# PSI Case Study

## PSI-CST


| | |
|---|---|
| Version: | MS11 [1.3.0] |
| Date: | 2025-04-23 |
| Reference: | PSI-CST |
| Total Pages: | 18 |

# Table of Contents



# List of Figures



# List of Tables

# 1 Document Meta Information

## 1.1 Document Signature Table

| | Name | Function | Company |
|---|---|---|---|
| Author | Norbert Czeranka | PSI Project Team | CGI |
| Author | Divya Chauhan | PSI Project Team | SES |
| Approval | Victoria McCarthy | Project Manager | SES |
| Approval | Hendrik Oppenberg | Technical Officer | CGI |
| Checked | Pepijn Witte | Quality Assurance Manager | CGI |

Table 1.1: Signature Table.

### 1.1.1 Document Change Record

#### 1.1.1.1 Changes

| Date | Version | author | message |
|---|---|---|---|
| 2023-04-27 | MS4 | David Valcarcel | Initial version |
| 2023-07-26 | MS5 | Norbert Czeranka | Restructurings, extracted roadmaps and portions of SDP, extracted case study |
| 2023-10-05 | MS6 | Divya Chauhan | Added information stemming from SES internal case study, improved descriptions of broker and provider PSS |
| 2024-01-25 | MS7 | Hendrik Oppenberg | Elaborated on matchmaking |
| 2024-09-11 | MS8 [1.2.0] | Hendrik Oppenberg | Public release adjustments. |
| 2024-12-09 | MS9 [1.2.1] | Hendrik Oppenberg | No update, just version bump. |
| 2025-02-03 | MS10 [1.2.2] | Wolfgang Robben | No update, just version bump. |
| 2025-04-23 | MS11 [1.3.0] | Wolfgang Robben | Anonymized providers and PSS operators. |

Table 1.2: DCR Table.

#### 1.1.1.2 Source Control

Changes to this document are tracked electronically. No signature is required by the authors. The following information can prove the integrity of the document and reveal any change.

| Repo | Date | Author | Branch | Hash |
|---|---|---|---|---|

Table 1.3: GIT Changelog Table.

Figure 1.1: DCR QR-Code.

## 1.2 Documents

### 1.2.1 Applicable Documents

| Acronym | Reference | Title | Version |
|---|---|---|---|
| | | | |

Table 1.4: Applicable Documents.

### 1.2.2 Reference Documents

| Acronym | Reference | Title | Version |
|---|---|---|---|
| PSI-DL | PSI-DL | PSI CGI Document List | current MS (doc version) |
| PSI-MADR | PSI-MADR | PSI Markdown Administrative Decision Records | see before |
| PSI-TAD | PSI-TAD | PSI Terms, Abbreviations and Definitions | see before |
| PSI-TOD | PSI-TOD | PSI Tasks and Operations Dictionary | see before |

Table 1.5: Reference Documents.

# 2 Introduction

The Pooling & Sharing Interfaces Definitions (PSID) project is an ESA co-funded effort to define a common standard for the interfaces of Pooling & Sharing Systems (PSS) for Satellite Communication (SatCom) services. A PSS is a digital platform for matchmaking (Gov)SatCom users' demands (both commercial and institutional) with (Gov)SatCom providers' offers. Bringing together multiple (Gov)SatCom providers in one platform makes the market transparent, thus allowing users to get an overview of the market and to compare different offers efficiently. Additionally, a PSS assists users with little knowledge about the (Gov)SatCom domain defining their requirements on the (Gov)SatCom services. Those two aspects combined allow for fast access to the services and an efficient usage of the available capacities. To accomplish this, a PSS steps in between the usual processes of finding a provider/supplier, requesting an offer, and ordering the desired products or services, either as a service broker or by pooling products and services from different providers and offering them as an intermediary or distributor. Subsequently, the PSS can be used to monitor the services and manage multiple missions in a single application.

Eventually, a PSS can also be used as (or manage) a community hub, i.e., a number of end users or customers with similar interest that *share* their common resources and utilize a commonly obtained *pool* of (Gov)SatCom capacities. This strategy increases the efficient usage of scarce resources further.

There are already different approaches on PSSs, that might lead to an unnecessary fragmentation of the market. Therefore, a common standard for the interfaces of a PSS is required to allow the interaction between those different PSSs and reduce the effort of (Gov)SatCom providers to offer their product and services via multiple PSSs to maximize their reach.

Such a standard needs to take care of the different interfaces involved in the aforementioned processes, i.e.,

The goal of this project is to mainly define aspect 1 and to develop a software mock-up as needed to validate the various interfaces being developed.

The PSI standard derives from the existing industry-standard "Open Digital Framework" of **TM Forum** alliance[1]. The "Open Digital Framework" is a reference framework for delivering online Information, Communications and Entertainment services to the telecom world. It empowers market participants to compete and cooperate. One of PSI's goals is to make this existing standard fit for the world of satellite communication.

The consortium for this project consists of the service & technology providers SES Techcom and CGI, as well as of the (Gov)SatCom operators SES, Hellas Sat, Hispasat, Hisdesat, and LuxGovSat, and Inmarsat being both a service & technology provider and a (Gov)SatCom operator.

[1] See https://www.tmforum.org/resources/reference/gb991-tm-forums-core-concepts-and-principles-v22-0-0/

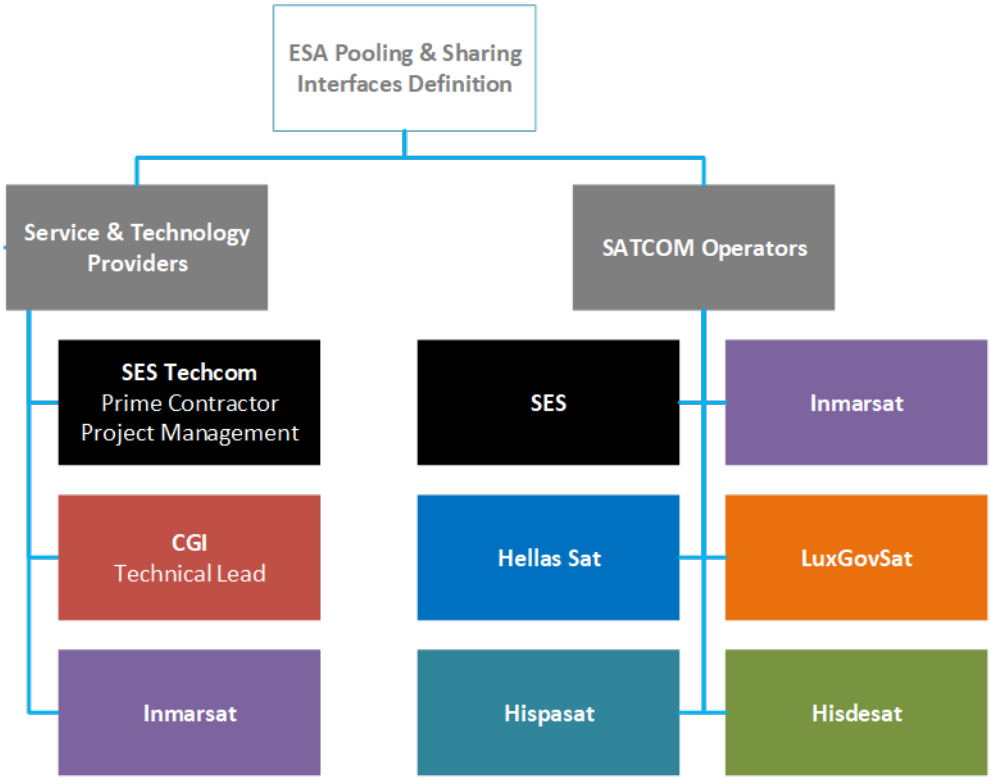


Figure 2.1: The PSI consortium.

## 2.1 Document Scope

This document describes a case study for the Pooling And Sharing Interfaces Standardisation Project (PSI). In this case study, we examine an organization (referred to as the “governance”) that faces a critical problem within the context of (Gov)SatCom (Government Satellite Communications) products. The issue at hand revolves around consumers of (Gov)SatCom products not receiving timely access, experiencing a lack of assurance regarding desired product attributes (e.g., security), and incurring suboptimal costs. The following sections heavily refer to terms, abbreviations and definitions defined in the [PSI-TAD].

### 2.1.1 Compiled Document

**NOTE**: THIS IS A COMPILED DOCUMENT [2]

This document has been compiled/generated from external sources and is not being written as-is. Therefore, any changes made within this compiled version of the document will be lost upon recompilation!

To make (permanent) changes, edit the respective sources directly or contact the PSID team.

### 2.1.2 Signature

Changes to this document are tracked electronically. No signature is required by the authors. The information in the “Source Control” chapter can prove the integrity of the document and reveal any change.

[2] Document compiled on 2025-04-23 12:37.

## 2.1.3 Release Notes

[[*TOC*]]

---

## 2.1.4 PSI Release Notes

### 2.1.4.1 Introduction

Welcome to the latest release of the Pooling and Sharing Interface (PSI) API!
This document outlines the new features, improvements, and important updates included in this version.

### 2.1.4.2 Key Highlights

The central focus of this release is the implementation of the **Mission Management ODA Blueprint**.
This component complements the mission-related APIs by providing a *reference implementation of graphical user interfaces* that help users specify their product and service requirements.

Designed with users in mind, this component uses templates to simplify mission creation and introduces a governance layer to facilitate and control the requirements collection process.
It's built as a standalone micro-frontend and can be easily integrated into existing OSS/BSS/PSS systems.

The interface includes multiple visualization modes:

Another major update in this release is the migration to **TM Forum APIs Version 5 (TMF5)**.
All APIs have been ported to the current TMF baseline.
However, TMF5 introduced some gaps in the Component Test Kit (CTK), resulting in partial test coverage for certain APIs. This limitation will be addressed once TM Forum updates the CTK.

Additionally, this release introduces **MEF-compatible APIs**, marking the beginning of convergence between MEF and TMF frameworks within PSI.
Our goal moving forward is to support both API standards in their respective areas.

### 2.1.4.3 What's New

#### 2.1.4.3.1 Newly Added APIs

#### 2.1.4.3.2 Updated APIs

#### 2.1.4.3.3 Added Requirements

### 2.1.4.4 Known Limitations

### 2.1.4.5 Feedback and Contributions

We appreciate your input!
If you experience any issues or have suggestions, please don't hesitate to contact us.
We also encourage community contributions to help enhance PSI further.

# 3 Case Study

## 3.1 Problem Identification

The governance has analyzed the situation and identified the root causes of these challenges:

Demand-side Fragmentation: A significant portion of the problem arises from the fragmentation on the demand side. This fragmentation is driven by the proliferation of small contracts initiated by a multitude of isolated customers. Interestingly, these customers essentially require the same (Gov)SatCom product. This situation results in inefficient resource allocation and product duplication.

Supply-side Fragmentation: The governance has also recognized fragmentation on the supply side. This issue emerges when (Gov)SatCom products with surplus capacity cannot be effectively shared among customers. Consequently, the inability to share these products hampers access and leads to underutilization of available resources.

## 3.2 Proposed Solution

The proposed solution entails the implementation of a Pooling and Sharing System (PSS). Under this system, (Gov)SatCom products will be consolidated into a shared pool, which will, in turn, enable the effective sharing of products among customers through product allocation and de-allocation procedures.

In order to facilitate a clear understanding of the proposed solution and its implementation, the governance has defined the following concepts needed for the Pooling and Sharing System (PSS):

The Governance has identified the following six key functions that it needs to perform as part of its solution:

The governance will perform some of these functions with the help of the broker software. The governance has decided that the main element of the broker software is a PSS, and that the PSS is the main enabler for automation and efficient execution of the functions. The governance has also decided to use the PSID APIs as part of their PSS, and as a fundamental tool on which to build the integration upon of all elements of the overall solution.

The following figure provides a visualization of some of the above functions: (Please refer the chapter “The role of the PSID APIs” of this document to understand the `PSIDxx` numbering.)

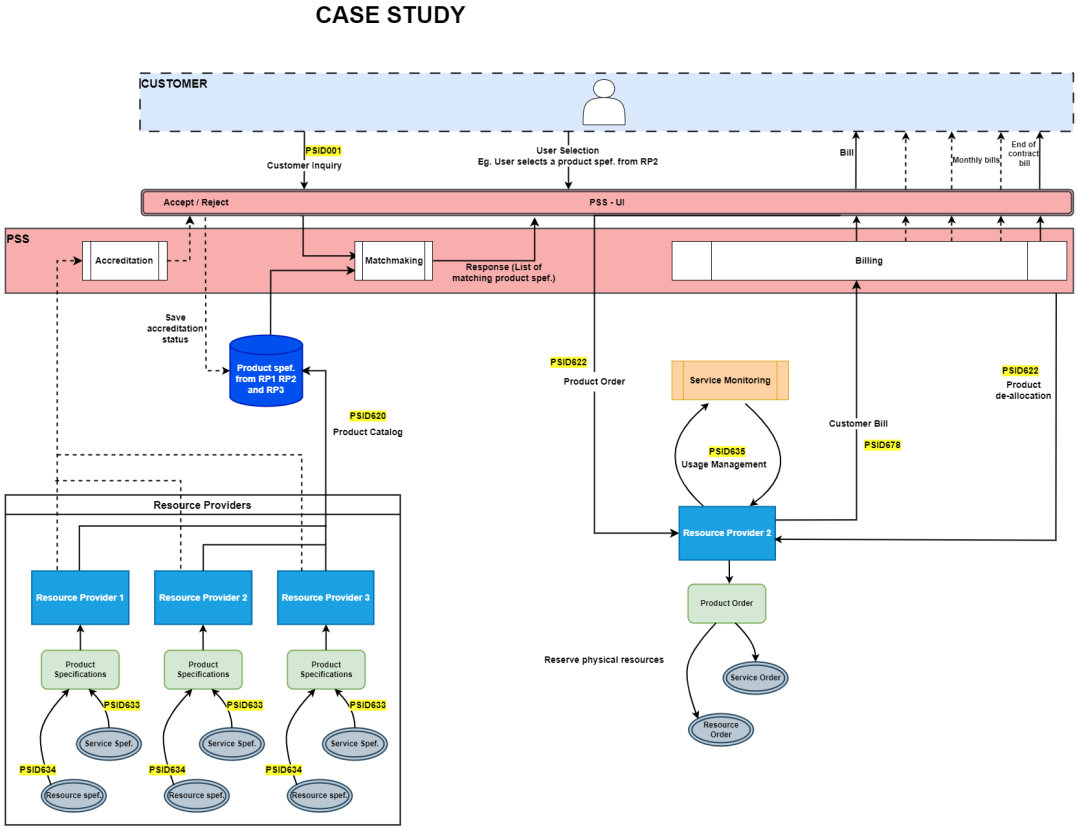


Figure 3.1: Case Study Diagram.

As can be seen in the figure above, the PSID APIs is the tool which enables machine-to-machine communication between Resource Provider Software and the PSS, which is the main element of the Broker Software.

The above figure illustrates the following steps:

These streamlined steps provide a clear overview of the process, emphasizing the key actions and interactions between the PSS, Resource Providers, and Customers.

## 3.3 PSS Architecture and Concept

A PSS aims to unify and centralise the (Gov)SatCom market and all associated interfaces. It consists of three main contexts:

Each context comprises a set of related subsystems. The *Frontend* contains the main interface to any actor of the PSS. The subsystems in the *Mission Context* target the main Use Cases of the customers and their service-needs. The *Business Context* refers to subsystems providing complimentary support to the main mission creation process, such as Customer Management.

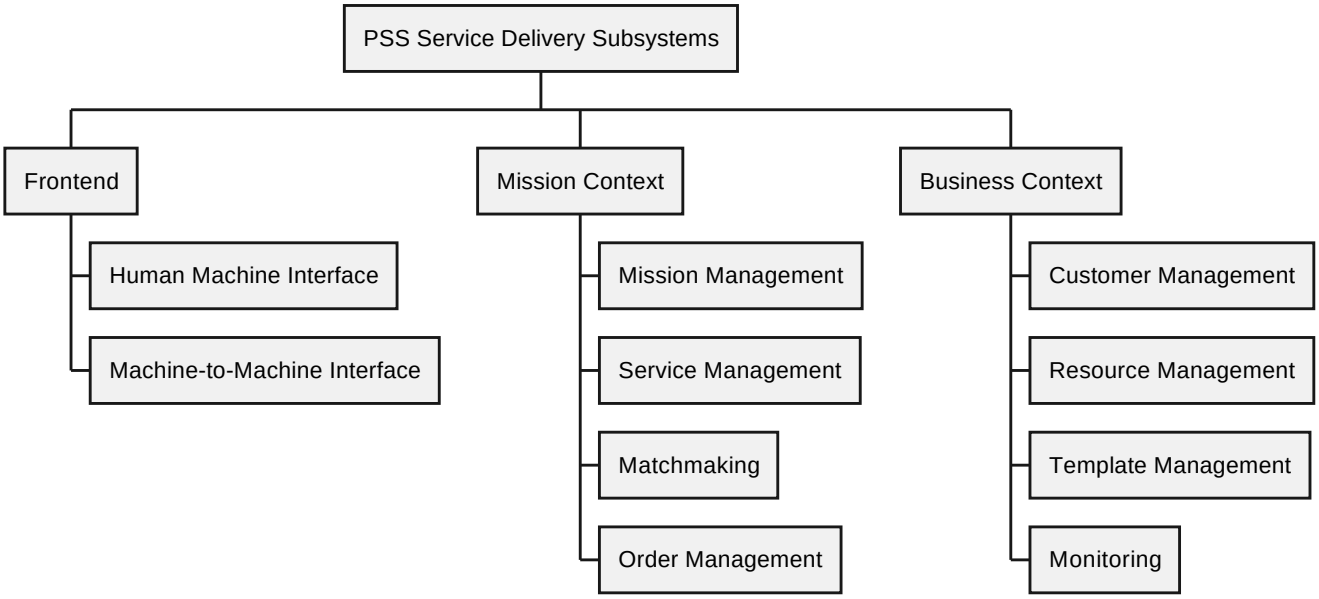


Figure 3.2: Service Delivery Subsystems.

## 3.4 Resource Provider Concept

The Resource Provider PSS system should essentially have the following modules:

## 3.5 The role of the PSID APIs

The desired outcome is that each provider who is compatible with this standard can connect to each PSS.

The PSS APIs can be organized in groups:

## 3.6 Description of Key Business Processes

This section provides a description of the key business processes.

### 3.6.1 Product Accreditation

In a scenario in which the broker platform has been completed (so all software, including the broker PSS and any others, and infrastructure is ready, all people are trained, all processes defined etc.), the next steps would be to certify an initial set of Resource Providers. For example, A-Sat and B-Sat might have expressed interest in participating in the broker program and have enhanced their software landscape to implement all relevant PSID APIs. In this case, once A-Sat and B-Sat are ready, a certification campaign would be performed involving the governance, A-Sat, and B-Sat. The campaign will aim to certify one or more A-Sat / B-Sat products as GOVSATCOM-compliant products (which would be somehow formally defined). As part of the campaign, the software components implementing PSID APIs of both Resource Providers would be certified via semi-automated testing campaigns. One of the outputs of the PSID initiative is a test suite for all PSID APIs. This test suite could be used and/or adapted for such certification

purposes. In any case, at the end of the certification campaign, the products, infrastructure, software, people, and processes of both Resource Providers would be certified and approved to participate in the broker platform.

### 3.6.2 Product Procurement

Once certification is completed, the governance will decide, perhaps based on some forecasting of potential demand, what products to procure from what Resource Providers. This will involve processes such as pricing negotiation, contracts, and others.

In the scope of this Case Study, the example will be for the Resource Provider system to push to the broker PSS system (PSS) the details of the procured products via PSID APIS, in the form of the PSID data model (Physical Resource Specifications, Logical Resource Specifications, Service Specifications, Product Specifications, Product Offerings, etc.). This will be a manual step.

### 3.6.3 Supply Aggregation

While the previous two business processes did not rely on the broker PSS nor on PSID APIs and were performed out of the scope of the broker software, once the products are procured, the broker and Resource Provider software can now leverage the PSID APIs to introduce automation into the process.

For it to be available in the broker platform, the A-Sat Product needs to be defined as per the PSID terms. Please refer to section “Specifications, Catalogs and Offerings” of the PSID “TAD (Terms, Abbreviations and Definitions)” document for a description of what PSID terms are used to define a given product.

An example definition of the A-Sat product is as follows - Name: Trunk mPOWERED product (only a general description is provided for practical purposes, as precise specifications would be too lengthy and not required for the purpose of this document):

In practice, the exact characteristics of what Product Offerings would be procured by the governance would be detailed and defined in contractual documents as part of the business process described in the previous step. In the same manner as the governance procured products from A-Sat, it would procure other products from other Resource Providers.

One possibility is for the Resource Provider systems to leverage the PSID APIs to publish the procured products to the broker PSS via the APIs. This process could be implemented in a manner such as that governance operators receive notifications of new published Product Offerings in the broker. The operators might then need to access the broker portal so to validate and approve the publishing of the Offerings.

In this manner, the broker PSS catalog would be populated semi-automatically by having all relevant Resource Providers push all previously procured products to the catalog. This would be a continuous process and would enable the PSS catalog to effectively create a pool of aggregated supply from all Resource Providers.

### 3.6.4 Demand Aggregation

The most probable scenario in which demand aggregation is achieved would be to leverage the fact that customers would use a graphical user interface (such as a web application that is part of the broker platform) to either browse the product offerings available in the pool or to submit a product inquiry.

The PSID APIs such as the Product Ordering Management API (PSID622) would be leverage internally in the PSS so to perform the process. Other scenarios in which the PSID APIs would be leveraged would be in the case that a quote is needed from one or more Resource Providers.

### 3.6.5 Product Allocation

Either matchmaking is delegated to the resource provider or the broker PSS would have some matchmaking algorithms that would propose a solution to the customer request. In the end, the customer would confirm the proposal and an order process would occur. In this case, several PSID APIs would be leveraged, mostly between the PSS and the Resource Provider systems. Most probably these would be APIs such as the Product Ordering Management API (PSID622) and the inventory-related APIs (Product Inventory Management API (PSID637), Service Inventory Management API (PSID638), Resource Inventory Management API (PSID639)).

The APIs would be used to ensure both the PSS pool and inventories are synchronized with Resource Provider systems and processes.

### 3.6.6 Product De-allocation

Once the product allocation has ended, a de-allocation step occurs. This could be in the case of service term expiry or premature service cancellation from customer's side. In a similar manner as the previous step, most probably the PSID APIs that would be leveraged would be APIs such as the Product Ordering Management API (PSID622) and the inventory-related APIs (Product Inventory Management API (PSID637), Service Inventory Management API (PSID638), Resource Inventory Management API (PSID639)).

# 4 A User Story: Demonstration of capabilities

The outlined case study shows how the interplay between different PSSs and different entities - a customer aiming at booking specific services to meet their communication needs, a provider offering resources or services via a PSS to broaden their reach to different user communities - can work adapting the PSI APIs. The User Story accompanying the Case Study showcases how the interaction between different PSSs can be implemented. The different roles of providers and PSS can be set as follows:

| Name | Role | Description |
|---|---|---|
| A-Sat | Provider | Provisioning of services |
| B-Sat | Provider | Provisioning of services |
| Provider PSS | Provider, PSS | Provisioning of services and PSS |
| Broker PSS | Customer, PSS | Demonstration user, Broker platform PSS |

Table 4.1: User Story players.

## 4.1 Step 1: Accreditation and service introduction

The broker PSS is importing data from both A-Sat and B-Sat. This can be done via JSON import if the services from A-Sat and B-Sat are being presented in a PSI API compatible JSON scheme format. If the optional demonstration step should be included, the services should include raw bandwidth services. After import, the required accreditation will be conducted for both providers.

The provider PSS sends both *resource specifications* and *service specifications* via the API, showcasing the API connection between the broker PSS and the provider PSS. Subsequently, the broker PSS displays the received data and conducts the accreditation for this provider as well. Finally, the provider PSS sends *product offerings* to the broker PSS via the aforementioned APIs as well. One product offering includes *internet access*, to enable step 2.

## 4.2 Step 2: Mission creation and product order

The demonstration user logs into the broker PSS. The user creates a mission specifying services that are matching the ones send by the provider PSS. Subsequently, the user performs the matchmaking. The broker PSS presents the user the product offering as introduced by the provider PSS in step 1. The user selects this offer and sends the product order.

## 4.3 Step 3: Order processing in provider PSS

The product order is transferred from the broker PSS, via the APIs, to the provider PSS. The new order is displayed in the provider PSS and further processed within the provider PSS. This might include showcasing how party management is handled within the provider PSS. Once the order is checked, a confirmation is send back to the broker PSS.

## 4.4 Optional: Showing additional matchmaking

Optionally another matchmaking action, using imported data in step 1 from A-Sat, can be shown on the example of raw bandwidth services. Potentially, the same services as those from the provider PSS can be shown with B-Sat derived services. The user refines the mission to include raw bandwidth services. The matchmaking in the broker PSS is being conducted once more, showing the raw bandwidth services from A-Sat as options. Potentially, if present, the services from B-Sat can be shown here as well, but it is demonstrated that those from A-Sat are chosen to show step 4.

## 4.5 Step 4: Order confirmation and start of Service

The provider PSS has confirmed the order. The broker PSS shows the order's confirmation status and also the service status.

## 4.6 Step 5: Integration tests

The user story has been completed. Within step 5, the capabilities for other endpoints are shown. This is done conducting the integration tests with a mock-up, i.e. mocking requests from customers or mocking further actions being conducted by potential providers or PSS <-> PSS interactions.

# Last Page of Document