PSI Case Study

PSI-CST

Version:	1.0.0
Date:	2024-04-08
Reference:	PSI-CST
Total Pages:	19

Table of Contents

1 Document Meta Information **3**
1.1 Document Change Record . . . 3
1.2 Documents . . . 3
1.2.1 Reference Documents . . . 3

2 Introduction **4**
2.1 Document Scope . . . 5
2.1.1 Compiled Document . . . 5
2.1.2 Signature . . . 5
2.1.3 PSI First Release Note - Version 1.0.0 . . . 6
2.1.4 Feedback and Contributions . . . 10

3 Case Study **11**
3.1 Problem Identification . . . 11
3.2 Proposed Solution . . . 11
3.3 PSS Architecture and Concept . . . 14
3.4 Resource Provider Concept . . . 14
3.5 The role of the PSID APIs . . . 15
3.6 Description of Key Business Processes . . . 15
3.6.1 Product Accreditation . . . 16
3.6.2 Product Procurement . . . 16
3.6.3 Supply Aggregation . . . 16
3.6.4 Demand Aggregation . . . 17
3.6.5 Product Allocation . . . 17
3.6.6 Product De-allocation . . . 17

List of Figures

1.1 DCR QR-Code. . . . 3
2.1 The PSI consortium. . . . 5
3.1 Case Study Diagram. . . . 13
3.2 Service Delivery Subsystems. . . . 14

List of Tables

1.1 Reference Documents. . . . 3

1 Document Meta Information

1.1 Document Change Record

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.

Figure 1.1: DCR QR-Code.

1.2 Documents

1.2.1 Reference Documents

Acronym	Reference	Title	Version
PSI-DL	PSI-DL	PSI Document List	1.0.0
PSI-CST	PSI-CST	PSI Case Study	1.0.0
PSI-REQ	PSI-REQ	Interface Requirements Document	1.0.0
PSI-TAD	PSI-TAD	Terms, Abbreviations and Definitions	1.0.0
PSI-TOD	PSI-TOD	Tasks and Operations Dictionary	1.0.0

Table 1.1: 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 SatCom users' demands (both commercial and institutional) with SatCom providers' offers. Bringing together multiple 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 SatCom domain defining their requirements on the SatCom services. Those two aspect 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 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 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.,

1. an interface between PSS and resource providers (satellite operators, service providers, or other PSSs),
2. an interface between the PSS and users, and
3. an interface between PSS and its own governance.

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 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 SatCom operators SES, Hellas Sat, Hispasat, Hisdesat, and LuxGovSat, and Inmarsat being both a service & technology provider and a SatCom operator.

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

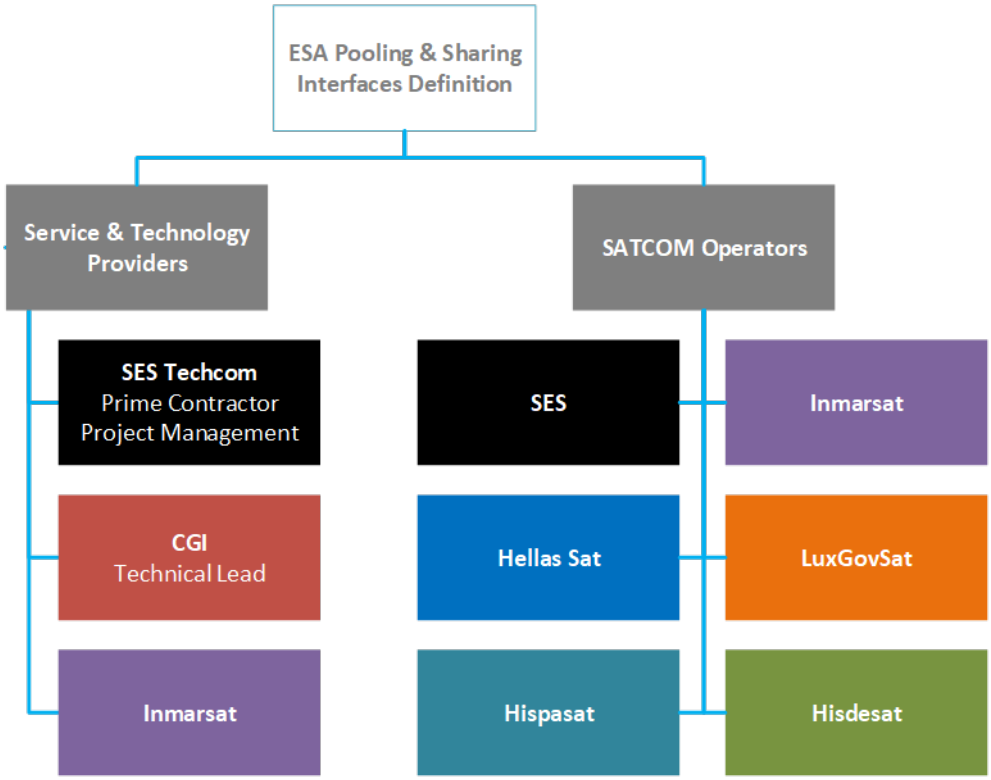

Figure 2.1: The PSI consortium.

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 GOVSATCOM (Government Satellite Communications) products. The issue at hand revolves around consumers of GOVSATCOM 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 2024-04-19 12:46.

2.1.3 PSI First Release Note - Version 1.0.0

2.1.3.1 Introduction

Welcome to the inaugural release of the Pooling and Sharing Interface API! This marks the beginning of an exciting journey. Below, you'll find details about the features, enhancements, and other important aspects of this release.

2.1.3.2 Key Highlights

- The PSI Standard provides a unified interface for satellite communication providers, enabling seamless integration and collaboration between various operators and systems.
- Version 1.0 includes essential features that have been developed in response to feedback from our consortium members and external observers.
  - TM Forum compatible catalog and inventory APIs
  - Inquiry API for distributed matchmaking between customer requirements and resource provider's products
  - Order process compliant with TM Forum
  - Distributed event handling
- This release focuses on raising awareness of the PSI Standard and gathering valuable user feedback to inform future enhancements.

2.1.3.3 What's New

- **PSI001 - Customer Inquiry Management API**: The Customer Inquiry Management API is wrapping the Catalog Management APIs to provide results based on an inquiry send by the customer. The Customer Inquiry Management API takes care of the handling of inquiries sent by a customer and responded by a PSS or provider.

  The PSS may provide different ways for the customer to create an inquiry, depending on the expertise of the user. These can range from just selecting from templates with commonly used product types, optionally customizing the characteristics or even the manual definition of the communication needs.

  Included REST APIs: * `customerInquiry`: Customer Inquiry API

- **PSI620 - Product Catalog Management API**: Based on TMF620 - Product Catalog Management API (Version 4.1.0).

  The Product Catalog Management API provides a standardized solution for rapidly adding partners' products to an existing Catalog. It brings the capability for Service Providers to directly feed partners systems with the technical description of the products they propose to them.

  The Product Catalog Management API provides the operation for the maintenance of product specifications available in the PSS, brought in by providers. A provider wants to utilize a PSS to offer their products to the users of the PSS. The products implement a product specification (describing general characteristics of the product), and they bundle one or more services and/or on-site resources. Therefore, a provider needs to be able to register(create) product specifications to the PSS, modify, remove or view them. Another PSS needs to be able to view the product specifications as well.

  The TM Forum API is extended by Product Template REST API.

Included REST APIs: * `/productOffering`: Product Offering API * `/productSpecification`: Product Specification API * `/productTemplate`: Product Template API

- **PSI620 - Trouble Ticket Management API**: Based on TMF621 - Trouble Ticket Management API (Version 4.0.0).

  The Trouble Ticket API provides a standardized client interface to Trouble Ticket Management Systems for creating, tracking and managing trouble tickets as a result of an issue or problem identified by a customer or another system. Examples of Trouble Ticket API originators (clients) include CRM applications, network management or fault management systems, or other Trouble Ticket management systems (e.g. B2B).

  The Trouble Ticket API provides the operation for tracking incident reports, complaints and other requests of customers and providers. They can be processed either by a PSS helpdesk operator if they concern the functionality of the PSS itself, or by the provider if they affect a SATCOM service. Most likely, the actual implementation is outsourced to an existing ticket system or the CRM.

  Included REST APIs: * `/troubleTicket`: Trouble Ticket API

- **PSI622 - Product Ordering Management API**: Based on TMF622 - Product Ordering Management API (Version 4.0.0).

  The Product Ordering API provides a standardized mechanism for placing a product order with all the necessary order parameters. The API consists of a simple set of operations that interact with CRM/Order Negotiation systems consistently. A product order is created based on a product offer that is defined in a catalog. The product offer identifies the product or set of products that are available to a customer, and includes characteristics such as pricing, product options and market. This API provides a task based resource to request order cancellation.

  The product order references the product offer and identifies any specific requests made by the customer.

  Included REST APIs: * `productOrder`: Product Order API

- **PSI632 - Party Management API**: Based on TMF632 - Party Management API (Version 4.0.0).

  The party API provides standardized mechanism for party management such as creation, update, retrieval, deletion and notification of events. A Party can be an individual or an organization that has any kind of relation with the enterprise. A Party is created to record individual or organization information before the assignment of any role. For example, within the context of a split billing mechanism, Party API allows creation of the individual or organization that will play the role of 3rd payer for a given offer and, then, allows consultation or update of their information.

  Included REST APIs: * `individual`: Individual API * `organization`: Organization API

- **PSI633 - Service Catalog Management API**: Based on TMF633 - Service Catalog Management API (Version 4.0.0).

  The Service Catalog Management API allows the management of the entire lifecycle of the service catalog elements.

  The TM Forum API is extended by Service Template REST API.

  Included REST APIs: * `/serviceSpecification`: Service Specification API * `/serviceTemplate`: Service Template API

- **PSI634 - Resource Catalog Management API** Based on TMF634 - Resource Catalog Management API (Version 4.1.0).

  The Resource Catalog Management API allows the management of the entire lifecycle of the Resource Catalog elements and the consultation of resource catalog elements during several processes such as ordering process.

The TM Forum API is extended by Resource Template REST API.

Included REST APIs: * `/resourceSpecification`: Resource Specification API * `/resourceTemplate`: Resource Template API

- **PSI637 - Product Inventory Management API**: Based on TMF637 - Product Inventory Management API (Version 4.0.0).

  The Product Inventory Management API provides standardized mechanism for product inventory management such as creation, update and retrieval of the representation of a product in the inventory. It also allows the notification of events related to product lifecycle.

  Included REST APIs: * `/product`: Product Inventory API

- **PSI638 - Service Inventory Management API**: Based on TMF638 - Service Inventory Management API (Version 4.0.0).

  The Service Inventory Management API provides standardized mechanism for service inventory management such as creation, update and retrieval of the representation of a service in the inventory. It also allows the notification of events related to service lifecycle.

  Included REST APIs: * `/service`: Service Inventory API

- **PSI639 - Resource Inventory Management API**: Based on TMF639 - Resource Inventory Management API (Version 4.0.0).

  The Resource Inventory Management API provides standardized mechanism for resource inventory management such as creation, update and retrieval of the representation of a resource in the inventory. It also allows the notification of events related to resource lifecycle.

  Included REST APIs: * `/resource`: Resource Inventory API

- **PSI657 - Service Quality Management API** Based on TMF657 Service Quality Management API (Version 4.0.0).

  The Service Quality Management API provides standardized mechanism for managing service level objectives (SLO) and service level specifications (SLS), which in turn are used to define service level agreements (SLAs) and declare monitoring of services and resources on provider side.

  Included REST APIs: * `serviceLevelObjective`: Service Level Objective API * `serviceLevelSpecification`: Service Level Specification API

- **PSI667 - Document Management API**: Based on TMF667 Document Management API (Version 4.0.0).

  The Document Management API provides the operations to synchronize documents and document versions across systems, i.e., between providers, customers and PSS. It also provides operations for uploading documents as well as for viewing of documents online. For example, a product offering of a provider is accompanied by a Service Level Agreement that should be shared with the customer via REST API, or when an order is concluded, an interface is required for sending the invoice.

  Included REST APIs: * `document`: Document Management API * `attachment`: Attachment Management API

- **PSI678 - Customer Bill Management API**: Based on TMF678 - Customer Bill Management API (Version 4.0.0).

  The Customer Bill Management API allows operations to find and retrieve one or several customer bills (also called invoices) produced for a customer also allows operations to find and retrieve the details of applied customer billing rates presented on a customer bill.

  It takes care of bills (invoices) produced for a customer for placed orders in the PSS. A customer bill or invoice is a document produced at the end of a regular back office process at the provider side which runs according to a

bill cycle definition. The customer bill contains information about the total amount due to be paid by a customer for the ordered product(s) during the billing period, the due date for the payment, and other information like the order and attachment references.

Included REST APIs: * `customerBill`: Customer Bill Management API

- **Stock Management API**: Based on TMF687 - Stock Management API (Version 4.1.0).

  The Stock Management API provides standardized mechanism for product stock management such as creation, update and retrieval of the representation of a product stock, reserve product stock, check or query product stock or adjust product stock. It also allows the notification of events related to them.

  The Stock Management API provides the operations to wrap the inventories to allow a PSS (on behalf of a customer) to check the availability of a provider's product. There are more operations that are performed internally on the provider side, which are not covered by the PSI but may be implemented consistently with TM Forum.

  Included REST APIs: * `checkProductStock`: Stock Check API

- **PSI688 - Event Management API**:

  Based on TMF688 - Event Management API (Version 4.0.0).

  The Event Management API provides a standardized client interface to the enterprise event management system for creating, managing and receiving service related events to (indicatively) drive automation workflows, notify other service providers for service outages and SLA violations, trigger Trouble Ticket creation, and enable more complex orchestration scenarios between management systems. The Event Management API can also be used to convey business level Events in support of other processes.

  Some processes between a PSS and a provider (or PSS and PSS), such as customer inquiries and orders, can take longer time to complete. For example, when a customer inquiry is created, the provider may require significant time to process and respond with an adequate product offering. Or, when a product order is placed by a customer, it can take hours to days for its state to change, e.g. from *inProgress* to *completed*.

  Inside a PSS (or a sophisticated provider system) the anticipated approach to propagate such state changes are message queues. A direct connection between these, although possible, would result in a strong coupling of the systems and major implications by the interface definition on the internal implementations. In order to avoid this, the Event Management defines how to exchange the information using REST.

  Note that this does not enforce the use of message queues. All named operations and endpoints can also be implemented in a monolithic application.

  Included REST APIs: * `hub`: Event Hub API * `listener`: Event Listener API

2.1.3.4 Known Limitations

1. Standardized JSON Schemas for resources, services, and products (including space assets and user terminals) are not available in the first release of PSI due do contractual obligations. They will be made available in the next release.
2. The Service Quality Management is rather basic. There is an ongoing effort to align this set of APIs with the results of a TM Forum Catalyst project. More information will follow in one of the next releases.
3. The Inquiry Management API implies the existence of a Mission Management Service. However, the available API implements only the outgoing interface. A full set of APIs to implement such a service are subject to an upcoming release.

2.1.4 Feedback and Contributions

We value your feedback! If you encounter any issues or have suggestions, please reach out. Additionally, we welcome contributions from the community.

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 GOVSATCOM 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 GOVSATCOM 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, GOVSATCOM 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):

- **Broker:** The complete solution combining physical locations, software, people, processes, and infrastructure so as to provide broker operators the means to monitor and operate broker business processes.
- **Broker Software (PSS):** The collection of software systems that share the overall goal of providing a cohesive and single solution to implementing broker business processes. Individually, each subsystem will implement one or more business processes. Collectively, the software subsystems work together to provide the overall functions of the broker solution.
- **Product:** Any process built around communication that is deemed to be useful for customers. These processes can be integrated into the core operations of the broker, enhancing the range of services and solutions offered to customers.
- **Resource Provider:** A “Resource Provider” is defined as any accredited entity responsible for offering Products within the framework of the broker solution. Accreditation procedures for these providers are established and managed by the governance, ensuring compliance with specified standards and quality.
- **Customer:** A “Customer” encompasses both individual users and groups of individuals who utilize Products offered through the broker. These Customers rely on the broker to access GOVSATCOM products.
- **Governance:** The organization responsible for defining and executing general governance processes of the broker, such as procuring Products, on-boarding Products to the Product portfolio or allocating products to customers.

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

1. **Provider and Product Accreditation:** The governance will perform Provider as well as Product Accreditation. This consists in ensuring that Resource Providers meet a set of requirements with respect to security, guarantee of access and product features. This will be a manual step(which can later can be automated).
2. **Product Procurement:** The governance will act as a central procurement department and it will procure suitable Products from various Resource Providers as part of supply aggregation. Once the governance has procured GOVSATCOM Products through a defined procedure, the Products become part of the broker pool, and can be allocated to Customers without the need to carry out an additional procurement procedure.
3. **Supply Aggregation:** The governance performs supply aggregation by procuring Products into a pool. The pool is the collection of procured Products that is kept ready to use. The strategic pooling of Products significantly enhances response times, especially in scenarios where Products provisioning and procurement costs are substantial.
4. **Demand Aggregation:** The governance will provide the means to Customers to order Products. Once these Products are within the pool, Customers can initiate orders, similar to an online shopping experience tailored for GOVSATCOM Products.
5. **Product Allocation:** The Governance will allocate Products to Customers via sophisticated aggregation of demand, matching, prioritization, and optimization techniques.
6. **Product De-allocation:** The governance will oversee the de-allocation of Products from Customers in cases when the contract ends or is terminated before completion.

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:

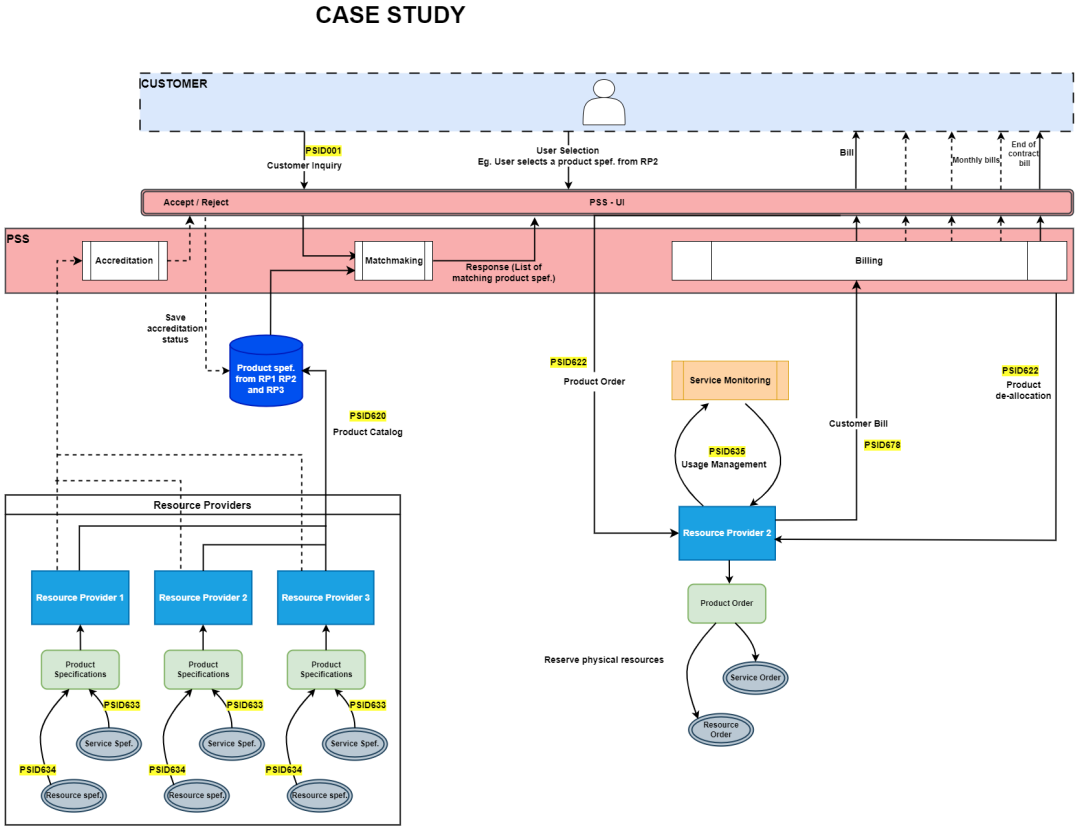

Figure 3.1: Case Study Diagram.

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:

- **STEP1 :** The governance performs the accreditation process for the Resource Providers in the PSS. After the accreditation is successful the Resource Providers are added in approved list.
- **STEP2 :** PSS performs supply aggregation from accredited Resource Providers - RP1, RP2 and RP3.
- **STEP3 :** All three Resource Providers publish their products via the PSID620 API to the PSS, and the Product information is stored in the PSS database.
- **STEP4 :** Customer sends the product requirements in form of an Inquiry using the PSID001 API.
- **STEP5 :** Matchmaking occurs at the PSS level, where the system identifies suitable products based on customer requirements.
- **STEP6 :** Customer is presented with a list of matching products through the PSS user interface.
- **STEP7 :** Customer selects a product and places an order.
- **STEP8 :** The PSS checks the selected product, associates it with the relevant Resource Provider (e.g., RP2), and creates a Product Order using the PSID622 API.
- **STEP9 :** RP2 receives the order, updates its inventory, and reserves the necessary physical resources for the product.

- **STEP10 :** Once the Order is placed and confirmation is sent, the contract is established. RP2 then performs the service monitoring and generates the bills monthly throughout the lifecycle of the contract.
- **STEP11 :** At the end of the contract, the PSS sends a Product Deallocation request to RP2 using the same PSID622 API initially used for allocation.

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 GOVSATCOM market and all associated interfaces. It consists of three main contexts:

- Frontend
- Mission Context
- Business Context

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.

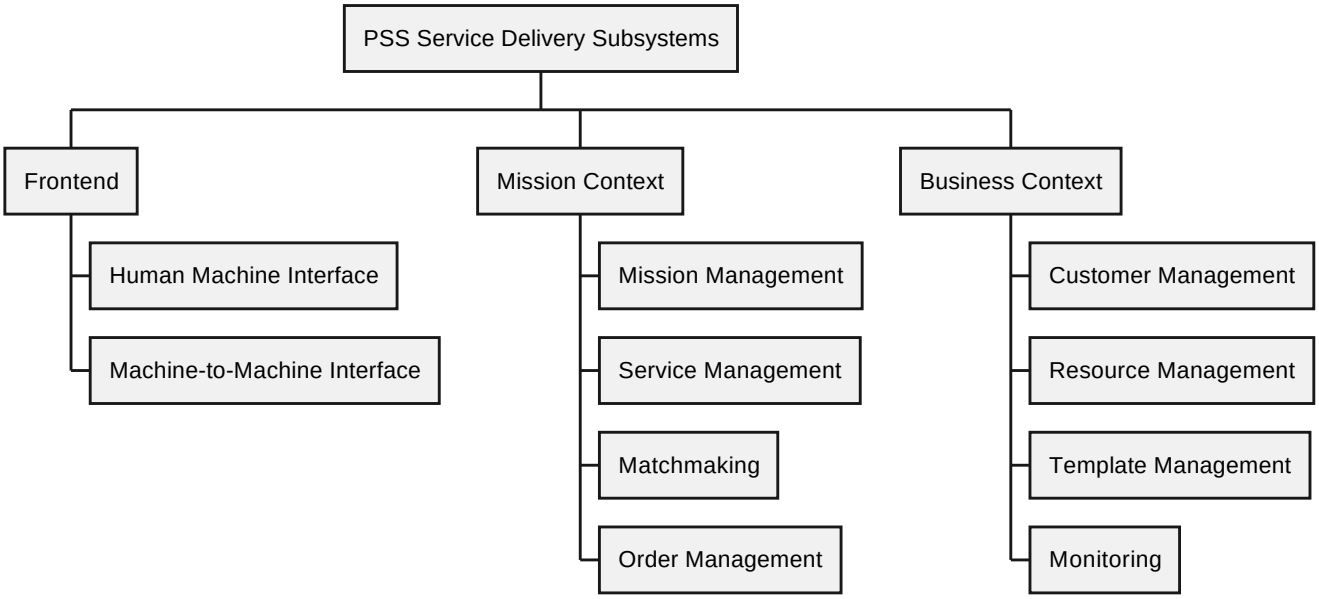

Figure 3.2: Service Delivery Subsystems.

3.4 Resource Provider Concept

The Resource Provider system should essentially have the following modules:

- Order Management System (OMS):
  This refers to a “backend” system responsible for processing of customer order establishment, customer qualification, customer order validation, customer order tracking & management, physically reserving resources related to the order, service monitoring throughout the lifecycle of the contract and generating the invoices(billing).
- Asset Management System (AMS):
  This refers to a “backend” system responsible for general management of assets, which are Products/Services/Resources. More specifically: Catalog/Inventory management, which describe assets (product-service-resources relations, SLAs, etc.) and the current state (i.e., location, allocation, etc.).

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:

- Catalog and Inventory Management
  - PSS620: Product Catalog Management API
  - PSS637: Product Inventory Management API
  - PSS633: Service Catalog Management API
  - PSS638: Service Inventory Management API
  - PSS634: Resource Catalog Management API
  - PSS639: Resource Inventory Management API
- Monitoring
  - PSS635: Usage Management API
- Ticketing
  - PSS621: Trouble Ticket Management API
- Order Management
  - PSS622: Product Ordering Management API
  - PSS679: Product Offering Qualification Management API
  - PSS648: Quote Management API
- Document Management
  - PSS667: Document Management API

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), 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, SES and Hellas 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 SES and Hellas Sat are ready, a certification campaign would be performed involving the governance and SES and Hellas Sat. The campaign will aim to certify one or more SES / Hellas 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 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 SES Product needs to be defined as per the PSID terms. Please refer to section “5.5 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 SES 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):

- Physical Resource Specifications: Intellian Terminal, 2 x 1.3m 20W, model mP130, Gateway infrastructure, etc.
- Logical Resource Specifications: Bandwidth allowing for a minimum combination of FWD and RTN of 50 Mbps, where FWD and RTN have variable pre-defined ratios.
- Service Specifications: Connectivity transport service based on Layer 2 Ethernet P2P E-Line service (E-Access/E-Transit), Layer 3 IP transport
- Product Specifications: Trunk mPOWERED

- Product Offerings: Trunk mPOWERED, intellian mP130, FWD 25 Mbps RTN 25 Mbps; Trunk mPOWERED, intellian mP130, FWD 35 Mbps RTN 35 Mbps; etc. One for each potential variation of FWD and RTN. The product offering will also contain details about SLAs (i.e., network uptime $>$ 99.5 %, Satellite Latency $<$ 150 ms, committed information rate (CIR) definitions, etc.).

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 SES, 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 (PSS622) 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. In this case the PSID API “Quote API” (PSS648) would be leveraged.

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 (PSS622) and the inventory-related APIs (Product Inventory Management API (TMF637), Service Inventory Management API (TMF638), Resource Inventory Management API (TMF639)).

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 (PSS622) and the inventory-related APIs (Product Inventory Management API (TMF637), Service Inventory Management API (TMF638), Resource Inventory Management API (TMF639)).

Last Page of Document