

CCSDS Mission Product Distribution Services - New interoperability standard for exchanging mission data products

**Dominik Marszk ^{*a}, Daniel Fischer ^a, Mehran Sarkarati ^a, Cesar Coelho ^b,
Rob Tapella ^c, Wai Phyoo ^c, Paul Ramirez ^c**

^a *European Space Agency, European Space Operations Centre, Darmstadt, Germany*

^b *CGI Deutschland B.V. & Co. KG, Darmstadt, Germany*

^c *Jet Propulsion Laboratory, California Institute of Technology, Pasadena, USA*

** Corresponding Author, dominik.marszk@esa.int*

Abstract

The Consultative Committee for Space Data Systems (CCSDS) has standardized a service-oriented architecture framework for interoperable mission operations between space agencies, called Mission Operations (MO) services. These services are being developed under the stewardship of the CCSDS Spacecraft Monitoring and Control (SM&C) Working Group, which is tasked with enabling interoperability, modularization, and reusability across mission operations software infrastructures. As part of its ongoing work, the SM&C WG is standardizing the Mission Product Distribution (MPD) services to facilitate consistent, automated, and cross-agency distribution of mission data products. As part of this framework, the CCSDS is now standardizing the Mission Product Distribution (MPD) services with the aim of achieving interoperability for the provisioning of data products between agencies. This is done by defining a set of services and interface data structures necessary to query, request, retrieve, and receive the data products.

The MPD services are motivated by the need to standardize cross-agency exchange of mission data products, which has traditionally relied on bespoke, mission-specific interfaces—often duplicative, fragile, and hard to scale. MPD resolves these issues by defining common services and data structures for querying and retrieving products, supporting both on-demand and subscription-based delivery via file transfer or service messaging, and building on the existing MO infrastructure.

This paper presents a technical summary of the specification of the MPD services, discusses the capabilities offered by the services, presents some of the use cases, and example applications. A comparison and high-level mapping to the existing product exchange interfaces like ESA Ground Operation System Data Disposition System (EDDS) and NASA AMMOS DataDrive is also made. Finally, the paper presents results from the first inter-agency implementation prototype and discusses potential future applications of the specification.

Keywords: CCSDS, MPD, Mission Products, EDDS, DataDrive

Acronyms/Abbreviations

API	Application Programming Interface
AMMOS	Advanced Multi-Mission Operations System
CCSDS	Consultative Committee for Space Data Systems
CLI	Command Line Interface
DD	DataDrive
EDDS	EGOS Data Disposition System
EGOS	ESA Ground Operation System
FTP	File Transfer Protocol
IAM	Identity and Access Management
LDAP	Lightweight Directory Access Protocol
MAL	Message Abstraction Layer
MO	Mission Operations
MPD	Mission Product Distribution
NASA	National Aeronautics and Space Administration
REST	Representational State Transfer
S3	Amazon Simple Storage Service
SANA	Space Assigned Numbers Authority
SCOS	Spacecraft Control & Operation System
SM&C	Spacecraft Monitoring and Control

SOAP	Simple Object Access Protocol
URI	Uniform Resource Identifier
WSDL	Web Services Description Language
XML	eXtensible Markup Language

1. Introduction

Modern space missions produce a vast and growing volume of mission data products, ranging from telemetry outputs to auxiliary files and processed payload datasets. Increasing number of these missions is collaborative, where major stakeholders are different countries or space agencies. This has created a growing demand for interoperable data exchange mechanisms between space agencies and their ground systems. While each agency has historically developed bespoke solutions to support internal workflows, the lack of standardized interfaces has posed significant challenges for inter-agency collaboration, tool integration, and automation of mission operations processes.

To address this gap, the CCSDS, through the Spacecraft Monitoring and Control (SM&C) Working Group, has standardised the Mission Product Distribution (MPD) Services specification [1]. The MPD defines a set of services for discovering, retrieving, and subscribing to mission data products using a common service-oriented architecture. This new standard leverages the CCSDS Mission Operations (MO) standards stack [2], enabling modular and scalable implementations that can be reused across missions, agencies, and toolchains.

The CCSDS Mission Operations services are a set of standard end-to-end services based on a service-oriented architecture defined by the Consultative Committee for Space Data Systems (CCSDS) and it is intended to be used for mission operations of future space missions. [2]

The architecture allows mission operation services to be specified in an implementation and communication agnostic manner. The core of the MO service framework is its Message Abstraction Layer (MAL) which ensures interoperability between mission operation services deployed by different implementations. The MO services are specified in compliance to a reference service model, using an abstract service description language, the MAL. This is similar to how Web Services are specified in terms of Web Services Description Language (WSDL). Produced machine-readable standard service specifications are then published in SANA registry [3][4]. For each concrete deployment, the abstract service interface must be bound to the selected software implementation and communication technology. [5]

The framework has been demonstrated in multiple space missions, notably the OPS-SAT mission and the Phi-Sat-2 mission. Additionally, the framework has been used in some interoperability projects between space agencies such as ESA, NASA, DLR, and CNES, in order to exchange data between them. [6][7][8]

The MPD Services specification introduces three core service interfaces—Product Retrieval, Order Management, and Product Order Delivery—which collectively enable consumers to query the availability of data products, establish standing delivery orders, and receive products via service or file-based mechanisms. Each service interface is precisely defined in terms of operations, message flows, and reusable data structures, ensuring consistency and semantic clarity across implementations.

This paper presents an in-depth overview of the MPD Services specification. It outlines the architecture and capabilities of each service, illustrates typical usage scenarios, and discusses interoperability with existing non-standard systems such as ESA's EGOS Data Disposition System (EDDS) and NASA AMMOS DataDrive. Furthermore, the paper shares initial findings from a multi-agency implementation prototype, highlighting lessons learned and the potential for broader standard adoption. Through this paper, we aim to demonstrate the value of MPD as a foundational enabler for future mission operations data ecosystems, especially in the context of international and inter-agency cooperation and distributed mission data systems.

High-level diagram showing MPD's placement in a typical MO stack is shown in Figure 1.

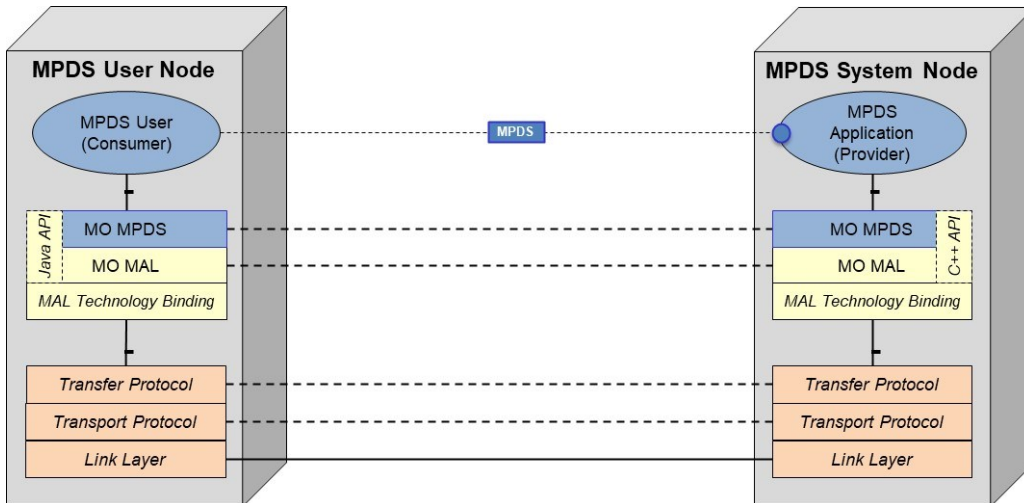


Figure 1. MO MPDS Services Generic Protocol Stack (reproduced from CCSDS Standard)

2. Overview of the CCSDS Mission Product Distribution (MPD) Standard

The CCSDS Mission Product Distribution (MPD) Services standard defines a suite of interoperable service interfaces intended for the exchange of mission data products between space agencies and mission stakeholders. It is built upon the CCSDS Mission Operations (MO) Service Framework and utilizes the Message Abstraction Layer (MAL) to define its data structures, ensuring consistent messaging semantics and extensibility across diverse implementations. The use of MO and MAL ensures building upon a rich service-oriented machine-readable interface description language, with ability to autogenerate consumer and provider interfaces in a number of languages like Java, C, C++, Go. The MAL specification can also be bound to a number of different transport and encoding technologies, with the aim of compatibility with variety of technology stacks seen across the industry.

While the standard is meant to be used on top of MO MAL by default—as it provides an end-to-end framework for interface definition, generation, and implementation—it is also not mandatory. The interface specification could be mapped onto any non-MO technology stack, effectively maintaining standard compatibility at the level of API and data structures.

The MPD services formalize how mission data products can be discovered, queried, retrieved on demand, or delivered automatically based on subscription-style standing orders.

At its core, MPD facilitates both pull-based and push-based product exchange patterns through three primary services:

- **Product Retrieval Service** for querying and downloading specific products,
- **Order Management Service** for submitting and managing standing orders, and
- **Product Order Delivery Service** for receiving notifications or direct delivery of products.

These services rely on a common information model that structures product metadata, filtering criteria, delivery preferences, and more. This chapter describes the MPD data model and individual services, and summarizes the supported operational use cases.

A high-level depiction of the MPD services and their relationships is shown in Figure 2, followed by the general mission product distribution concept in Figure 3.

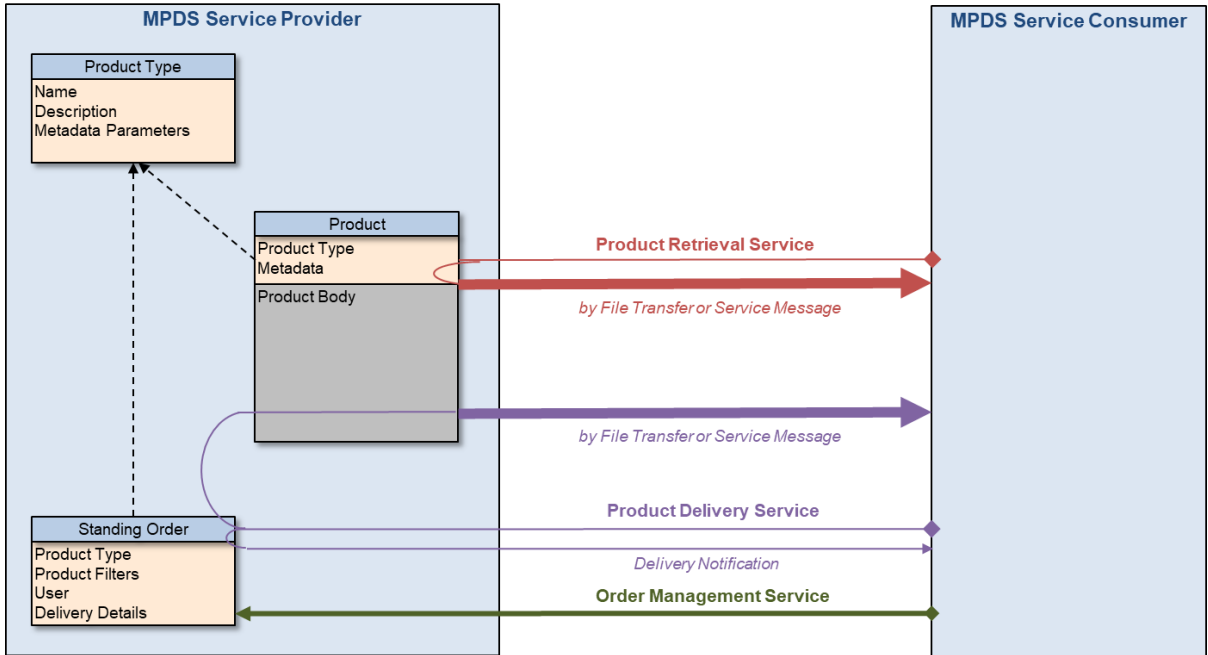


Figure 2. Relations between different Mission Product Distribution services (reproduced from CCSDS Standard)

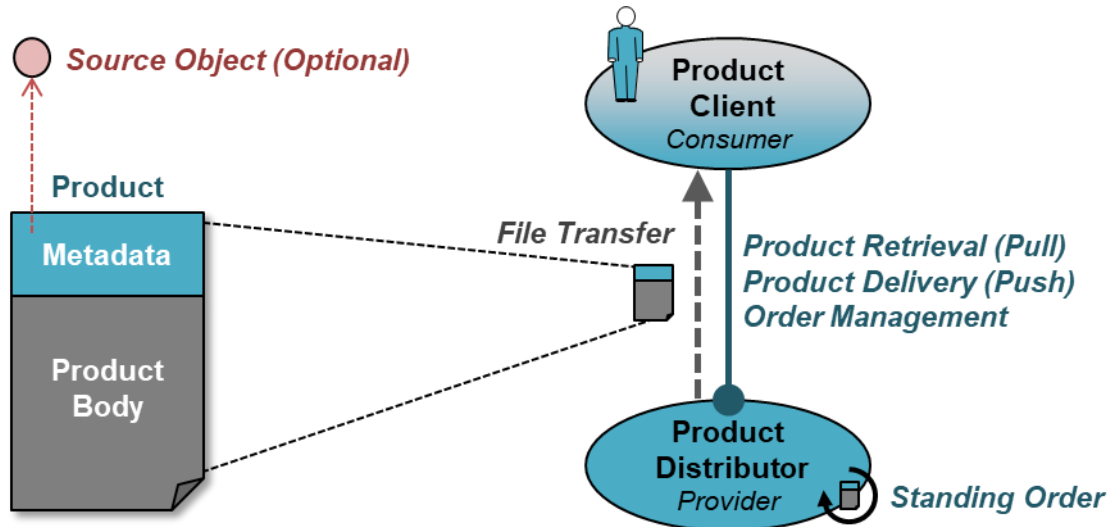


Figure 3. Mission Product Distribution Concept (reproduced from CCSDS Standard)

2.1. MPD Information Model Overview

The MPD information model distinguishes between product definitions (types) and actual product instances. Central to this model are three main concepts:

- **ProductType** defines a category of data products. It includes a name, optional description, and a list of metadata attribute definitions (**AttributeDef**), each specifying a name, type, and optional units and description.
- **Product** represents a specific data product instance. It references a **ProductType**, carries timestamps (**creationDate**, **contentDate**), optional origin metadata (source), attribute values, file metadata, and the product body itself.
- **StandingOrder** specifies a long-lived product subscription. It includes a product filter, delivery method (e.g., file transfer or service message), validity window, and delivery target.

To support selection and filtering of products, the model includes:

- **ProductFilter:** Filters products based on type, domain, sources, and attribute filters.
- **TimeWindow:** Defines temporal constraints for creation and content date ranges.
- **AttributeFilter:** Provides three filter variants—**ValueRange**, **ValueSet**, and **StringPattern**—to refine searches using product metadata.

These components are leveraged across the service interfaces, enabling flexible, fine-grained product discovery and delivery configurations.

A simplified UML diagram illustrating the relationships between these concepts is shown in Figure 4.

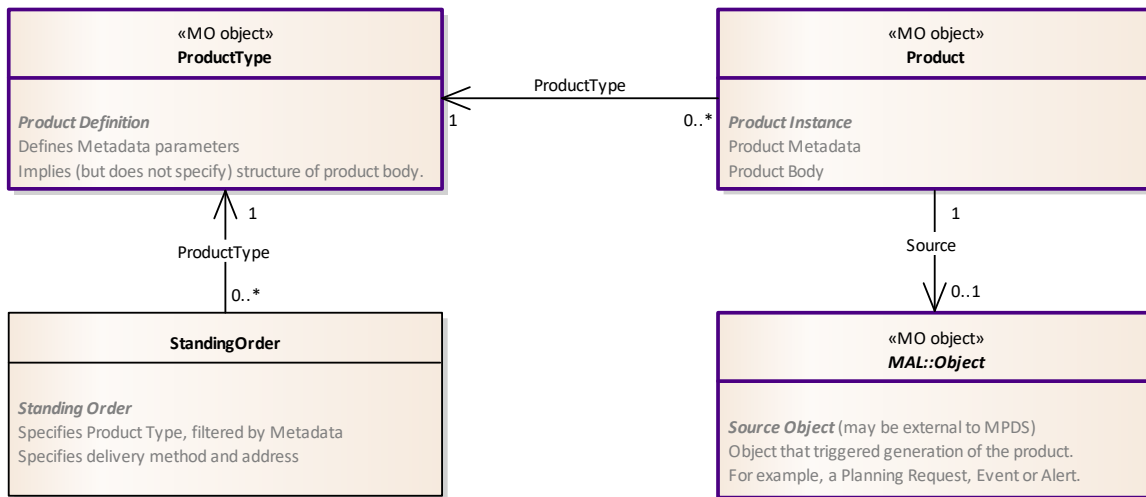


Figure 4 MPD Information Model Overview (reproduced from CCSDS Standard)

2.2. MPD Service Specifications

The MPD standard defines three services, each with a clearly delineated role in the distribution workflow. The service interfaces are grounded in the MAL interaction patterns and allow for different communication styles including synchronous requests and publish-subscribe notifications.

2.2.1. Product Retrieval Service

The Product Retrieval Service enables clients to:

- Browse available products via **listProducts**, which returns a list of **ProductMetadata** filtered by type, creation/content date, source, and metadata attributes.
- Retrieve specific products using **getProducts**, which returns complete Product objects through a progress interaction.
- Trigger file-based delivery of selected products using **deliverProductFiles**, which initiates transfer to a specified URI [9] and sends back per-file delivery status metadata.

This service supports ad hoc (pull-based) interactions and is particularly suited for use cases such as retrospective data access or batch retrieval of archived products.

2.2.2. Order Management Service:

The Order Management Service provides functionality to:

- List existing standing orders (**listStandingOrders**) for a given user and domain.
- Submit a new standing order (**submitStandingOrder**), specifying filter criteria, delivery method, target location, and validity period.
- Cancel an existing standing order (**cancelStandingOrder**) by order ID.

Standing orders allow consumers to subscribe to a continuous stream of products matching specific criteria, reducing the need for repetitive queries and manual downloads.

2.2.3. Product Order Delivery Service

This service complements the Order Management Service and enables actual delivery of subscribed products. It supports two mechanisms:

- **notifyProductDelivery** publishes notifications when products are delivered by file transfer, including metadata, filename, destination URI, and success status.
- **deliverProducts** delivers full product instances directly via the service interface, using a publish-subscribe model.

Consumers can subscribe to these publications based on user and order ID filters. This service is essential for real-time or near-real-time product delivery in mission operations scenarios.

2.3. Use Cases

The MPD standard is designed to support a range of common data distribution patterns in mission operations:

- **On-Demand Product Retrieval:** Users query available products and retrieve them via **getProducts** or **deliverProductFiles**.
- **Scripted Pulling:** Automation clients periodically execute queries and downloads, using the Retrieval Service interfaces.
- **Subscription-Based Delivery:** Consumers define standing orders to receive products as they are generated.
- **Multiple Delivery Methods:** Delivery can occur via:
 - **Service Message:** Products or notifications are transferred directly through the service interface.
 - **File Transfer:** Products are delivered to a nominated URI, with optional notification.

These use cases enable both interactive and automated product distribution workflows, accommodating diverse operational requirements.

3. Comparison with Existing Interfaces

To validate the conceptual completeness and applicability of the CCSDS Mission Product Distribution (MPD) Services, this section compares the MPD against the interfaces of two established, non-standard product exchange systems operationally used within major space agencies. These comparisons help illustrate both the alignment and divergence between current operational interfaces and the MPD standard, while at the same time, suggests how MPD can serve as a unifying abstraction.

The following sub-sections present the ESA **EGOS Data Disposition System (EDDS)**, and the NASA **AMMOS DataDrive**. Both are widely used within their respective agencies and cover a variety of mission operations use cases.

3.1. EGOS Data Disposition System (EDDS)

The EGOS Data Disposition System (EDDS) is a core ESA component used to manage the distribution of mission telemetry, auxiliary data, and derived products. EDDS provides automated file-based delivery to designated endpoints based on user subscriptions. The interface is defined using a SOAP-based WSDL service and accompanying XML schema, which describe a set of operations supporting data requests, job submission and tracking, catalogue queries, and user account management. EDDS also offers a REST-based service. Mapping of particular high-level functionalities between the two interfaces is done in Table 1.

Table 1. Mapping EDDS Concepts to MPD

EDDS Concept	MPD Equivalent
batchRequest() / streamRequest()	submitStandingOrder() (Order Management Service)
getResponse()	notifyProductDelivery(), deliverProducts() (Order Delivery Service)
getJobs(), getStatus()	listStandingOrders()
getCatalogue()	listProducts() + ProductFilter
FileServerDelivery, ServerDelivery	Delivery methods: FILETRANSFER, SERVICE COMPLETE
File metadata embedded in responses	File metadata embedded in ProductMetadata structure

3.1.1. Key Differences and Observations

- **Pull vs Push Semantics:** EDDS supports both push and pull oriented delivery, which is the same for MPD.
- **Search and Metadata Structure:** While EDDS focuses on generating products on demand, it has no advanced filtering and metadata searching options.
- **Jobs vs Standing Orders:** EDDS Job corresponds to MPD Standing Order concept.

- **Notifications:** The EDDS typically does not forward live delivery notifications corresponding to the notifications from `notifyProductDelivery()` operation of MPD. Job status can be polled instead.
- **Server-to-Server Transfers:** EDDS allows direct server-to-server transfers, which corresponds to MPD concepts.
- **Delivery Methods:** MPD allows products to be delivered as full service objects or just the metadata (e.g., `SERVICE_JUST_METADATA`), which puts it in par with EDDS.
- **Authorization Model:** EDDS utilises simple login credential flow that yield authorisation tokens. MPD assumes integration with CCSDS MAL Access Control for fine-grained control, though specifics are implementation-defined.

The MPD provides a higher degree of service abstraction and interface standardization compared to the EDDS interface. The use of MAL-based interactions and formally defined service operations enables improved interoperability and reuse. However, wrapping of EDDS into MPD requires mapping existing request/response structures and job tracking semantics into the MPD information model.

Furthermore, some of the EDDS product data types (e.g. `PktEv`, `PktTc`, `PktTm`) and associated filters are highly specific to the data model and interfaces of the underlying ESA control systems (SCOS and EGS-CC), and might require some compromises when mapping them onto MPD.

3.2. NASA AMMOS DataDrive

NASA AMMOS DataDrive is a cloud-based file metadata and distribution system used to support the downlink, storage, organization, and retrieval of operational and science mission data products. Its architecture comprises an ElasticSearch metadata index, an S3-based file store, and a set of access clients (web interface, CLI, middleware). Mapping of particular high-level functionalities between the two interfaces is done in Table 2.

Table 2. Mapping DataDrive concepts to MPD

DataDrive Concept	CCSDS MPD Equivalent
ObjectType (JSON schema)	ProductType
File	Product (with body = file)
ElasticSearch metadata index	ProductMetadata, AttributeDef
Search by metadata	listProducts with ProductFilter
Saved search	Can be indirectly emulated as StandingOrder
Subscription (CLI-based)	StandingOrder + Delivery Service
File download	getProducts, getProductFiles
Auto-download via CLI	StandingOrder with local pull
Notification (websocket logs)	notifyProductDelivery, deliverProducts
Client-side download	MPD can deliver to client URI or via message
Folder browsing by path	Can be emulated via metadata filtering

3.2.1. Key Differences and Observations

- **Pull vs Push Semantics:** DataDrive is fundamentally pull-oriented. Subscriptions are active only while the CLI is running and download files locally. In contrast, MPD supports server-managed push delivery through standing orders and can move data from provider to consumer endpoints directly, enabling inter-server transfers.
- **Search and Metadata Structure:** DataDrive allows faceted searches across multiple product types. MPD filters are defined per-ProductType. This reflects a design tradeoff: MPD favors formal, type-safe filtering, while DataDrive favors dynamic discovery and full-text flexibility.
- **Saved Searches vs Standing Orders:** DataDrive's saved searches are disconnected from subscriptions. In MPD, the standing order combines filter and delivery configuration in a single construct. An extension to MPD could introduce reusable, named filters as a concept to close this gap.
- **Notifications:** While both systems offer notification mechanisms, MPD formalizes these using MAL's publish-subscribe interactions. DataDrive relies on log messages and websocket events during CLI execution. A persistent, replayable notification mechanism is absent in DataDrive but standardized in MPD.
- **Server-to-Server Transfers:** MPD supports remote delivery via `deliverProductFiles` and standing orders. DataDrive assumes client-side download; direct server-to-server transfers are not supported out of the box.
- **Delivery Methods:** MPD allows products to be delivered as full service objects or just the metadata (e.g., `SERVICE_JUST_METADATA`), offering a richer set of delivery strategies. DataDrive always delivers raw files.

- **Authorization Model:** DataDrive enforces access via AWS IAM and LDAP group-based path policies. MPD assumes integration with CCSDS MAL Access Control for fine-grained control, though specifics are implementation-defined.

Overall, DataDrive and MPD share substantial conceptual overlap, and many DataDrive patterns could be encapsulated by an MPD adapter layer. A set of identified functionality gaps and proposed mitigations is shown in Table 3.

Table 3. Summary of MPD and DataDrive functionality gaps and proposed mitigations

Functionality Gap	Gap location (MPD or DD)	Notes / Action
No browse-by-folder	MPD	Can be emulated via listProducts filtering on path metadata.
No multi-ProductType search	MPD	Not supported by design.
No object-based file delivery	DD	File-only. Can wrap downloads in objects at adapter level.
No provider-side subscriptions	DD	Subscriptions are local only. Server-side standing orders in MPD are superior.
No persistent notification queue	DD	MPD offers built-in pub-sub via notifyProductDelivery.
Cannot pause/resume subscriptions	MPD	Enhancement under discussion (retain resume metadata in standing orders).
No user-specified local delivery (auto)	MPD	Would require wrapper CLI or Consumer-initiated delivery method.

3.3. Summary of Comparison

The comparative analysis of ESA’s EDDS and NASA’s DataDrive against the CCSDS MPD standard reveals substantial conceptual alignment in how space agencies manage and disseminate mission data products. While each legacy system exhibits agency-specific implementations and operational constraints, MPD offers a unifying abstraction that encapsulates the core functionalities of both. Summary of comparison is shown in table Table 4.

Table 4. Summary of comparison between 3 interfaces

Functional Area	EDDS	DataDrive	MPD Standard
Product Search / Filtering	Catalogue queries, param filters	ElasticSearch metadata query	listProducts() + ProductFilter
On-Demand Retrieval	getResponse()	Manual or scripted downloads	getProducts(), deliverProductFiles()
Subscription / Automation	Batch/stream requests, jobs	CLI subscriptions	submitStandingOrder()
Push Delivery	FTP/SFTP delivery, notifications	CLI download + event logs	notifyProductDelivery(), deliverProducts()
Product Metadata	Custom per-mission structures	JSON schema + Elastic mappings	ProductMetadata, AttributeDef
Delivery Status	Job status / download result	CLI logs, no persistent status	PUB-SUB event via service interaction, no persistent status
Client Interfaces	WSDL/SOAP, mission adapters	Web UI, CLI, REST proxy	MAL bindings and autogenerated APIs in: Java, C, Go, C++. Possible other technology bindings

Both EDDS and DataDrive reflect valuable operational experience in mission data distribution but lack the formal standardization and extensibility of MPD. The MPD specification represents a converged model capable of encapsulating their core behaviours while introducing higher interoperability, and service abstraction. Adoption of MPD would unify the interfaces across agencies but also reduce integration overhead for missions requiring inter-agency collaboration.

4. Inter-Agency Prototyping and Validation

To ensure the robustness, clarity, and unambiguity of the CCSDS Mission Product Distribution (MPD) standard, ESA and NASA's Jet Propulsion Laboratory (JPL) jointly committed to a formalized prototyping effort with two independent prototype implementations by each agency — a mandatory step prior to Blue Book publication. This

prototyping phase took place throughout 2024 and is expected to be completed in 2025, running in parallel with the CCSDS Agency Review.

The primary objective of the prototype was to validate the completeness and consistency of the specification through practical implementation. A shared test harness was designed, comprising a jointly developed MPD Consumer component and an extensible interface for connecting MPD Provider implementations. Both parties populated their provider backends with identical dummy mission products and metadata, enabling harmonized validation.

The prototype architecture relied on ESA's Java MO Services stack, employing a process-to-process transport layer for streamlined integration. File transfers were deliberately scoped to local host operations (i.e., all valid URIs prefixed with `file://`) to simplify testing and avoid external system dependencies. While integrations with operational systems such as EDDS and DataDrive were considered, the prototype remained self-contained by design. Nonetheless, the prototypes can be potentially integrated with EDDS or DataDrive in future activities.

A comprehensive suite of test cases was derived from the Blue Book, ensuring that all normative example scenarios were implemented. The test suite was designed to maximize coverage of all service operations defined in the autogenerated stubs, validating both push and pull workflows.

The prototype architecture is illustrated in Figure 5. The test is orchestrated using a test harness, built on top of Java MO Framework, with common consumer implementation, implementing the test cases. Both prototype providers from ESA and NASA are implemented independently and executed against the same test suite. They are exposing identical dummy sets of backend data and configuration.

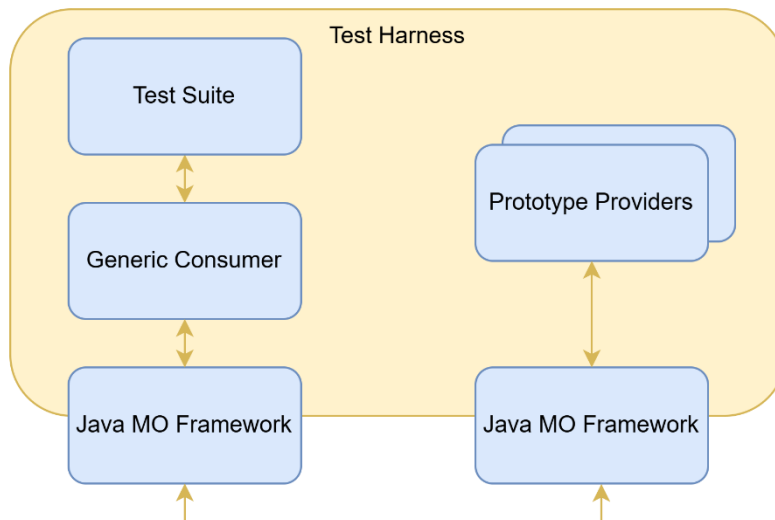


Figure 5 High level MPD prototype architecture

The initial results confirmed the overall clarity and feasibility of the standard. However, several ambiguities surfaced during implementation—such as structural naming inconsistencies, the misapplication of nullable attributes, and inappropriate field types. These issues were swiftly resolved via updates to the data model, contributing directly to refinements in the specification. The prototyping phase captured all the identified issues in the CCSDS prototype report to be delivered as part of the prototyping activity.

5. Potential Future Applications and Integrations

The CCSDS Mission Product Distribution (MPD) standard is designed to address long-standing interoperability challenges in the distribution of mission data products. Its service-oriented abstraction and formal specification make it broadly applicable across multiple mission domains and architectural layers. Looking ahead, MPD is well positioned to serve as a foundation for next-generation data exchange infrastructures across agency boundaries, operational segments, and mission lifecycles.

5.1. Application Scenarios

Several categories of mission operations stand to benefit directly from MPD adoption:

- **Mission Data System ↔ Payload Ground Processor Interfaces:** Many missions maintain separate ground chains for spacecraft operations and payload processing. MPD offers a clean separation of concerns by defining standardized delivery contracts between the mission control system and downstream payload processing centers.
- **Distributed Operations and Hosted Payload Scenarios:** As hosted payload missions become more common, telemetry and auxiliary data must often be routed from the hosting organization to the payload owner. MPD enables such handovers using standing orders or direct retrieval interfaces, decoupled from internal file formats or transport protocols.
- **Inter-Agency Data Sharing:** MPD was designed from the outset to support cross-agency data exchange. Its information model is rich enough to capture metadata and product semantics across diverse missions, while the underlying CCSDS Mission Operations (MO) Framework ensures consistency and reusability. In particular, collaboration scenarios such as ESA ↔ NASA or commercial-to-agency data exchange (e.g., Earth observation constellations) can be streamlined using the MPD interfaces.

These application domains illustrate MPD's value as a general-purpose product exchange layer within increasingly modular and federated mission architectures.

5.2. *Benefits of MPD for New Missions*

Future missions integrating MPD as part of their ground architecture can realize a number of strategic and practical advantages:

- **Elimination of Custom Interfaces:** By adopting MPD as a baseline interface for product dissemination, projects avoid the need to define, develop, and maintain bespoke APIs or file transfer contracts.
- **Tool and System Interoperability:** MPD defines reusable data structures and service operations, enabling plug-and-play integration of common components such as consumers, order brokers, or file staging nodes.
- **Cross-Agency Consensus:** As a CCSDS standard co-developed and endorsed by multiple space agencies, MPD represents an interoperable interface model that can be reliably adopted by international mission teams and stakeholders.
- **Future-Proofing and Governance:** The use of the MO Framework ensures alignment with other CCSDS services and provides a solid foundation for long-term extensibility.

These factors make MPD particularly attractive for missions aiming to minimize integration risk while maximizing future reusability.

5.3. *Future Enhancements and Considerations*

While the current version of MPD fulfils its design objectives, several areas of potential enhancement have been identified for future review cycles:

- **Enhanced Filtering and Querying:** Although the current ProductFilter model supports attribute- and time-based constraints, future enhancements may introduce Boolean logic, support for multi-ProductType queries, or advanced temporal scopes.
- **Formalization of ProductType Registries:** Standardizing common ProductType definitions through a SANA registry would enable greater semantic alignment across missions and simplify consumer-side integration.
- **Product Information Service:** An optional companion service could expose available ProductTypes, attribute definitions, or data lineage information in a discoverable form, facilitating introspection and tooling.
- **Delivery Protocol Flexibility:** MPD already supports protocol-independent delivery, where the destination URI prefix [9] (e.g., `file://`, `sftp://`, `https://`) informs the implementation-specific delivery mechanism. This model could be extended or profiled to ensure consistent semantics across deployments.

The CCSDS SM&C Working Group is tracking proposed enhancements and community feedback via a public issue tracker [10]. As with other CCSDS standards, MPD will be subject to periodic review cycles, with a five-year window offering time to observe patterns of adoption and identify areas for formal evolution.

6. **Conclusions**

As the complexity of modern space missions increases and inter-agency collaboration becomes more common, the need for standardized, interoperable mechanisms for mission product distribution has become critical. Historically, ground segments have relied on custom-built interfaces, leading to duplicated effort, costly integration cycles, and limited tool portability. The MPD standard directly addresses these challenges by defining a flexible, service-oriented framework for querying, retrieving, and delivering mission data products.

At the time of writing, the standard is in the final stages of CCSDS Agency Review, which is a mandatory step for each book proposed to be released as a CCSDS standard, as outlined in Organization and Processes for CCSDS [11]. A joint prototype developed by ESA and NASA/JPL validated the specification against a broad set of use cases, and a Prototype Report (Yellow Book) is currently in preparation. The standard is expected to be published as a CCSDS Blue Book in Q3–Q4 2025, marking its formal release and suitability for operational adoption.

MPD specifies three core services—**Product Retrieval, Order Management, and Product Order Delivery**—alongside a robust information model built for extensibility and semantic clarity. These services enable both pull-based and subscription-based workflows, abstracting delivery mechanisms and metadata structures behind formally defined operations.

Beyond its technical completeness, MPD's design is rooted in the CCSDS MO Framework and Message Abstraction Layer (MAL). This approach ensures **technology independence**: while the inter-agency prototype presented in this paper used a Java implementation, equivalent stacks in C, C++, and Go exist and are fully interoperable. Code generation from the machine-readable service specifications enables rapid onboarding for new consumers and providers, without reimplementing core middleware logic.

The practical value of MPD was confirmed through a collaborative prototype developed by ESA and NASA/JPL. This effort validated all normative use cases and demonstrated cross-implementation compatibility. It also reinforced MPD's ability to streamline cross-institutional integration by avoiding bespoke interface definitions—an effort often underestimated in scope and complexity.

The conceptual mapping to existing systems such as ESA EDDS and NASA AMMOS DataDrive shows that MPD encompasses their capabilities while offering a unified, formal interface. This highlights MPD not only as a technical solution, but also as a **strategic enabler of inter-agency data ecosystems**. By facilitating standard-based product exchange, MPD lays the foundation for long-term harmonization across agency, mission, and contractor boundaries.

We formally recommend that space missions, ground segment developers, and data service providers adopt MPD for current and future projects requiring mission product dissemination. As the standard approaches Blue Book publication, stakeholders are invited to contribute to its evolution through the open CCSDS issue tracker [10].

References

- [1] CCSDS 522.2-R-2, Mission Operations - Mission Data Product Distribution Services, CCSDS Red Book, July 2024.
- [2] CCSDS 520.0-G-3, Mission Operations Services Concept, CCSDS Green Book, December 2010.
- [3] SANA Registry for MO Schemas: <http://sanaregistry.org/r/moschemas/>
- [4] SANA Registry Link for MPD XML Definition: <https://beta.sanaregistry.org/r/moschemas/area009-v001-MPD.xml>
- [5] CCSDS 521.0-B-3, Mission Operations Message Abstraction Layer, CCSDS Blue Book, March 2024.
- [6] OPS-SAT: Preparing for the Operations of ESA's First NanoSat, D Evans, AT Lange, JL Feiteirinha, J Nörtemann, C Coelho, 14th International Conference on Space Operations, .
- [7] Φ-Sat-2: Onboard AI Apps for Earth Observation, A Marin, C Coelho, F Deconinck, I Babkina, N Longepe, M Pastena, Space and Artificial Intelligence 2021
- [8] CCSDS Mission Operations Directory service: Paving the road for a new world of services' discoverability, C Coelho, S Cooper, M Sarkarati, M Merri, O Koudelka, 4S Symposium
- [9] T. Berners-Lee, R. Fielding, and L. Masinter, Uniform Resource Identifier (URI): Generic Syntax, RFC 3986, Reston, Virginia, ISOC, January 2005.
- [10] CCSDS SM&C WG Issue Tracker: <https://github.com/ccsdsmo/smc-backlog/issues>
- [11] CCSDS A02.1-Y-4, Organization and Processes for the Consultative Committee for Space Data Systems, CCSDS Yellow Book, April 2014