverage of the User Request at the end of the test is **32,6%**, which is as expected inferior to the value with empty Mission Chains (57,26%).

6.4.4.2.1.3 Use Case 1c

The initial dispatch is equivalent to the one of Use Case 1b.

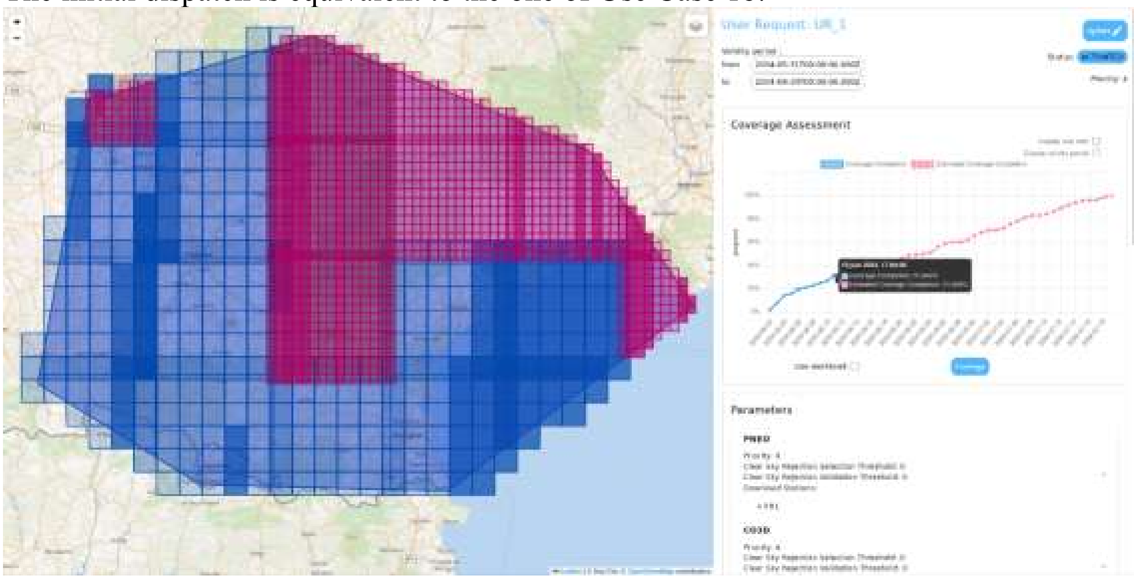


Figure 84 : Completion of Occitania with workload after the tenth plan for Use Case 1c

The global coverage of the User Request at the end of the test is **31,64%,** compared to 54,63% without workload.

6.4.4.2.1.4 Use Case 2

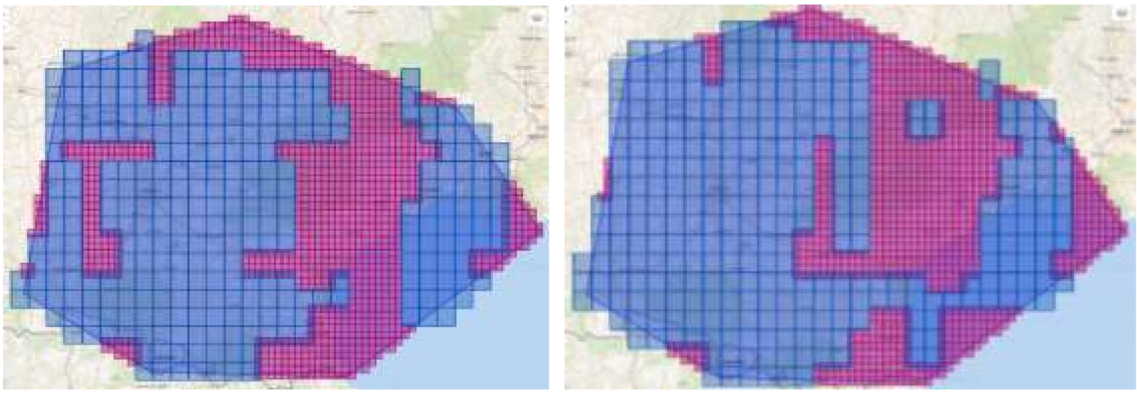


Figure 85 : Initial dispatch of Occitania with (on the right) and without (on the left) workload

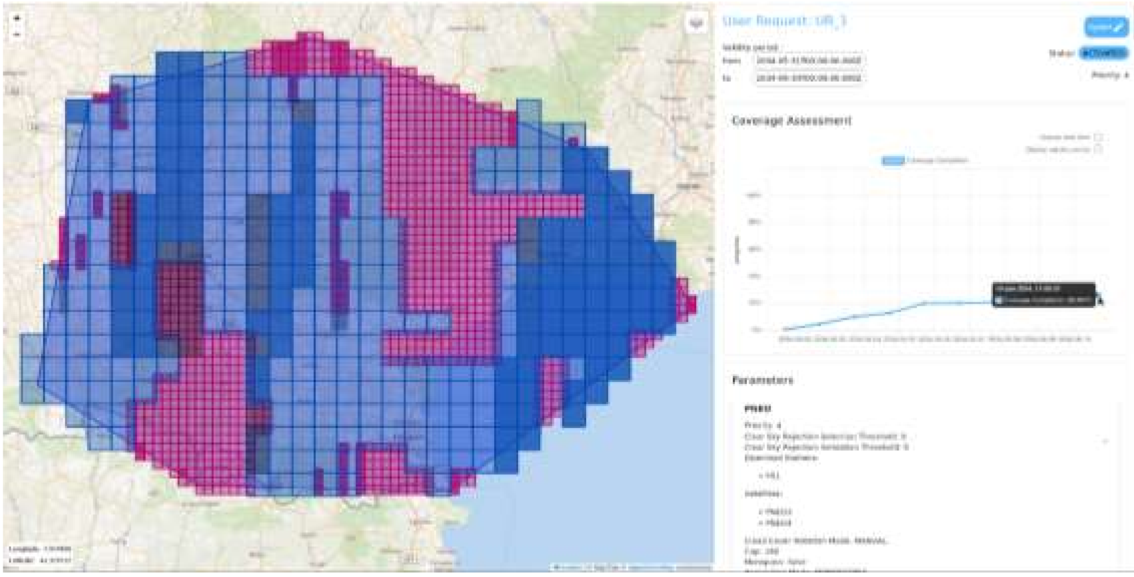


Figure 86 : Completion of Occitania with workload after the tenth plan for Use Case 2

The global coverage of the User Request at the end of the test is **26,46**%, compared to 51,89% without workload.

6.4.4.2.2 KPI#2 Area Waste

6.4.4.2.2.1 Use Case 1a

The “Outside Area Waste” at the end of the test is **5,84%**.

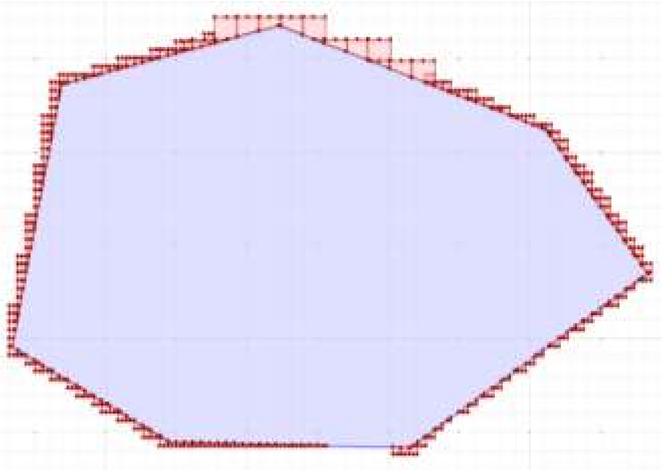


Figure 87 : Outside Area Waste Loaded Use Case 1a

The "Intra Area Waste" at the end of the test is **15,46%**.

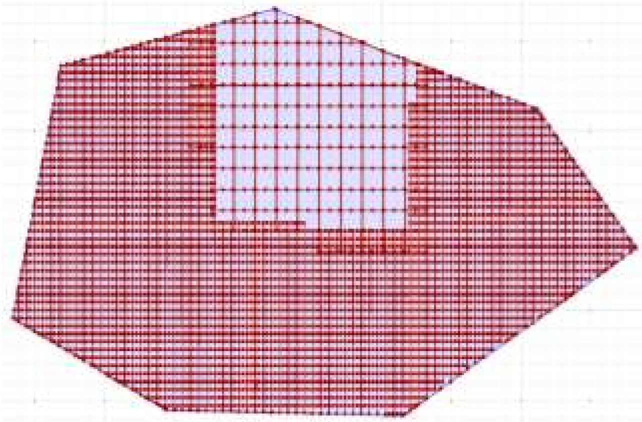


Figure 88 : Intra Area Waste Loaded Use Case 1c

The "Inter-Mission Area Waste" at the end of the test is **5,41%**.

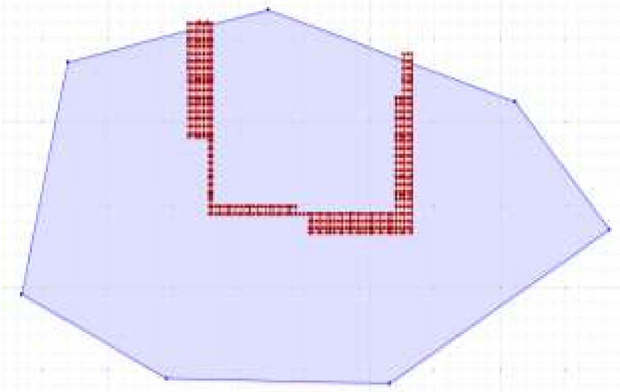


Figure 89 : Inter Mission Area Waste Loaded Use Case 1a

6.4.4.2.2.2 Use Case 1b

The "Outside Area Waste" at the end of the test is **11,05%**.

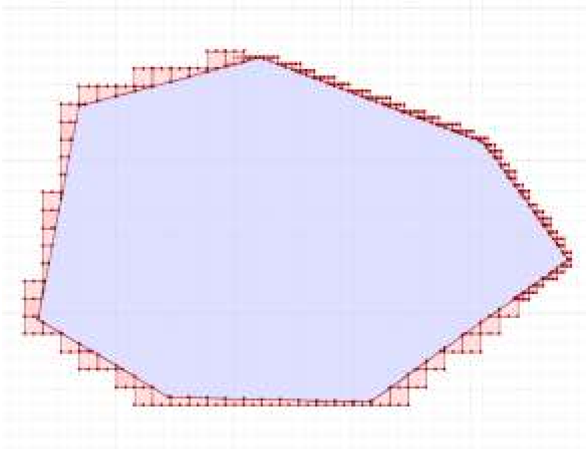


Figure 90: Outside Area Waste Loaded Use Case 1b

The “Intra Area Waste” at the end of the test is **39,11%**.

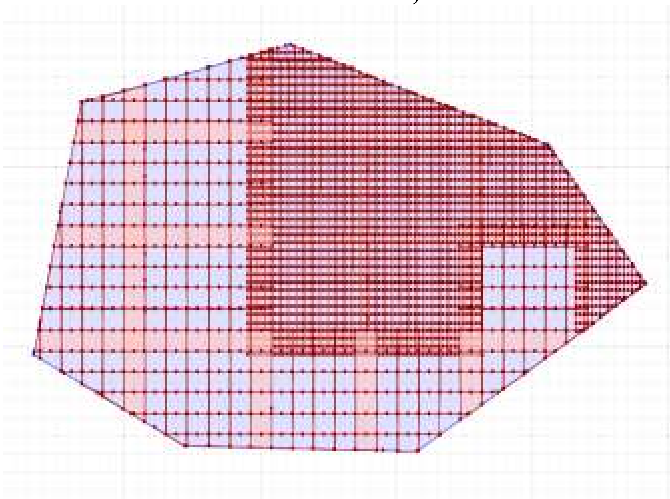


Figure 91: Intra Waste Loaded Use Case 1b

The “Inter-Mission Area Waste” at the end of the test is **10,94%**.

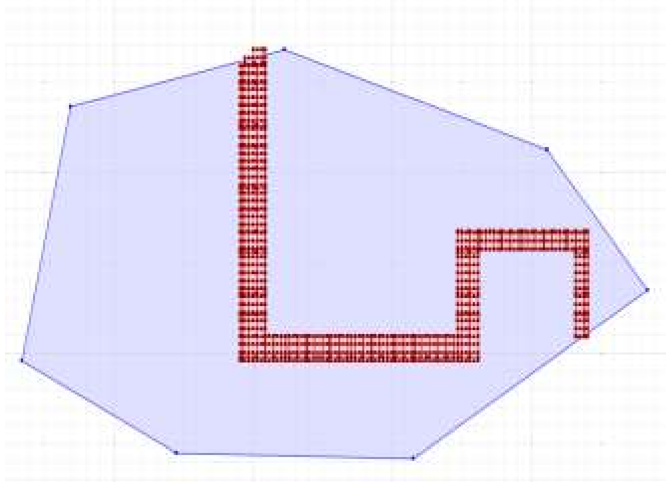


Figure 92 : Inter Mission Waste Loaded Use Case 1b

6.4.4.2.2.3 Use Case 1c

The “Outside Area Waste” at the end of the test is **10,97%**.

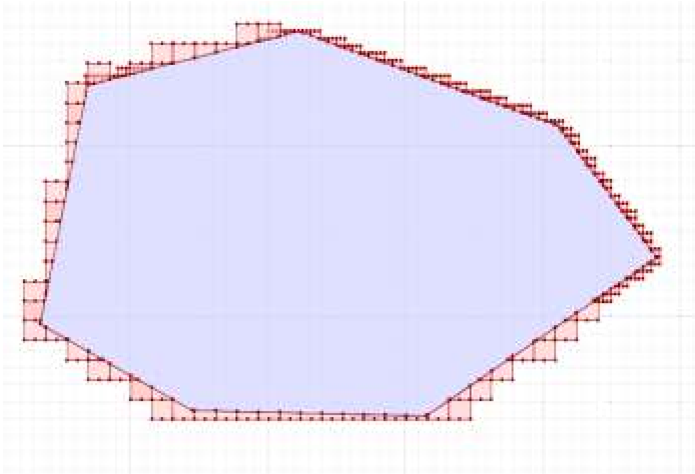


Figure 93 : Outside Area Waste Loaded Use Case 1c

The “Intra Area Waste” at the end of the test is **39,81%**.

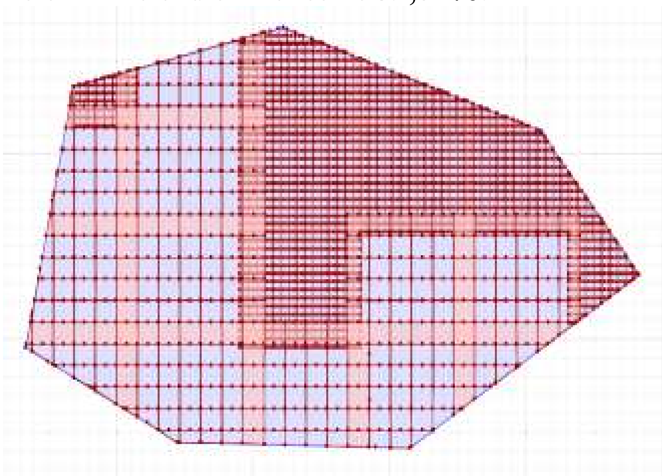


Figure 94: Intra Area Waste Loaded Use Case 1c

The “Inter-Mission Area Waste” at the end of the test is **11,40%**.

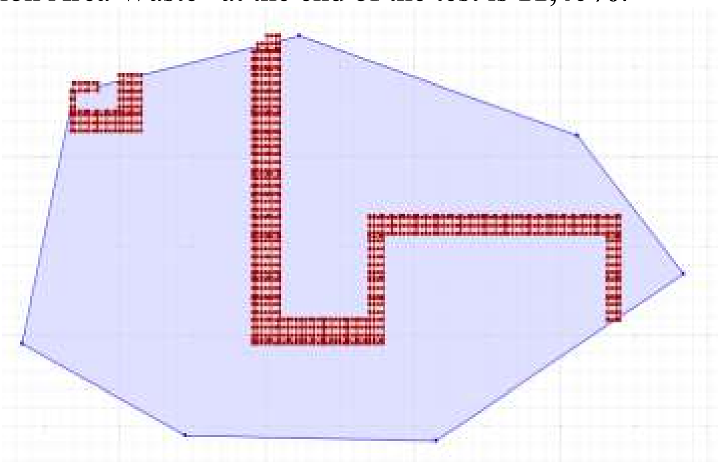


Figure 95: Intra Mission Area Waste Loaded Use Case 1c

6.4.4.2.2.4 Use Case 2

The “Outside Area Waste” at the end of the test is **9,10%**.

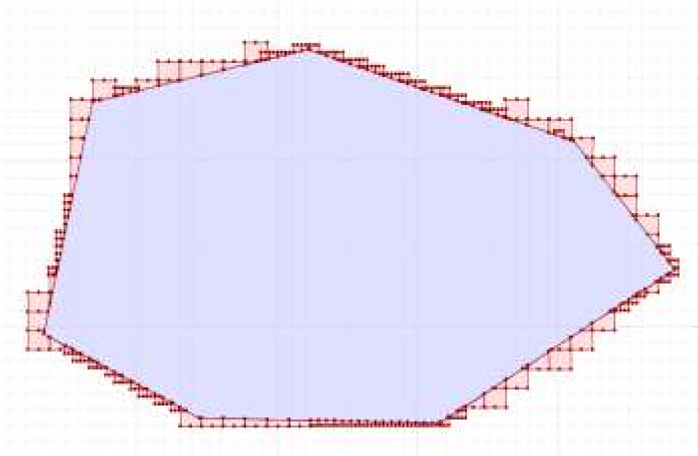


Figure 96 : Outside Area Waste Loaded Use Case 2

The "Intra Area Waste" at the end of the test is **10,98%**.

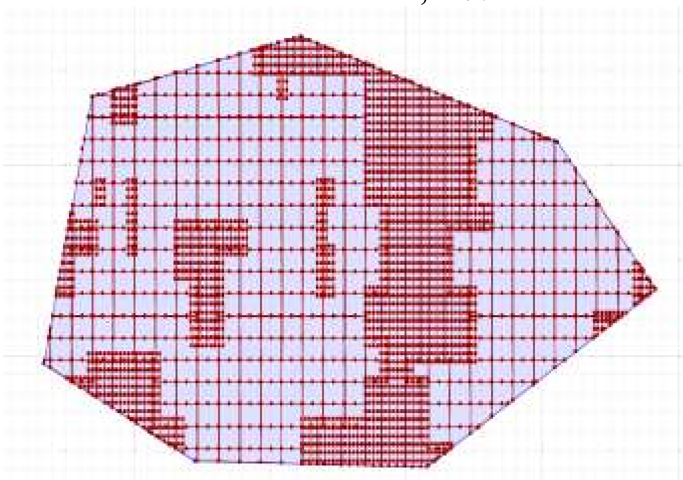


Figure 97 : Intra Area Waste Loaded Use Case 2

The "Inter-Mission Area Waste" at the end of the test is **6,33%**.

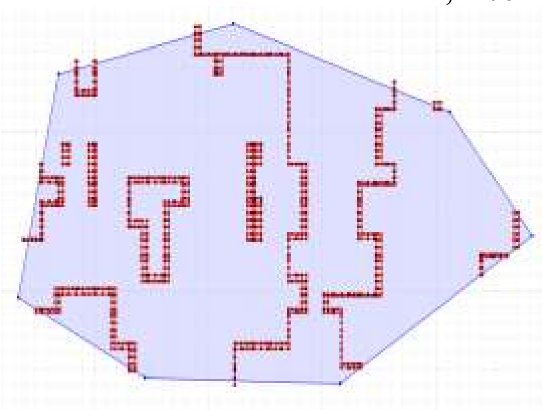


Figure 98 : Inter Mission Area Waste Loaded Use Case 2

6.4.4.2.3 KPI#3 Mission Chains solicitation

6.4.4.2.3.1 Use Case 1a

	CREATE	CANCEL	FOLLOW-UP	WORKLOAD	ALL
MISSION2	1	0	10	11	**22**

MISSION1	5	0	50	11	**66**

Table 39 : Mission Chains solicitation Occitania Loaded UC1a

Even if the manual dispatch is the same than for the Occitania scenario, the Programming Request on Mission 2 is never completed and therefore the workloads are retrieved the three last days of test. The total number of external calls is slightly higher for the scenario Occitania with loaded Mission Chain.

6.4.4.2.3.2 Use Case 1b

	CREATE	CANCEL	FOLLOW-UP	WORKLOAD	ALL
MISSION2	12	0	120	11	**144**
MISSION1	10	0	100	11	**121**

Table 40 : Mission Chains solicitation Occitania Loaded UC1b

6.4.4.2.3.3 Use Case 1c

	CREATE	CANCEL	FOLLOW-UP	WORKLOAD	ALL
MISSION2	17	5	130	21	**173**
MISSION1	15	5	116	21	**157**

Table 41 : Mission Chains solicitation Occitania Loaded UC1c

6.4.4.2.3.4 Use Case 2

	CREATE	CANCEL	FOLLOW-UP	WORKLOAD	ALL
MISSION2	381	305	767	11	**1464**
MISSION1	1184	954	2469	11	**4618**

Table 42 : Mission Chains solicitation Occitania Loaded UC2

6.4.4.2.4 KPI#4 Time Computation

6.4.4.2.4.1 Use Case 1a

	Mean (in seconds)
Global Coverage Completion Estimation computation	18.30047778
MISSION2 – Coverage Completion Estimation computation	2.533133371
MISSION1 – Unitary Assessment	17.57941858

Table 43 : Time computation Occitania Loaded Uc1a

6.4.4.2.4.2 Use Case 1b

	Mean (in seconds)
Global Coverage Completion Estimation computation	35.43912507
MISSION2 – Coverage Completion Estimation computation	2.364989486
MISSION1 – Unitary Assessment	15.41910543
Dispatch	68.90048834

Table 44 : Time computation Occitania Loaded Uc1b

6.4.4.2.4.3 Use Case 1c

	Mean (in seconds)
Global Coverage Completion Estimation computation	34.64422856
MISSION2 – Coverage Completion Estimation computation	2.346184605
MISSION1 – Unitary Assessment	15.56800994

Dispatch	68.94235469
Redispatch	62.38504729

Table 45 : Time computation Occitania Loaded Uc1c

The computation time are globally like the computation time of the Occitania scenario. The additional workload doesn't have impact on the computation time.

6.4.4.2.4.4 Use Case 2

	Mean (in seconds)
Redispatch	51.85360114

Table 46 : Time computation Occitania Loaded Uc2

6.4.4.2.5 KPI#5 Resource Usage

6.4.4.2.5.1 Use Case 1a

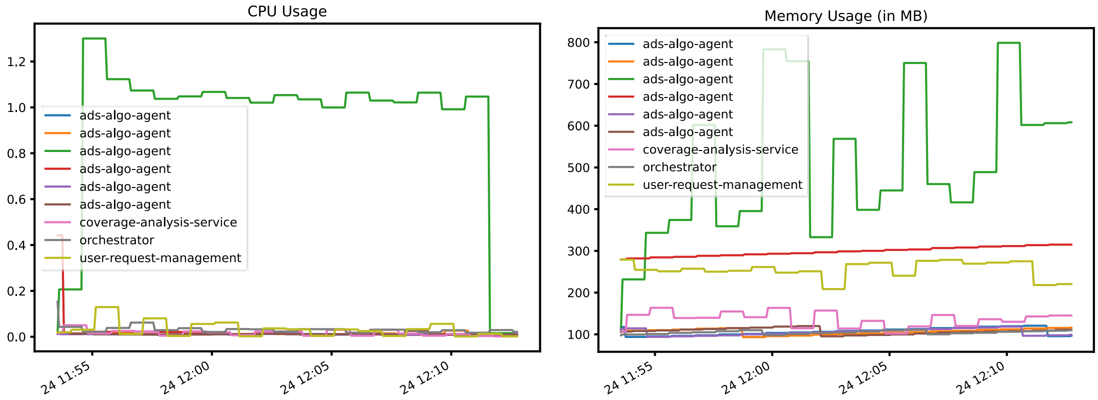


Figure 99 : Resources usage Occitania Loaded for Use Case 1a

6.4.4.2.5.2 Use Case 1b

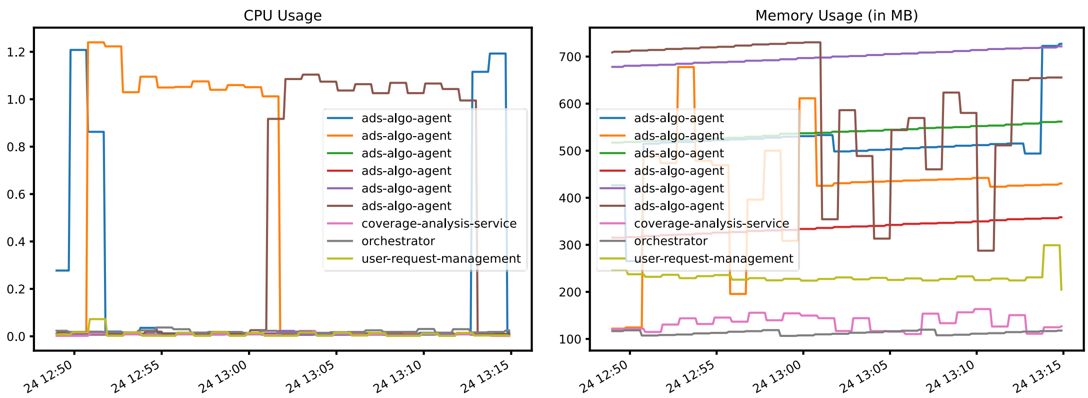


Figure 100 : Resource usage Occitania Loaded for Use Case 1b

6.4.4.2.5.3 Use Case 1c

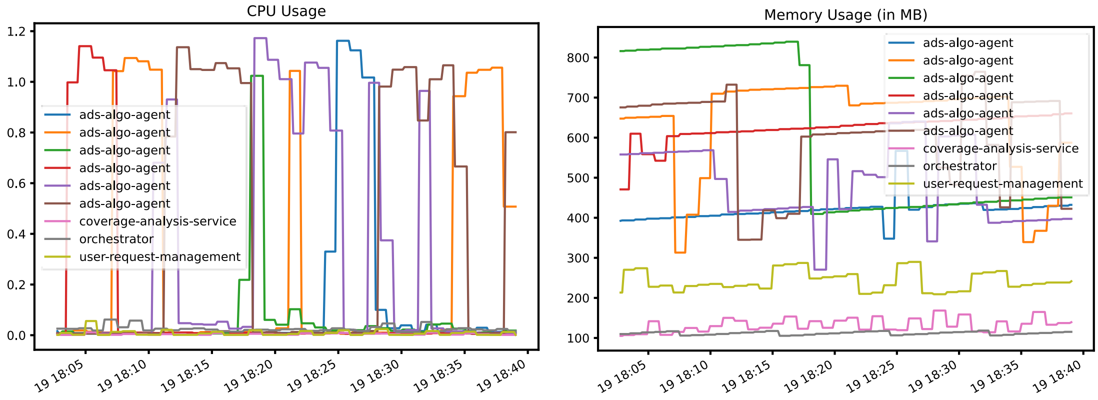


Figure 101 : Resources usage Occitania Loaded for Use Case 1c

6.4.4.2.5.4 Use Case 2

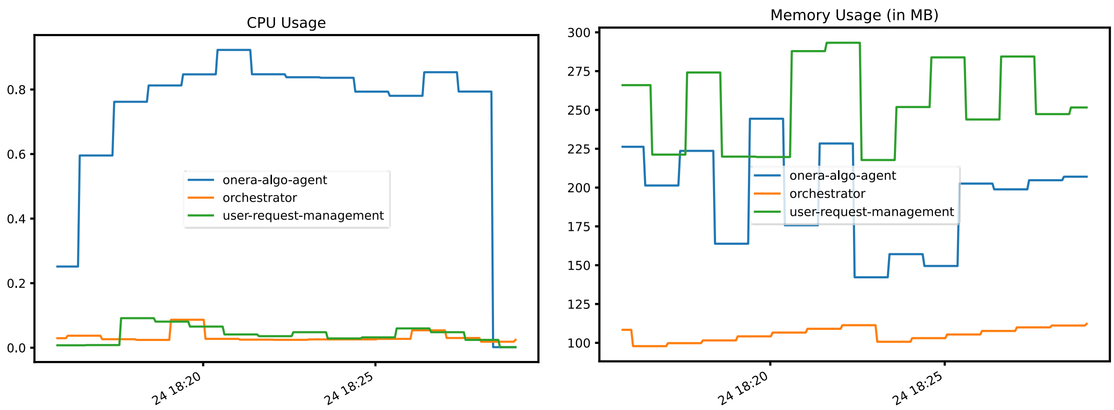


Figure 102 : Resource usage Occitania Loaded for Use Case 2

6.4.4.2.6 KPI#6 Load balancing

6.4.4.2.6.1 Use Case 1a

	MISSION1		MISSION2		Ratio
	Activation	Cancel	Activation	Cancel	
Initial Dispatch	5	0	1	0	0.2

Table 47 : Load balancing Occitania Loaded Uc1a

6.4.4.2.6.2 Use Case 1b

	MISSION1		MISSION2		Ratio
	Activation	Cancel	Activation	Cancel	
Initial Dispatch	10	0	12	0	1.2

Table 48 : Load balancing Occitania Loaded Uc1b

6.4.4.2.6.3 Use Case 1c

	MISSION1		MISSION2		Ratio
	Activation	Cancel	Activation	Cancel	
Initial Dispatch	10	0	12	0	1.2
Redispatch 1	0	1	1	0	1.4
Redispatch 2	0	0	0	0	1.4
Redispatch 3	1	3	3	1	2.1

Redispatch 4	0	0	0	0	2.1
Redispatch 5	4	0	0	4	1.0
Redispatch 6	0	0	0	0	1.0
Redispatch 7	0	0	0	0	1.0
Redispatch 8	0	1	1	0	1.2
Redispatch 9	0	0	0	0	1.2
Redispatch 10	0	0	0	0	1.2

Table 49 : Load balancing Occitania Loaded Uc1c

The average ratio is 1.35.

6.4.4.2.6.4 Use Case 2

	MISSION1		MISSION2		Ratio
	Activation	Cancel	Activation	Cancel	
Redispatch 1	229	0	67	0	1.5
Redispatch 2	120	86	30	29	1.3
Redispatch 3	113	116	35	34	1.3
Redispatch 4	87	84	28	25	1.4
Redispatch 5	85	101	37	34	1.5
Redispatch 6	68	76	21	21	1.6
Redispatch 7	91	91	31	32	1.5
Redispatch 8	98	72	32	29	1.5
Redispatch 9	72	127	38	33	2.0
Redispatch 10	114	88	33	30	1.8

Table 50 : Load balancing Occitania Loaded Uc2

The average ratio is 1.5.

7 INTEGRATION WITH OTHER DOMINO-E APPLICATIONS

Another step in the validation campaign of the Coverage Service is its integration with other DOMINO-E applications. As the complete End-to-End test scenario has already been described in another deliverable (RD1), we will only focus on the Coverage Service part in this section.

7.1 End-to-end scenario, focused on the Coverage Service

As the main objective of the End-to-end scenario is to prove the capacity of the applications to work together, the test has been run only with the Coverage Service in Use Case 1c mode.
During this scenario, the Mission Chain simulators are configured to use the planning simulation described in 3.3, as a high accuracy of the planning is not needed for End-to-end test purposes.

Here are the steps of the End-to-end scenario, from the Coverage Service point of view:

1. Submission of a User Request on the Sichuan area, coming from the VAS
2. Dispatch of this area between PNEO and CO3D Mission Chains and submission of the according Programming Request
3. Simulation of 10 days of planning by the Mission Chains Simulators
4. Follow-up of the Programming Requests to retrieve their progression
5. Coverage Reassessment to compute the updated estimation of coverage
6. Redispatch to update the dispatch according to the updated inputs
7. Step 3 to 6 are repeated 10 days after

7.2 Interactions between components

The following schema describes the interactions between DOMINO-E applications, focused on interactions of the Coverage Service. To help the understanding of the complete loop, the Mission Chain Simulators have been added in the schema.

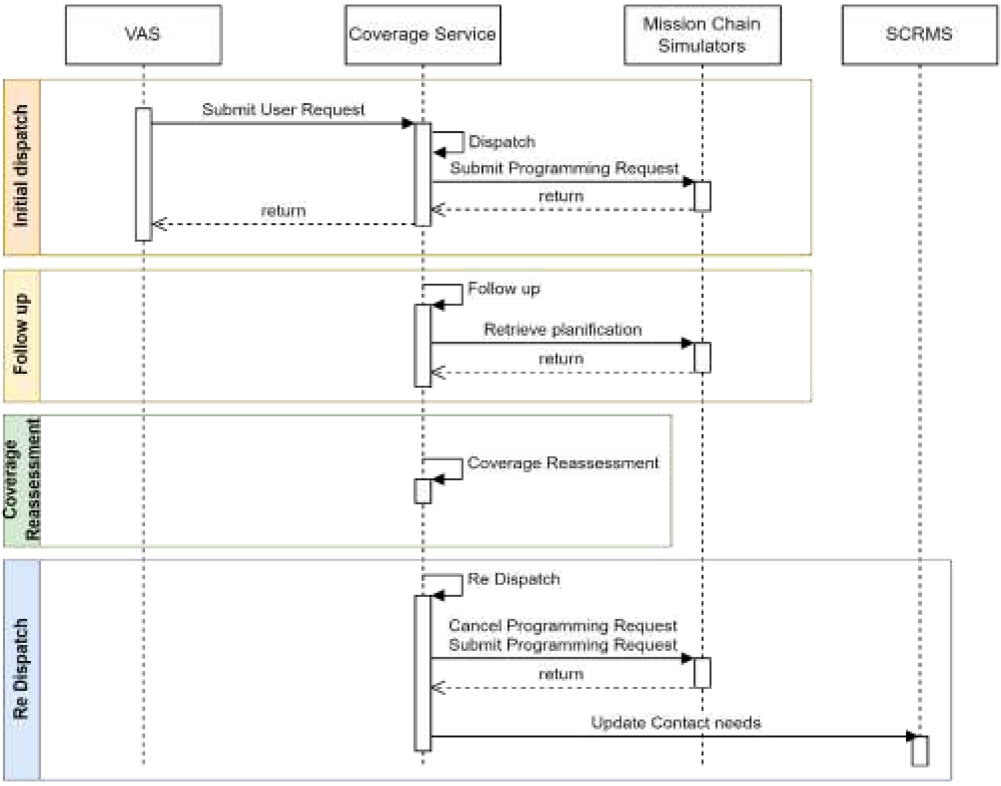


Figure 103: End-to-end interactions between components

During the submission, a User interacts with the UAS containing the VAS until its User Request is ready to be submitted. The UAS then sends the creation request to the Coverage Service, which will dispatch this User Request between the wanted Mission Chains. Once the submission of all the Programming Requests terminated, it answers the dispatch the UAS, which displays the result on a map.
Then, periodically, the Coverage Service retrieves the progression of the submitted Programming Requests amongst the Mission Chains.
It also periodically triggers the re-computation of the coverage estimation, to match the updated progression of the Programming Requests.
During the periodic redispatch, the Coverage Service launches a new dispatch with all the activated User Request to define a new optimised dispatch result according to the progression updates. At the end of the redispatch, all the Programming Requests are retrieved and transformed into contact needs that will be sent to the SCRMS.

7.3 Results

After the submission coming from the UAS, the User Request is dispatched and the result of this dispatch is displayed in the following screenshot:

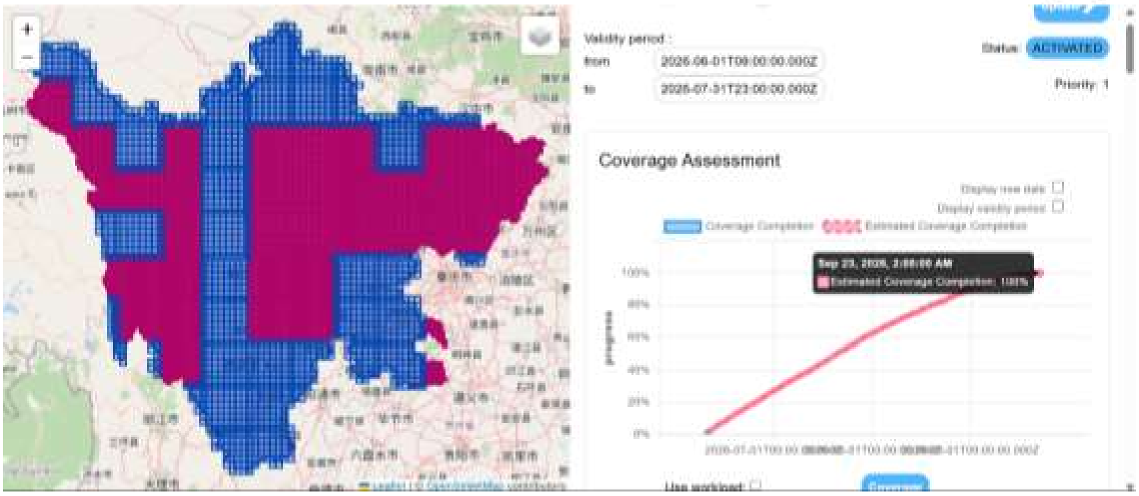


Figure 104 : Initial dispatch of Sichuan area

After the 20 days of simulated planning, the final state of the User Request is a follow:

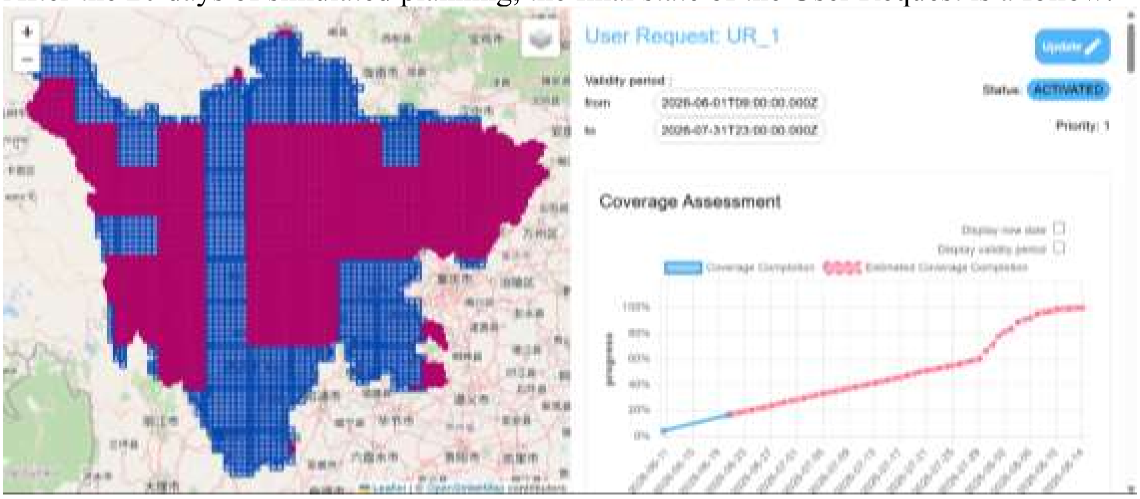


Figure 105 : Final state of the End-to-end test

After each redispatch, the contact needs have been sent to the SCRMS.

Finally, this end-to-end test allowed us to verify the interactions between the DOMINO-E components. A more complete analysis of the results is available on the deliverable RD1.

8 Results Analysis

8.1 Exploitation of test results – Summary

The goal of this chapter is to exploit the results of the precedent section and summarize some of the conclusions that have been drawn from those results.

Here is a graph showing all the result for the KPI of Coverage Completion on each scenario and each Use Cases:

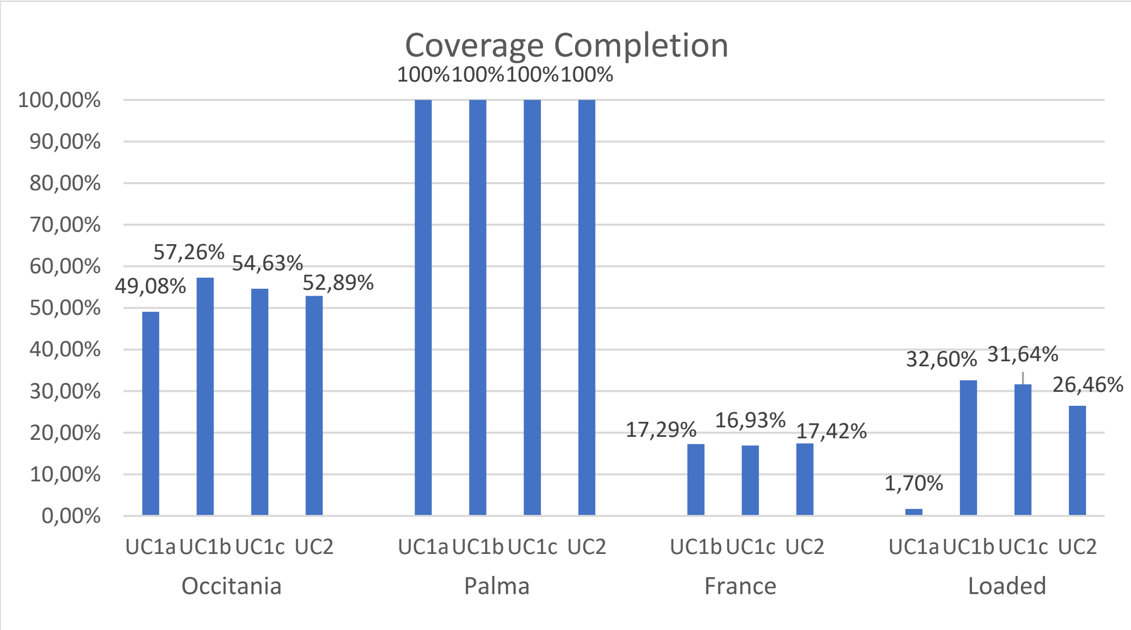


Figure 106 : Coverage Completion – Summary

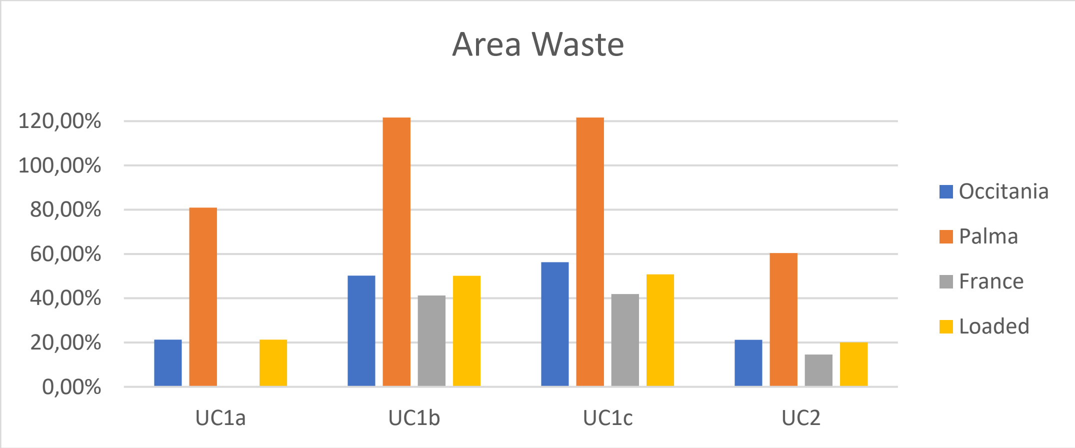
General observations:

- The load has an impact on the final coverage completion
- The bigger the area, the slower the completion

Comparisons :

- Coverage Completion are globally similar
- The UC1b (without redispatch) is permanentely better than UC1c and UC2 (with redispatch)
- UC2 in « Loaded » mode is a little less performant than the other Use Cases

The two following graphs show the evolution of the area waste KPI on each scenario per Use Cases, with or without Mission Chains optimizations:



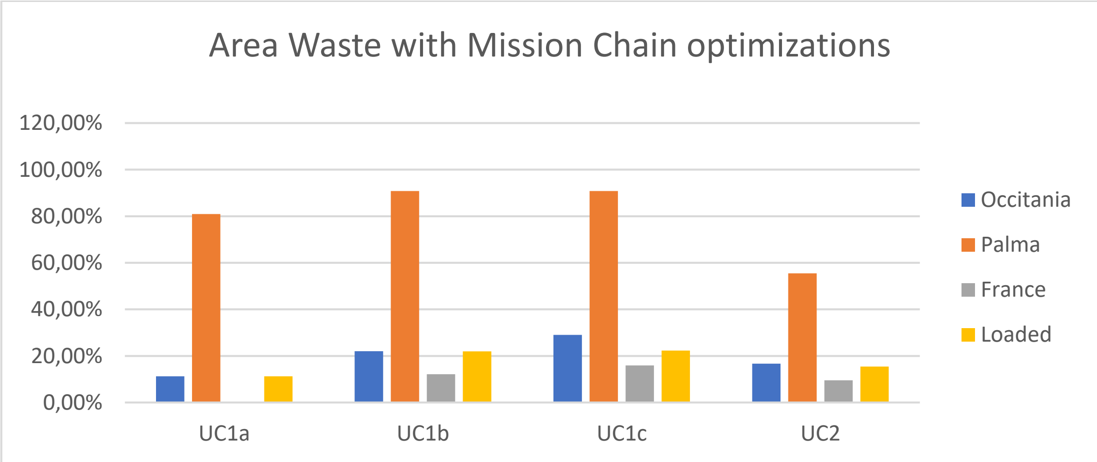


Figure 107 : Area Waste – Summary

General observations:

- The smaller the area, the bigger the Area Waste
- Considering Mission Chains optimizations considerably reduces the Area Waste percentage

Comparisons :

- UC1 generates more area waste than UC2 (due to misalignment during the splitting of the User Request)

The following graphs shows the number of external calls performed during the tests by Use Cases:

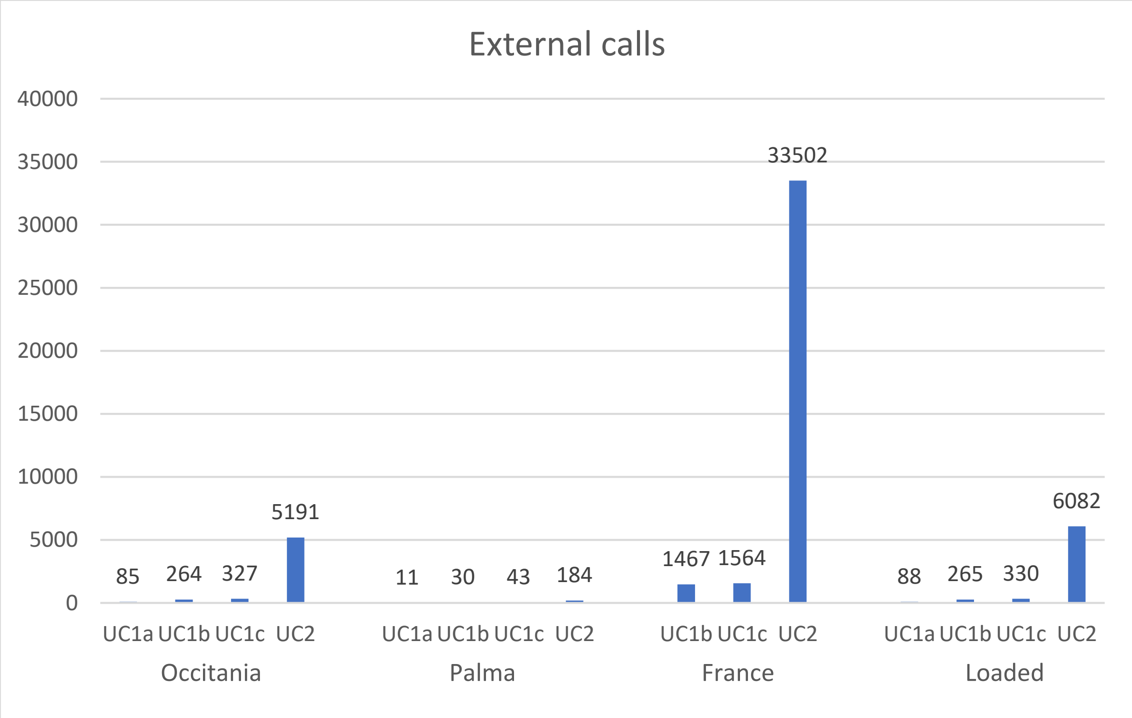


Figure 108 : External calls – Summary

General observations:

- The bigger the area the more important the number of interactions with external components

Comparisons :

- The redispatch generates more interaction with external components
- UC2, by its design, generates a lot of external calls. However, using the precedent dispatch would reduce the number of changing dispatches

In the following graphs, the time of computation of the initial dispatch (only for UC1c) and the average of computation time for redispatches are displayed:

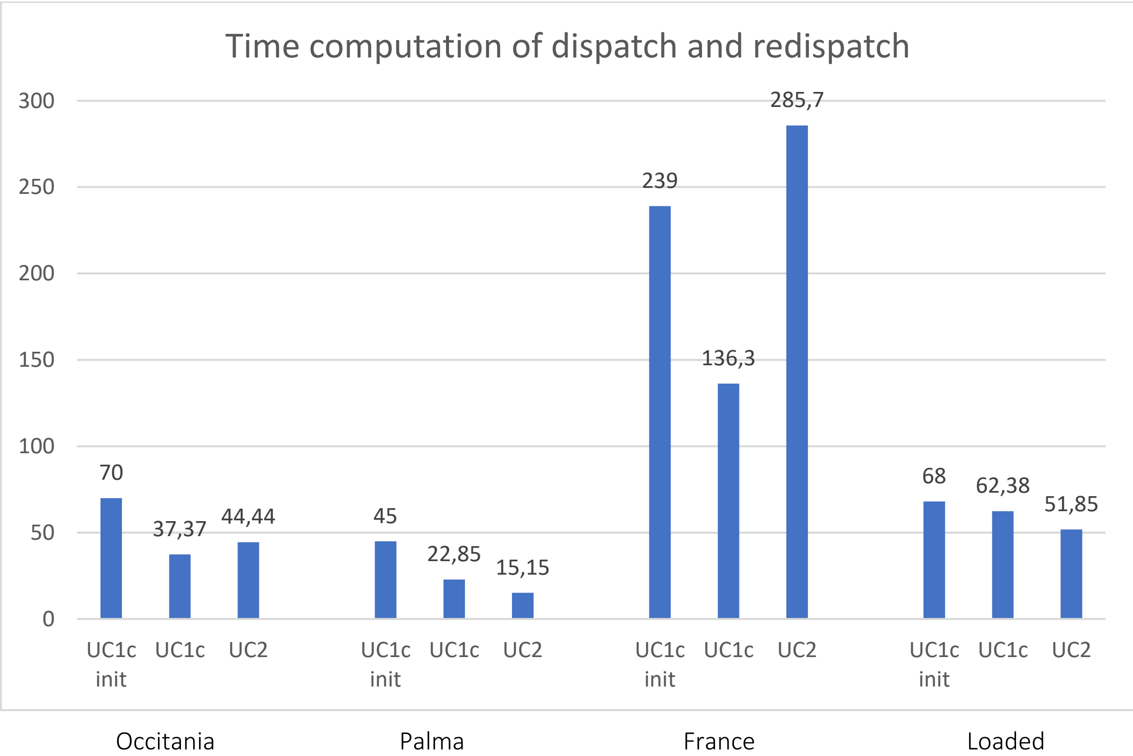


Figure 109 : Time of dispatch - Summary

General observations:

- The bigger the area, the bigger the time of dispatch

Comparisons :

- UC1 allows to configure the number of iterations of the algorithms only: the time of dispatch is then proportional to the size of the area. UC2 allows to configure the number of iterations of the algorithms AND a time of dispatch: It allows more predictability of computation time.
- UC1 reuses the initial dispatch performed, its redispatches time are therefore reduced.

The two following graphs show, for the scenario of Occitania (on the left) and for France (on the right), the CPU usage of algo-agent in Use Case 1c:

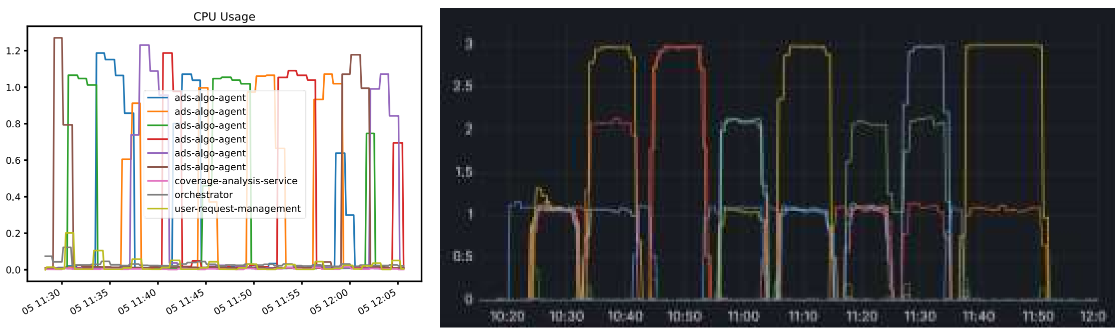


Figure 110 : CPU Usage between Occitanie and France Scenarios for UC1c

General observations :

- The bigger the area, the more resources used.

Comparisons :

- UC1 requires a lot of algorithmic calls (Coverage Completion, Dispatchs..) and some of them can be parallized. The parallelization is useful only on bigger areas.
- As the Use Case 2 algorithms are not parallelized, few conclusions can be drawn of its CPU usage.

In the following graphs, the load balance ratio is displayed for each Use Case and each scenario

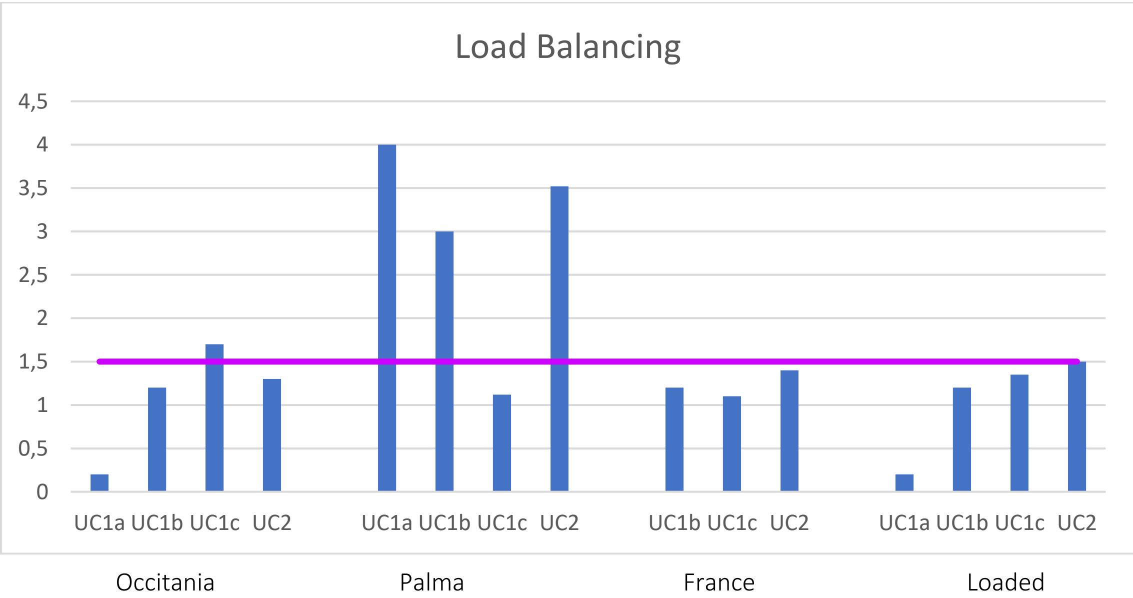


Figure 111 : Load Balancing - Summary

General observations:

- The smaller the area, the worst the load balancing

8.2 Coherence of algorithmic estimations with Mission Chains real behaviour

8.2.1 Splitting

8.2.1.1 Use Case 1

Some differences of splitting have been observed between coverage algorithmic results and Mission Chains results. We can witness a misalignment between the dispatched area and the sub-split performed during submission on the Mission Chains. For example, in the following image, we can see the dispatch of an area by the algorithms and its sub-split by the two Mission Chains. We can see that the misalignment of the dispatch leads to a superposition at the frontier of the Programming Requests.

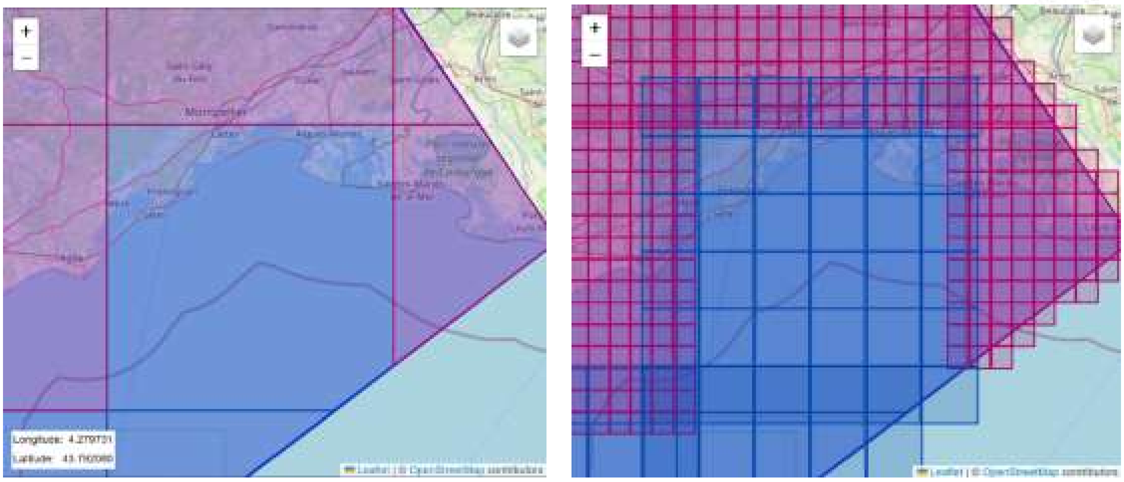


Figure 112 : Dispatch of an area by algorithms (on the left) and its sub-split over the Mission Chains (on the right)

<u>Cause:</u>
The goal of the UseCase 1b and 1c is to find, for a User Request, a split of the initial area that will be submitted between the different Mission Chains.
Ideally, the subarea size is a multiple of the size of each Mission Chains unitary area (AcqR). But this constraint is hardly fulfilled when the precision of the size of the AcqR of each Mission Chains is high. Therefore, an approximation to the nearest kilometres is made.
This approximation causes a misalignment of the subarea of the coverage and the splitting of the Mission Chains.
This problem occurs only because we chose a certain type of splitting called "WORLD_LAYERED_SPLIT", to map the split on a fixed pre-computed world grid. This grid is not aligned between dispatch algorithms and Mission Chain algorithms which lead to this misalignment.

<u>Consequences:</u>
The main consequence of this misalignment is that some areas are submitted several times for the same Mission Chains or submitted on different Mission Chain.
In the previous example, we can see that some of the red meshes are completely covered by blue ones. When neighbour areas are submitted on the same Mission, the same area can be submitted up to 4 times, as shown in the following screenshots.

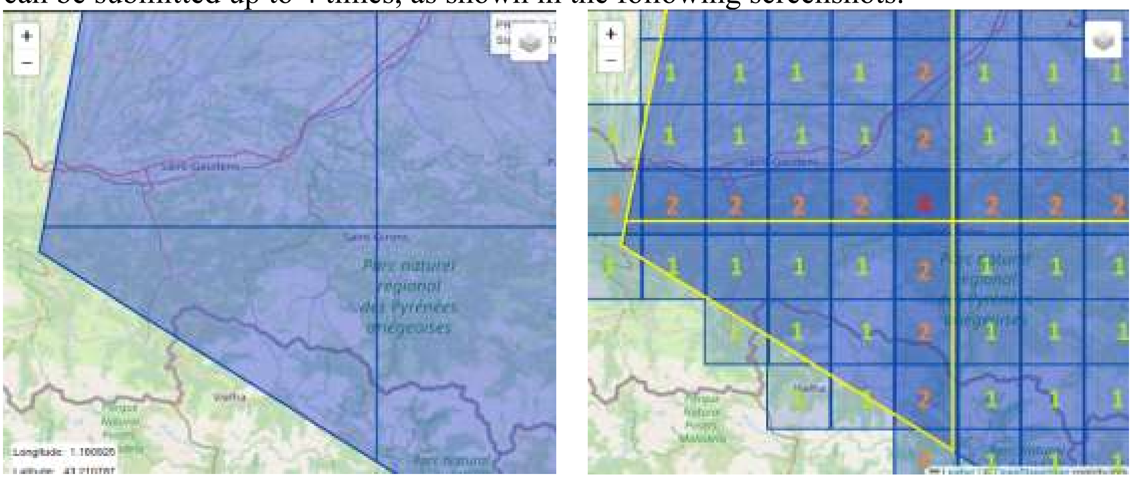


Figure 113 : Split on MISSION2 by algorithms (on the left) and its sub-split once the area submitted on the Mission Chain (on the right)

This causes a non-necessary overload of the Mission Chains and will make the planning estimation less accurate.

Remediation:
A first mitigation for this problem is that some Mission Chains can detect the overlapping of several submitted area and only plan once the corresponding area. This will cancel the impact of the areas planned several times on the same Mission Chain.
However, the problem of submission on different Mission Chain remains.

A remediation to this problem would be to change the split method of the requests. If we use the DYNAMIC_FIXED_ORIENTATION method (instead of WORL_LAYERED_SPLIT), which is a split aligned on the satellite orientation, the splitting would be more similar.

8.2.1.2 Use Case 2

Unlike the Use Case 1, the algorithms of Use Case 2 require a mapping of the entire earth by all the addressed Mission Chains to operate. In the Coverage Service prototype, the files are furnished at the deployment of the algorithms.
This method ensures the adequation of the algorithm and the Mission Chains splitting.

8.2.1.3 Comparison of the two algorithmic solutions

To sum up, the splitting of the algorithmic solution of Use Case 2 is **very accurate,** but also **very complex to configure** because it requires a first step of furniture of each Mission Chain splitting of the entire earth.
By contrast, the splitting of the algorithmic solution of Use Case 1 is **less accurate** but **loosely coupled** with the Mission Chains as only a few parameters are needed in its configuration.
The algorithms of Use Case 1 will also **be more easily extensible** to other splitting method, taking the orientation of the satellite into account for example, unlike the algorithms of Use Case 2, which have a fixed splitting of the Earth.

8.2.2 Planning estimation

In each algorithmic solution, an estimation of the planning of each Mission Chain is realised to choose the dispatch solution optimizing the completion date. It can therefore be interesting to compare the prevision of planning with the real planning of the Mission Chains.

8.2.2.1 Use Case 1

The misalignment of the splitting obviously has an impact on the planning estimation. In addition, we have fewer tools to analyse the access considered in the planning than in the algorithmic implementation of the Use Case 2. Due to these two last causes, the analysis is complicated to conclude.

However, we can compare the planning performed through UseCase 1b and the one performed through UseCase 1c and its redispatches.

For example, when we focus on the 6th plan performed on the Occitania region, here are the two results of planning (in green) for UC1c (right) and UC1b (left):

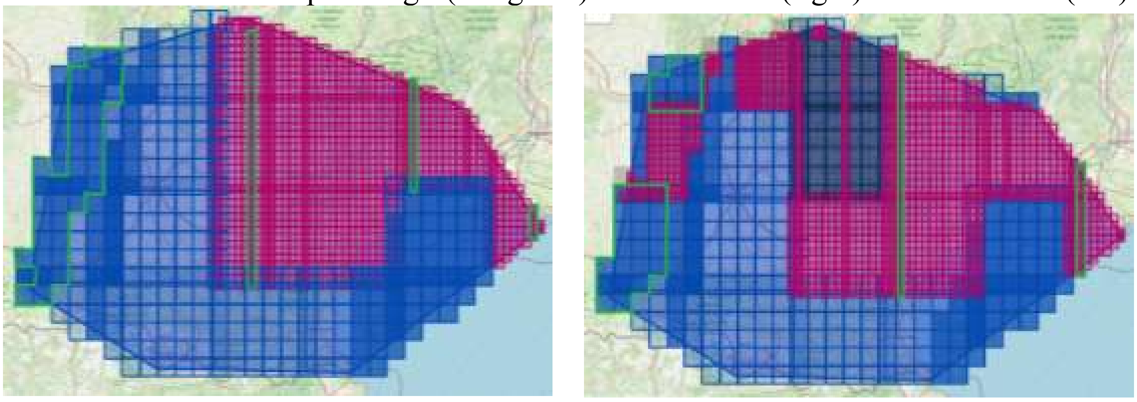


Figure 114 : Planification results of 6th plan for UC1b (left) and UC1c (right)

In the following unique visualization, we can highlight the planning result with UC1b (yellow) and the planning result with UC1c (green).

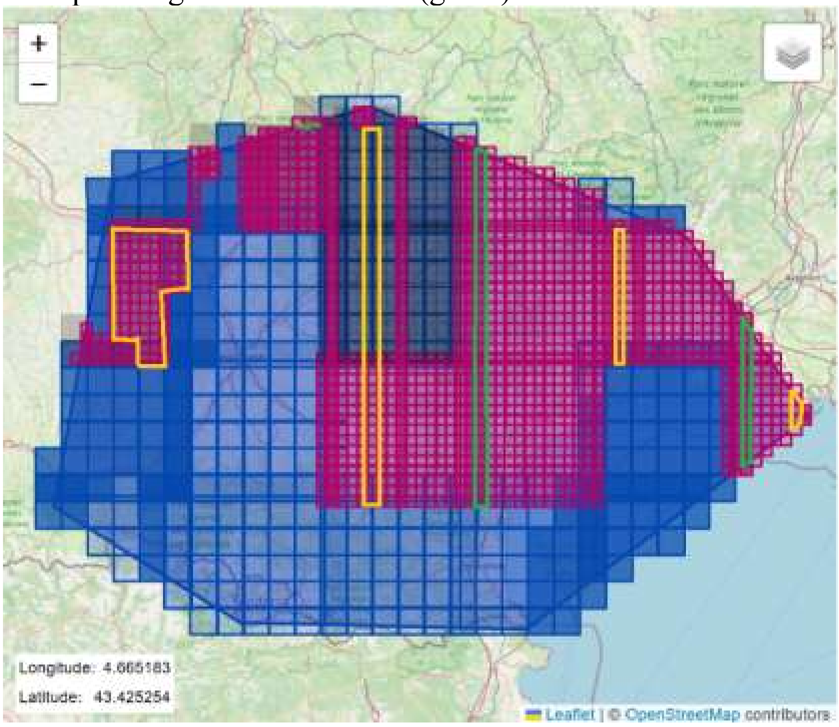


Figure 115 : Focus on the planning difference on 6th plan

Specifically on this plan iteration, the UseCase 1c decided to dispatch the area on the left side of the UserRequest on CO3D whereas PNEO constellation would have been able to acquire all this area

This behaviour can be explained by the approximation of planning, the need to balance the dispatch between the missions or meteorological hazards.

Further analysis must be conducted to determine the part of the behaviour which is due to an inaccurate estimation (that can be reduced) or to a justifiable behaviour (that must be kept).

However, the estimation date of completion for requests are continuously better with the UseCase 1c than with the UseCase 1b.

8.2.2.2 Use Case 2

Differences of planning have been observed during tests.
For example, on the test “Occitania Coverage”, the algorithms were forecasting an access on the satellite M2_2 on the next planning day, but the MISSION2 Mission Chain didn’t plan anything on this satellite.

<u>Cause:</u>
This difference of planning is coming from the test chronology:

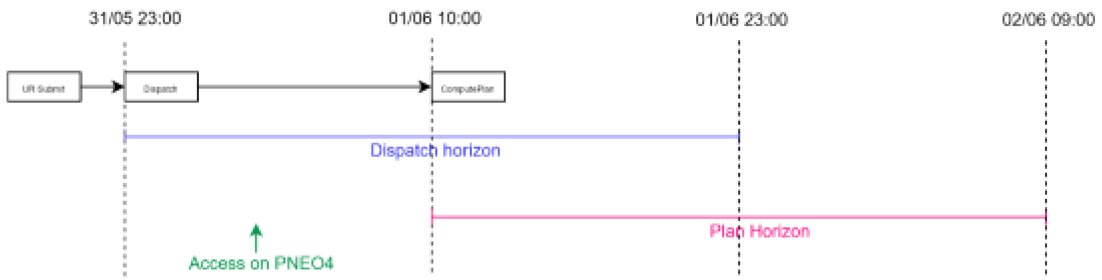


Figure 116 : Test chronology for Use Case 2

In our test, the Dispatch is launched in the middle of the night at 23:00. Whereas the next plan is computed only at 10:00 in the morning of the next day.
During the dispatch, the algorithms predict an access around 05:00 (represented by the green arrow in the previous schema) which is not contained in the plan horizon. Therefore, the estimation is not exactly accurate with the real planning.
However, this case can appear only on the first dispatch and plan computation. As soon as the second dispatch is launched, the plan horizon will completely cover the dispatch horizons, as show in the next schema.

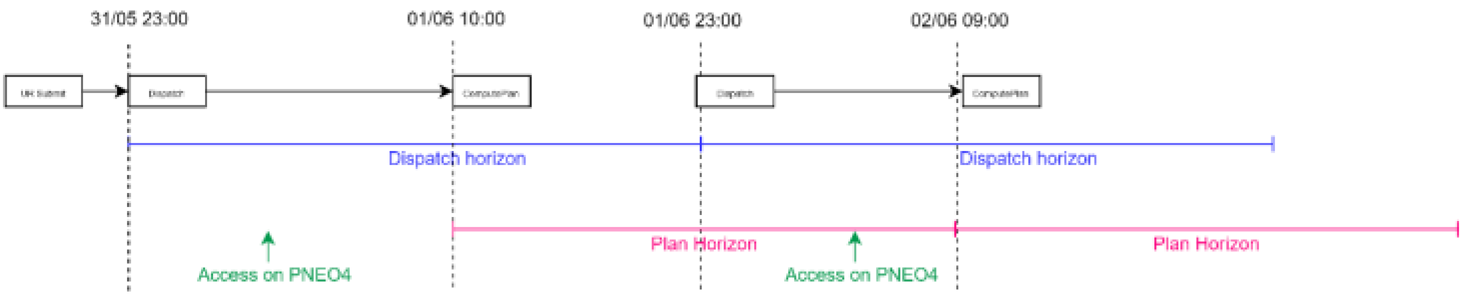


Figure 117 : Second dispatch and plan chronology

The access on M2_2 found on the second dispatch will already have been planned on the first plan. The estimation and the real planning will match.

Nevertheless, some limitations connected to the chronology can be found, even if not experienced: in the next schema, we’ve shown two cases where the estimation and the real planning will be desynchronised.

- Red arrow case:

In this case, the second dispatch predict an access on a satellite that wasn't plan of the first plan (for example, if a new request is submitted after the first plan computation). This access will never be really planned.

- Green arrow case:

In this case, an access is planned on the first plan but not predicted by the second dispatch (for example, this access is less urgent than other newly submitted requests).

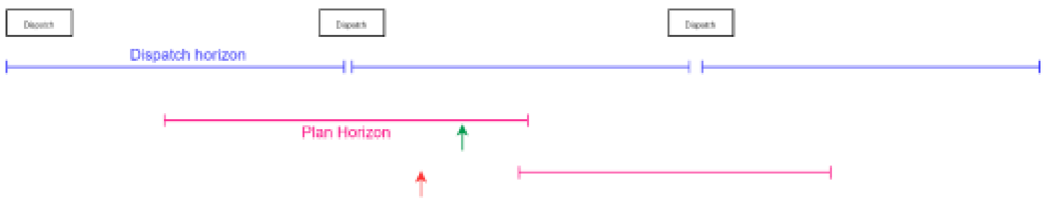


Figure 118 : Chronology limitation

<u>Consequences:</u>
Consequently, the estimation will be less accurate and therefore the optimized decisions made during the dispatch will be less precise.

<u>Remediation:</u>
To remedy the previous explained cases, we identified an improvement axis for the DOMINO architecture.
The algorithms would need as an input, for each satellite, the date of the next upload slot. It will therefore be possible for them to identify the accesses to be ignored during the dispatch computation. In our previous example, the access represented by the "red arrow" will be ignored and the dispatch horizon will start only at the next plan computation.
However, this upload slot date must be furnished either by the SCRMS or the Mission Chains directly. The interaction will have to be described in the DOMINO architecture.

To avoid the green arrow case, where the planning estimation of the second dispatch can differ from the planning of the first plan computation, the Coverage Service would also need to store, for each satellite, the date of the previous plan computation. It will then be able to identify the newly submitted requests and take them into account only from the date of the next plan computation.

8.2.2.3 Comparison of the two algorithmic solutions

To sum up, the estimation of planning can be improved on both Use Cases.

By solving its splitting differences, the Use Case 1c will improve its accuracy of planning estimation. After that, even if the planning estimation on a specific programming request can be less accurate because of the imprecision of the inputs, the planning estimation of the overall User Requests can be relatively accurate because the sum of the estimations balances the possible inaccuracies of the unitary estimation. The Use Case 1c will probably be **less accurate** but **easily configurable**.

The Use Case 2 will need more inputs to refine its estimation of planning (download slots forecast plan, real satellite orbits). It will then probably be **more accurate** but also **more complex to configure**.

8.3 Stabilization of the Use Case 2 dispatch result

We have seen through the different scenario that, each time, the Use Case 2 algorithmic solution generates changes each time a redispatch is launched, in opposition to the Use Case 1c, very stable.
The time of dispatch of the Use Case 1c is also very unstable: when no changes are detected, the time of dispatch is reduced. This optimization will be applicable to UC2 if the dispatch results are stabilized.

Causes:
By its implementation, the Use Case 2 algorithmic solution inserts hazard inside of the decision results. In addition, the smaller size of the programming request increases this effect.

Remediation:
A possible solution, to be implemented in future programs, can be to consider the previous dispatch result and build a different solution starting from an existing state instead of a blank page. It will help stabilize the dispatch solutions and reduce the number of external calls performed to the mission chains.

8.4 Efficiency of redispatch

In all the scenarios we ran during the validation campaign with realistic Mission Chains, the Coverage percentage of the request at the end of the test is inferior for UC1c and UC2 than for UC1b.
However, in all those scenarios, whereas we see the complete coverage completion date moves back each time a re-computation is performed on UC1b, this date remains stable in UC1c. Even if, after 10 days on plan computation, the actual coverage is inferior for Use Cases with redispatches compared to the Use Case without, the forecast complete coverage date is earlier.
Another point to highlight is the absence of weather forecasts consideration during the scenarios: it is a point that can disadvantage the Use Cases with redispatch, as they are configured to manage meteorological hazards during the completion.
In RD5, the efficiency of redispatch have been demonstrated in simulated environments, some steps need to be conducted to prove the reach the same level of effectiveness in more realistic environments.

8.5 Comparison with existing Federation models

8.5.1 Comparison with only one Mission Chain taking the area

To have a point of comparison, we submitted the France country on only MISSION1 and MISSION2. The result of this submission is displayed in the two following screenshots:

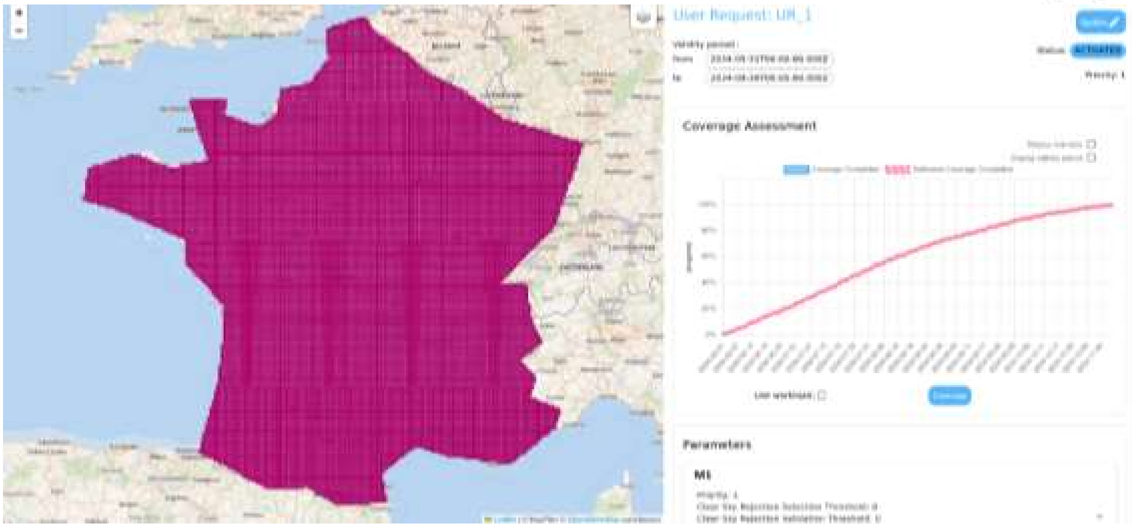


Figure 119 : France submission only on MISSION1

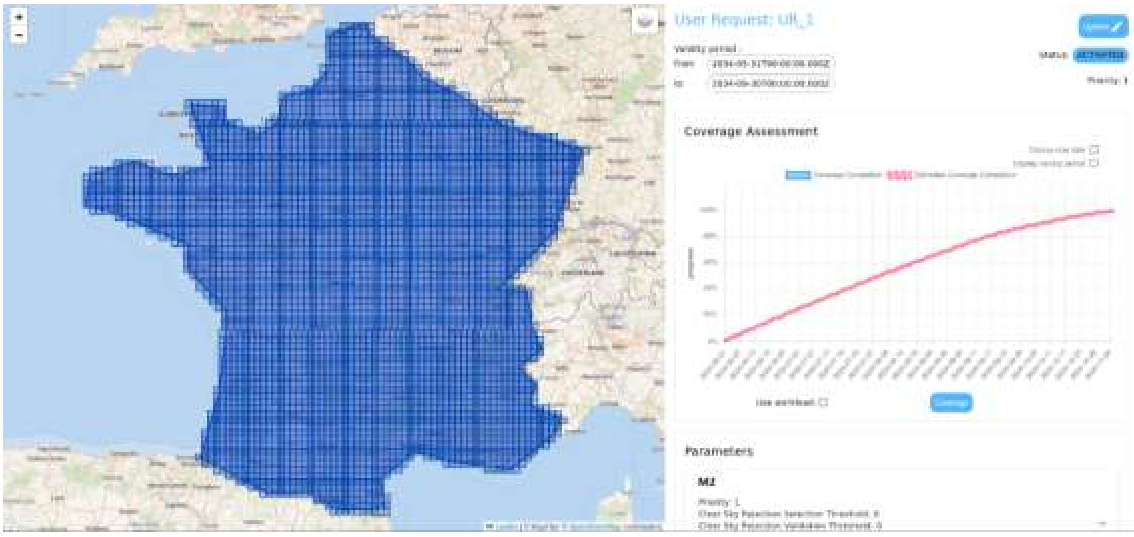


Figure 120: France submission only on MISSION2

The estimation date for the complete coverage is at 08/11/2034 for MISSION1 only and 05/11/2034 for MISSION2.
On submission with both missions, the estimation date for the complete coverage is 09/09/2034. It brings to the fore the improvement of the complete coverage date thanks to the dispatch process.

8.5.2 Comparison with the two Mission Chains taking the entirety of the area

Another possible easy solution for this optimization problem is to submit the complete User Request area on all the Mission Chains we want to address. Each mission will plan all the area, and the Federation application oversees the merging of the result of all those missions planning with an optimised final product.
This way of proceeding ensures complete coverage completion date a little delayed compare to a dispatch strategy result (as the Missions can take time to plan some areas that would not have been submitted on them with the dispatch strategy). However, it generates a **lot of Area Waste** and a **flooding of the Mission Chains**, compared to the dispatch solution implemented for DOMINO-E.

8.6 Improvements axis

In addition to the previously cited remediations of existing problems, we can raise other improvements axis:

- For the moment, the Coverage Service considers a limited number of User Request types (Monoscopic, World Layered Split requests). To fulfil the operational needs of this application, all the interesting requests type should be processed: STEREO/TRISTEREO requests, requests with split method of type Dynamic Fixed Orientation…
- Discussion on DOMINO-X interface about “identifiers”: In the DOMINO-X interfaces, the main identifier used to communicate between the component is the “guid”. In actual Ground Segments, the notion of “external identifier” exists, which allow for each component to have a copy of an object with a guid inside its own perimeter and still discuss with other components about this specific object through the “external identifier”. This discussion must be conducted during the refinement phase of the DOMINO-X APIs.
- Some functionalities, mandatory in operational environments, must be develop (Purge Management…)

9 ANNEXES

9.1 Example of test sequence diagram

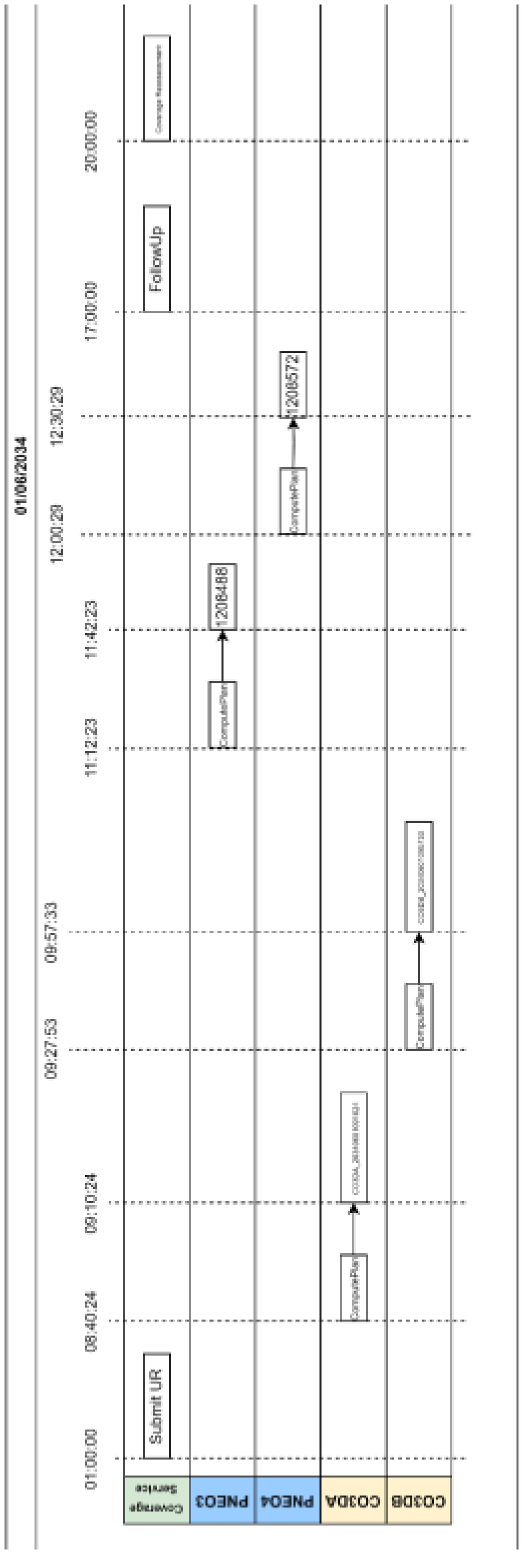


Figure 121 : First day of the test scenario sequence diagram for UC1b

9.2 Occitania Scenario progress

9.2.1 Use Case 1c

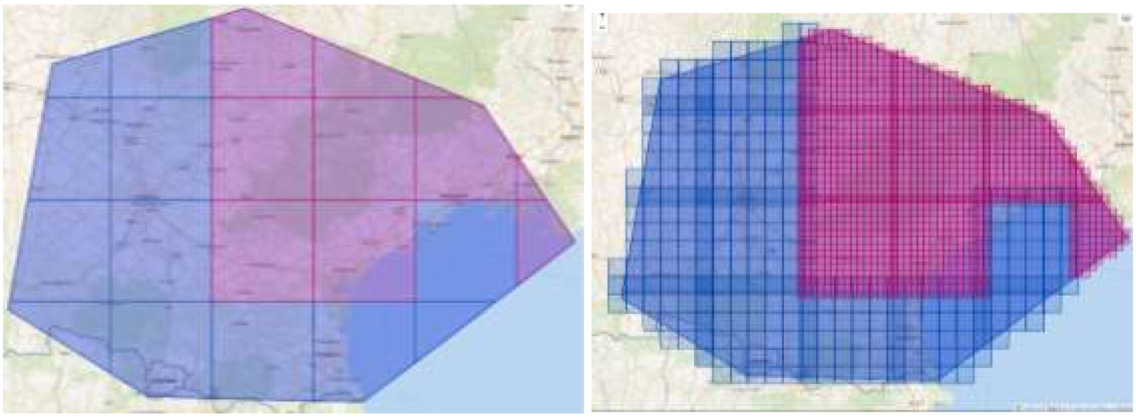


Figure 122 : Occitania scenario Uc1c - After Initial Dispatch

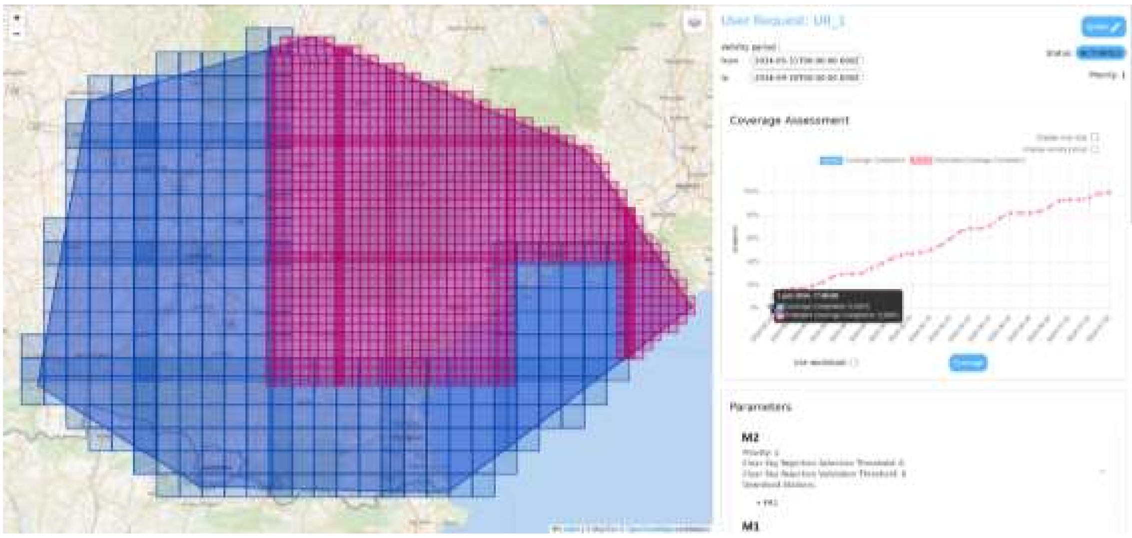


Figure 123 : Scenario Occitania Uc1c - After Plan #1

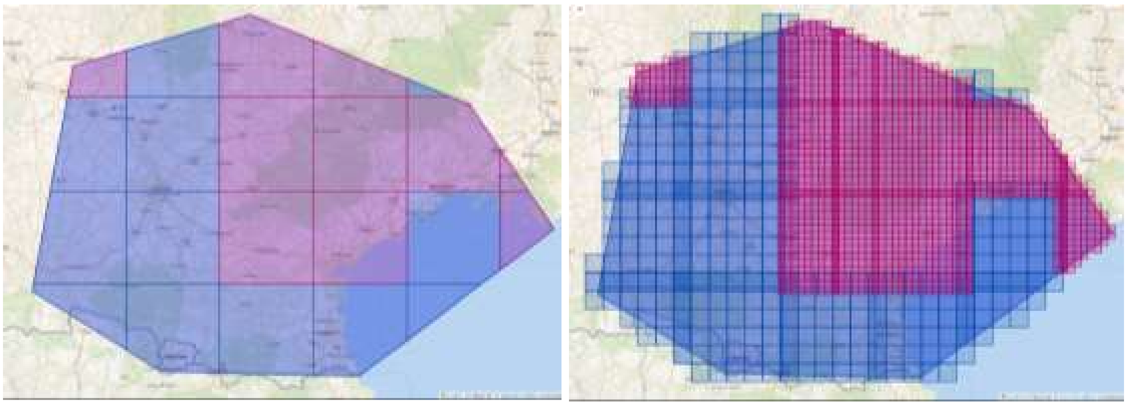


Figure 124 : Occitania scenario Uc1c - After Redispatch #1

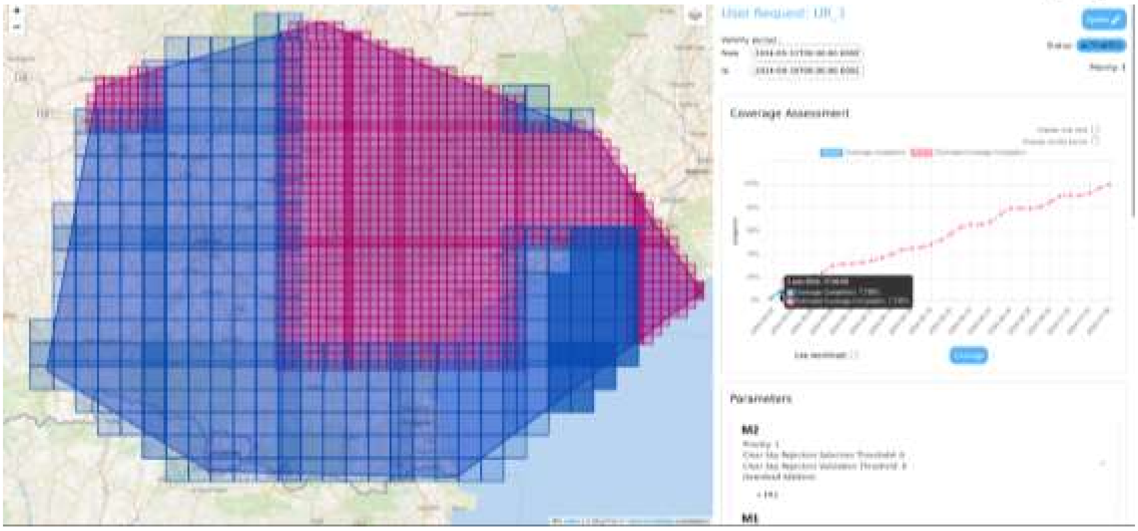


Figure 125: Scenario Occitania Uc1c - After Plan #2

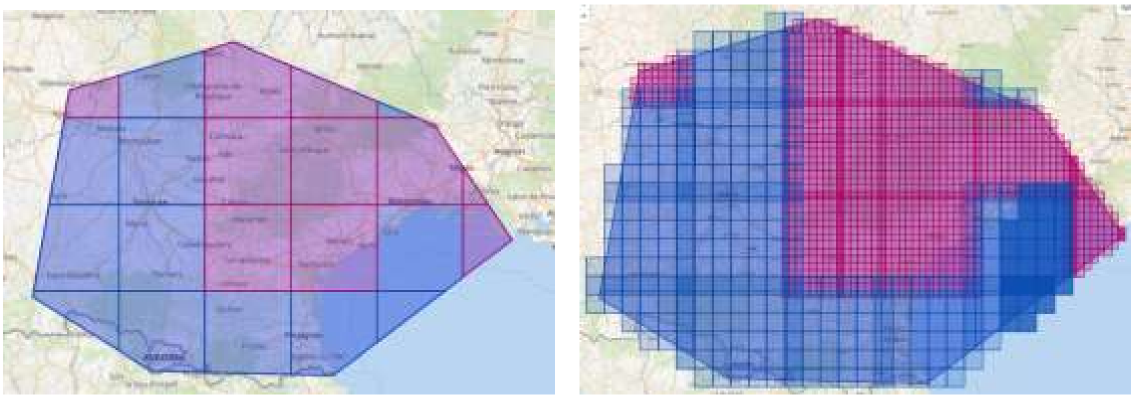


Figure 126 : Occitania scenario Uc1c - After Redispatch #2

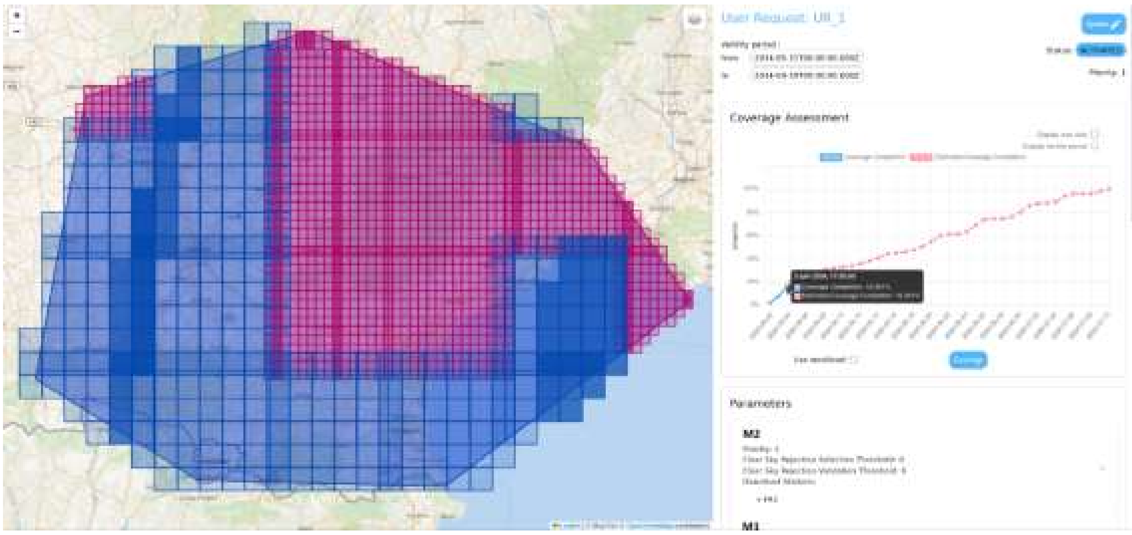


Figure 127 : Scenario Occitania Uc1c - After Plan #3

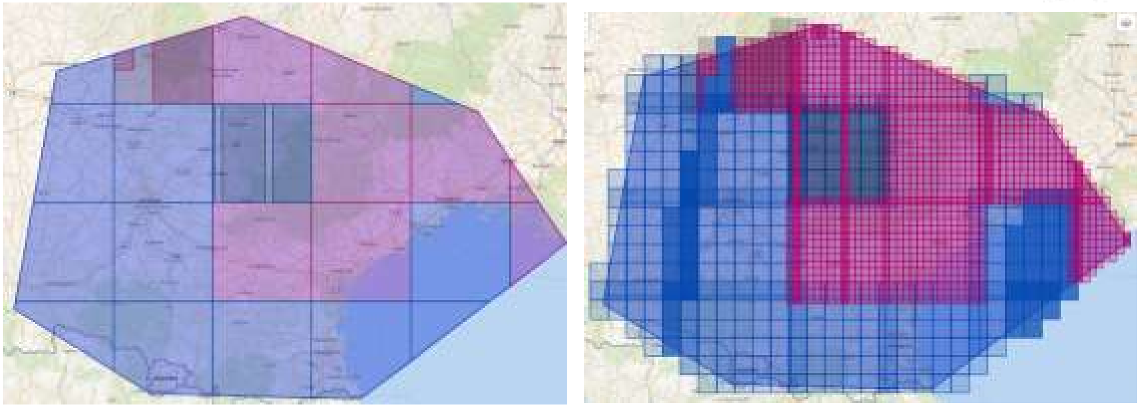


Figure 128 : Occitania scenario Uc1c - After Redispatch #3

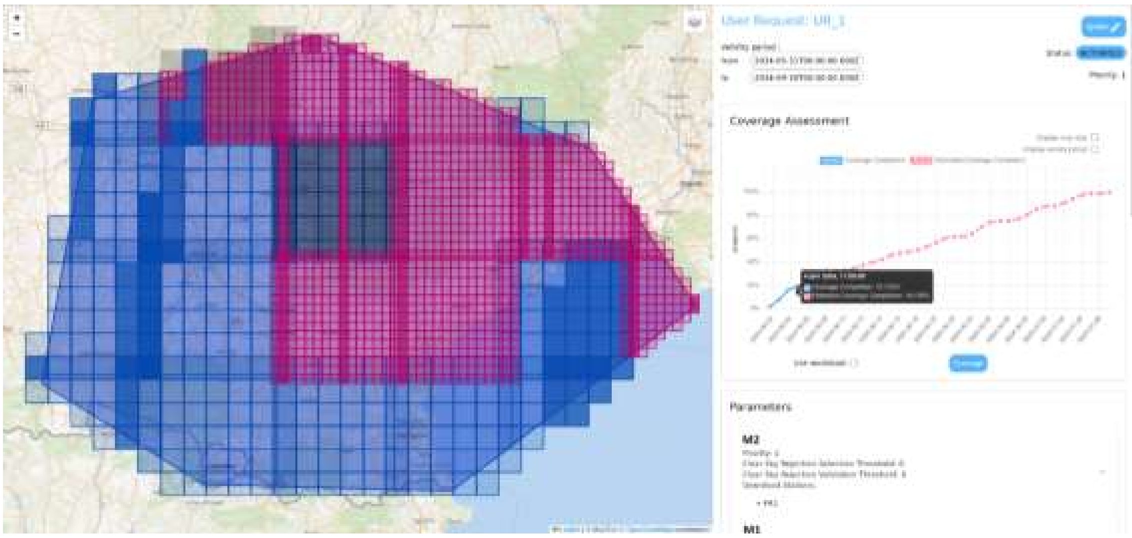


Figure 129 : Scenario Occitania Uc1c - After Plan #4

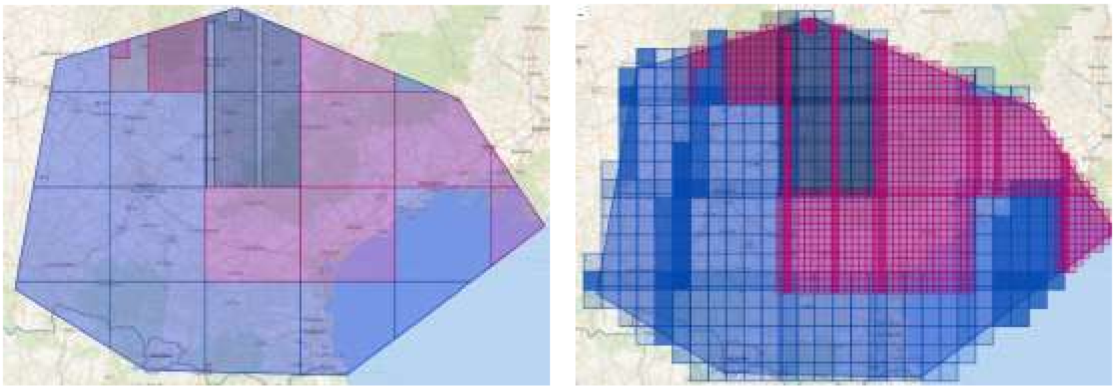


Figure 130: Occitania scenario Uc1c - After Redispatch #4

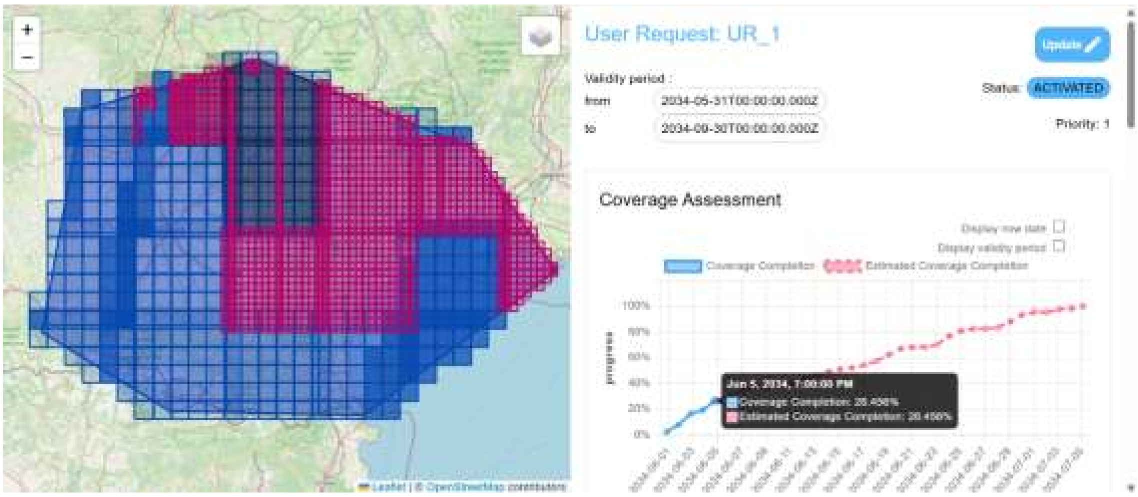


Figure 131 : Scenario Occitania Uc1c - After Plan #5

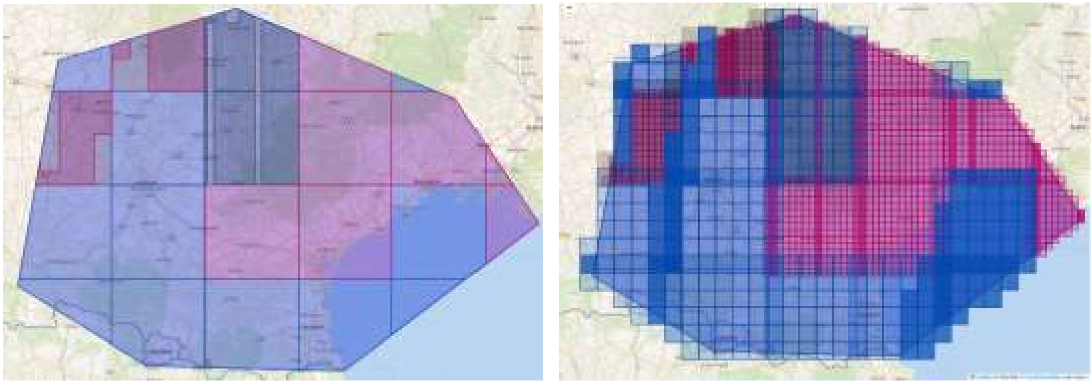


Figure 132 : Occitania scenario Uc1c - After Redispatch #5

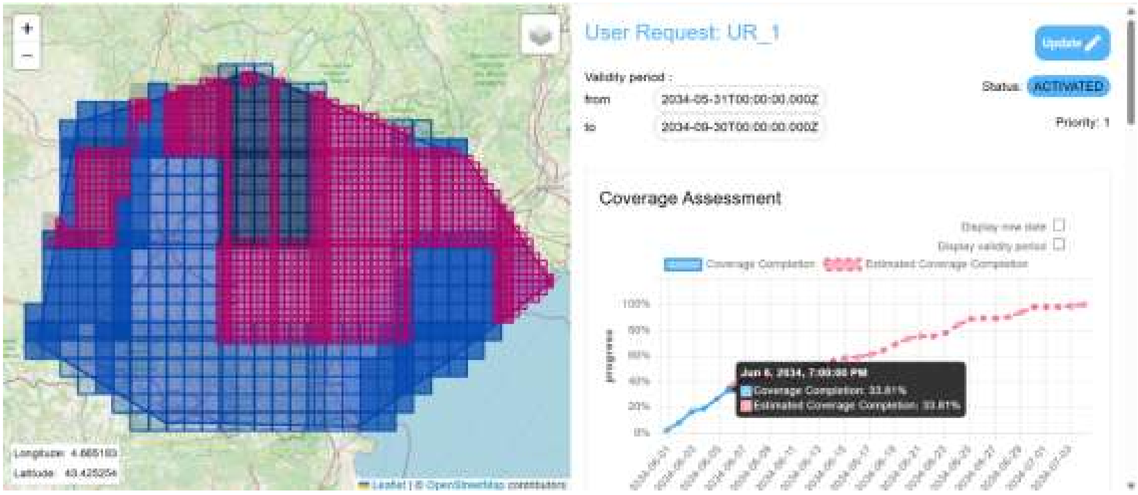


Figure 133 : Scenario Occitania Uc1c - After Plan #6

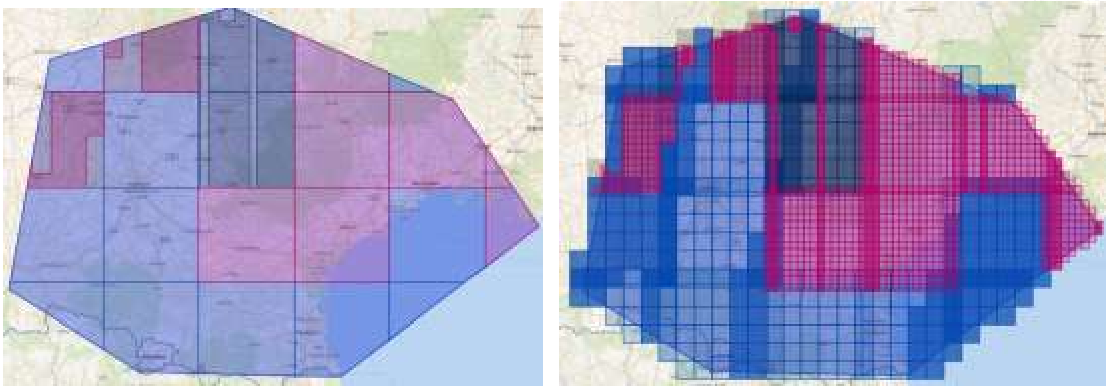


Figure 134 : Occitania scenario Uc1c - After Redispatch #6

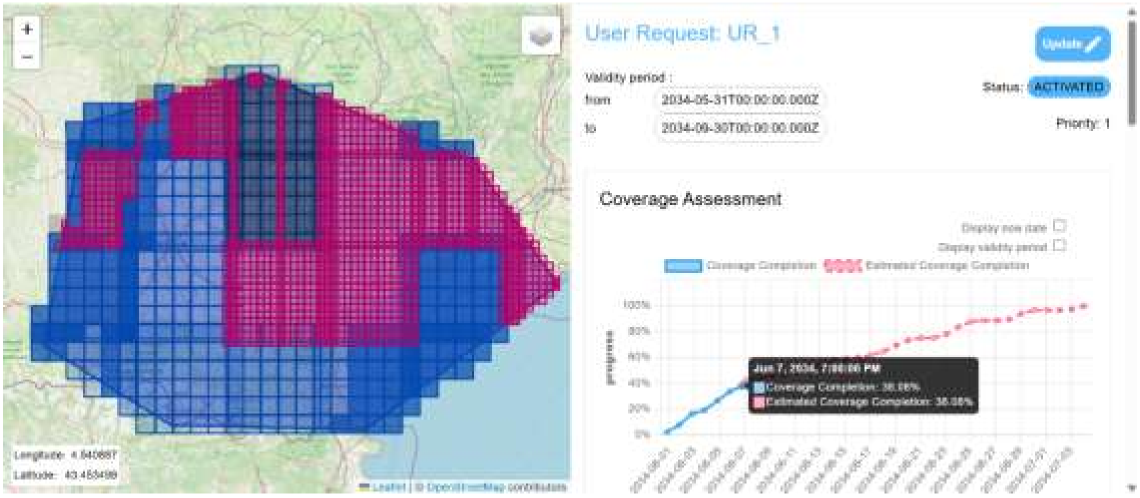


Figure 135 : Scenario Occitania Uc1c - After Plan #7

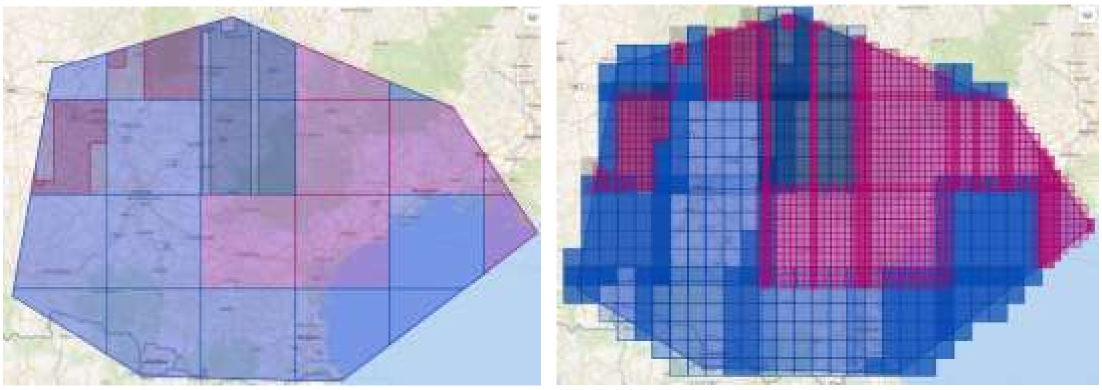


Figure 136 : Occitania scenario Uc1c - After Redispatch #7

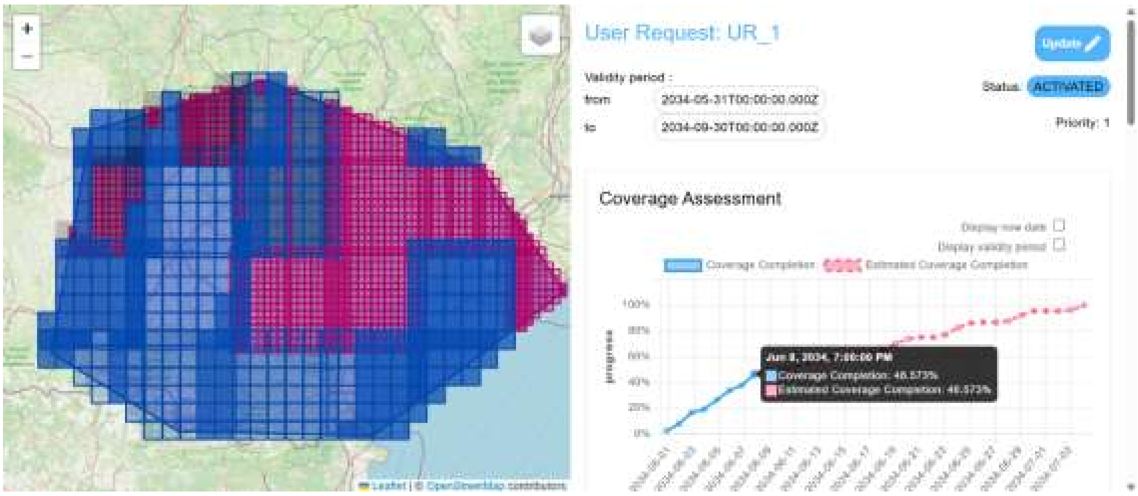


Figure 137 : Scenario Occitania Uc1c - After Plan #8

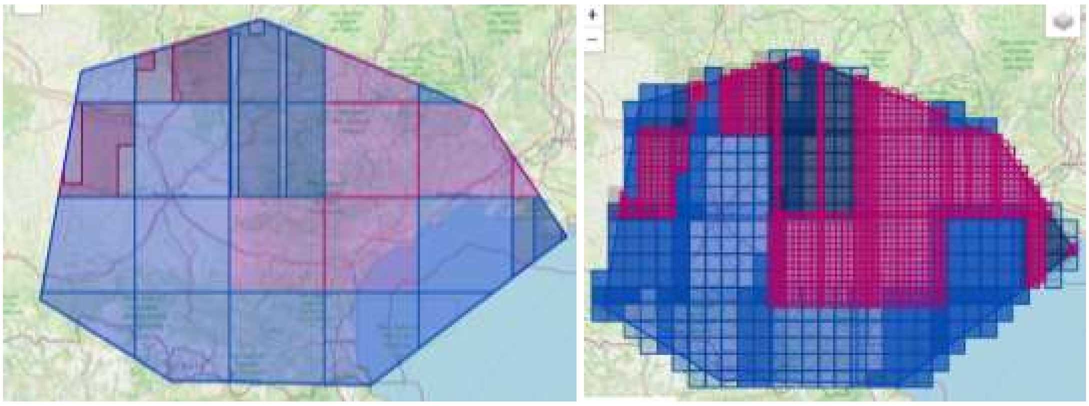


Figure 138 : Occitania scenario Uc1c - After Redispatch #8

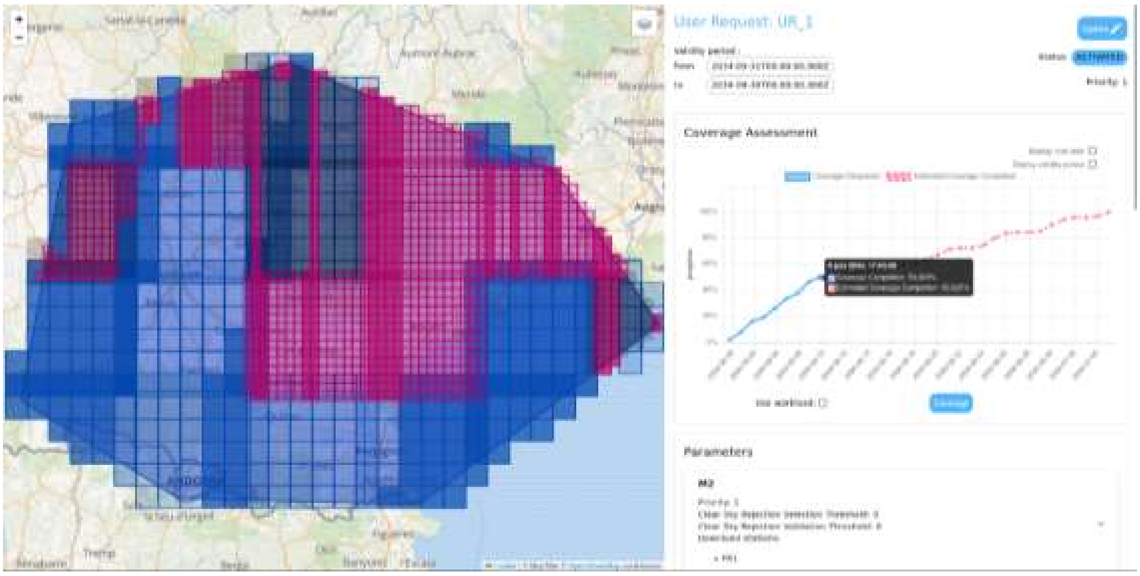


Figure 139 : Scenario Occitania Uc1c - After Plan #9

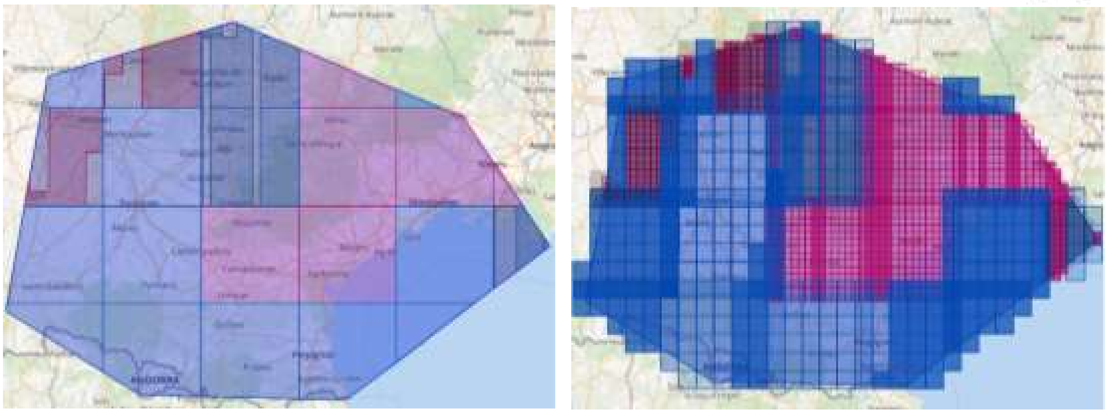


Figure 140 : Occitania scenario Uc1c - After Redispatch #9

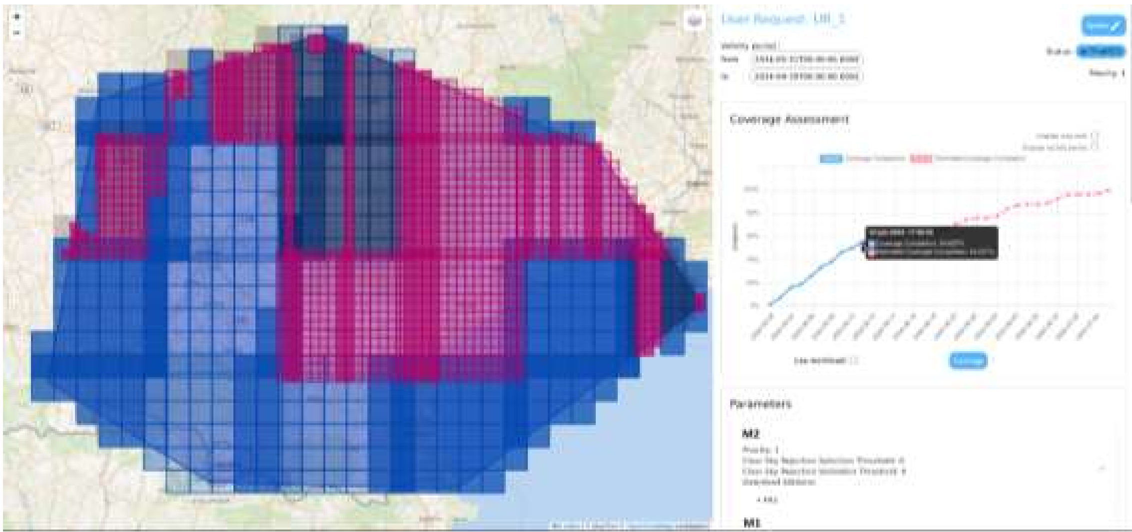


Figure 141 : Scenario Occitania Uc1c - After Plan #10

9.2.2 Use Case 2

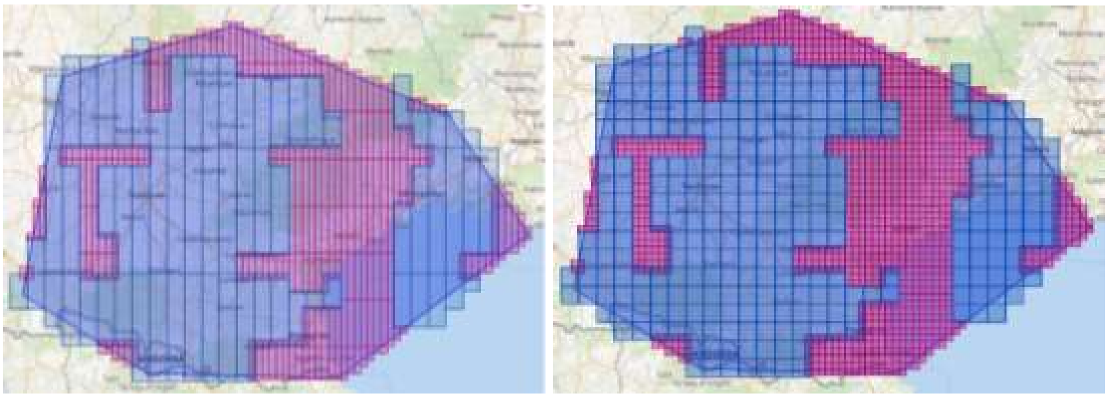


Figure 142 : Occitania scenario Uc2 - After Redispatch #1

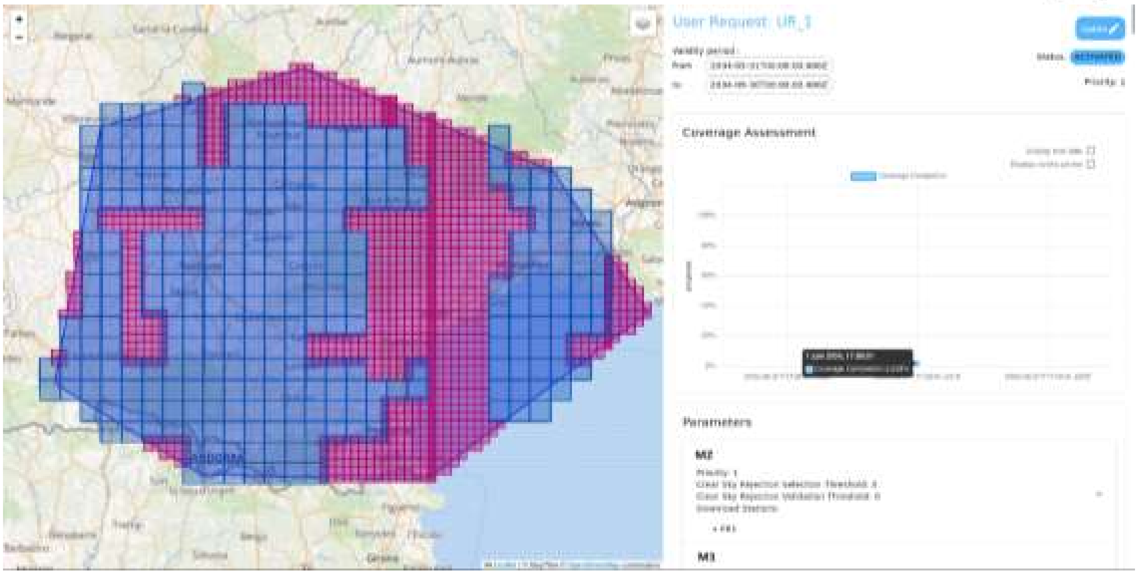


Figure 143 : Scenario Occitania Uc2 - After Plan #1

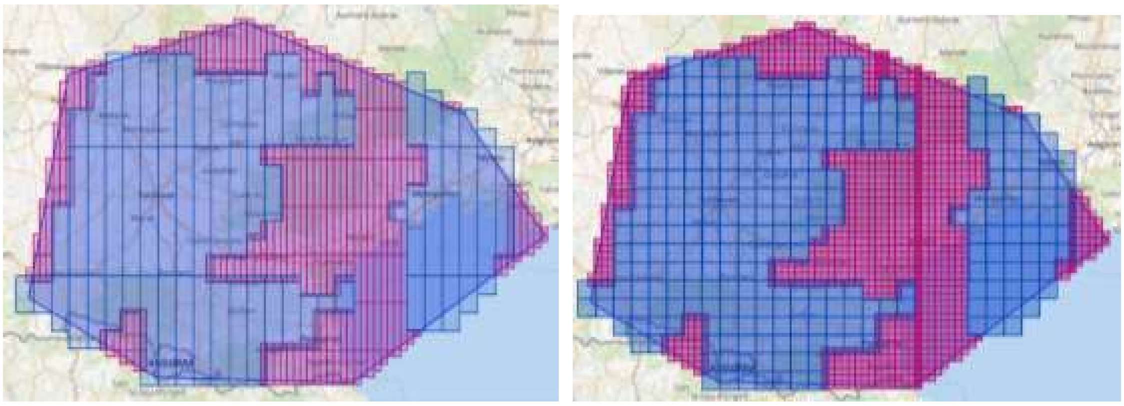


Figure 144 : Occitania scenario Uc2 - After Redispatch #2

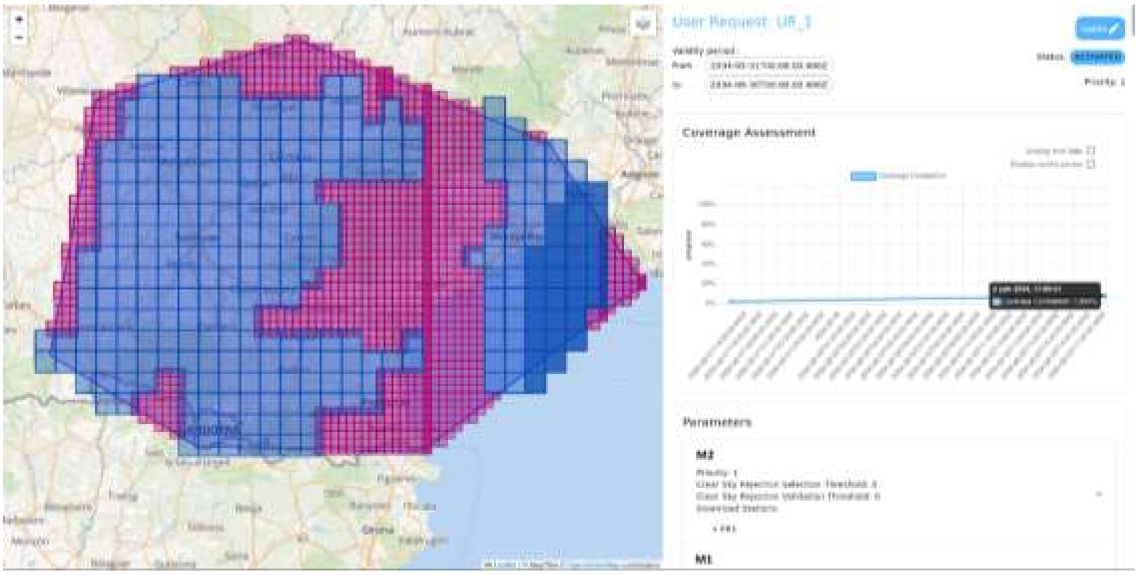


Figure 145 : Scenario Occitania Uc2 - After Plan #2

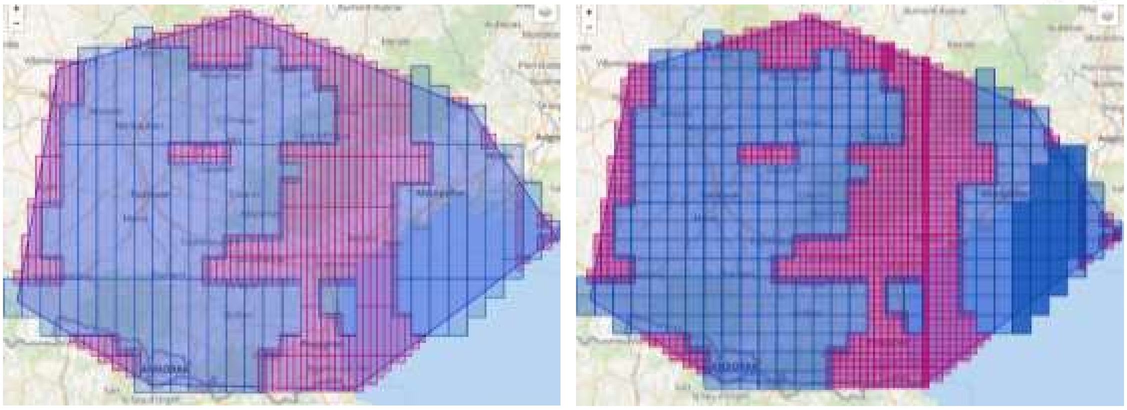


Figure 146 : Occitania scenario Uc2 - After Redispatch #3

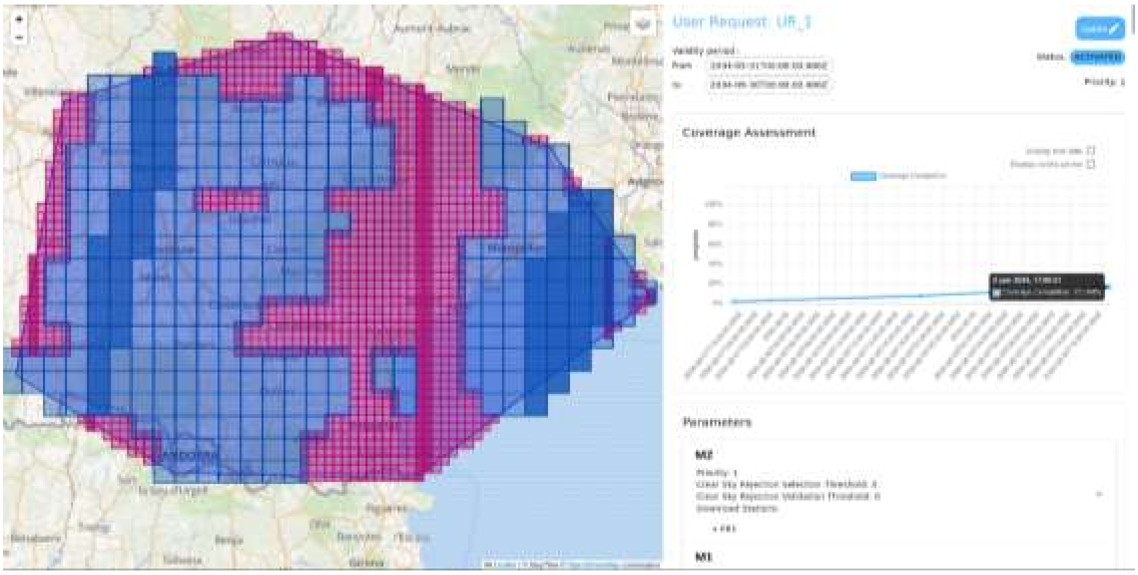


Figure 147 : Scenario Occitania Uc2 - After Plan #3

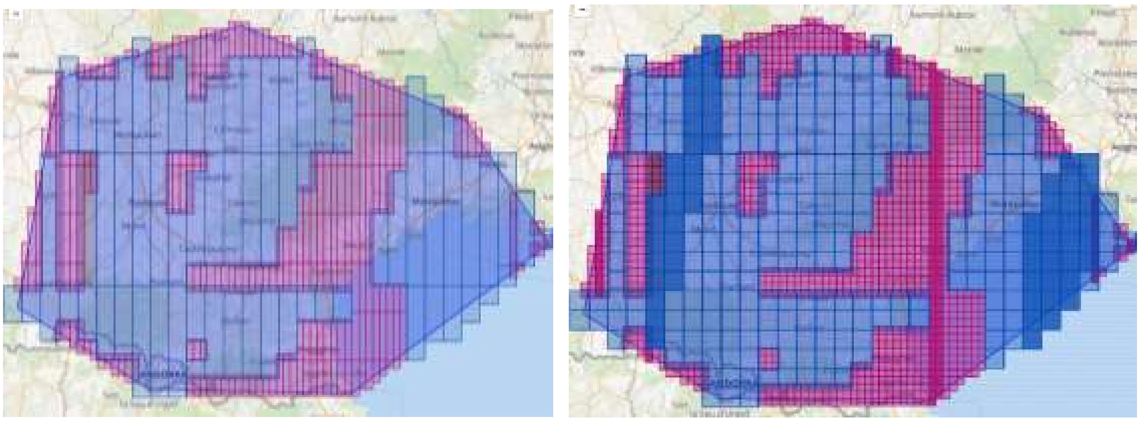


Figure 148 : Occitania scenario Uc2 - After Redispatch #4

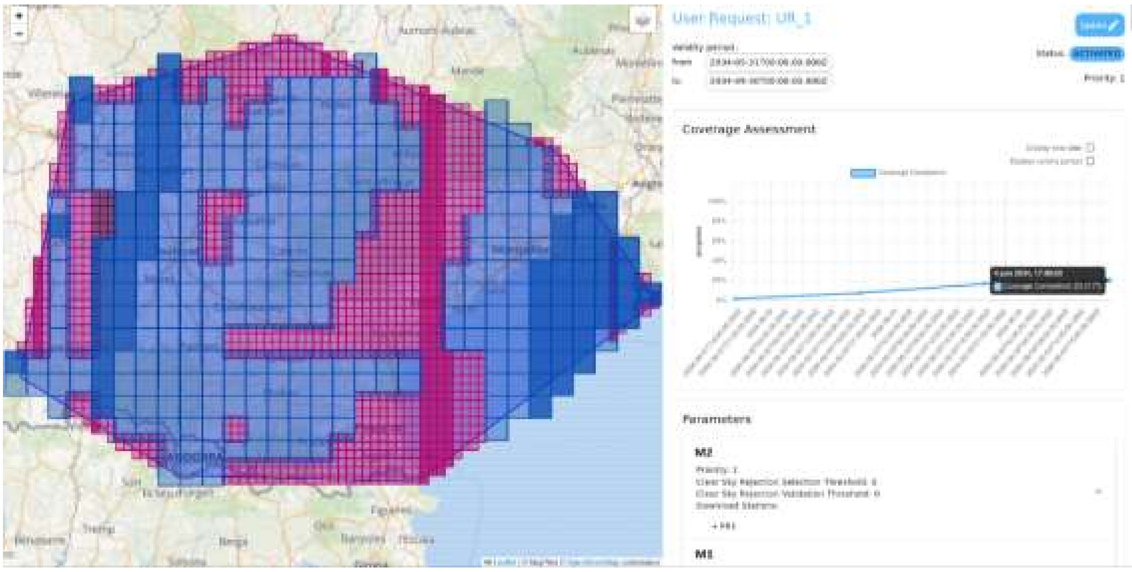


Figure 149 : Scenario Occitania Uc2 - After Plan #4

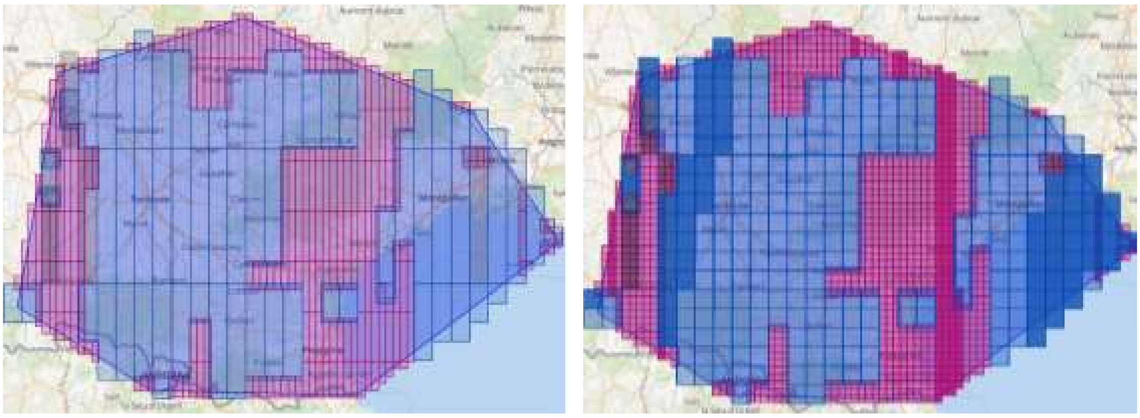


Figure 150 : Occitania scenario Uc2 - After Redispatch #5

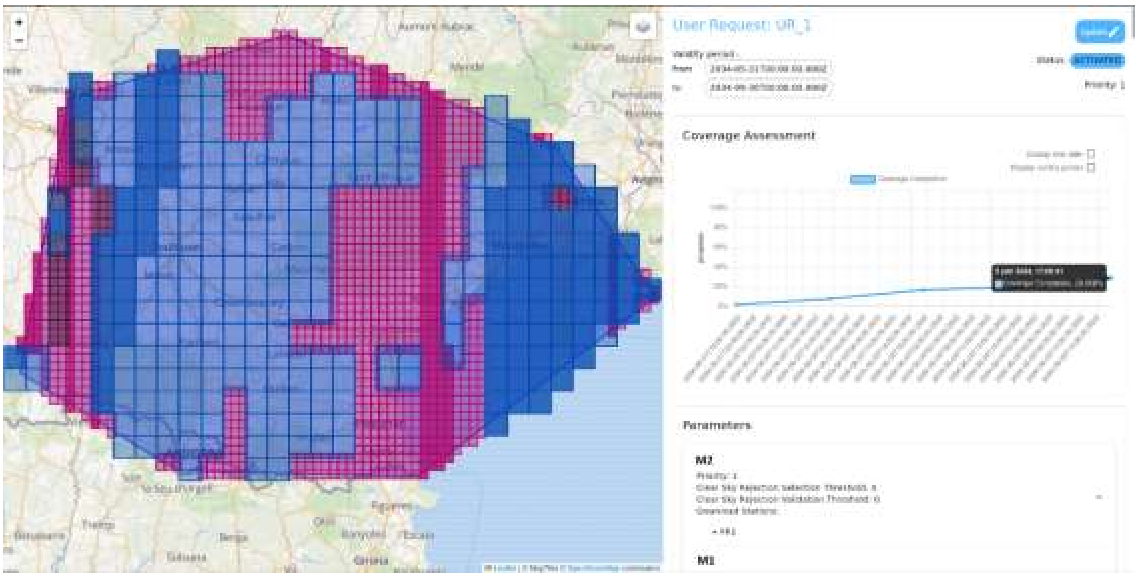


Figure 151 : Scenario Occitania Uc2 - After Plan #5

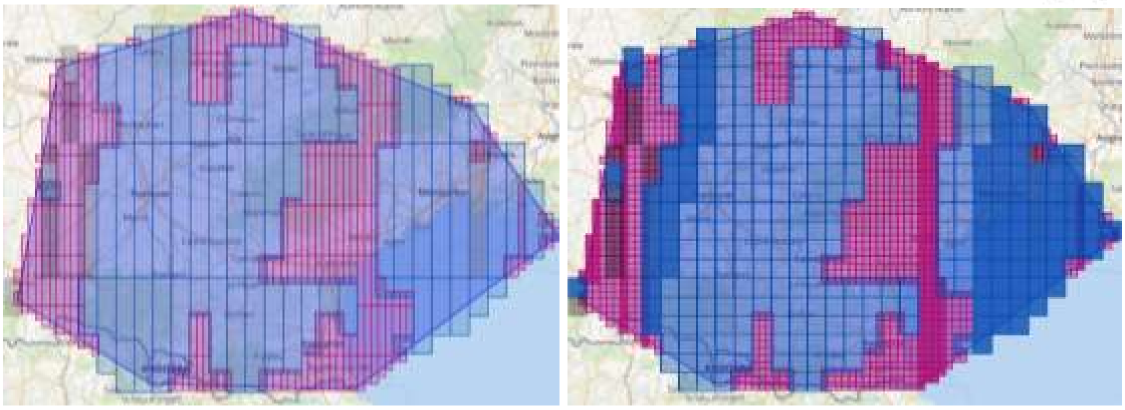


Figure 152 : Occitania scenario Uc2 - After Redispatch #6

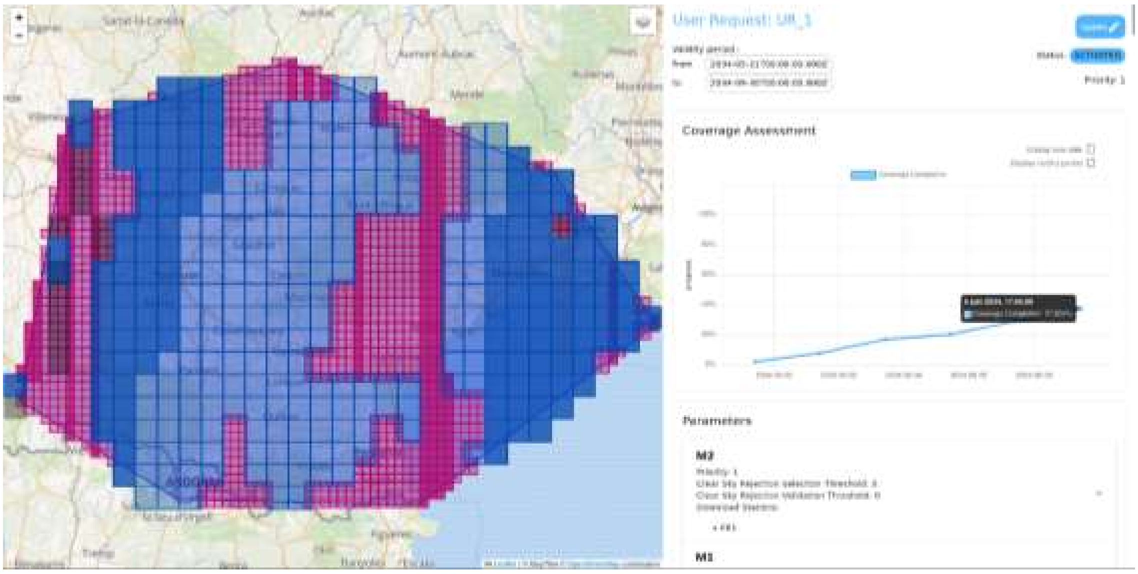


Figure 153 : Scenario Occitania Uc2 - After Plan #6

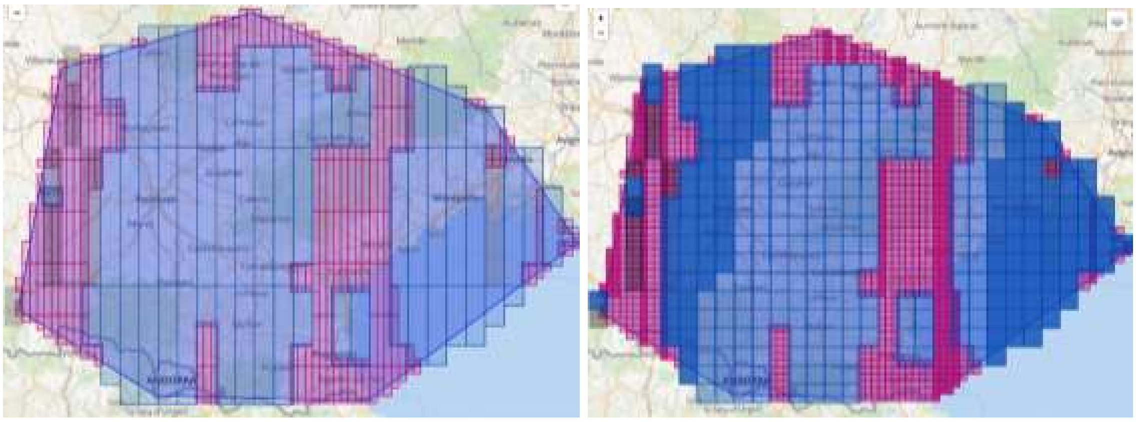


Figure 154 : Occitania scenario Uc2 - After Redispatch #7

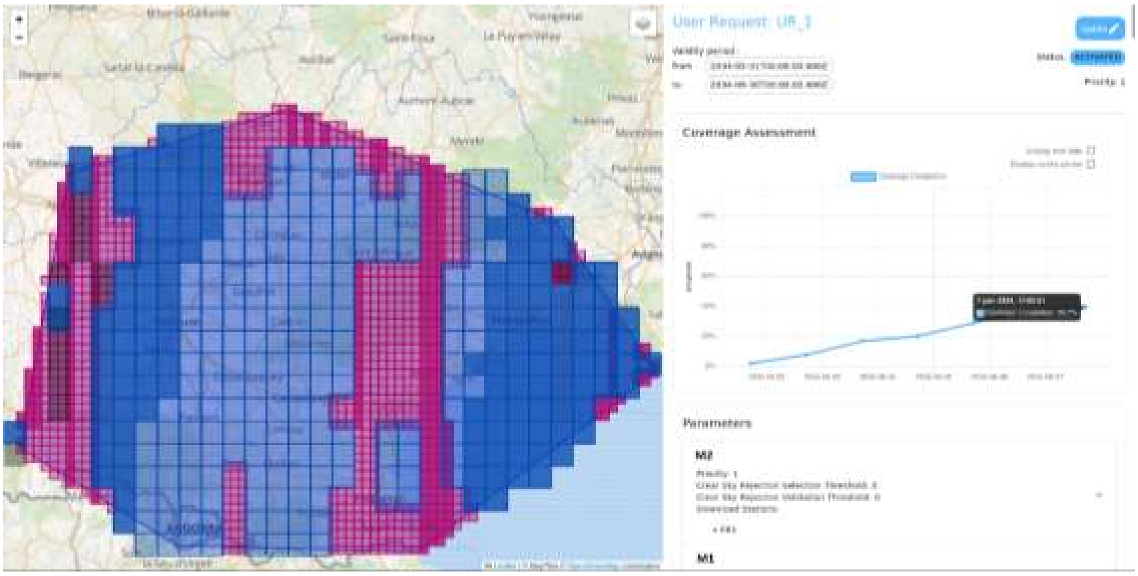


Figure 155 : Scenario Occitania Uc2 - After Plan #7

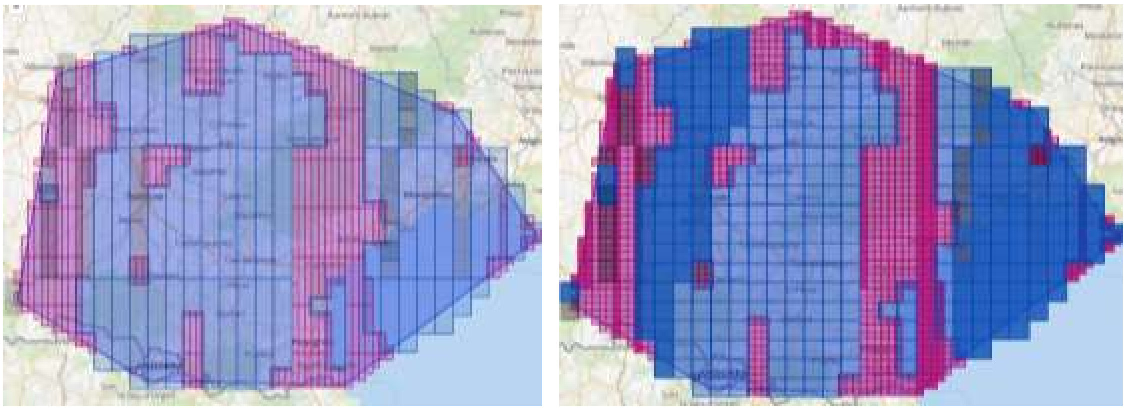


Figure 156 : Occitania scenario Uc2 - After Redispatch #8

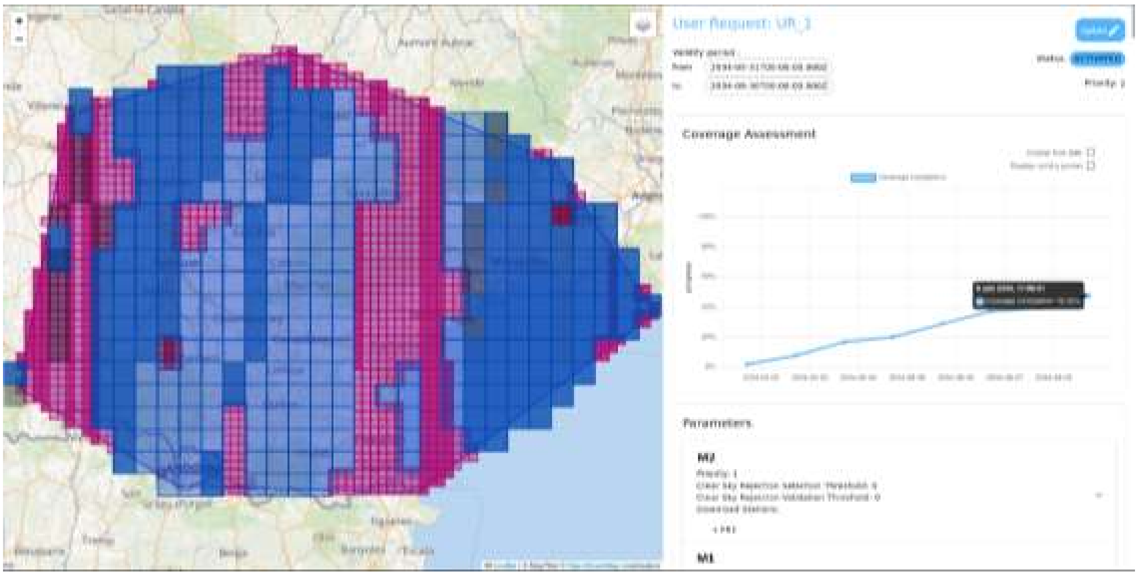


Figure 157 : Scenario Occitania Uc2 - After Plan #8

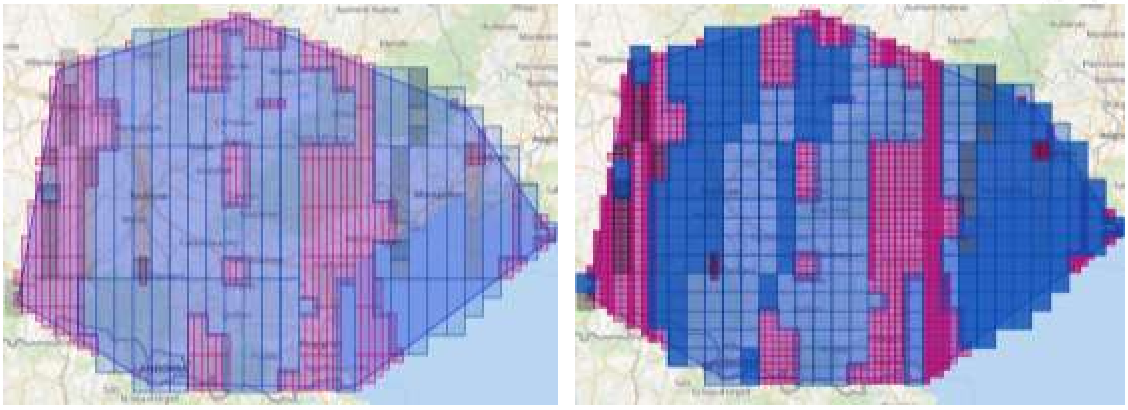


Figure 158 : Occitania scenario Uc2 - After Redispatch #9

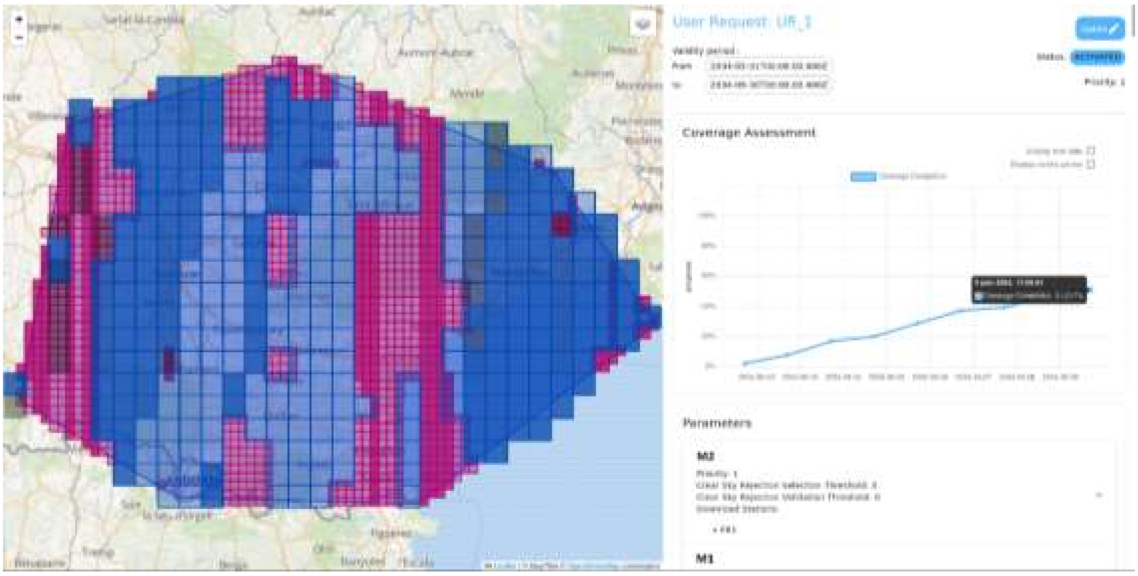


Figure 159 : Occitania scenario Uc2 - After Plan #9

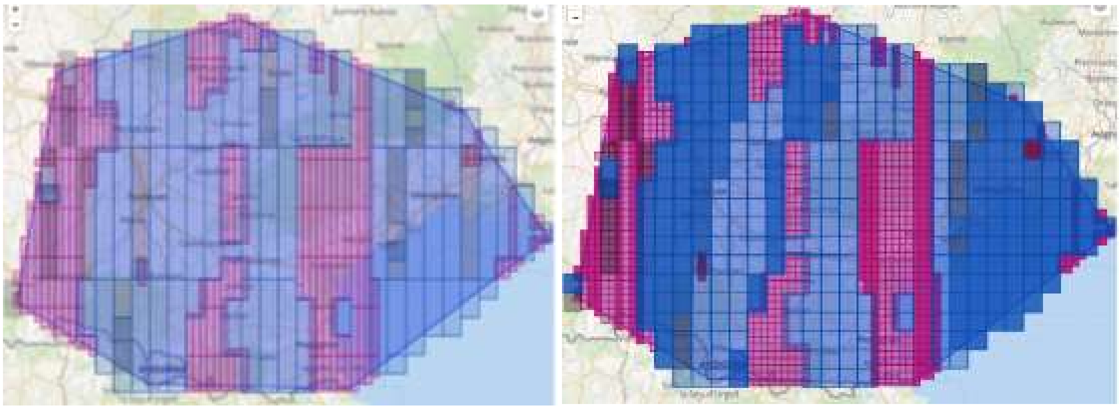


Figure 160 : Occitania scenario Uc2 - After Redispatch #10

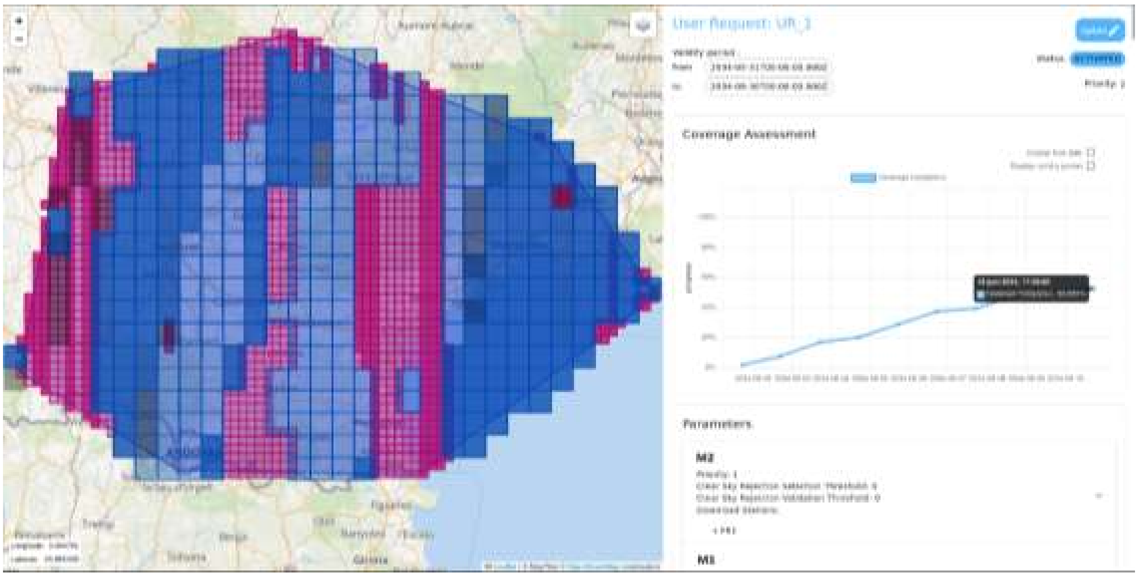


Figure 161 : Occitania scenario Uc2 - After Plan #10

9.3 Palma Scenario progress

9.3.1 Use Case 1c

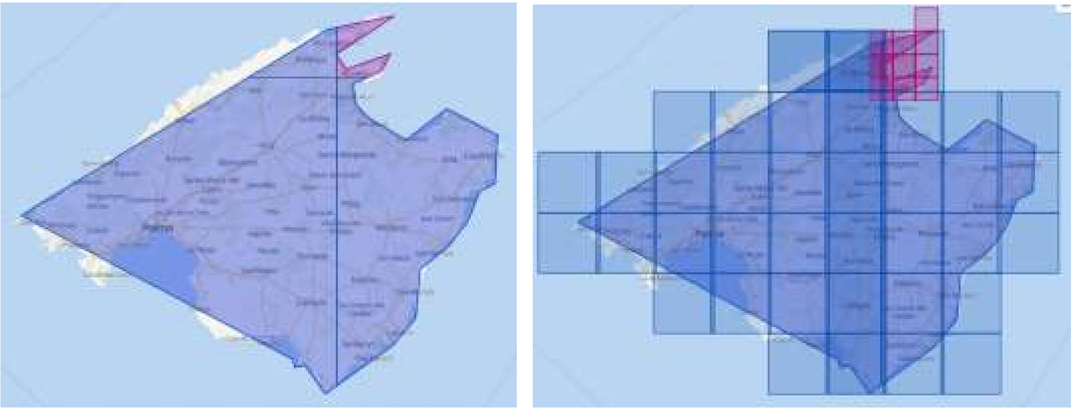


Figure 162 : Palma scenario Uc1c - After Initial Dispatch

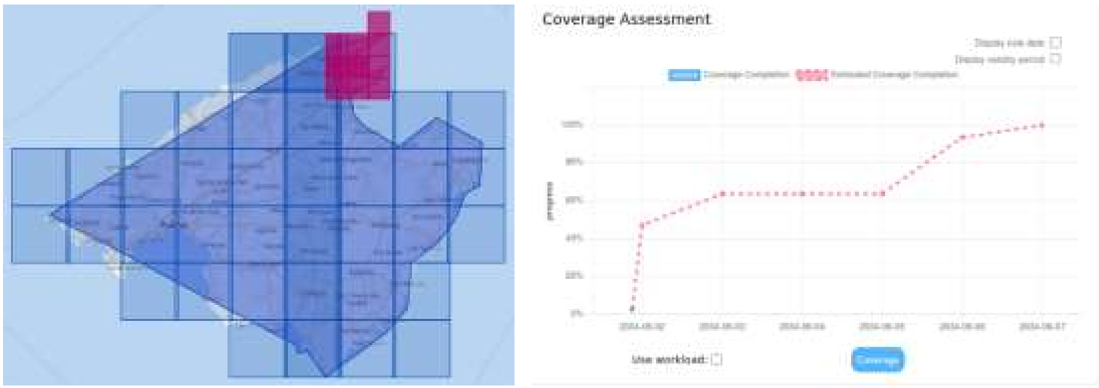


Figure 163 : Palma scenario Uc1c - After Plan #1

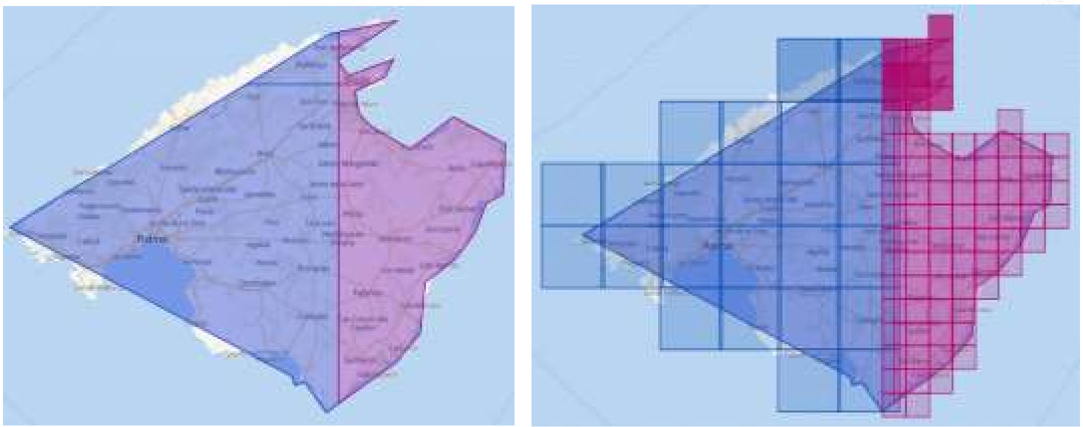


Figure 164 : Palma scenario Uc1c - After Redispatch #1

Figure 165 : Palma scenario Uc1c - After Plan #2

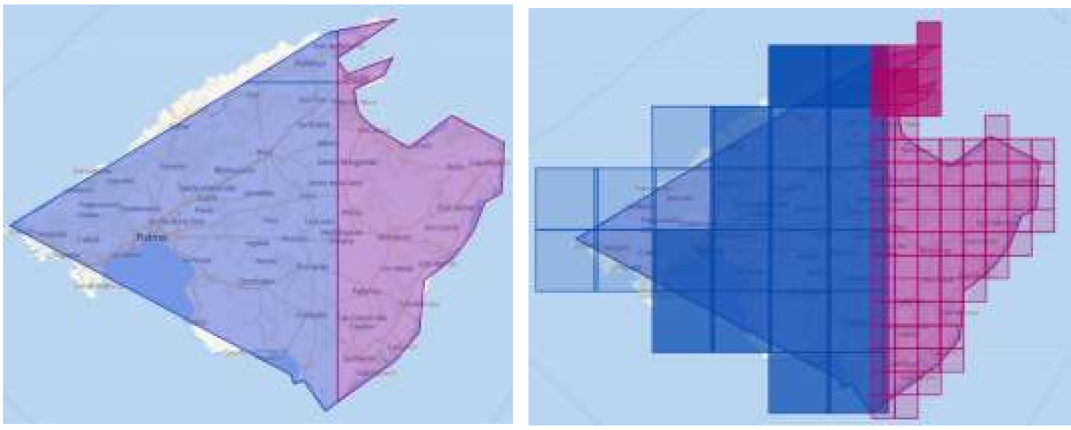


Figure 166 : Palma scenario Uc1c - After Redispatch #2

Figure 167 : Palma scenario Uc1c - After Plan #3

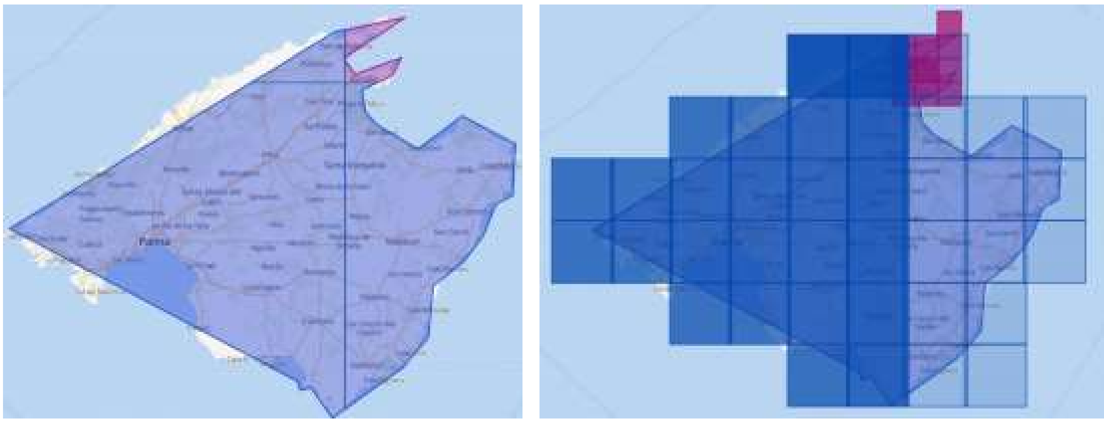


Figure 168 : Palma scenario Uc1c - After Redispatch #3

Figure 169: Palma scenario Uc1c - After Plan #4

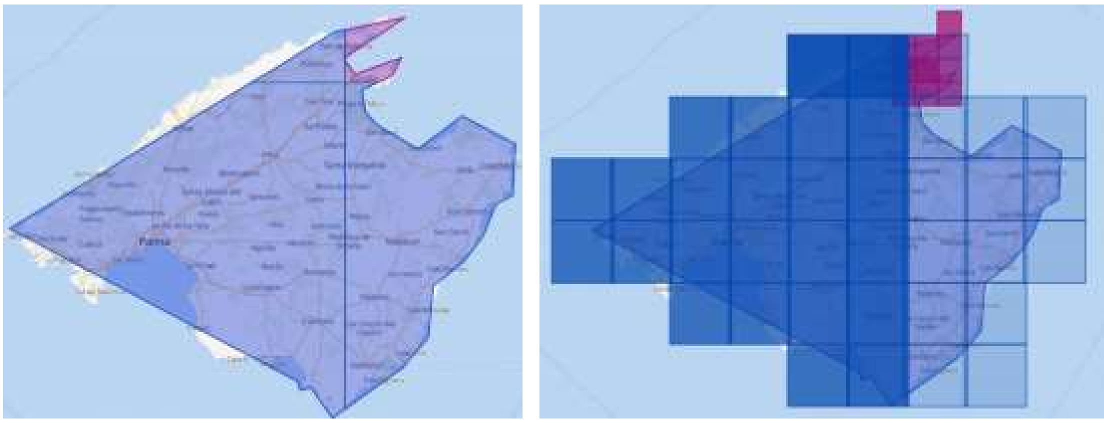


Figure 170 : Palma scenario Uc1c - After Redispatch #4

Figure 171 : Palma scenario Uc1c - After Plan #5

9.3.2 Use Case 2

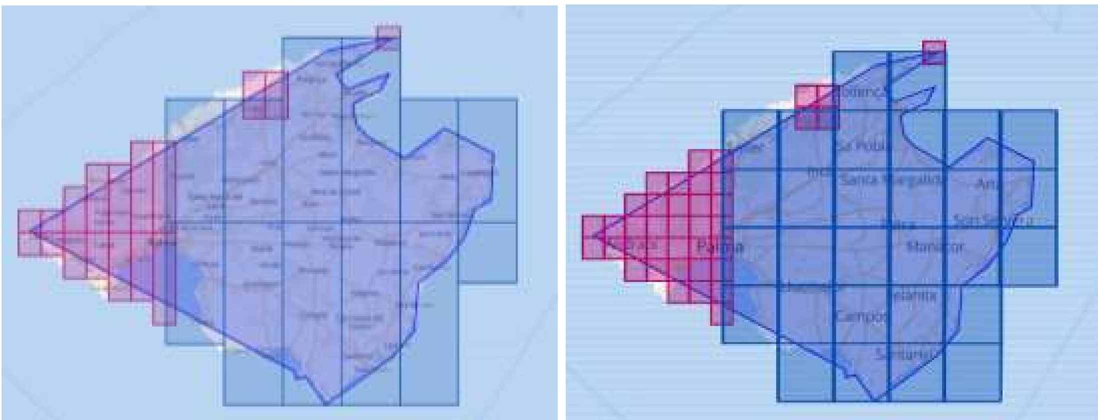


Figure 172 : Palma scenario Uc2 - After Redispatch #1

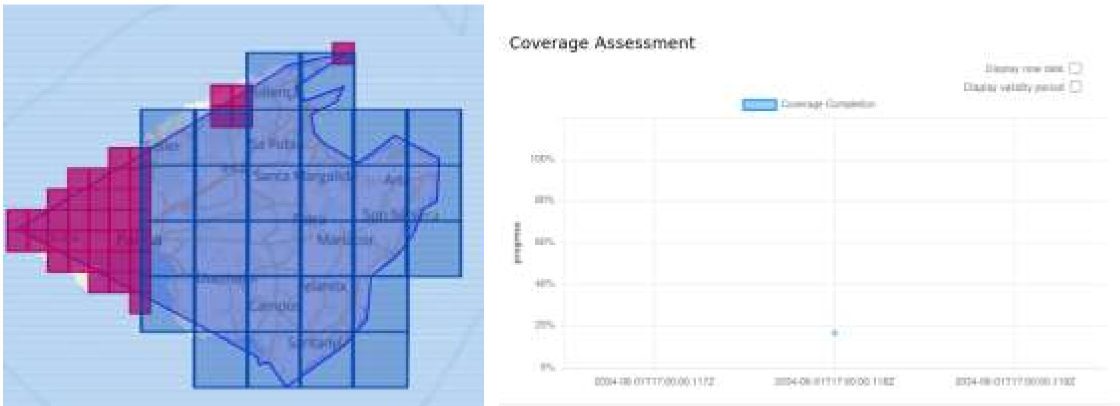


Figure 173 : Palma scenario Uc2 - After Plan #1

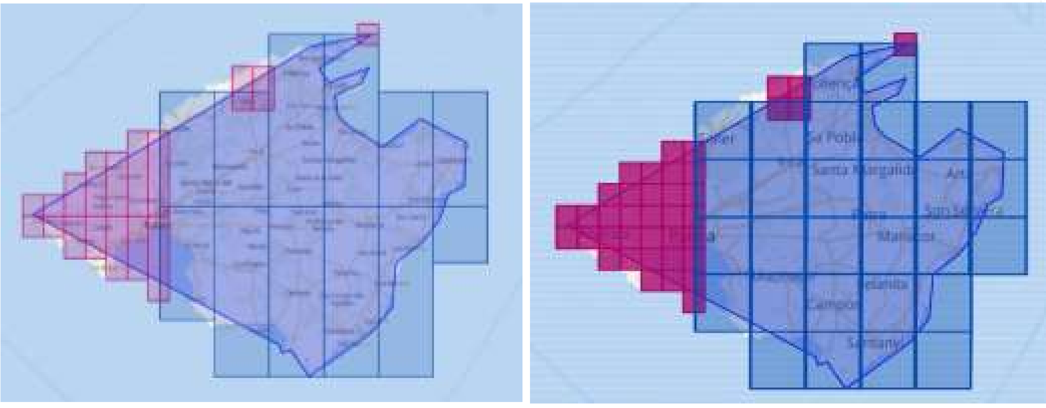


Figure 174 : Palma scenario Uc2 - After Redispatch #2

Figure 175 : Palma scenario Uc2 - After Plan #2

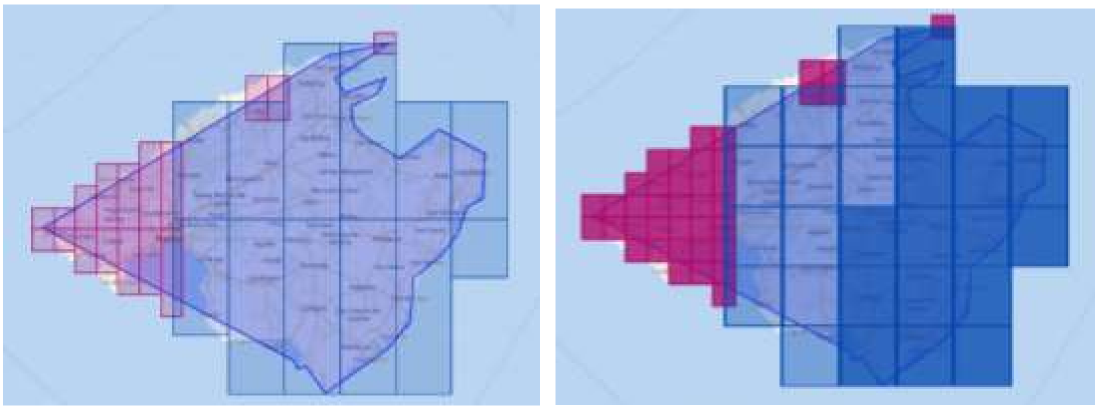


Figure 176 : Palma scenario Uc2 - After Redispatch #3

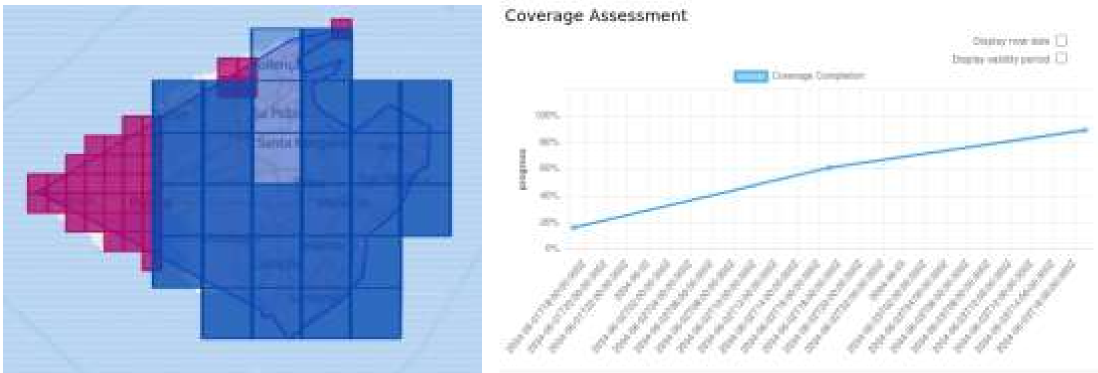


Figure 177 : Palma scenario Uc2 - After Plan #3

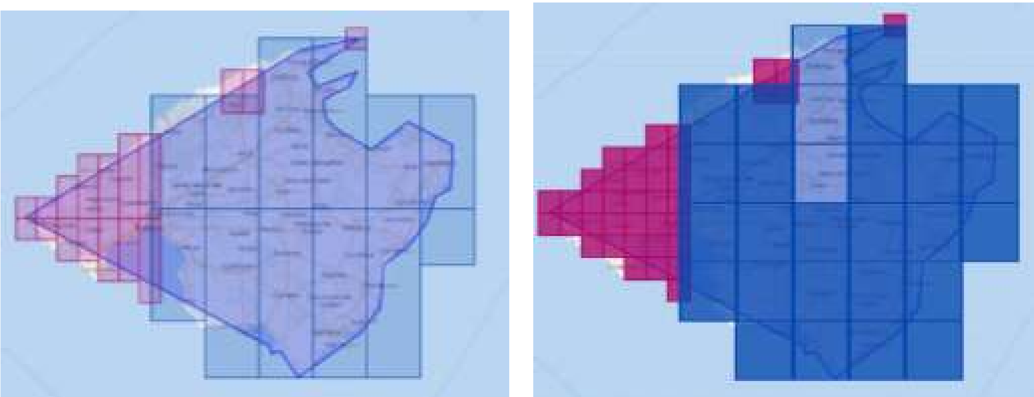


Figure 178 : Palma scenario Uc2 - After Redispatch #4

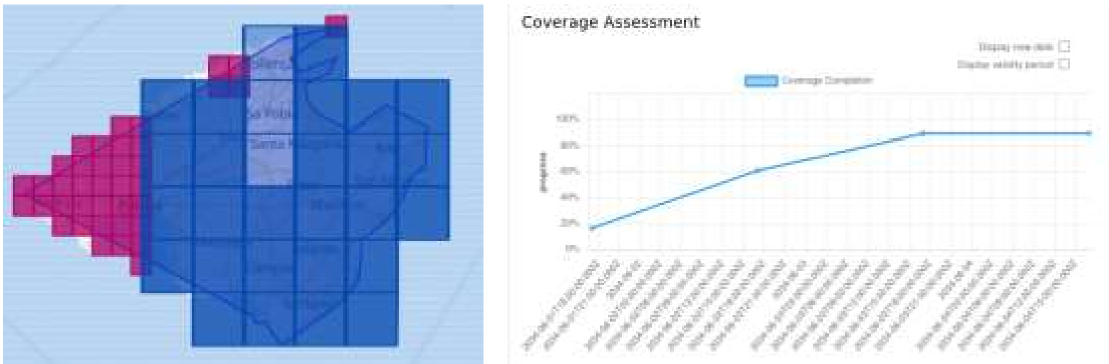


Figure 179 : Palma scenario Uc2 - After Plan #4

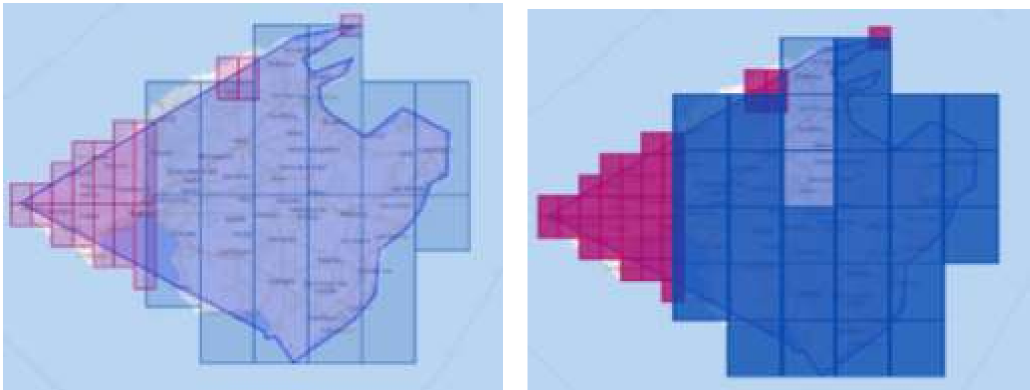


Figure 180 : Palma scenario Uc2 - After Redispatch #5

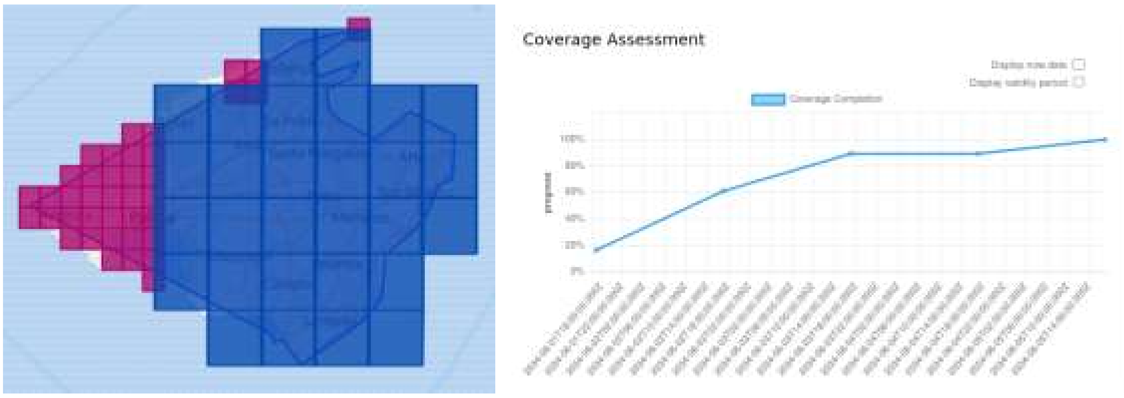


Figure 181 : Palma scenario Uc2 - After Plan #5