PSI Document List

PSI-DL

Version:	1.0.0
Date:	2024-04-08
Reference:	PSI-DL
Total Pages:	12

Table of Contents

List of Figures

List of Tables

1   Document Meta Information

1.1   Document Change Record

1.1.1   Changes

Date	Version	author	message
2024-04-09	1.0.0	Hendrik Oppenberg	Release of Version 1.0.0

Table 1.1: DCR Table.

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.

Figure 1.1: DCR QR-Code.

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/
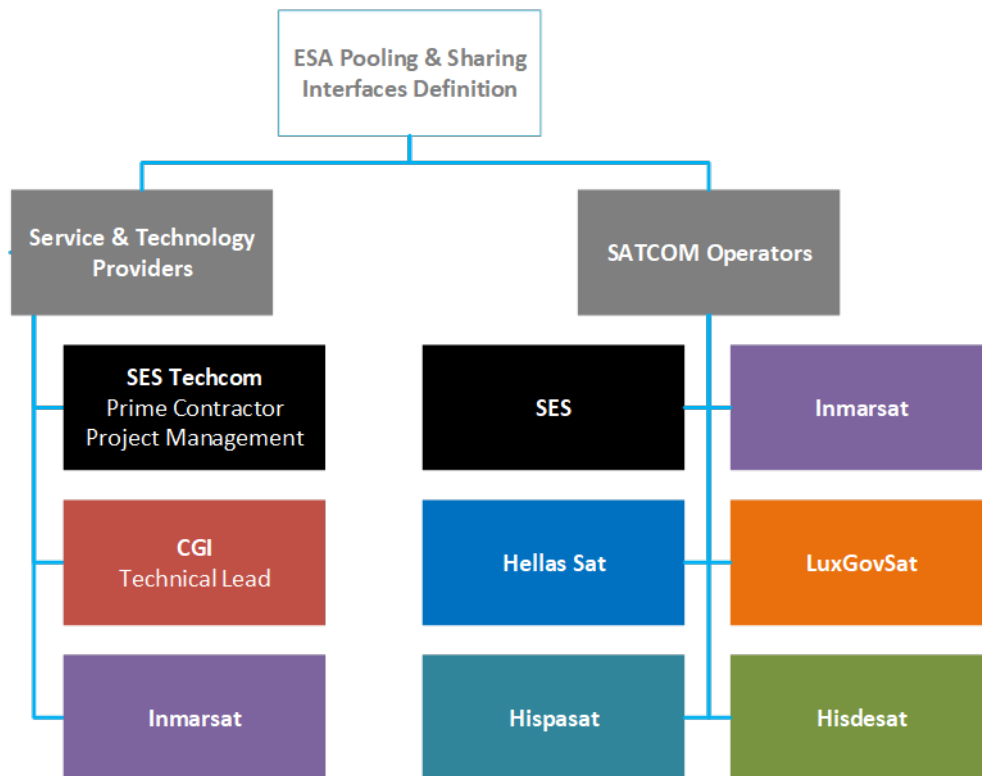
Figure 2.1: The PSI consortium.

2.1 Scope of Document

This document constitutes the document list of the Pooling & Sharing Interface (PSI).

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 12:45.

2.2 Release Notes

2.2.1 PSI First Release Note - Version 1.0.0

2.2.1.1 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.

2.2.1.2 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.

2.2.1.3 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

2.2.1.4   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.

2.2.2   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 List of PSI Documents (DL)

The following table keeps track of all produced documents and their current status. Resulting from the agile methodology, the actual document list is extended by additional documents. Refer the table above for referencing.

Doc-Ref	Document title	Version
PSI-DL	PSI Document List	1.0.0
PSI-CST	PSI Case Study	1.0.0
PSI-GID	Graphical Interface Description	1.0.0
PSI-ICD	Interface Control Document	1.0.0
PSI-REQ	Interface Requirements Document	1.0.0
PSI-SLF	Software License File	1.0.0
PSI-TAD	Terms, Abbreviations and Definitions	1.0.0
PSI-TOD	Tasks and Operations Dictionary	1.0.0
PSI-READFIRST	Suggested Reading Order	1.0.0

Table 3.1: Document list of the PSI project.

Last Page of Document