



INTERFACE CONTROL DOCUMENT

Copernicus Space Component

Production Interface Delivery Point Specification

Reference	ESA-EOPG-EOPGC-IF-3
Issue/Revision	2.0
Date of Issue	26/04/2023
Status	Issued

CHANGE LOG

Production Interface Delivery Point Specification	Issue Nr.	Revision Number	Date
Initial version	1	0	01/08/19
Update for review comments	1	1	20/09/19
Update following Checkpoint comments, inclusion of ODPRIIP extension and addition of Annex 2 and 3	1	2	12/02/20
Alignment of Attribute entities	1	3	17/02/20
Update to ProductionType and Query by Name, Attributes, Geographic Criteria; Update to Authentication methods and User Entity; Addition of Monitoring and Reporting chapter	1	4	30/04/20
Update to align to LTA ICD 1.4 for LTA start of operations	1	5	09/09/20
Addition of OriginDate property	1	6	25/11/20
Update following Checkpoint 2020 and alignment with CSC Common Entity Definition Document	1	7	19/04/21
Alignment of Subscription query examples	1	8	02/07/21
Update for "Checkpoint 2022"	1	9	03/10/22
Update following "Checkpoint 2022" comments	2	0	26/04/23

CHANGE RECORD

Issue Number 1	Revision Number 1		
Reason for change	Date	Pages	Paragraph(s)
Update for review comments	Sept. 2019	Multiple	Multiple

Issue Number 1	Revision Number 2		
Reason for change	Date	Pages	Paragraph(s)
Product property 'CreationDate' renamed to 'PublicationDate'	Dec. 2019	Multiple	Multiple
Update to Attribute Entity Description table	Dec. 2019	14	3.2, Table 2
Update to clarify query examples, including the addition of Query by Name	Dec. 2019	20	3.3
Update of Query by Attributes examples	Dec. 2019	21	3.3.1.6
Update for (OD)PRIP subscriptions	Dec. 2019	27	3.5
Update for On-Demand Processing	Dec. 2019	Multiple	4, Annex 1
Update for quota details	Dec. 2019	46	5.1.2

Addition of OData Metadata Description	Jan. 2020	58	Annex 2
Addition of Compliance Test Suite	Feb. 2020	62	Annex 3

Issue Number 1		Revision Number 3	
Reason for change	Date	Pages	Paragraph(s)
Update to align Attribute entities	Feb. 2020	Multiple	Figure 2, Annex 2

Issue Number 1		Revision Number 4	
Reason for change	Date	Pages	Paragraph(s)
Update to Query by Name	Mar. 2020	20	3.3.1.1
Update to Query by Attributes	Mar. 2020	21	3.3.1.6
Change of LastQueryDate to LastNotificationDate	Mar. 2020	27	Figure 4, Table 3
Modifications to the User Entity	Mar. 2020	44	5.1
Addition of Monitoring and Reporting	Mar. 2020	47	6
Update to Query by Geographic Criteria	Apr. 2020	21	3.3.1.5
DateTimeOffset Attribute update	Apr. 2020	Multiple	Figure 2, Annex 2
Update to ProductionType	Apr. 2020	Multiple	3.3.1.3, Annex 2

Issue Number 1		Revision Number 5	
Reason for change	Date	Pages	Paragraph(s)
Minor update to Query by Geographic Criteria	May 2020	20	3.3.1.5
Update to the ProductionType Values	May 2020	Multiple	Table 1, Annex 2
Addition of AUXIP ICD and Glossary to Reference Documents	Aug. 2020	8	1.4
Modification of 'substringof' to 'contains' to align with OData v4.01	Aug. 2020	Multiple	Multiple
Modification of DateAttribute to DateTimeOffsetAttribute	Aug. 2020	58	Annex 2
Modification of "count" to "@odata.count" in the examples to align with OData v4.01	Aug. 2020	Multiple	Multiple
Update to the use of quotation marks in the Request and Response examples	Aug. 2020	Multiple	Multiple
Addition of BooleanAttribute	Aug. 2020	Multiple	3.2, Annex 2
Addition of Metric entity, Intersects function, and gauge and counter Enum types to Annex 2	Aug. 2020	58	Annex 2
Update to the use of quotation marks in the Request and Response examples	Aug. 2020	Multiple	Multiple
Update to the Production Summary Reporting Statistics	Aug. 2020	Multiple	Table 12, Table 13

Update to Authorization and Authentication	Sep. 2020	10	2.2
--	-----------	----	-----

Issue Number 1		Revision Number 6	
Reason for change	Date	Pages	Paragraph(s)
Addition of OriginDate property	Nov. 2020	Multiple	3.2, Table-1, 3.3.1.2

Issue Number 1		Revision Number 7	
Reason for change	Date	Pages	Paragraph(s)
Addition of support for the “and”, “or”, “not”, “in” operators	Apr. 2021	22	3.3.1.7
Update on the management and publication of the Catalogue Exports	Apr. 2021	25	3.3.3
Addition of ServiceAlias property to the User Entity Model	Apr. 2021	44	5.1
Addition of Event Recording and Event Entity Model	Apr. 2021	-	-
Update to Annex 3: Interface Test Suite	Apr. 2021	62	Annex 3

Issue Number 1		Revision Number 8	
Reason for change	Date	Pages	Paragraph(s)
Alignment of Subscription examples, removing the “Products?\$filter=” syntax from “FilterParam”	Jun. 2021	25, 26	3.5

Issue Number 1		Revision Number 9	
Reason for change	Date	Pages	Paragraph(s)
Addition of [AD-7] to [AD-11]	Oct. 2022	8	1.3
Addition of GeoFootprint property	Oct. 2022	14	3.2, Figure 2, Table 1
Update of Attribute Entity Description	Oct. 2022	17	3.2, Table 2
Inclusion of \$expand=Attributes query example	Oct. 2022	17	3.2
Updates and alignment of Annex 2	Oct. 2022	58	Annex 2

Issue Number 2		Revision Number 0	
Reason for change	Date	Pages	Paragraph(s)
Update of [AD-7] in line with the new Traceability Service	Apr. 2023	8	1.3

Update of Figure 2 to align with Table 1 for OriginDate and Table 2 for BooleanAttribute	Apr. 2023	14	3.2
Update of Example of \$expand=Attributes Query Response	Apr. 2023	17	3.2
Default WorkflowOption property changed to optional	Apr. 2023	33	4.1, Table 6
EstimatedDate ProductionOrder property changed to optional	Apr. 2023	33	4.1, Table 7
Addition of OrderQuota and DownloadQuota to Table 9, in line with CSC Common Entity Definition	Apr. 2023	44	5.1, Table 9
Removal of Event Recording functionality	Apr. 2023	n/a	6.2

Table of contents:

1 INTRODUCTION.....	7
1.1 Purpose and Scope	7
1.2 Structure of the Document	7
1.3 Applicable Documents	8
1.4 Reference Documents	8
1.5 Definitions.....	9
1.6 List of Acronyms	9
2 OVERVIEW.....	10
2.1 Production Interface Delivery Point (PRIP) and On-Demand Production Interface Delivery Point (ODPRIP) Description	10
2.2 Interface Configuration.....	10
3 PRIP INTERFACE DESCRIPTION	12
3.1 Nominal Use Scenario Description.....	12
3.2 Product Entity Model.....	14
3.3 Query Products Catalogue	20
3.3.1 Query Products.....	20
3.3.1.1 Query by Name.....	20
3.3.1.2 Query by Product Origin Date	21
3.3.1.3 Query by Product Publication Date	21
3.3.1.4 Query by Sensing Date	21
3.3.1.5 Query by Geographic Criteria	21
3.3.1.6 Query by Attributes	21
3.3.1.7 Additional Options.....	22
3.3.2 Query Products Response	23
3.3.3 Catalogue Export	25
3.4 Product Download	26
3.5 Subscription	27
4 ON-DEMAND PROCESSING INTERFACE DESCRIPTION	31
4.1 On-Demand Entity Model and Properties.....	33
4.2 Workflow and ProductionOrder Examples	39
4.2.1 Step 1 – Workflow Query	39
4.2.2 Step 2 – ProductionOrder Submission	40
5 CLIENT ADMINISTRATION.....	44
5.1 User Entity Model	44
5.1.1 SystemRole	45
5.1.2 Quota	46
6 MONITORING AND REPORTING	47
6.1.1 Reporting.....	47
6.1.1.1 Systematic and On-Demand Production Reporting.....	48
6.1.1.2 On-Demand Production Reporting	49
6.1.2 Production Metrics.....	52
ANNEX 1: WORKFLOW TAILORING	57
ANNEX 2: ODATA METADATA DESCRIPTION	58
ANNEX 3: INTERFACE CONFORMANCE TEST SUITE	62

1 INTRODUCTION

1.1 Purpose and Scope

The purpose of this document is to specify an https RESTful Application Programming Interface (API) through which Sentinel (or Auxiliary) data products may be discovered and downloaded by authorised users from a Production Interface delivery Point (PRIP). PRIPs are intended as the standard pick-up points for systematically-produced Sentinel data products from within the PDGSs, replacing the diverse interfaces currently in place inside each Sentinel PDGS. The API is (optionally) extended to manage on-demand production, whereby an order triggers a production that is delivered through the Interface delivery Point. In this case the API is referred to within this document as the On-Demand Production Interface delivery Point (ODPRIP).

The specification of this API is intended for:

- The discovery and download of systematically produced data products,
- The discovery of the processing services available for a given type of parent product
- The submission of on-demand processing requests for pre-identified parent products
- Monitoring of on-demand processing requests

It is anticipated that this will be further refined for future additional services for which the Production Interface Point concept is valid. For example, a possible extension might cover on-demand processing batch requests and the possibility to trigger the production for a given discreet area (i.e. a tile) and time, rather than for a selected parent product. The current version of this document introduces a preliminary design of the ODPRIP which will be subsequently revised.

This specification uses the latest OData v4 standard for REST-ful APIs. The API interface is intended to be compliant with the OData v4 standard, however, it is not necessary to fully implement the OData v4 standard. Compliance to the ICD can be considered as an API contract with a set of API OData v4 style calls, which have to be supported, that shall follow OData query conventions such as URL syntax, filter syntax, content accessing conventions, etc.

1.2 Structure of the Document

This document is structured as follows:

- Section 1 (this section): Introduction, providing document structure, reference documents and definitions/acronyms
- Section 2: Provides an overview of the (On-Demand) Production Interface Delivery Point, its interface with users and the nominal product retrieval scenario
- Section 3: Description of the PRIP-client interface
- Section 4: Description of the ODPRIP-client interface
- Section 5: Description of the client administration interface
- Section 6: Description of monitoring and reporting
- Annexes: Contain Workflow tailoring, an OData metadata description and compliance test suites

1.3 Applicable Documents

- [AD-1] CSC Sentinel-1 Product Unit Definition and Metadata ICD [ESA-EOPG-EOPGC-SP-01]
- [AD-2] CSC Sentinel-2 Product Unit Definition and Metadata ICD [ESA-EOPG-EOPGC-SP-02]
- [AD-3] CSC Sentinel-3 Product Unit Definition and Metadata ICD [ESA-EOPG-EOPGC-SP-03]
- [AD-4] CSC Sentinel-5P Product Unit Definition and Metadata ICD [ESA-EOPG-EOPGC-SP-04]
- [AD-5] CSC POD Product Unit Definition and Metadata ICD [ESA-EOPG-EOPGC-SP-5]
- [AD-6] CSC Common Entity Definition Document [ESA-EOPG-EOPGC-IF-5]
- [AD-7] Copernicus Data Space Ecosystem - Traceability Service Interface Control Document [CDSE-TRC-TSY]
- [AD-8] OData Version 4.01. Part 1: Protocol
<http://docs.oasis-open.org/odata/odata/v4.01/odata-v4.01-part1-protocol.html>
- [AD-9] OData Version 4.01. Part 2: URL Conventions
<http://docs.oasis-open.org/odata/odata/v4.01/odata-v4.01-part2-url-conventions.html>
- [AD-10] OData JSON Format Version 4.01
<http://docs.oasis-open.org/odata/odata-json-format/v4.01/odata-json-format-v4.01.html>
- [AD-11] "The GeoJSON Format", RFC 7946, August 2016.
<http://tools.ietf.org/html/rfc7946>

1.4 Reference Documents

- [RD-1] OData Documentation – OData – the Best Way to REST – v4.01 ISO standard
<http://www.odata.org/documentation/>
- [RD-2] OData Protocol
<http://docs.oasis-open.org/odata/odata/v4.01/odata-v4.01-part1-protocol.html>
- [RD-3] CSC Long Term Archive Interface Control Document [ESA-EOPG-EOPGC-IF-2]
- [RD-4] OGC Earth Observation Metadata profile of Observations & Measurements
<http://docs.opengeospatial.org/is/10-157r4/10-157r4.html>
- [RD-5] OGC EO Dataset Metadata GeoJSON(-LD) Encoding Standard
<https://docs.ogc.org/is/17-003r2/17-003r2.html>
- [RD-6] Data Protection (GDPR)
https://ec.europa.eu/info/law/law-topic/data-protection_en



[RD-7] The OAuth 2.0 Authorization Framework – Password Credentials Grant, October 2012, Internet Engineering Task Force (IETF)
<https://tools.ietf.org/html/rfc6749#section-4.3>

[RD-8] Auxiliary Data Interface Delivery Point ICD [ESA-EOPG-EOPGC-IF-10]

1.5 Definitions

API - Application Programming Interface, is a set of routines, protocols, and tools for building software applications.

Client – any HTTP client performing HTTP/OData requests to the PRIP

Product – Packaged set of data files corresponding to EO or auxiliary product format specification

Quota – refers to measures to restrict the over usage of an API (limiting number of parallel downloads, orders etc.).

Use Case – is a list of actions or event steps, typically defining the interactions between an actor and a system, to achieve a goal.

1.6 List of Acronyms

AD	– Applicable Document
API	– Application Programming Interface
AUXIP	– AUXiliary Interface delivery Point
CSC	– Copernicus Space Component
DSIB	– Data Session Information Block
E2E	– End-to-End
EDIP	– EDRS Interface delivery Point
EO	– Earth Observation
ESA	– European Space Agency
GS	– Ground Segment
HTTP(S)	– Hypertext Transfer Protocol (Secure)
ID	– Identifier
ICD	– Interface Control Document
JSON	– JavaScript Object Notation
LTA	– Long Term Archive
OData	– Open Data Protocol
ODP	– On-Demand Processing
ODPRIP	– On-Demand PROduction Interface delivery Point
PRIP	– PROduction Interface delivery Point
RD	– Reference Document
REST	– Representational State Transfer
TBC	– To Be Confirmed
URI	– Uniform Resource Identifier
UTC	– Coordinated Universal Time
UUID	– Universally Unique Identifier
XBIP	– X-Band Interface delivery Point

2 OVERVIEW

2.1 Production Interface Delivery Point (PRIP) and On-Demand Production Interface Delivery Point (ODPRIP) Description

Production Interface Delivery Points (PRIP) are pick-up points for newly published Sentinel data products, located inside the Sentinel Ground Segments, which allow clients to straightforwardly discover and retrieve available products through standard interfaces, in this case an OData RESTful API. Clients of the PRIPs are intended to be bulk downloaders whose principal use case requires systematic (or at least regular) retrieval of newly published Sentinel data products. Examples of PRIP clients include the Sentinel Data Hubs (which in turn make the products available to end users), Long Term Archives (LTAs), and possibly clients requiring data for Instrument and Product Performance investigations.

PRIPs are the first point of retrieval for newly published Sentinel data products and, as such, they are co-located with the production service. They make new products available in the agreed packaged unit, with any compression as described in [AD-1], [AD-2], [AD-3], [AD-4], [AD-5], with the products only being published once the product package is fully available. Products are supplied on a rolling archive, for example of 3 to 5 days, providing a buffer allowing the recovery of data in case of unavailability (the precise requirements are to be specified in the corresponding Technical Requirements, according to the system budget and operational needs). The API and standard use scenario allow authorised users to discover, with a simple set of filters, the list of products which need to be retrieved in batches, for example since the last check, and then to perform the downloads.

A PRIP may be extended to manage on-demand production (in which case it is referred to as ODPRIP in this document). ODPRIPs allow clients to trigger the generation of a higher level user product (typically Level 1 or 2) from an archived lower level product (typically Level-0), for those higher level products not available online on the rolling archive. The ODPRIP processes the request, requests and retrieves the applicable archived product from the LTA, applies the requested processing and makes the output product available for download by the client.

2.2 Interface Configuration

Only entitled clients are allowed to request products from the PRIP. As part of every request to the PRIP, the client triggering the requests will identify themselves via *Explicit identification*.

The API shall support the following two methods of authentication:

- HTTP Basic Authentication
- OAuth 2.0

HTTP Basic Authentication represents a simple and easy method, consisting of a client placing the username and password inside the authorization header of the request. The username and password are encoded with base64, an encoding technique that converts the username and password into a set of 64 characters in order to



ensure safe transmission. It is mandatory to use Basic Authentication in conjunction with HTTPS in order to provide sufficient security to prevent unauthorized users from discovering the authentication information.

Although HTTP Basic Authentication should be supported for the purpose of testing and initial phases, it shall not represent the main authentication method used. Operationally, OAuth 2.0 is to be used (<https://tools.ietf.org/html/rfc6749>), as it is a more secure and powerful system, allowing for scalability of security. OAuth allows access tokens and refresh tokens to be issued to clients by an authorization server. The client then uses the access token to access the interface delivery point API. The OAuth flow / grant type, to be used is the Password Credentials grant: <https://tools.ietf.org/html/rfc6749#section-4.3>.

The definition of new clients on the PRIP side is done by configuration creating them with the necessary attributes. The attributes to be configured per client may be:

- **User Credentials** (username, password) for authentication
- **User Quota** – PRIPs requests are limited by a quota mechanism. Section 5.1.2 should be referred to for further details.

It is not intended to require an internal subsetting of products visible to any client (if such a need is identified then a possible model is outlined in the Common Entity Definition Document [AD-6]). Quota mechanisms are outlined in section 5.1.2 in order to avoid any misuse of the PRIP.

Clients download products from a PRIP using HTTPs protocols.

3 PRIP INTERFACE DESCRIPTION

This section provides detail on the specific interfaces between the PRIP client and the PRIP.

Where applicable, examples of message content are provided.

3.1 Nominal Use Scenario Description

The nominal use scenario for discovery and download of products from a PRIP is shown in the figure below.

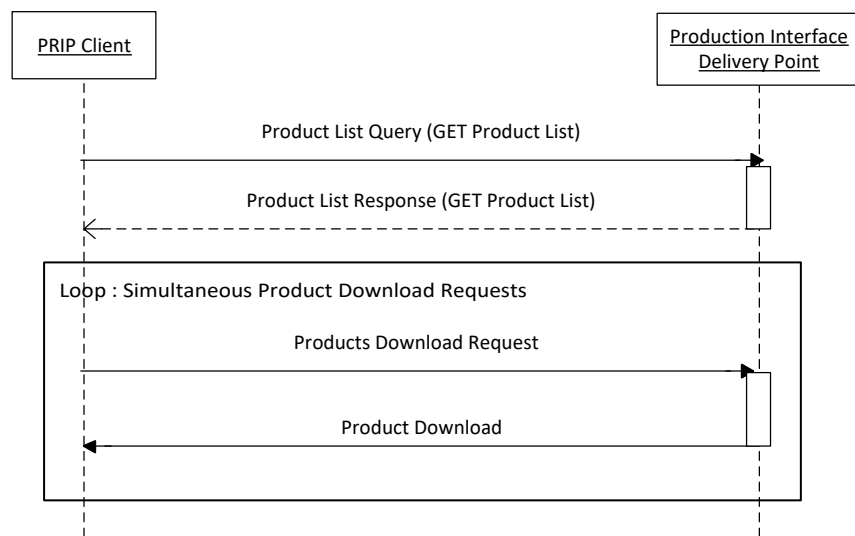


Figure 1: Nominal Product Retrieval Scenario from the Production Interface Delivery Point

- In the first step, the PRIP user uses the API to request a list of products currently available for download from the PRIP by sending a *GET Products List Query*. As PRIP users are in general systematic downloaders of all or a subset of Sentinel data products, these requests are sent regularly to ensure that a complete picture of all relevant published products is maintained at user-level. The query can include a filter by the content of the product name, and by publication time. Time range filtering examples include:
 - All products published after a certain date-time. Likely to be used for more regular requests, to discover the products published in the time interval since the previous request.
 - All products published in a certain date-time range (start/stop). For example, a client that prioritizes download of latest data, uses date-time ranges to download and fill gaps in the archive e.g. following an outage which took place over a known time period
 - All products (no time range).
- In response, the PRIP uses the *GET Products List Response* to provide the user with the requested list of products. For each product on the list, a limited set of product property metadata is provided. This metadata is intended to give the user a partial picture of each product to assist in retrieval. Fields will include: Product ID (logical file identifier), Product Name, Content Length (size of the downloadable file), Publication (Creation) Date, Eviction Date (time period for which data will remain on PRIP



before deletion), Checksums (algorithm, value and date) and Content Date (product sensing start and end times).

- Following receipt of the *GET Products List Response*, the PRIP user can optionally check if any of the products reported in the list have already been retrieved. Then, Product Download Requests are sent to the PRIP via the API for the required products. Products will be downloaded in the pre-defined packages for the particular mission and product type.

Requests are limited by a quota mechanism, e.g. limiting the user a maximum number of connections in parallel.

NB: The nominal scenario for on-demand requests to an ODPRIP is covered in Section 4.

3.2 Product Entity Model

The figure below shows the PRIP basic Product Entity model.

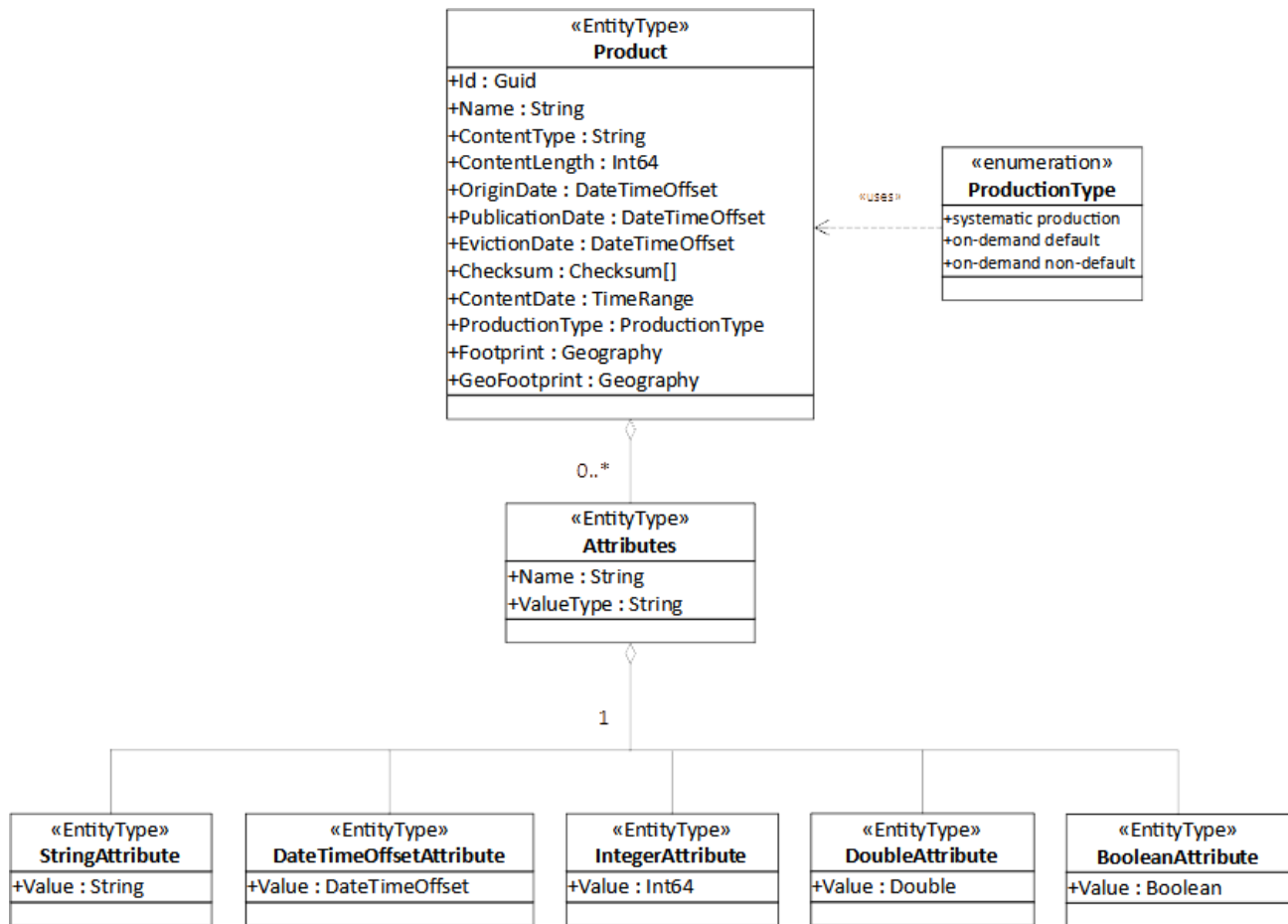


Figure 2: Product Entity Model

Product Properties	Type	M	Description	Example
Id	Guid ¹	Y	It is a universally unique identifier (UUID). The Id is a local identifier for the product instance within the PRIP, assigned by the Production system.	2b17b57d-fff4-4645-b539-91f305c27c69
Name	String	Y	Data file name	S1A_IW_SLC__1SDV_20160117T103451_20160117T103518_009533_00DD94_D46A.zip
ContentType	String	Y	The Mime type of the product	application/octet-stream

¹ For this, and for all other instances of the use of 'Guid' for IDs in this document, it should be noted that the IDs are UUIDs (Universally Unique Identifiers) of the (OData) Guid type.

ContentLength	Int64	Y	Actual size in bytes (B) of the downloadable file	4737286945
OriginDate	DateTimeOffset	Y	Timestamp recording when the entire downlinked raw data for the downlink session containing the data take is available at the acquisition interface point (XBIP and EDIP), and corresponds to the last DSIB availability date at the XBIP/EDIP	2018-01-17T14:16:03.788Z
PublicationDate	DateTimeOffset	Y	Publication date and time of the data file (time at which the file becomes visible to the user). Time is in UTC in the format YYYY-MM-DDThh:mm:ss.sssZ	2018-01-17T14:46:03.788Z
EvictionDate	DateTimeOffset	N	Date when the data file will be removed from the rolling archive. Time is in UTC in the format YYYY-MM-DDThh:mm:ss.sssZ The EvictionDate is optional but should be provided if a rolling policy is in place.	2018-01-22T18:00:00.000Z
Checksum (Algorithm, Value and ChecksumDate)	Checksum[]	Y	Represents the known checksums for the product's physical date, providing a unique value for supporting download integrity check. At least MD5 checksum, including ChecksumDate, is mandatory.	"Checksum": [{ "Algorithm": "MD5", "Value": "E8A303BF3D85200514F727DB60E7DB65" "ChecksumDate": "2018-01-22T18:00:00.000Z" }]
ContentDate	TimeRange	Y	The sensing range period. Start and end times are in UTC in the format YYYY-MM-DDThh:mm:ss.sssZ	"ContentDate": { "Start": "2016-01-17T10:34:51.922Z", "End": "2016-01-17T10:35:18.872Z" }
ProductionType	Production Type Enum	Y	ProductionType values: - <i>systematic_production</i> - <i>on-demand default</i> - <i>on-demand non-default</i> Where: - <i>systematic_production</i> means that standard systematic production has been applied - <i>on-demand default</i> means that the production is the result of an on-demand processing request	on-demand default

			<p>with default workflow options applied.</p> <p>- <i>on-demand non-default</i> means that the production is the result of an on-demand processing request with non-default workflow options applied.</p> <p>NB: while '<i>on-demand non-default</i>' signifies that options have been applied, it does not indicate precisely which options. The selected options will, however, be reflected within the attributes of the product.</p>	
Footprint²	Geography	N	Footprint of the product	<pre>geography'SRID=4326; Polygon((), (-41.15749 66.766701, -31.740927 67.629135, -31.479883 66.860405, -40.616844 66.011871, -41.15749 66.766701))'</pre>
GeoFootprint³	Geography	N	<p>Mandatory for georeferenced products, following the definition in RFC7946, with the following modification, according to [OData JSON Format Version 4.01 – Section 7.1]:</p> <ul style="list-style-type: none"> Keys should be ordered with type first, then coordinates, then any other keys <p>“Polygon” is used only as an example here, as it represents the most common type of GeoFootprint. However, the type will correspond to the type of footprint of the product, which can also be e.g. LineString, MultiLineString, etc.</p>	<pre>"GeoFootprint":{ "type": "Polygon", "coordinates": [[[-59.3169, 2.6367], [-63.105, -14.0539], [-60.8506, -14.4245], [-57.1309, 2.3269], [-59.3169, 2.6367]]] }</pre>

Table 1: Product Properties in GET Products List Response

² Will be deprecated in favour of GeoFootprint starting from TBC date

³ To be implemented starting from January 2023

Attribute Properties	Type	M	Description	Example
Name	String	Y	String name of the attribute	platformShortName
ValueType	String	Y	The type of attribute. This shall correspond to the following ⁴ : - String - Integer - DateTimeOffset - Boolean - Double	String
Value	Depends on ValueType		The value of the attribute, depending on the ValueType	SENTINEL-2

Table 2: Attributes Entity Description

As Attribute values will require different data types, they are based on derived entity types such as StringAttribute, DateTimeOffsetAttribute, IntegerAttribute etc (see Figure 2).

For the full list of applicable attributes, refer to [AD-1], [AD-2], [AD-3], [AD-4], [AD-5]. Note that in the case of the PRIP, the set of attributes necessary to be queryable will be outlined in the Extended Compliance Tests (to be specified in Annex 3), although all Attributes should be reported through the interface.

The response to an `$expand=Attributes` query shall include at least the attribute properties listed in Table 2.

Example of \$expand=Attributes Query:

```
https://<service-root-uri>/odata/v1/Products?$filter=contains(Name,'IW_RAW__ON') and
ContentDate/Start gt 2022-06-26T00:00:00.000Z and ContentDate/End lt 2022-06-
26T10:00:00.000Z&$top=1&$expand=Attributes
```

Example of \$expand=Attributes Query Response:

```
HTTP/1.1 200 OK
{
  "@odata.context": "$metadata#Products(Attributes())",
  "value": [
    {
      "Id": "06c91249-b275-4811-9a55-a8d11b534945",
      "Name":
        "S1A_IW_RAW__ONSH_20220626T050533_20220626T051038_043829_053B7F_203C.SAFE.zip",
      "EvictionDate": "2022-06-31T23:59:59.999Z",
      "OriginDate": "2022-06-26T06:14:55.468Z",
      "PublicationDate": "2022-06-26T06:30:34.558Z",
      "ContentLength": 47649797,
      "ContentType": "application/zip",
      "ContentDate": {
        "Start": "2022-06-26T05:05:33.863Z",
        "End": "2022-06-26T05:10:38.849Z"
      }
    }
  ],
}
```

⁴ Only these values shall be allowed (e.g. Edm.DateTimeOffset or DateTimeOffsetAttribute is not allowed)

```

    "Checksum": [
      {
        "Algorithm": "MD5",
        "Value": "235f8cac56705aee78b4c8bbb45c7604",
        "ChecksumDate": "2022-06-26T06:14:55.468Z"
      }
    ],
    "ProductionType": systematic_production,
    "Footprint": "geography'SRID=4326;POLYGON((-59.3169 2.6367,-63.105 -14.0539,-60.8506 -14.4245,-57.1309 2.3269,-59.3169 2.6367))",
    "GeoFootprint":{
      "type": "Polygon",
      "coordinates": [
        [
          [
            -59.3169,
            2.6367
          ],
          [
            -63.105,
            -14.0539
          ],
          [
            -60.8506,
            -14.4245
          ],
          [
            -57.1309,
            2.3269
          ],
          [
            -59.3169,
            2.6367
          ]
        ]
      ]
    },
    "Attributes": [
      {
        "Name": "polarisationChannels",
        "ValueType": "String",
        "Value": "HH"
      },
      {
        "Name": "productType",
        "ValueType": "String",
        "Value": "IW_RAW__ON"
      },
      {
        "Name": "cycleNumber",
        "ValueType": "Integer",
        "Value": 265
      },
      {
        "Name": "relativeOrbitNumber",
        "ValueType": "Integer",
        "Value": 43829
      },
      {
        "Name": "processingCenter",
        "ValueType": "String",
        "Value": "S1 Production Service-SERCO"
      }
    ]
  }

```

```

    {
      "Name": "completionTimeFromAscendingNode",
      "ValueType": "Double",
      "Value": 4646455.264
    },
    {
      "Name": "processingDate",
      "ValueType": "DateTimeOffset",
      "Value": "2022-06-26T06:11:22.535Z"
    },
    {
      "Name": "datatakeID",
      "ValueType": "Integer",
      "Value": 342911
    },
    {
      "Name": "orbitNumber",
      "ValueType": "Integer",
      "Value": 43829
    },
    {
      "Name": "productConsolidation",
      "ValueType": "String",
      "Value": "FULL"
    },
    {
      "Name": "endingDateTime",
      "ValueType": "DateTimeOffset",
      "Value": "2022-06-26T05:10:38.849Z"
    },
    {
      "Name": "qualityDataObjectID",
      "ValueType": "String",
      "Value": "measurementData"
    },
    {
      "Name": "coordinates",
      "ValueType": "String",
      "Value": "-59.3169 2.6367,-63.105 -14.0539,-60.8506 -14.4245,-57.1309
2.3269,-59.3169 2.6367'"
    },
    {
      "Name": "orbitDirection",
      "ValueType": "String",
      "Value": "ASCENDING"
    },
    {
      "Name": "sliceProductFlag",
      "ValueType": "Boolean",
      "Value": false
    },
    {
      "Name": "beginningDateTime",
      "ValueType": "DateTimeOffset",
      "Value": "2022-06-26T05:05:33.863Z"
    },
    {
      "Name": "instrumentConfigurationID",
      "ValueType": "Integer",
      "Value": 7
    },
    {
      "Name": "startTimeFromAscendingNode",

```

```

        "ValueType": "Double",
        "Value": 4341469.328
    },
    {
        "Name": "instrumentShortName",
        "ValueType": "String",
        "Value": "SAR"
    },
    {
        "Name": "productClass",
        "ValueType": "String",
        "Value": "N"
    },
    {
        "Name": "platformShortName",
        "ValueType": "String",
        "Value": "SENTINEL-1"
    },
    {
        "Name": "platformSerialIdentifier",
        "ValueType": "String",
        "Value": "A"
    }
  ]
}

```

3.3 Query Products Catalogue

The Query Products Catalogue function is achieved through standard APIs according to the Entity Model described in Section 3.3. By default the Products query are to be ordered by Publication Date, in an ascending order (e.g. any query for new products published since a previous query will find the list ordered from oldest to newest)

3.3.1 Query Products

The main query possibilities are outlined below, filter conditions may be combined.

3.3.1.1 Query by Name

The most basic way to query products within the PRIP is to perform queries based on the filename, using string functions.

Function	Description
contains	The contains function returns records with names containing a particular string at any position
endswith	The endswith function returns true if the first parameter string value ends with the second parameter string value, otherwise it returns false
startswith	The startswith function returns true if the first parameter string value starts with the second parameter string value, otherwise it returns false

The list of products starting with the filename “S1B_EW_” can be retrieved as follows:

```
https://<service-root-uri>/odata/v1/Products?$filter=startswith(Name, 'S1B_EW_')
```



The following query will return all products containing “S3B_SL_1_RBT”:

```
https://<service-root-uri>/odata/v1/Products?$filter=contains(Name, 'S3B_SL_1_RBT')
```

3.3.1.2 Query by Product Origin Date

The list of products by OriginDate (i.e. time at which the products are available at the acquisition interface delivery point):

```
https://<service-root-uri>/odata/v1/Products?$filter=OriginDate gt 2020-05-15T00:00:00.000Z
```

3.3.1.3 Query by Product Publication Date

The list of products resulting from the systematic production and published in the PRIP since a reference time:

```
https://<service-root-uri>/odata/v1/Products?$filter=PublicationDate gt 2017-05-15T00:00:00.000Z and ProductionType eq OData.CSC.ProductionType'systematic_production'
```

3.3.1.4 Query by Sensing Date

The list of products filtered by sensing date criteria can be retrieved for example as follows:

```
https://<service-root-uri>/odata/v1/Products?$filter=ContentDate/Start gt 2019-05-15T00:00:00.000Z and ContentDate/End lt 2019-05-16T00:00:00.000Z
```

3.3.1.5 Query by Geographic Criteria

The list of products filtered by geographical criteria can be retrieved as follows:

```
https://<service-root-uri>/odata/v1/Products?$filter=OData.CSC.Intersects(area=geography'SRID=4326;POLYGON((-127.89734578345 45.234534534,-127.89734578345 45.234534534,-127.89734578345 45.234534534,-127.89734578345 45.234534534))')
```

It must be noted that this is a non-native OData function. It is introduced as a custom function extension:

[http://docs.oasis-open.org/odata/odata/v4.01/odata-v4.01-part1-](http://docs.oasis-open.org/odata/odata/v4.01/odata-v4.01-part1-protocol.html#sec_ActionFunctionExtensibility)

[protocol.html#sec_ActionFunctionExtensibility](http://docs.oasis-open.org/odata/odata/v4.01/odata-v4.01-part1-protocol.html#sec_ActionFunctionExtensibility); with the following signature:

Edm.Boolean OData.CSC.Intersects(Edm.GeographyPolygon,Edm.GeographyPolygon).

3.3.1.6 Query by Attributes

It is important that each PRIP uses the same metadata element from the product. [AD-1], [AD-2], [AD-3], [AD-4], [AD-5] provide the definition of the minimum metadata that shall be indexed for each product, and their origin within the product.

The list of products associated with a particular attribute Name value can be obtained as follows:

```
https://<service-root-uri>/odata/v1/Products?$filter=Attributes/OData.CSC.ValueTypeAttribute/any(att:att/Name eq '[Attribute.Name]' and att/OData.CSC.ValueTypeAttribute/Value eq '[Attribute.Value]')
```



Where the *ValueTypeAttribute* is the defined type for the Named attribute, e.g. StringAttribute, DateTimeOffsetAttribute, IntegerAttribute.

Products can also be queried by using a combination of attributes as in the example below:

```
https://<service-root-  
uri>/odata/v1/Products?$filter=Attributes/odata.CSC.StringAttribute/any(att:att/Name eq  
'productType' and att/odata.CSC.StringAttribute/Value eq 'MSI_L1C_TL') and  
Attributes/odata.CSC.StringAttribute/any(att:att/Name eq 'processorVersion' and  
att/odata.CSC.StringAttribute/Value eq '02.08')
```

3.3.1.7 Additional Options

Query functions shall be supported on all properties of the Product according to the OData \$filter query option. Results ordering shall be supported according to the \$orderby option; result paging shall be supported according to the \$top, \$skip and \$count options, with the page size configurable to support at least 1000 returned results per page.

Additionally, the (OD)PRIP shall support the “and”, “or”, “not” and “in” operators. More details on these options are provided below.

\$orderby

The \$orderby system query option allows clients to request resources in either ascending order using asc or descending order using desc. If asc or desc not specified, then the resources will be ordered in ascending order. For example:

```
https://<service-root-uri>/odata/v1/Products?$orderby=PublicationDate desc
```

Will return a list of products ordered by PublicationDate in descending order.

\$top

The \$top system query option allows clients to specify the maximum (non-negative integer) number of items returned from a query. In the case of the PRIP, \$top shall accept a maximum value of at least 1000.

For example:

```
https://<service-root-uri>/odata/v1/Products$top=1000&$filter=startswith(Name, 'S2')
```

Will return the first 1000 Sentinel-2 products currently available on the PRIP.

\$skip

The \$skip system query option allows clients to specify a (non-negative integer) number of items excluded from the start of a query result, i.e. the query will start returning items from the provided integer+1.

For example:

```
https://<service-root-uri>/odata/v1/Products$skip=100&$filter=startswith(Name, 'S2')
```

Will return the all Sentinel-2 products currently available on the PRIP, excluding the first 100.

\$top and \$skip are often applied together; in this case \$skip is always applied first regardless of the order in which they appear in the query.



\$count

The \$count system query option allows clients to request a count of the matching resources included with the resources in the response: the number of the matching resources is returned as result.

The \$count query option is useful to know the number of entities which are identified by the resource path section of the URI after having applied some filters.

For example:

```
https://<service-root-uri>/odata/v1/Products$count=true&$filter=startswith(Name, 'S2')
```

Will return the number of Sentinel-2 products currently available on the PRIP.

Operators “and”, “or”, “not”, “in”

The “and” operator allows clients to combine different filter functions.

Example:

```
https://<service-root-uri>/odata/v1/Products?$filter=startswith(Name, 'S1') and  
endswith(Name, '.EOF.zip')
```

The “or” operator allows clients to apply different filter values on the same filter function. The “or” operator shall not be used on different functions.

Example:

```
https://<service-root-uri>/odata/v1/Products?$filter=contains(Name, 'S1_RAW__0S') or  
contains(Name, 'S2_RAW__0S')
```

The “not” operator allows clients to omit certain results from the query.

Example:

```
https://<service-root-uri>/odata/v1/Products?$filter=startswith(Name, 'S1') and not  
contains(Name, 'WV_SLC__1')
```

The “in” operator allows for a shorthand way of writing multiple “eq” expressions joined by “or”.

Example:

```
https://<service-root-  
uri>/odata/v1/Products?$filter=Attributes/OData.CSC.StringAttribute/any(att:att/Name eq  
'productType' and att/OData.CSC.StringAttribute/Value in ('AUX_WAV', 'AUX_ICE', 'AUX_WND'))
```

3.3.2 Query Products Response

The GET Products List response is an http response to the GET product request. As recommended by the OData specification, the PRIP OData service should supports responses JSON.

HTTP Response Envelope

The following overall HTTP status codes may be returned with the response:

- 200 OK: if the request is accepted and a response can be returned



- 400 Bad Request
- 401 Unauthorized: if the requesting user is unauthorised to submit a GET Products List query
- 404 Not Found
- 429 Too Many Requests: if a quota is exceeded (see section 5.1.2)
- 500 Internal Server Error

In the case where the query is accepted but no products are found (for example if nothing is found in the date range in the query) then the 200 OK code is returned with an empty response body, instead of returning the 404 Not Found code.

Products Properties

For each product element in the Products List, the set of properties provided is listed in Table 1 above. It is assumed that at the time of response, all products returned in a Request Products List Response are currently available online at the PRIP, i.e. EvictionDates will always be in the future.

Example (JSON format):

```
HTTP/1.1 200 OK
{
  "@odata.context": "$metadata#Products",
  "value": [
    {
      "@odata.context": "$metadata#Products",
      "@odata.mediaContentType": "application/octet-stream",
      "Id": "e872683a-ea5b-455d-bfb2-fb304cbfbacb",
      "Name": "S1A_EW_OCN_2SDH_20171231T194900_20171231T194913_019951_021F8C_5392.zip",
      "ContentType": "application/octet-stream",
      "ContentLength": 4063869137,
      "OriginDate": "2016-12-25T12:48:00.048Z",
      "PublicationDate": "2016-12-25T13:18:00.048Z",
      "EvictionDate": "2016-12-28T13:18:00.048Z",
      "Checksum": [
        {
          "Algorithm": "MD5",
          "Value": "E8A303BF3D85200514F727DB60E7DB65",
          "ChecksumDate": "2016-12-28T13:18:00.048Z"
        }
      ]
      "ContentDate": {
        "Start": "2016-01-17T10:34:51.922Z",
        "End": "2016-01-17T10:35:18.872Z"
      }
      "Footprint": "geography'SRID=4326;Polygon((-70.19146 -33.512962,-72.916611 -32.859627,-72.366188 -31.255634,-69.690193 -31.896481,-70.19146 -33.512962))'",
      "GeoFootprint": {
        "type": "Polygon",
        "coordinates": [
          [
            [-41.15749, 66.766701],
            [-31.740927, 67.629135],
            [-41.15749, 66.766701]
          ]
        ]
      }
    }
  ]
}
```



```

        [-31.479883, 66.860405],
        [-40.616844, 66.011871],
        [-41.15749, 66.766701]
    ]
}
    }
    "ProductionType": "systematic_production"
}
]
}

```

3.3.3 Catalogue Export

Catalogue data can be exported from the PRIP in a compact format for cross-reference with other GS elements. The elements included in the catalogue export are the product properties and attribute properties, and shall be presented in JSON format corresponding to the query.

```
Products?$expand=Attributes&$format=json
```

Recognising that this may be a resource consuming query it may be preferred to cache such results in sorted lists of JSON files as routinely updated snapshots e.g. per sensing date or per publication date., whereby exported files are organized in folders according to the following hierarchical structure:

- *[SSS]*: Satellite Platform short name, 3 characters (e.g. S1A, S2A, ...)
- *[YYYY]*: Sensing Year, 4 digits (e.g. 2014, 2015, ...)
- *[MM]*: Sensing Month, 2 digits (e.g. 01, 02, ...)

The month folder contains the set of exported files. There is a data file for each day of the month. Each data file lists the products having the same 'sensing start' date.

The exported file naming convention and format is:

<SSS>_<YYYYMMDD>_<SYS>_<PROV>_catalogue_<yyyymmddhhmmss>.json

where:

- <SSS> 3 characters corresponding to the satellite platform short name S1A, S1B, S2A etc.
- <YYYYMMDD> 8 digits corresponding to the sensing start date of all the products listed in the file
- <SYS> 3 characters corresponding to the Production system: PRO, AUX, POD
- <PROV> 4 characters corresponding to the Provider (e.g. WXYZ)
- <yyyymmddhhmmss> 14 digits corresponding to the year, month, day and UTC time of the catalogue snapshot. The products listed in the exported file are those that have been published in the PRIP up until this date.

The frequency of the snapshot should be e.g. nightly.

These lists shall be made available through a simple HTTPS file interface, and accessible by the clients with the same credentials as for the (OD)PRIP. The lists shall be maintained from the start of production, i.e. maintaining the record even when the data are removed from the PRIP due to the rolling period.

3.4 Product Download

Product download over the OData API is initiated using the 'Id' for each product returned in the GET Products List Response. The correct URI for the download of a single product is:

```
https://<service-root-uri>/odata/v1/Products(Id)/$value
```

Where *Id* is the UUID assigned per product.

The download is considered in the OData protocol through the /\$value URL (**Error! Reference source not found.** §11.2.3 “To address the media stream represented by a media entity, clients append /\$value to the resource path of the media entity URL. Services may redirect from this canonical URL to the source URL of the media stream.”).

The PRIP response supports the following HTTP status codes as a minimum:

- 200 OK
- 307 Temporary Redirect (if applicable)
- 400 Bad Request
- 404 Product not found or no longer available
- 401 Unauthorized
- 429 Too Many Requests: if the download quota is exceeded (see section 5.1.2)
- 500 Internal Server Error

The PRIP supports single unit downloads only. No component downloads of, for example, metadata, are assumed. However, http byte range requests on full product downloads are allowed, enabling pause/resume of downloads or the efficient split of large download into multiple chunks, and may be added as a Range Header to the Download Request, eg: "Range: bytes=0-1023"

3.5 Subscription

The Subscription function allows the request for automated notifications of future products entering the (OD)PRIP, according to a set of filter parameters supplied within the subscription request. Once the notification has been received, the client can request the download of the product.

The sequence diagram below illustrates the subscription process.

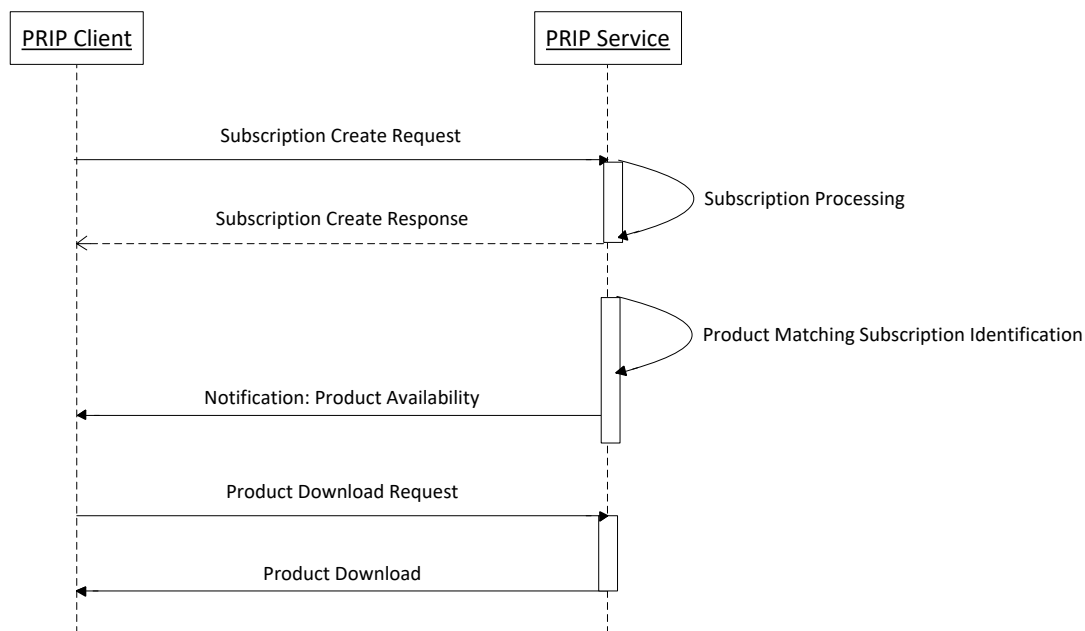


Figure 3: Subscription Nominal Sequence

The nominal scenario can be described in the following steps:

- **STEP 1 – Subscription Create**

The scenario is initiated by a PRIP client submitting a create subscription request to the PRIP service. The request contains the filter parameters of the subscription (product types, time range, footprint etc) and a NotificationEndpoint to which product download notifications will be sent. The request is processed by the PRIP Service, and a response is returned, which includes a unique identifier for the particular subscription request, a status (initially *running*), the filter parameters and the submission date/time of the request.

- **STEP 2 – Product Notification and Product Retrieval**

For each subscription with status *running*, the PRIP identifies new products matching the provided filter criteria. Upon identification, the PRIP sends a notification of product availability to the NotificationEndpoint provided, containing the product ID, name and ProductionType, the subscription ID and the notification date. The client is then free to submit a product Download Request to trigger the download of the product (see section 3.4).

- A subscription continues to identify products matching the filter parameters until it is paused or cancelled.
- In case the client endpoint is unavailable and the subscription notification fails, there is no retry.

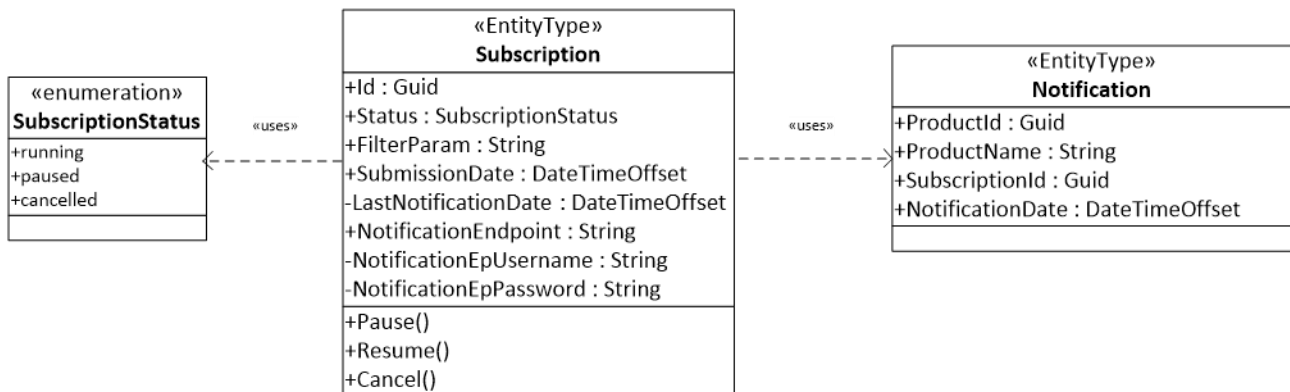


Figure 4: Subscription Model

Subscription Properties	Type	M	Description	Example
Id	Guid	Y	The Id is a local unique identifier for the Subscription instance within the PRIP, assigned upon Subscription creation.	2b17b57d-fff4-4645-b539-91f305c27e11
Status	Subscription Status enumeration	Y	SubscriptionStatus value (TBC): <ul style="list-style-type: none"> - <i>running</i> - <i>paused</i> - <i>cancelled</i> 	running
FilterParam	String	Y	The filter parameters of the Subscription (refers to the \$filter= parameter of any Products? query)	contains(Name, 'MSIL2A_') and PublicationDate gt '2019-07-01T00:00:00.000Z' and PublicationDate lt '2019-09-01T00:00:00.000Z'
SubmissionDate	DateTimeOffset	Y	Date and time at which the Subscription was received by the PRIP. Time is in UTC in the format YYYY-MM-DDThh:mm:ss.sssZ	2018-01-17T14:46:03.788Z
LastNotificationDate	DateTimeOffset	N	Date and time corresponding to the last time the Subscription was notified. Time is in UTC in the format YYYY-MM-DDThh:mm:ss.sssZ Only provided following the initial Subscription query.	2018-02-05T09:00:00.000Z
NotificationEndpoint	String	Y	URI used by the PRIP for subscription notifications	Any URI, e.g. https://myservice/notification
NotificationEpUsername	String	N	The username associated with the EndPoint URI provided	myserviceuser

			Mandatory if NotificationEndpoint requires authentication	
NotificationEpPassword	String	N	<p>The password associated with the EndPoint URI provided</p> <p>Mandatory if NotificationEndpoint requires authentication</p> <p>Shall be suitably secured to not be disclosed in subsequent queries via the OData interfaces.</p>	*****

Table 3: Subscription Entity Description

Notification Properties	Type	M	Description	Example
ProductId	Guid	Y	The unique identifier of the product being notified. Maps to Product.Id	2b17b57d-fff4-4645-b539-91f305c27c69
ProductName	String	Y	The data file name of the product being notified. Maps to Product.Name	S1A_IW_SLC__1SDV_20160117T103451_20160117T103518_009533_00DD94_D46A.zip
SubscriptionId	Guid	Y	The identifier of the Subscription being notified. Maps to Subscription.Id	2b17b57d-fff4-4645-b539-91f305c27e11
NotificationDate	DateTime Offset	Y	Date and time at which the notification was generated. Time is in UTC in the format YYYY-MM-DDThh:mm:ss.sssZ	2019-02-05T09:00:00.000Z

Table 4: Subscription Notification Entity Description

i-i Subscription Create: Example Request:

```
POST
http://<service-root-uri>/odata/v1/Subscriptions
{
  "@odata.context": "$metadata#Subscription/$entity",
  "FilterParam": "contains(Name, '_IW_SLC_') and PublicationDate gt 2021-07-01T00:00:00.000Z and PublicationDate lt 2021-07-02T00:00:00.000Z",
  "NotificationEndpoint": "https://myservice/notification",
  "NotificationEpUsername": "myserviceuser",
  "NotificationEpPassword": "*****"
}
```

i-ii Subscription Create: Example Response (JSON format):

```
HTTP/1.1 201 Created
{
  "@odata.context": "$metadata#OData.CSC.Subscription",
  "Id": "2b17b57d-fff4-4645-b539-91f305c27e11",
  "Status": "running",
  "FilterParam": "contains(Name, '_IW_SLC_') and PublicationDate gt 2021-07-01T00:00:00.000Z and PublicationDate lt 2021-07-02T00:00:00.000Z",
  "SubmissionDate": "2021-06-30T18:25:30.000Z",
  "NotificationEndpoint": "https://myservice/notification"
}
```

ii-i Example Subscription Notification:

```
POST
http://myservice/notification
{
  "@odata.context": "$metadata#Notification/$entity",
  "ProductId": "e872683a-ea5b-455d-bfb2-fb304cbfbacb",
  "ProductName":
  "S1B_IW_SLC__1SDV_20210701T235308_20210701T235335_003545_006104_0DD6",
  "SubscriptionId": "2b17b57d-fff4-4645-b539-91f305c27e11",
  "NotificationDate": "2021-07-02T00:01:09.000Z"
}
```

4 ON-DEMAND PROCESSING INTERFACE DESCRIPTION

The On-Demand Processing API allows a client to request the on-the-fly processing of a parent (input) product into a higher level (output) product. More specifically, this is achieved by creating a ProductionOrder entity which specifies the product to be used as input, the workflow to apply, and the options if any for customising the workflow.

In the context of the On-Demand Processing, a workflow corresponds to a specific version of a product processor.

The figure below shows the sequence by which an on On-Demand Processing request is made through a ProductionOrder and an output product is retrieved.

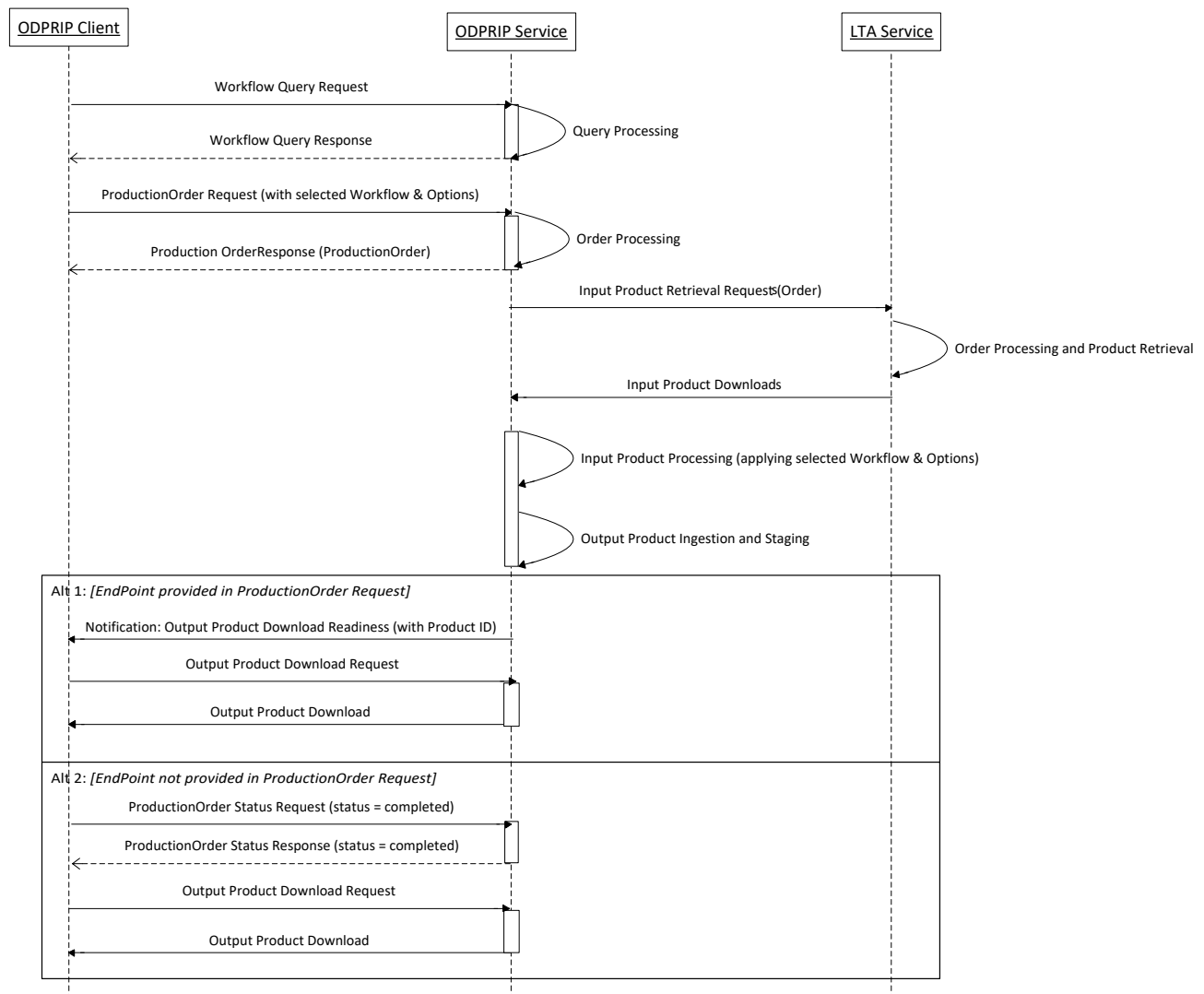


Figure 5: On-Demand Processing Request Nominal Sequence

The nominal scenario can be described in the following steps:

- **STEP 1: Workflow Query**

Following the initial identification of the input product to which on-demand processing is to be applied, the ODPRIIP Client may query the ODPRIIP Service for relevant Workflows to apply, if this is not already known. The query response provides information on the identified Workflows, including the unique identifier, name, textual description and input and output products types. A set of user options may also be included within a Workflow, which constrain how the requested processing may be modified and override the default options of the Workflow. The default set of options applied are also provided.

There are two basic scenarios in which on-demand processing may be applied. The first is that a product that has been previously produced and catalogued has been evicted from the data access service and has to be recreated. The second is that a higher level processing may be applied to a product available in the catalogue.

- **STEP 2: ProductionOrder Submission**

Having selected the input product and the Workflow to be triggered plus applicable options, the Client submits a request to create a ProductionOrder towards the ODPRIIP Service. As well as the parameters just listed, the ProductionOrder may also include an optional priority and an optional endpoint through which the Client can be notified about output product download readiness. Once received, the ProductionOrder is processed by the ODPRIIP Service and a response is returned which, in addition to the input parameters, includes a unique identifier for the particular ProductionOrder, a status (initially *queued*) and status message, a submission date and an estimated date (for completion). At this stage the user's quota is checked against the ProductionOrder: should insufficient remain the ProductionOrder will be rejected and the client will need to manage any later resubmission.

For the first definition of this interface it is assumed (and modelled) that a single ProductionOrder contains a single input product, which gives rise to a single output product and that a ProductionOrder identifies a single Workflow to be applied.

- **STEP 3: ProductionOrder Processing**

Following receipt of a ProductionOrder, the ODPRIIP Service generates a request for a product Order to retrieve the input product and sends to the LTA Service. The interfaces between the ODPRIIP and LTA, up to the download of the input product from the LTA Interface Delivery Point, are fully covered in the LTA ICD [RD-1] and hence are not described in detail in this document. At this point the status of the ProductionOrder changes to *in_progress*.

Following retrieval of the input product, the ODPRIIP applies the processing and options identified in the ProductionOrder and applicable workflow. The output product, once available, is then ingested and staged. The ProductionOrder entity is also updated at this stage to change the status to *completed* and to include the following additional properties: the output product name and identifier, the completed date, the eviction date and the order output size.

- **STEP 4: Output Product Retrieval**

If a notification endpoint was provided, the ODPRIP Service will send an Output Product Download Readiness Notification containing the Product ID of the output product. The Client is then free to submit a product Download Request. Alternatively, if no endpoint was provided, the Client polls the ODPRIP Service to determine the status of a ProductionRequest. Once the status has been updated to *completed*, the output product can be identified and can be downloaded as indicated in section 3.4.

The Client can use the *ProductionType* property within the product to identify whether non default processing options have been applied to an output product, and hence whether the product should be made available to all downstream users from the Data Access Service (in the case where default options are applied) or only to the requesting user via the Data Access Service (in the case where non default options have been applied).

4.1 On-Demand Entity Model and Properties

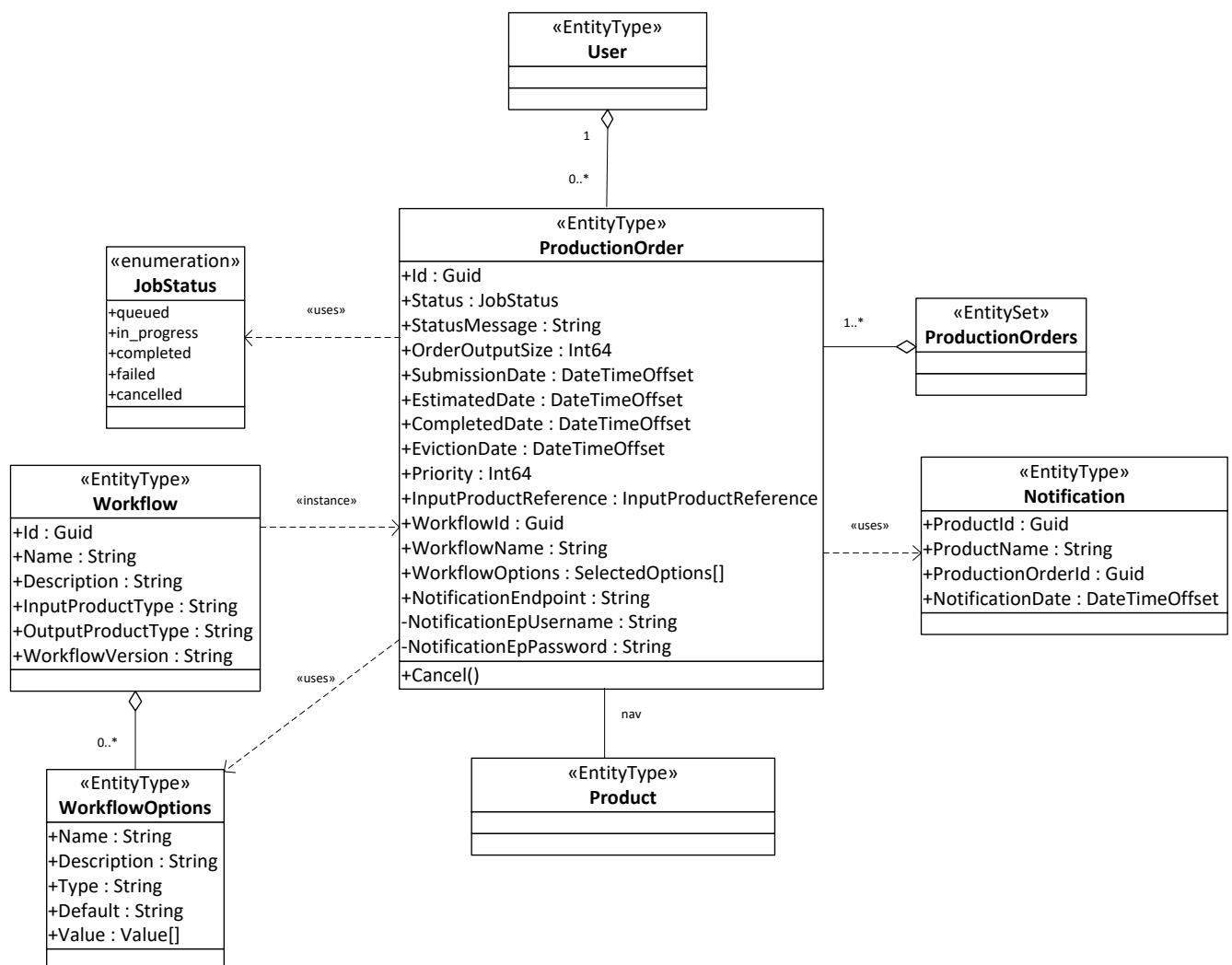


Figure 6: ProductionOrder and Workflow Model

Workflow Properties	Type	M	Description	Example
Id	Guid	Y	The Id is a local unique identifier for the Workflow instance within the ODPRIIP, assigned upon Workflow creation.	2b17b57d-ffff4-4645-b539-91f305c27c69
Name	String	Y	Short name of the workflow	S2_L1C_L2A
Description	String	Y	Textual description of the workflow, including details of the processor version and configuration applicable	Product processing from Sentinel-2 L1C to L2A. Processor V2.3.6
InputProductType	String	Y	Product Type of the Input Product	S2MSI1C
OutputProductType	String	Y	Product Type of the Output Product	S2MSI2A
WorkflowVersion	String	Y	Version number applicable to the workflow	1.2

Table 5: Workflow Entity Description

Each Workflow may include zero or more client-selectable WorkflowOptions. Note that it is assumed that all workflows will present an array with a list of available discrete Values. Only one of these values may be selected per WorkflowOption.

WorkflowOptions Properties	Type	M	Description	Example
Name	String	Y	The name of the option	"ImageFormat"
Description	String	Y	Textual description of the option	"Image Format option for the production output, GEOTIFF, JPG"
Type	String	Y	Type of the option	"String"
Default	String	N	The default value of the option, if there is one	"GEOTIFF"
Value	Value[]	Y	Array representing all possible values of the option: <ul style="list-style-type: none"> - Option 1 - Option 2 - Option 3 - etc 	"Value": ["GEOTIFF", "JPG"]

Table 6: WorkflowOptions Entity Description

ProductionOrder Properties	Type	M	Description	Example
Id	Guid	Y	The Id is a local unique identifier for the ProductionOrder instance within the ODPRIP, assigned upon ProductionOrder creation.	2b17b57d-fff4-4645-b539-91f305c27c69
Status	JobStatus enumeration	Y	<p>JobStatus value:</p> <ul style="list-style-type: none"> - <i>queued</i> - <i>in_progress</i> - <i>completed</i> - <i>failed</i> - <i>cancelled</i> <p>Where:</p> <ul style="list-style-type: none"> - <i>queued</i> means that the ordered production request is in the queue to be performed - <i>in_progress</i> means that the ordered production request is being performed - <i>completed</i> means that that the ordered output product has been produced is available for download - <i>failed</i> means the production of the ordered output product has failed - <i>cancelled</i> means that the ProductionOrder has been cancelled by the user (while in status <i>in_progress</i>) 	<i>in_progress</i>
StatusMessage	String	Y	<p>Text message providing additional information on the returned status, e.g. the reason for a failure. Example values are:</p> <p><i>If status = 'queued':</i> ‘request is queued for processing’</p> <p><i>If status = 'in_progress':</i> ‘request is under processing’</p> <p><i>If status = 'completed':</i> ‘requested output product is available’</p> <p><i>If status = 'failed':</i> ‘production has failed’ ‘input product currently unavailable’ ‘input product not found on LTA’</p> <p><i>If status = cancelled:</i></p>	requested output product is available

			'request cancelled by user'	
OrderOutputSize	Int64	N	Actual size in bytes (B) of the output Product composing the Order. Only provided if the ProductionOrder is in status <i>completed</i>	4737286945
SubmissionDate	DateTime Offset	Y	Date and time at which the ProductionOrder was received by the ODPRIP. Time is in UTC in the format YYYY-MM-DDThh:mm:ss.sssZ	2018-01-17T14:46:03.788Z
EstimatedDate	DateTime Offset	N	Estimated date and time when the product will be available for download from the ODPRIP. Time is in UTC in the format YYYY-MM-DDThh:mm:ss.sssZ The EstimatedDate is assigned upon initial ProductionOrder response and is not subsequently updated. The EstimatedDate is assigned on the basis of the queue when the ProductionOrder is processed (the actual CompletedDate could be impacted by subsequent incoming higher priority orders)	2018-01-22T18:00:00.000Z
CompletedDate	DateTime Offset	N	Date and time when the product was available for download from the ODPRIP. Time is in UTC in the format YYYY-MM-DDThh:mm:ss.sssZ Only provided if the ProductionOrder is in status <i>completed</i>	2018-01-22T18:00:00.000Z
EvictionDate	DateTime Offset	N	Date when the Product related to the order will be removed from the ODPRIP. Time is in UTC in the format YYYY-MM-DDThh:mm:ss.sssZ Only provided if the ProductionOrder is in status <i>completed</i>	2018-01-22T18:00:00.000Z
Priority	Int64	Y	Priority of the ProductionOrder. It is an integer from 1-100, where 100 is the highest priority. There is always a priority associated to a ProductionOrder: if	50

			<p>it is not set within the ProductionOrder Request, then it is automatically set to the default priority assigned to the client. If the ProductionOrder Request sets a priority higher than the highest priority assigned to the client then it is automatically set at the highest priority assigned.</p> <p>Production orders are processed in order of their priority. Some clients e.g. Copernicus Services may have a higher assigned priority than others.</p>	
InputProductReference (Reference, ContentDate)	InputProductReference	Y	<p>Complex type used to unambiguously identify the input product. It contains:</p> <ul style="list-style-type: none"> - <i>Reference</i> (String): Identifier of the product, typically the product file name or other similar value - <i>ContentDate</i> (TimeRange): The sensing range period. Start and end times are in UTC in the format YYYY-MM-DDThh:mm:ss.sssZ <p>At least one of the <i>Reference</i> or the <i>ContentDate</i> must be included.</p> <p>The optionality and precise definition of the <i>InputProductReference</i> should be described in an explicit tailoring, see for example Annex A.</p>	<pre>"InputProductReference": { "Reference": " S1A_IW_SLC_1SDV_ 20160117T103451_ 20160117T103518_009533_ 00DD94_D46A.zip", "ContentDate": { "Start": "2016-01- 17T10:34:51.922Z", "End": "2016-01- 17T10:35:18.872Z" } }</pre>
WorkflowId	Guid	Y	<p>The Id is a local unique identifier for the Workflow instance within the ODPRIP which is applicable to the ProductionOrder</p> <p>Maps to Workflow.Id</p>	2b17b57d-fff4-4645-b539-91f305c27c69
WorkflowName	String	Y	<p>Short name of the workflow</p> <p>Maps to Workflow.Name</p>	S2_L1C_L2A
WorkflowOptions (Name, Value)	WorkflowOptions[]		<p>Selection of applicable options from the Workflow. Each option is provided in the format (taken from the WorkflowOptions entity):</p>	<pre>WorkflowOptions: [{ "Name": "ImageFormat", "Value": "GEOTIFF" }, </pre>

			<i>Name:Value</i> If no options are selected, the default value is applied in each case.	<pre>{ "Name": "Resolution", "Value": "60" }</pre>
NotificationEndpoint	String	N	URI used by the ODPRIP for product download readiness notifications, should these be required. If not provided, no notifications will be sent.	<code>https://myservice/notification</code>
NotificationEndpointUsername	String	N	The username associated with the Endpoint URI provided. Mandatory if NotificationEndpoint requires authentication	<code>myserviceuser</code>
NotificationEndpointPassword	String	N	The password associated with the Endpoint URI provided. Mandatory if NotificationEndpoint requires authentication Shall be suitably secured to not be disclosed in subsequent queries via the OData interfaces.	<code>*****</code>

Table 7: ProductionOrder Entity Description

Notification Properties	Type	M	Description	Example
ProductId	Guid	Y	The unique identifier of the product being notified. Maps to Product.Id	2b17b57d-fff4-4645-b539-91f305c27c69
ProductName	String	Y	The data file name of the product being notified. Maps to Product.Name	S1A_IW_SLC__1SDV_20160117T103451_20160117T103518_009533_00DD94_D46A.zip
ProductionOrderId	Guid	Y	The identifier of the ProductionOrder being notified. Maps to ProductionOrder.Id	2b17b57d-fff4-4645-b539-91f305c27e11
NotificationDate	DateTime Offset	Y	Date and time at which the notification was generated. Time is in UTC in the format YYYY-MM-DDThh:mm:ss.sssZ	2019-02-05T09:00:00.000Z

Table 8: ProductionOrder Output Product Notification Entity Description

4.2 Workflow and ProductionOrder Examples

4.2.1 Step 1 – Workflow Query

i-i Example Request:

The following query shows the request for, as an example, the list of processing services (Workflows) supported for a given type of parent (input) product, in this case Sentinel2- ‘S2MSI1C’ products. It ‘expands’ the result to show the WorkflowOptions.

```
GET
http://<service-root-uri>/odata/v1/Workflows?filter=InputProductType eq
'S2MSI1C'&expand=WorkflowOptions
```

i-ii Example Response (JSON format):

In this example response, a single workflow containing a single option is returned:

```
HTTP/1.1 200 OK
{
  "@odata.context": "$metadata#Workflows/$entity",
  "value":
  [
    {
      "Id": "2b17b57d-fff4-4645-b539-91f305c83z91",
      "Name": "S2_L1C_L2A",
      "Description": "Product processing from Sentinel-2 L1C to L2A. Processor
V2.3.6",
      "InputProductType": "S2MSILC",
      "OutputProductType": "S2MSI2A",
      "WorkflowVersion": "1.2",
      "WorkflowOptions":
      [
        {
          "Name": "ImageFormat",
          "Description": "Image Format option for the production output: GEOTIFF, JPG",
          "Type": "String",
          "Default": "GEOTIFF",
          "Value":
          [
            "GEOTIFF",
            "JPG"
          ]
        }
      ]
    }
  ]
}
```

4.2.2 Step 2 – ProductionOrder Submission

ii-i Example ProductionOrder Request:

The following example request shows the Client requesting creation of a ProductionOrder via the OData.CSC.Order method. The mandatory InputProductReference is supplied, as well as the WorkflowId, WorkflowName and WorkflowOptions and Priority fields (default values are used if the latter two fields are not supplied).

```
POST
"http://<service-root-uri>/odata/v1/ProductionOrder/OData.CSC.Order"
{
  "InputProductReference":
  {
    "Reference": "S2B_MSIL1C_20191025T085939_N0208_R007_T37VCC_20191025T112031.zip",
    "ContentDate":
    {
      "Start": "2016-01-17T10:34:51.922Z",
      "End": "2016-01-17T10:35:18.872Z"
    }
  }
  "Priority": 50,
  "WorkflowId": "6c18b57d-fgk4-1236-b539-12h305c26z89",
  "WorkflowName": "S2_L1C_L2A",
  "WorkflowOptions":
  [
    {
      "Name": "ImageFormat",
      "Value": "JPG"
    }
  ]
  "NotificationEndpoint": "https://myservice/notification",
  "NotificationEpUsername": "myserviceuser",
  "NotificationEpPassword": "*****"
}
```

ii-ii Example ProductionOrder Submission Response:

The following response is an example of what would be sent in the case that the ProductionOrder request in ii-i is successfully accepted. The response describes the newly created ProductionOrder:

```
HTTP/1.1 201 Created
{
  "@odata.context": "$metadata#OData.CSC.Order",
  "value":
  [
    {
      "Id": "2b17b57d-fff4-4645-b539-91f305c26x53",
      "Status": "queued",
      "StatusMessage": "request is queued for processing",
      "SubmissionDate": "2019-03-27T13:55:12.390Z",
      "EstimatedDate": "2019-03-27T14:02:51.390Z",
    }
  ]
}
```



```

"Priority": 50,
"InputProductReference":
{
  "Reference": "S2B_MSIL1C_20191025T085939_N0208_R007_T37VCC_20191025T112031.zip",
  "ContentDate":
  {
    "Start": "2016-01-17T10:34:51.922Z",
    "End": "2016-01-17T10:35:18:872Z"
  }
}
"WorkflowId": "6c18b57d-fgk4-1236-b539-12h305c26z89",
"WorkflowName": "S2_L1C_L2A",
"WorkflowOptions":
[
  {
    "Name": "ImageFormat",
    "Value": "JPG"
  }
]
"NotificationEndpoint": "https://myservice/notification"
]
}

```

ii-iii ProductionOrder Retrieval Request Examples:

This request retrieves a particular ProductionOrder by its Id. The response is a single ProductionOrder.

```

GET
https://<service-root-uri>/odata/v1/ProductionOrders(Id)

```

This request retrieves all current ProductionOrders submitted by the Client. The response is an array of ProductionOrders.

```

GET
https://<service-root-uri>/odata/v1/ProductionOrders

```

This request retrieves all current ProductionOrders whose status is '*completed*'. This request can be used to identify output products ready for download.

```

GET
https://<service-root-uri>/odata/v1/ProductionOrders?$filter=Status eq
OData.CSC.JobStatus'completed'

```

ii-iv ProductionOrder Retrieval Request – Response for a Completed ProductionOrder

The following shows the ProductionOrder response returned for a status=*'completed'*.

```

HTTP/1.1 200 OK

```

```
{
"@odata.context": "$metadata#ProductionOrder/$entity",
"value":
[
  {
    "Id": "2b17b57d-fff4-4645-b539-91f305c26x53",
    "Status": "completed",
    "StatusMessage": "requested output product is available",
    "OrderOutputSize": 23444841,
    "SubmissionDate": "2019-03-27T13:55:12.390Z",
    "EstimatedDate": "2019-03-27T14:02:51.390Z",
    "CompletedDate": "2019-03-27T14:12:12.400Z",
    "EvictionDate": "2019-04-01T14:12:12.400Z",
    "Priority": 50,
    "InputProductReference":
    {
      "Reference": "S2B_MSIL1C_20191025T085939_N0208_R007_T37VCC_20191025T112031.zip",
      "ContentDate":
      {
        "Start": "2016-01-17T10:34:51.922Z",
        "End": "2016-01-17T10:35:18.872Z"
      }
    }
  }
]
"WorkflowId": "6c18b57d-fgk4-1236-b539-12h305c26z89",
"WorkflowName": "S2_L1C_L2A",
"WorkflowOptions":
[
  {
    "Name": "ImageFormat",
    "Value": "JPG"
  }
]
"NotificationEndpoint": "https://myservice/notification"
}
}
```

And thus the product download details are available according to the navigation link:

```
GET
https://<service-root-uri>/odata/v1/ProductionOrder(2b17b57d-fff4-4645-b539-91f305c26x53)/Product
```

And the download may be triggered accordingly:

```
GET
https://<service-root-uri>/odata/v1/ProductionOrder(2b17b57d-fff4-4645-b539-91f305c26x53)/Product/$value
```

ii-v ProductionOrder Output Product Notification

The example below shows a Notification, sent when an output product related to a ProductionOrder is available for download (i.e. the ProductionOrder is in status = *'completed'*).



POST

http://myservice/notification

```
{
  "@odata.context": "$metadata#Notification/$entity",
  "ProductId": "e872683a-ea5b-455d-bfb2-fb304cbfbacb",
  "ProductName": "S1B_IW_SLC__1SDV_20161224T235308_20161224T235335_003545_006104_ODD6",
  "ProductionOrderId": "9e17b57d-fff4-4645-b539-91f305c76g22",
  "NotificationDate": "2019-05-30T22:18:09.000Z"
}
```

5 CLIENT ADMINISTRATION

The PRIP service enforces that only properly authenticated clients are accepted and that the access rights are applied.

If a client calls a PRIP service function without permission, it is informed with a corresponding return message.

5.1 User⁵ Entity Model

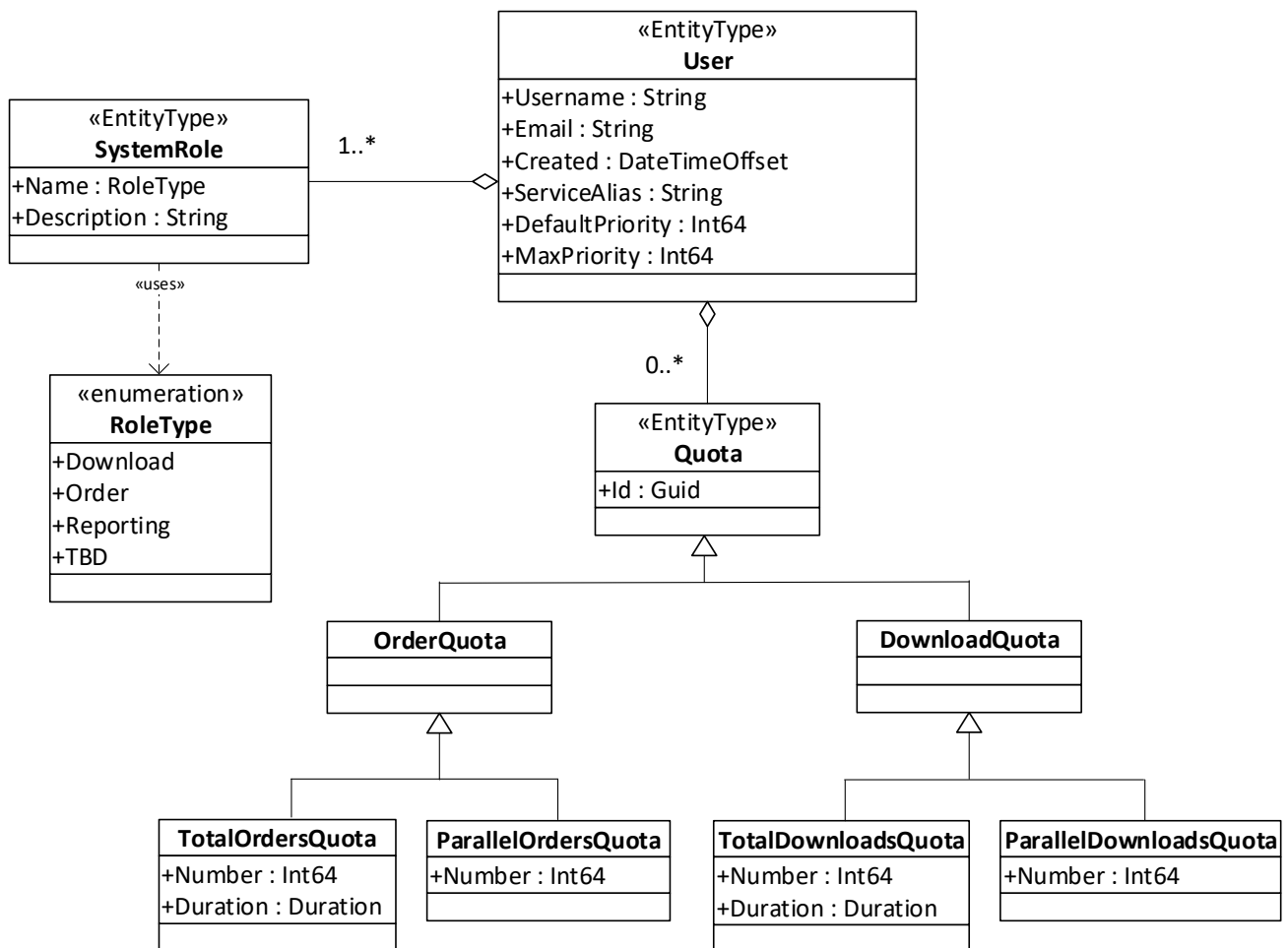


Figure 7: User Entity Model

The User entity is an entity that allows to define the system role of the client and associate any quota restrictions. The model describes the minimum properties that shall be used to control user access, according to the business agreement. The User is identified by the key property Username. The properties assigned to each User entity are described in the table below.

⁵ In this section, the User entity represents the Client

User Properties	Type	M	Description	Example
Username (key)	String	Y	Username of the User	DataDissem4CollGSHub
Email	String	Y	Email of the User	datahubops@xyz.com
Created	DateTimeOffset	Y	Date/time of user creation in the (OD)PRIP. Time is in UTC in the format YYYY-MM-DDThh:mm:ss.sssZ	2019-02-22T18:15:23.000Z
ServiceAlias	String	N	Generic string to describe the client	DataAccessColhub
DefaultPriority	Int64	N	The default order priority assigned to the User. It is an integer from 1-100 (TBC), where 100 is the highest priority. Mandatory if the user has the 'Order' SystemRole.	50
MaxPriority	Int64	N	The maximum priority which can be assigned to an order by the User. It is an integer from 1-100 (TBC), where 100 is the highest priority. It must be equal to or higher than the DefaultPriority Mandatory if the user has the 'Order' SystemRole.	100
OrderQuota	Int64	N	Refers to the maximum number of orders for allowed to be processed in parallel (i.e. in status in_progress) which a user may have ongoing at a particular time.	20
DownloadQuota	Int64	N	Refers to the maximum number of downloads that may be performed in parallel by the user.	20

Table 9: User Entity Description

Usernames associated email are to be managed as generic indicators of client and not personal information. The ServiceAlias value can be identical to the Username, unless indicated otherwise by the Agency.

The SystemRole and Quotas sub-entities, assigned for registered Users, are described in the sections below.

5.1.1 SystemRole

The SystemRole entity describes the rights and capabilities granted to a client. The properties of the SystemRole entity are described in the table below:

SystemRole Properties	Type	M	Description	Example
Name (key)	RoleType enumeration	Y	Assignment of the rights and capabilities granted to a user, including:	Download

			<ul style="list-style-type: none"> - Download - Order - Reporting - TBD 	
Description	String	Y	Textual description of the role	Allowing the user to perform queries and make downloads

Table 10: SystemRole Entity Description

All registered clients must have one or more roles assigned. The roles available for assignment to clients may be:

Client Services

- **Download** – The client may perform queries on the products available at the PRIP and perform downloads
- **Order** – as above, but additionally the user can order products via the ODPRIP interfaces

Administration Services

- **(Monitoring &) Reporting** – Monitoring and Reporting functionality, including the permission to perform queries on all properties.

5.1.2 Quota

User quota is assigned on a per user basis. Quota is assigned and applied to downloads for all users, but may also be separately assigned applied to ProductionOrders for users with the ‘Order’ SystemRole. Therefore, users may be assigned the OrderQuota’ and ‘DownloadQuota’, as shown below. ‘OrderQuota’ and ‘DownloadQuota’ are not linked: they are applied separately and independently at the order and download stages respectively. The quotas are not mutually exclusive and multiple or none may be assigned.

Order Quota (applicable to users with either ‘Download’ or ‘Order’ SystemRoles)

- *TotalOrdersQuota*: Maximum number of authorized ProductionOrders which can be made in a defined period
- *ParallelOrdersQuota*: Maximum number of separate ProductionOrders with status ‘in_progress’ which can be running in parallel.

Download Quota (only applicable to users with the ‘Download’ SystemRole)

- *TotalDownloadsQuota*: Maximum volume (GB) transfer within a defined period
- *ParallelDownloadsQuota*: Maximum number of separate downloads which can be performed in parallel

Should the quota be exceeded the 429 *Too Many Requests* code will be returned and the client must manage resubmission of the ProductionOrder or download request at a later date.

NB: if a ProductionOrder is still in *queued* status, it may be cancelled without impact on the order quota.

6 MONITORING AND REPORTING

The Monitoring and Reporting role allows for the collection of data, for analysis and reporting, of the Production system activity through API queries on all Products, Production Orders (On-Demand Production) and Subscriptions managed by the system. When used for the reporting function the API allows the collection of data concerning the overall use of the system by registered users; used for the monitoring function the API allows to regularly collect information concerning indicators which may be used to assess whether the routine operation of the system is running within the expected bounds.

The parameters obtained for Monitoring and Reporting through the API, with example requests, are detailed in the sections below.

6.1.1 Reporting

The table below provides a possible schema for routine collection of reporting information, example queries are developed further. The time period is indicative, depending on the strategy for end-to-end monitoring and reporting.

System Entity	Reporting Statistic	Time Period	Entity Parameters Collected
Products	Products generated and published	10 minutes	<ul style="list-style-type: none"> - Count - Name - ContentLength (Product size) - OriginDate - PublicationDate - ContentDate/Start - ProductionType
ProductionOrder (On-Demand Production only)	Production Orders submitted	10 minutes	<ul style="list-style-type: none"> - Count - Username - Id (ProductionOrder) - SubmissionDate - InputProductReference - WorkflowName
	Production Orders completed and time for completion	10 minutes	<ul style="list-style-type: none"> - Count - Username - Id (ProductionOrder) - OrderOutputSize - SubmissionDate - CompletedDate - InputProductReference - WorkflowName - ProductName
	Production Orders failed	10 minutes	<ul style="list-style-type: none"> - Count - Username - Id (ProductionOrder) - SubmissionDate - InputProductReference

			- WorkflowName
	Production Orders cancelled	10 minutes	- Count - Username - Id (ProductionOrder) - SubmissionDate - InputProductReference - WorkflowName
	Production Orders pending for over 1 day	10 minutes	- Count - Username - Id (ProductionOrder) - SubmissionDate - InputProductReference - WorkflowName
Subscriptions	Subscription in status <i>running</i> (total)	10 minutes	- Count
	Subscriptions submitted (daily)	10 minutes	- Count - Username - Id (Subscription)

Table 11: Production Reporting Statistics

Examples of the queries used to retrieve the described statistics are presented below. Descriptions are limited to a single entity and can straightforwardly be modified to the other entities reported on where necessary.

6.1.1.1 Systematic and On-Demand Production Reporting

i Products published per day

The \$count notation is used to return the total products published in a particular day, and the \$select notation to select the product properties to be returned.

i-i Request

```
https://<service-root-uri>/odata/v1/Products?$count=true&$filter=PublicationDate gt
2019-01-17T00:00:00.000Z and PublicationDate lt 2019-01-
18T00:00:00.000Z&$select=Name,ContentLength,OriginDate,PublicationDate,ContentDate/Start,
ProductionType
```

i-ii Response (JSON format)

In the example, the count of products returned has been limited to two.

```
HTTP/1.1 200 OK
{
  "@odata.context": "$metadata#Products/$entity",
  "@odata.count": 2,
  "value":
  [
    {
      "@odata.mediaContentType": "application/octet-stream",
      "Name": "S1B_IW_SLC__1SDV_20161224T235308_20161224T235335_003545_006104_0DD6",
      "ContentType": "application/octet-stream",
```



```

"ContentLength": 4063869137,
"OriginDate": "2019-01-17T08:48:00.048Z",
"PublicationDate": "2019-01-17T09:18:00.048Z",
"ContentDate":
{
  "Start": "2019-01-17T05:30:00.060Z",
}
"ProductionType": "systematic_production"
}
{
"@odata.mediaContentType": "application/octet-stream",
"Name": "S1B_IW_SLC__1SDV_20191224T235516_20191224T235892_003545_006104_0DP9",
"ContentType": "application/octet-stream",
"ContentLength": 4423569265,
"OriginDate": "2019-01-17T09:45:00.048Z",
"PublicationDate": "2019-01-17T10:25:00.066Z",
"ContentDate":
{
  "Start": "2019-01-17T08:34:51.922Z",
}
"ProductionType": "systematic_production"
}
]
}

```

6.1.1.2 On-Demand Production Reporting

i Production Orders Submitted per day

The \$expand notation is used to return the required properties of the Product and User entities related to each ProductionOrder matching the filter criteria.

i-i Request

```

https://<service-root-uri>/odata/v1/ProductionOrders?$count=true&$filter=SubmissionDate
gt 2019-01-17T00:00:00.000Z and SubmissionDate lt 2019-01-18T00:00:00.000Z&$select=Id,
SubmissionDate,InputProductReference,WorkflowName&$expand=Users($select=Username)

```

i--ii Response (JSON format)

```

HTTP/1.1 200 OK
{
  "@odata.context": "$metadata#ProductionOrders/$entity",
  "@odata.count": 1,
  "value":
  [
    {
      "@odata.context": "$metadata#Odata.CSC.Order",
      "Id": "2b17b57d-fff4-4645-b539-91f305c27c43",
      "SubmissionDate": "2019-01-17T13:55:12.390Z",
      "InputProductReference":
      [
        {

```

```

"Reference":
  "S2B_OPER_MSI_L1C_TL_MPS__20191001T112642_A013418_T34UGA_N02.08.tar",
"ContentDate":
  {
    "Start": "2019-10-01T09:23:08.000200Z",
    "End": "2019-10-01T09:26:30.000235Z"
  }
]
}
"WorkflowName": "S2_L1C_L2A",
"Users":
[
  {
    "@odata.mediaContentType": "application/octet-stream",
    "Username": "jroberts12"
  }
]
}
]
}

```

ii Orders in status ‘completed’ per day and time of completion

ii-i Request

```

https://<service-root-uri>/odata/v1/ProductionOrders?$count=true&$filter=CompletedDate gt
2019-01-17T00:00:00.000Z and CompletedDate lt 2019-01-18T00:00:00.000Z
&$select=Id,OrderOutputSize,SubmissionDate,CompletedDate,InputProductReference,WorkflowNa
me&$expand=Products($select=Name),Users($select=Username)

```

ii-ii Response (JSON format)

```

HTTP/1.1 200 OK
{
  "@odata.context": "$metadata#ProductionOrders/$entity",
  "@odata.count": 1,
  "value":
  [
    {
      "@odata.context": "$metadata#OData.CSC.Order",
      "Id": "2b17b57d-f ff4-4645-b539-91f305c27c12",
      "OrderOutputSize": "4737286945",
      "SubmissionDate": "2019-01-17T13:55:12.390Z",
      "CompletedDate": "2019-01-17T15:02:22.561Z",
      "InputProductReference":
      {
        "Reference":
        "S2B_OPER_MSI_L1C_TL_MPS__20191001T112642_A013418_T34UGA_N02.08.tar",
        "ContentDate":
        {
          "Start": "2019-10-01T09:23:08.000200Z",
          "End": "2019-10-01T09:26:30.000235Z"
        }
      }
      "WorkflowName": "S2_L1C_L2A",
      "Products":
      [
        {

```

```

"@odata.mediaContentType": "application/octet-stream",
"Name": "S2B_OPER_MSI_L2A_TL_MPS__20191001T112642_A013418_T34UGA_N02.08"
}
]
"Users":
[
{
"@odata.mediaContentType": "application/octet-stream",
"Username": "jroberts12"
}
]
}
]
}

```

iii Orders Pending for over 1 Day

Checks for all Production Orders which have a Submission date prior to the start of the reporting query day and a status still *in_progress*.

iii-i Request

```

https://<service-root-uri>/odata/v1/ProductionOrders?$count=true&$filter=SubmissionDate
lt 2019-01-17T00:00:00.000Z and Status eq
OData.CSC.JobStatus'in_progress'&$select=Id,SubmissionDate,InputProductReference,Workflow
Name&$expand=Users($select=Username)

```

iii-ii Response

```

HTTP/1.1 200 OK
{
  "@odata.context": "$metadata#ProductionOrders/$entity",
  "@odata.count": 1,
  "value":
  [
    {
      "@odata.context": "$metadata#OData.CSC.Order",
      "Id": "2b17b57d-fff4-4645-b539-91f305c27c99",
      "SubmissionDate": "2019-01-15T22:25:33.491Z",
      "InputProductReference":
      {
        "Reference":
        "S2B_OPER_MSI_L1C_TL_MPS__20191001T112642_A013418_T34UGA_N02.08.tar",
        "ContentDate":
        {
          "Start": "2019-10-01T09:23:08.000200Z",
          "End": "2019-10-01T09:26:30.000235Z"
        }
      }
    }
  ],
  "WorkflowName": "S2_L1C_L2A",
  "Users":
  [
    {
      "@odata.mediaContentType": " application/octet-stream",

```

```

    "Username": "jroberts12"
  }
]
}
]
}

```

Download Reporting

Due to the flexibility inherent with the download via byte ranges the download reporting may be managed through the summary reporting outlined below.

6.1.2 Production Metrics

The high-level statistics listed below are typical of those which will be made available by the Production Service towards a dashboard, for example, summarizing the current status of the system and its operations. Since such statistics would be relatively intensive to calculate on demand it is assumed that they will be pre-calculated on a routine basis (e.g. daily, hourly, every 10 minutes) and cached, rather than dynamically retrieved via standard API queries. The following OData Metrics Entity provides the metrics at a reporting frequency outlined in Table 12.

Metrics Property	Type	Description
Name	String	Name of metric according to a standard naming convention
Timestamp	DateTimeOffset	Date/time of metric reporting. Time is in UTC in the format YYYY-MM-DDThh:mm:ss.sssZ
MetricType	Enum	Gauge or Counter
Gauge	String	Value of Gauge at reporting timestamp (mandatory if MetricType=Gauge)
Count	Int64	Value of Counter at reporting timestamp (mandatory if MetricType=Counter)

Metric	Description	Type	Frequency
<productionType>.<productType>.<platformShortName>.<platformSerialIdentifier>.size	Cumulative volume of <productType> by <productionType> of mission platform (<platformShortName>.<platformSerialIdentifier>) produced in Bytes	Counter	10 minute update
<productionType>.<productType>.<platformShortName>.<platformSerialIdentifier>.count	Cumulative number of <productType> by <productionType> of mission platform (<platformShortName>.<platformSerialIdentifier>) produced	Counter	10 minute update
<productionType>.<platformShortName>.<platformSerialIdentifier>.size	Cumulative volume of data produced by <productionType> for mission platform (<platformShortName>.<platformSerialIdentifier>) in Bytes	Counter	Hourly update
<productionType>.<platformShortName>.<platformSerialIdentifier>.completed	Cumulative number of products produced by <productionType> for mission platform	Counter	Hourly update

	(<platformShortName>.<platformSerialIdentifier>)		
<productionType>.<platformShortName>.<platformSerialIdentifier>.failed	Cumulative number of products failed to be produced by <productionType> for mission platform (<platformShortName>.<platformSerialIdentifier>)	Counter	Hourly update
Download.<productType>.<platformShortName>.<platformSerialIdentifier>.<ServiceAlias>.size	Cumulative size of <productType> by <productionType> of mission platform (<platformShortName>.<platformSerialIdentifier>) downloaded (by <ServiceAlias>) in Bytes	Counter	10 minute update
Download.<productType>.<platformShortName>.<platformSerialIdentifier>.<ServiceAlias>.completed	Cumulative number of <productType> downloads by <productionType> of mission platform (<platformShortName>.<platformSerialIdentifier>) completed (by <ServiceAlias>)	Counter	10 minute update
Download.<productType>.<platformShortName>.<platformSerialIdentifier>.<ServiceAlias>.failed	Cumulative number of <productType> downloads by <productionType> of mission platform (<platformShortName>.<platformSerialIdentifier>) failed (by <ServiceAlias>)	Counter	10 minute update
SensingToPublication.Daily.min.time.<productType>.<platformShortName>.<platformSerialIdentifier>	Daily minimum time difference in seconds between sensing time (ContentDate/End) and PRIP PublicationDate of <productType> of mission platform (<platformShortName>.<platformSerialIdentifier>) (sliding window of 24 hours). This is only applicable to the Systematic Production.	Gauge	10 minute update
SensingToPublication.Daily.max.time.<productType>.<platformShortName>.<platformSerialIdentifier>	Daily maximum time difference in seconds between sensing time (ContentDate/End) and PRIP PublicationDate of <productType> of mission platform (<platformShortName>.<platformSerialIdentifier>) (sliding window of 24 hours). This is only applicable to the Systematic Production.	Gauge	10 minute update
SensingToPublication.Daily.avg.time.<productType>.<platformShortName>.<platformSerialIdentifier>	Daily average time difference in seconds between sensing time (ContentDate/End) and PRIP PublicationDate of <productType> of mission platform (<platformShortName>.<platformSerialIdentifier>) (sliding window of 24 hours). This is only applicable to the Systematic Production.	Gauge	10 minute update
SensingToPublication.Monthly.min.time.<productType>.<platformShortName>.<platformSerialIdentifier>	Monthly minimum time difference in seconds between sensing time (ContentDate/End) and PRIP PublicationDate of <productType> of mission platform (<platformShortName>.<platformSerialIdentifier>) (sliding window of last month).	Gauge	Hourly update

	This is only applicable to the Systematic Production.		
SensingToPublication.Monthly.max.time.<productType>.<platformShortName>.<platformSerialIdentifier>	Monthly maximum time difference in seconds between sensing time (ContentDate/End) and PRIP PublicationDate of <productType> of mission platform (<platformShortName>.<platformSerialIdentifier>) (sliding window of last month). This is only applicable to the Systematic Production.	Gauge	Hourly update
SensingToPublication.Monthly.avg.time.<productType>.<platformShortName>.<platformSerialIdentifier>	Monthly average time difference in seconds between sensing time (ContentDate/End) and PRIP PublicationDate of <productType> of mission platform (<platformShortName>.<platformSerialIdentifier>) (sliding window of last month). This is only applicable to the Systematic Production.	Gauge	Hourly update
OriginToPublication.Daily.min.time.<productType>.<platformShortName>.<platformSerialIdentifier>	Daily minimum time difference in seconds between OriginDate and PRIP PublicationDate of <productType> of mission platform (<platformShortName>.<platformSerialIdentifier>) (sliding window of 24 hours). This is only applicable to the Systematic Production.	Gauge	10 minute update
OriginToPublication.Daily.max.time.<productType>.<platformShortName>.<platformSerialIdentifier>	Daily maximum time difference in seconds between OriginDate and PRIP PublicationDate of <productType> of mission platform (<platformShortName>.<platformSerialIdentifier>) (sliding window of 24 hours). This is only applicable to the Systematic Production.	Gauge	10 minute update
OriginToPublication.Daily.avg.time.<productType>.<platformShortName>.<platformSerialIdentifier>	Daily average time difference in seconds between OriginDate and PRIP PublicationDate of <productType> of mission platform (<platformShortName>.<platformSerialIdentifier>) (sliding window of 24 hours). This is only applicable to the Systematic Production.	Gauge	10 minute update
OriginToPublication.Monthly.min.time.<productType>.<platformShortName>.<platformSerialIdentifier>	Monthly minimum time difference in seconds between OriginDate and PRIP PublicationDate of <productType> of mission platform (<platformShortName>.<platformSerialIdentifier>) (sliding window of last month). This is only applicable to the Systematic Production.	Gauge	Hourly update
OriginToPublication.Monthly.max.time.<productType>.<platformShortName>.<platformSerialIdentifier>	Monthly maximum time difference in seconds between OriginDate and PRIP PublicationDate of <productType> of	Gauge	Hourly update

	mission platform (<platformShortName>.<platformSerialIdentifier>) (sliding window of last month). This is only applicable to the Systematic Production.		
OriginToPublication.Monthly.avg.time.<productType>.<platformShortName>.<platformSerialIdentifier>	Monthly average time difference in seconds between OriginDate and PRIP PublicationDate of <productType> of mission platform (<platformShortName>.<platformSerialIdentifier>) (sliding window of last month). This is only applicable to the Systematic Production.	Gauge	Hourly update
Service.<KPI>.value	KPI value	Gauge	

Table 12: Systematic and On-Demand Production Summary Reporting Statistics

In addition to the metrics reported in the table above, the following statistics shall be provided for the On-Demand Production.

Metric	Description	Type	Frequency
SubmissionToCompletion.Daily.min.time	Minimum time from ProductionOrder SubmissionDate to CompletedDate on the On-Demand PRIP (sliding window of 24 hours)	Gauge	10 minute update
SubmissionToCompletion.Daily.max.time	Maximum time from ProductionOrder SubmissionDate to CompletedDate on the On-Demand PRIP (sliding window of 24 hours)	Gauge	10 minute update
SubmissionToCompletion.Daily.avg.time	Average time from ProductionOrder SubmissionDate to CompletedDate on the On-Demand PRIP (sliding window of 24 hours)	Gauge	10 minute update
SubmissionToCompletion.Monthly.min.time	Minimum time from ProductionOrder SubmissionDate to CompletedDate on the On-Demand PRIP (sliding window of last month)	Gauge	Hourly update
SubmissionToCompletion.Monthly.max.time	Maximum time from ProductionOrder SubmissionDate to CompletedDate on the On-Demand PRIP (sliding window of last month)	Gauge	Hourly update
SubmissionToCompletion.Monthly.avg.time	Average time from ProductionOrder SubmissionDate to CompletedDate on the On-Demand PRIP (sliding window of last month)	Gauge	Hourly update
Service.<KPI>.value	KPI value	Gauge	

Table 13: On-Demand Production Summary Reporting Statistics





ANNEX 1: WORKFLOW TAILORING

A template for tailoring the Workflow and WorkflowOptions entities, with an example filled, is shown below.

Workflow Description													WorkflowOptions						Comments
General					Input Product					Output Product									
					Input Product Type	Input Product Description	Input Product Reference												
Reference Included?	Reference Example	ContentDate Included?	Output Product Type	Output Product Description			Output Product Name Example	#	Name	Description	Type	Option Values	Default Value						
1		S2_L1C_L2A	Product Processing from Sentinel-2 L1C to L2A. Processor v02.04.00	1.1	S2MSI1C	Sentinel-2 Level 1C (Tile ID)	Y	S2B_OPER_MSI_L1C_TL_MPS__20180222T110232_A005038_T35TNK_N02.06	N	S2MSI2A	Sentinel-2 Level 2A		Option 1	Resolution	Resolution of the L2A product to be processed	Int	60 - only 60m resolution will be processed 20 - 60m & 20m resolutions will be processed 10 - all resolutions will be processed	60	1) The input product filename is considered the PDI identifier within the single tile compact format and not the actual filename.
													Option 2						
													Option 3						
													Option 4						
													Option 5						
													etc						



ANNEX 2: ODATA METADATA DESCRIPTION

```
<?xml version="1.0"?>
<edmx:Edmx Version="4.0">
  <edmx:DataServices>
    <Schema Namespace="OData.CSC">
      <EnumType Name="JobStatus" IsFlags="false" UnderlyingType="Edm.Int32">
        <Member Name="queued" Value="4"/>
        <Member Name="in_progress" Value="0"/>
        <Member Name="completed" Value="1"/>
        <Member Name="failed" Value="2"/>
        <Member Name="cancelled" Value="3"/>
      </EnumType>
      <EnumType Name="ProductionType" IsFlags="false" UnderlyingType="Edm.Int32">
        <Member Name="systematic_production" Value="0"/>
        <Member Name="on-demand default" Value="1"/>
        <Member Name="on-demand non-default" Value="2"/>
      </EnumType>
      <EnumType Name="EnumType" IsFlags="false" UnderlyingType="Edm.Int32">
        <Member Name="Gauge" Value="0"/>
        <Member Name="Counter" Value="1"/>
      </EnumType>
      <Function Name="Intersects" IsBound="true">
        <Parameter Name="product" Type="OData.CSC.Product"/>
        <Parameter Name="area" Type="Edm.Geography"/>
        <ReturnType Type="Edm.Boolean" />
      </Function>
      <EnumType Name="SubscriptionStatus" IsFlags="false" UnderlyingType="Edm.Int32">
        <Member Name="running" Value="0"/>
        <Member Name="paused" Value="1"/>
        <Member Name="cancelled" Value="2"/>
      </EnumType>
      <EntityType Name="StringAttribute" BaseType="OData.CSC.Attribute">
        <Property Name="Value" Type="Edm.String"/>
      </EntityType>
      <EntityType Name="IntegerAttribute" BaseType="OData.CSC.Attribute">
        <Property Name="Value" Type="Edm.Int64"/>
      </EntityType>
      <EntityType Name="DateTimeOffsetAttribute" BaseType="OData.CSC.Attribute">
        <Property Name="Value" Type="Edm.DateTimeOffset"/>
      </EntityType>
      <EntityType Name="DoubleAttribute" BaseType="OData.CSC.Attribute">
        <Property Name="Value" Type="Edm.Double"/>
      </EntityType>
      <EntityType Name="BooleanAttribute" BaseType="OData.CSC.Attribute">
        <Property Name="Value" Type="Edm.Boolean"/>
      </EntityType>
      <EntityType Name="Workflow">
        <Key>
          <PropertyRef Name="Id"/>
        </Key>
      </EntityType>
    </Schema>
  </edmx:DataServices>
</edmx:Edmx>
```



```

<Property Name="Id" Type="Edm.Guid"/>
<Property Name="Name" Type="Edm.String"/>
<Property Name="Description" Type="Edm.String"/>
<Property Name="InputProductType" Type="Edm.String"/>
<Property Name="OutputProductType" Type="Edm.String"/>
<Property Name="WorkflowVersion" Type="Edm.String"/>
</EntityType>
<EntityType Name="WorkflowOptions">
  <Key>
    <PropertyRef Name="Id"/>
  </Key>
  <Property Name="Name" Type="Edm.String"/>
  <Property Name="Description" Type="Edm.String"/>
  <Property Name="Type" Type="Edm.String"/>
  <Property Name="Default" Type="Edm.String"/>
  <Property Name="Value" Type="Edm.String"/>
</EntityType>
<EntityType Name="ProductionOrder">
  <Key>
    <PropertyRef Name="Id"/>
  </Key>
  <Property Name="Id" Type="Edm.Guid"/>
  <Property Name="Status" Type="OData.CSC.JobStatus"/>
  <Property Name="StatusMessage" Type="Edm.String"/>
  <Property Name="OrderOutputSize" Type="Edm.Int64"/>
  <Property Name="SubmissionDate" Type="Edm.DateTimeOffset"/>
  <Property Name="EstimatedDate" Type="Edm.DateTimeOffset"/>
  <Property Name="CompletedDate" Type="Edm.DateTimeOffset"/>
  <Property Name="EvictionDate" Type="Edm.DateTimeOffset"/>
  <Property Name="Priority" Type="Edm.Int64"/>
  <Property Name="InputProductReference" Type="Edm.ComplexType "/>
  <Property Name="WorkflowId" Type="Edm.Guid"/>
  <Property Name="WorkflowName" Type="Edm.String"/>
  <Property Name="WorkflowOptions" Type="OData.CSC.WorkflowOptions"/>
  <NavigationProperty Name="Product" Type="OData.CSC.Product"/>
</EntityType>
<EntityType Name="Subscription">
  <Key>
    <PropertyRef Name="Id"/>
  </Key>
  <Property Name="Id" Type="Edm.Guid"/>
  <Property Name="Status" Type="OData.CSC.SubscriptionStatus"/>
  <Property Name="FilterParam" Type="Edm.String"/>
  <Property Name="SubmissionDate" Type="Edm.DateTimeOffset" Precision="3"/>
  <Property Name="LastNotificationDate" Type="Edm.DateTimeOffset" Precision="3"/>
</EntityType>
<EntityType Name="Product" HasStream="true">
  <Key>
    <PropertyRef Name="Id"/>
  </Key>
  <Property Name="Id" Type="Edm.Guid"/>
  <Property Name="Name" Type="Edm.String"/>
  <Property Name="ContentType" Type="Edm.String"/>
  <Property Name="ContentLength" Type="Edm.Int64"/>

```



```

<Property Name="OriginDate" Type="Edm.DateTimeOffset" Precision="3"/>
<Property Name="PublicationDate" Type="Edm.DateTimeOffset" Precision="3"/>
<Property Name="EvictionDate" Type="Edm.DateTimeOffset" Precision="3"/>
<Property Name="Checksum" Type="Collection(OData.CSC.Checksum)"/>
<Property Name="ContentDate" Type="OData.CSC.TimeRange"/>
<Property Name="ProductionType" Type="OData.CSC.ProductionType"/>
<Property Name="Footprint" Type="Edm.Geography"/>
<Property Name="GeoFootprint" Type="Edm.Geography"/>
<NavigationProperty Name="Attributes" Type="Collection(OData.CSC.Attribute)"/>
</EntityType>
<EntityType Name="Attribute" Abstract="true">
  <Key>
    <PropertyRef Name="Id"/>
  </Key>
  <Property Name="Name" Type="Edm.String"/>
  <Property Name="ValueType" Type="Edm.String"/>
</EntityType>
<EntityType Name="Metric">
  <Key>
    <PropertyRef Name="Name"/>
  </Key>
  <Property Name="Name" Type="Edm.String"/>
  <Property Name="Timestamp" Type="Edm.DateTimeOffset" Precision="3"/>
  <Property Name="MetricType" Type="OData.CSC.EnumType"/>
  <Property Name="Gauge" Type="Edm.String"/>
  <Property Name="Counter" Type="Edm.Int64"/>
</EntityType>
<ComplexType Name="TimeRange">
  <Property Name="Start" Type="Edm.DateTimeOffset" Nullable="false" Precision="6"/>
  <Property Name="End" Type="Edm.DateTimeOffset" Nullable="false" Precision="6"/>
</ComplexType>
<ComplexType Name="Checksum">
  <Property Name="Algorithm" Type="Edm.String" Nullable="false"/>
  <Property Name="Value" Type="Edm.String" Nullable="false"/>
  <Property Name="ChecksumDate" Type="Edm.DateTimeOffset"/>
</ComplexType>
<ComplexType Name="Property">
  <Property Name="Name" Type="Edm.String" Nullable="false"/>
  <Property Name="Value" Type="Edm.String" Nullable="false"/>
</ComplexType>
<ComplexType Name="InputProductReference">
  <Property Name="Reference" Type="Edm.String" Nullable="true"/>
  <Property Name="ContentDate" Type="OData.CSC.TimeRange" Nullable="true"/>
</ComplexType>
<Action Name="Pause" IsBound="true">
  <Parameter Name="Subscription" Type="OData.CSC.Subscription" Nullable="false"/>
</Action>
<Action Name="ProductionOrder" IsBound="true">
  <Parameter Name="Product" Type="OData.CSC.Order" Nullable="false"/>
  <Parameter Name="Priority" Type="Edm.Int32"/>
  <Parameter Name="NotificationEndpoint" Type="Edm.String"/>
  <Parameter Name="NotificationEpUsername" Type="Edm.String"/>
  <Parameter Name="NotificationEpPassword" Type="Edm.String"/>
  <ReturnType Type="OData.CSC.Order"/>

```



```

</Action>
<Action Name="Resume" IsBound="true">
  <Parameter Name="Subscription" Type="OData.CSC.Subscription" Nullable="false"/>
  <ReturnType Type="OData.CSC.Subscription"/>
</Action>
<Action Name="Cancel" IsBound="true">
  <Parameter Name="ProductionOrder" Type="OData.CSC.Product" Nullable="false"/>
  <ReturnType Type="OData.CSC.Order"/>
</Action>
<Action Name="Cancel" IsBound="true">
  <Parameter Name="Subscription" Type="OData.CSC.Subscription" Nullable="false"/>
  <ReturnType Type="OData.CSC.Subscription"/>
</Action>
<EntityContainer Name="Container">
  <EntitySet Name="ProductionOrders" EntityType="OData.CSC.Order">
    <NavigationPropertyBinding Path="Product" Target="Products"/>
  </EntitySet>
  <EntitySet Name="Products" EntityType="OData.CSC.Product">
    <NavigationPropertyBinding Path="Attributes" Target="Attributes"/>
  </EntitySet>
  <EntitySet Name="Subscriptions" EntityType="OData.CSC.Subscription">
    <NavigationPropertyBinding Path="User" Target="Users"/>
  </EntitySet>
  <EntitySet Name="Attributes" EntityType="OData.CSC.Attribute"
IncludeInServiceDocument="false"/>
  <EntitySet Name="StringAttributes" EntityType="OData.CSC.StringAttribute"
IncludeInServiceDocument="false"/>
  <EntitySet Name="IntegerAttributes" EntityType="OData.CSC.IntegerAttribute"
IncludeInServiceDocument="false"/>
  <EntitySet Name="DoubleAttributes" EntityType="OData.CSC.DoubleAttribute"
IncludeInServiceDocument="false"/>
  <EntitySet Name="BooleanAttributes" EntityType="OData.CSC.BooleanAttribute"
IncludeInServiceDocument="false"/>
  <EntitySet Name="DateTimeOffsetAttributes"
EntityType="OData.CSC.DateTimeOffsetAttribute" IncludeInServiceDocument="false"/>
  <EntitySet Name="Metrics" EntityType="OData.CSC.Metric"/>
</EntityContainer>
</Schema>
</edmx:DataServices>
</edmx:Edmx>

```

ANNEX 3: INTERFACE CONFORMANCE TEST SUITE

A test suite that can facilitate the implementation and be used for testing the Interface delivery Point is available at <https://github.com/SercoSPA/CSC-Ground-Segment-Interface-Delivery-Point-Test-Suite>

The current implementation is designed to check the capability of the Interface delivery Point, and it is implemented as a Postman (<https://www.getpostman.com/>) Collection JSON file (format v2.1).

The test suite is provided as a set of OData requests to be sent to the Interface delivery Point, and a series of test scripts for verifying the received responses.

The test suite is quite broad and it is suitable for multiple interfaces (e.g. LTA, AUXIP, Production). It includes OData requests on: EO product types, EO product properties and attributes, as well as other main entities defined in the ICDs. The test suite may not include all possible queries specific to a certain interface. Self-tests may be performed using the Postman collection or it can be used by Clients to test existing interfaces.