

# ***HSA-HCSS ICD***

---

**prepared by/préparé par** Juan Carlos Segovia, Jesus Salgado, Maria Henar Sarmiento, Javier Castellanos

**reference/référence** HERSCHEL-HSC-DOC-1069

**issue/édition** 1

**revision/révision** 8

**date of issue/date d'édition** 28-Mar-2012

**status/état** Issue

**Document type/type de document**

**Distribution/distribution**

**A P P R O V A L**

Title <i>Titre</i>	HSA – HCSS ICD	issue <i>issue</i>	1	revision <i>revision</i>	8
-----------------------	----------------	-----------------------	---	-----------------------------	---

author <i>auteur</i>	Juan Carlos Segovia, Jesus Salgado, Maria Henar Sarmiento, Javier Castellanos	date <i>date</i>	28-Mar-2012
-------------------------	---	---------------------	-------------

approved by <i>approuvé par</i>	Christophe Arviset, Stephan Ott	date <i>date</i>	
------------------------------------	---------------------------------	---------------------	--

**D I S T R I B U T I O N L I S T**

designation / <i>désignation</i>	name/ <i>nom</i>
HSA/HSC	hsa@sciops.esa.int (See <a href="http://www.sciops.esa.int/WebPortal/Pages/alias.php?hsa">http://www.sciops.esa.int/WebPortal/Pages/alias.php?hsa</a> )
HSC	Jorgo Bakker ( <a href="mailto:jbakker@sciops.esa.int">jbakker@sciops.esa.int</a> )

**C H A N G E L O G**

reason for change / <i>raison du changement</i>	issue/ <i>issue</i>	revision/ <i>revision</i>	date/ <i>date</i>
New 'scope' metadata field added	1	1	12-Sep-2008
Added Physical descriptions diagrams	1	1	12-Sep-2008
Added 'HCSS Product Version' and 'SPG Software Identifier' to Data Dictionary section	1	2	04-Mar-2009
Added explanations about SPG software identifier to several sections	1	2	04-Mar-2009
Added I tarball specifications	1	2	04-Mar-2009
Updated messages MSG-3-001 and MSG-3-002 related to I/F 3	1	2	04-Mar-2009
Replaced AD-7 by HERSCHEL-HSC-DOC-1081	1	2	04-Mar-2009

<i>reason for change /raison du changement</i>	<i>issue/issue</i>	<i>revision/revision</i>	<i>date/date</i>
Updated physical descriptions	1	2	04-Mar-2009
CRC xml attribute (Ingestion Request File) is removed	1	2	04-Mar-2009
Updated Common Metadata table	1	2	04-Mar-2009
Changed VO-Table by VOTable	1	2	04-Mar-2009
Changed 'I.esac.esa.int:8080' by 'archives.esac.esa.int/I'	1	2	04-Mar-2009
Added 'importing' directory to Ingestion Request Area	1	2	04-Mar-2009
Updated MSG-2.1-007/008 HCSS_TRACK_ID resource class. New functionality: query by HCSS_CLASS_TYPE	1	2	04-Mar-2009
Updated 'stage' area specification	1	2	04-Mar-2009
Updated file name verification message MSG-2.1-005	1	2	04-Mar-2009
Updated use case UC-001	1	2	04-Mar-2009
Added authentication procedure for retrieving products	1	2	04-Mar-2009
Added product location inside a tar file	1	2	04-Mar-2009
Removed open issues	1	2	04-Mar-2009
Added HAIIO HTTP Error codes	1	2	04-Mar-2009
FTP server specified	1	2	04-Mar-2009
I/F-3 working with on-demand results	1	2	04-Mar-2009
Included LAST_TRACK_VERSION parameter	1	3	29-Sep-2009
Ingestion Request File XML 'description' attribute changed to 'implied' (optional)	1	3	29-Sep-2009
Added postcard handling	1	3	20-Jan-2010
MSG-3-001 Starting of the HUI message format changed	1	3	20-Jan-2010
Data dictionary updated: changed HCSS-HSA Writeable pool Application explanation	1	3	06-Apr-2010
Objects updated	1	3	06-Apr-2010
MSG-2.3-002 explanation updated	1	3	06-Apr-2010
Physical descriptions updated	1	3	06-Apr-2010

<i>reason for change /raison du changement</i>	<i>issue/issue</i>	<i>revision/revision</i>	<i>date/date</i>
Added cus_mode to Ingestion Request Object	1	3	06-Apr-2010
Added Common Ingestion Request example	1	3	06-Apr-2010
Added two open issues related to Ingestion Request Object	1	3	15-Apr-2010
Changed PLASTIC to SAMP (available in HSA v3.1 and HCSS v1.2.7)	1	3	15-Apr-2010
Updated Common Metadata table	1	3	15-Apr-2010
Added XML Request Ingestion File retrieval	1	4	14-Jul-2010
Added Metadata retrieval	1	4	14-Jul-2010
Added LIMIT request parameter	1	4	24-Nov-2010
Added SHOW_PROPRIETARY request parameter	1	4	24-Nov-2010
Added Virtual TAR retrieval	1	4	24-Nov-2010
Specified HUI JNLP URL	1	5	7-Apr-2011
Updated Ingestion Request file XML attributes	1	5	7-Apr-2011
Added optional 'file_absolute_path' XML item attribute	1	5	27-Apr-2011
Specified Common Metadata string sizes	1	6	17-May-2011
Added 'extended' XML attribute	1	7	19-Aug-2011
Added Observation Metadata queries through HAIO	1	8	28-Feb-2012
Added procedure to retrieve Observations through HAIO	1	8	28-Feb-2012
Added new HAIO metadata response examples	1	8	28-Feb-2012
Added new HAIO HTTP error codes	1	8	28-Feb-2012
Replaced HBL references by QM	1	8	28-Feb-2012
Added new HAIO VOTABLE error messages	1	8	28-Feb-2012
Added 'User-Agent' HTTP header format	1	8	28-Mar-2012
Created allowed query operators appendix	1	8	28-Mar-2012
Added open issue related to Common Metadata string sizes.	1	8	28-Mar-2012

**CHANGE RECORD**

Issue: 1 Revision: 3

<i>reason for change/raison du changement</i>	<i>page(s)/page(s)</i>	<i>paragraph(s)/paragraph(s)</i>
Added two new concepts to 'Data Dictionary' section: 'HCSS Product Version' and 'SPG Software identifier'	Section 2.1	
Added explanations about Product Version handling	Sections 5.1, 5.2 and 9.4	
Added Tarball specifications	Appendix F	
Changed message broker register client message identifier (MSG-3-002) to 'HIPE'	Section 2.3.3.1	
Modified message 'Starting of the HUI' (MSG-3-001)	Section 2.3.3.2	
Updated I/F 3 process descriptions	Section 6.3.2	
Updated Herschel Science Archive User Requirements link (AD-7)	Section 1.3	
Updated physical descriptions	Sections 7.1, 7.2, 7.4.1 and 7.4.2	
CRC xml attribute is removed	Sections 8, 10.1 and 10.1.2	
Updated Common Metadata table	Section 9.2	
New Ingestion Request Area 'importing' directory	Sections 2.2.6 and 6.1	
Updated MSG-2.1-007/008 HCSS_TRACK_ID resource class.	Sections 2.3.2.4 and 11.7	
Replaced 'stage' by 'Ingestion Request Area'	Section 2.2.6	
Updated file name verification message MSG-2.1-005: added HCSS_CLASS_TYPE	Section 2.3.2.3	
Updated UC-001: added observation tarball references generation step	Section 4.1.1	
Added authentication procedure, objects, messages and explanations.	Sections 2.1, 2.2.2, 2.3.2.6, 2.3.2.7, 4.3, 5.2 and 6.2	
Added two new xml attributes (distribution_path and distribution_filename) for specifying a product tar file location.	Sections 10.1, 10.1.2, 10.1.3 and Appendix F	
Removed Open Issues	Sections 1.2, 1.5, 2.2.8, 2.3.2.3 and 8	

reason for change/ <i>raison du changement</i>	page(s)/ <i>page(s)</i>	paragraph(s)/ <i>paragraph(s)</i>
Added HAIO HTTP Error codes	Sections 2.2.2, 2.3.2.6, 2.3.2.7,	
FTP server specified	4.3.3, 5.2, 6.2 and Appendix G	
I/F-3 working with on-demand results	Section 7.4.1	
I/F-3 working with on-demand results	Sections 2.3.3.5 and 6.3.3	
Included LAST_TRACK_VERSION parameter	Sections 2.3.2.1 and 11.1	
Ingestion Request File XML 'description' attribute	Section 10.1.2	
changed to 'implied' (optional)		
Added Product Postcard object	Sections 2.1, 2.2.1, 2.2.6 and	
Added Product Postcard handling	2.2.14	
Added Product Postcard handling	Sections 4.1.1, 4.1.4, 5.1, 6.1,	
MSG-3-001 Starting of the HUI message: changed	10.1 and 10.1.2	
the way properties are passed.	Sections 2.3.3.1 and 6.3.2	
Data dictionary updated: changed HCSS-HSA	Section 2.1	
Writeable pool Application explanation		
Objects updated	Sections 2.2.9, 2.2.10 and 2.2.14	
MSG-2.3-002 explanation updated	Section 2.3.2.7	
Physical descriptions updated	Sections 7.1 and 7.2	
Added cus_mode to Ingestion Request Object	Section 10.1.2	
Added Common Ingestion Request example	Section 10.1.2	
Added two open issues related to Ingestion Request	Sections 2.2.9 and 8	
Object		
Changed PLASTIC to SAMP (available in HSA v3.1	Sections 1.4, 2.1, 2.3.3, 2.3.3.2,	
and HCSS v1.2.7)	2.3.3.3, 5.3, 6.3.1 and 6.3.3	
Updated Common Metadata table	Section 9.2	
Added XML Ingestion File and Metadata retrieval	Sections 1.2, 2.2.2, 2.3.2.6, 4.3,	
Added LIMIT request parameter	5.2, 6.2, 7.6.5 and 11.11	
Added SHOW_PROPRIETARY request parameter	Sections 2.3.2.1, 6.2, 11.1,	
Added Virtual TAR retrieval	Appendix H and Appendix J.	
Added HUI JNLP URL	Sections 2.3.2.1, 6.2, 11.1 and	
Updated Ingestion Request file XML attributes	11.6	
	Sections 1.2, 2.1, 2.2.15, 4.3.3,	
	7.6.6 and Appendix I	
	Section 2.3.3.1	
	Section 10.1.2	

<i>reason for change/raison du changement</i>	<i>page(s)/page(s)</i>	<i>paragraph(s)/paragraph(s)</i>
Added optional 'file_absolute_path' XML item attribute	Sections 10.1 and 10.1.2	
Added string sizes column to Common Metadata table	Section 9.2	
Added 'extended' XML attribute	Sections 4.1.1, 4.1.3, 5.1, 6.1, 10.1, 10.1.2, 13.2 and Appendix K	
Replaced HBL references by QM	Sections 2.1 and 6.2	
Added explanation about observation metadata queries	Sections 1.2, 2.2.2, 2.3.2.1, 2.3.2.6 (inserted new section), 4.2, 4.3, 5.2, 6.2	
Added explanation about observation request	Section 2.3.2.7	
Added new HAIO metadata response examples	Appendix C	
Added new HAIO HTTP Error codes	Section 6.2 and Appendix G	
Added new VOTABLE Errors	Appendix H	
Added 'User-Agent' HTTP header format	Appendix G, H and L	
Created allowed query operators appendix	Section 2.3.2.1 and Appendix M	
Added open issue related to Common Metadata string sizes.	Sections 8 and 9.2	

## **T A B L E O F C O N T E N T S**

<b>1</b>	<b>INTRODUCTION .....</b>	<b>1</b>
1.1	Purpose.....	1
1.2	Scope.....	1
1.3	Applicable documentation .....	2
1.4	Reference documentation.....	2
1.5	Acronyms .....	2
<b>2</b>	<b>DOMAIN OBJECTS .....</b>	<b>3</b>
2.1	Data dictionary .....	3
2.2	Objects .....	6
2.2.1	Product Ingestion (I/F-1).....	6
2.2.2	Direct HAIO access (I/F-2).....	7
2.2.3	IA – HUI (I/F-3).....	8
2.2.4	SPG On-demand reprocessing (I/F-4).....	9
2.2.5	Ingestion Storage Area.....	10
2.2.6	Ingestion Request Area .....	10
2.2.7	On-Demand Storage Area.....	11
2.2.8	On-Demand Request Area .....	11
2.2.9	Ingestion Request File.....	11
2.2.10	Ingestion Request Error File .....	12
2.2.11	On-demand Request File.....	13
2.2.12	On-demand Request Result File .....	13
2.2.13	HSA tarball .....	13
2.2.14	Product Postcard File .....	14
2.2.15	Virtual Tar.....	14
2.3	Message formats .....	15
2.3.1	Product Ingestion (I/F-1).....	15
2.3.1.1	<b>MSG-1.1-001: Ingestion Request .....</b>	<b>15</b>
2.3.1.2	<b>MSG-1.1-002: Ingestion Request error.....</b>	<b>15</b>
2.3.2	Direct HAIO Access (I/F-2).....	15
2.3.2.1	<b>MSG-2.1-001/002: HAIO Product Metadata Queries .....</b>	<b>15</b>
2.3.2.2	<b>MSG-2.1-003/004: HAIO HCSS_CLASS_TYPE Resource Class Query .....</b>	<b>17</b>
2.3.2.3	<b>MSG-2.1-005/006: HAIO PRODUCT Resource Class Query .....</b>	<b>18</b>
2.3.2.4	<b>MSG-2.1-007/008: HAIO HCSS_TRACK_ID Resource Class Query.....</b>	<b>19</b>
2.3.2.5	<b>MSG-2.1-009/010: HAIO PRODUCT_METADATA Resource Class Query .....</b>	<b>20</b>
2.3.2.6	<b>MSG-2.1-011/012: HAIO Observation Metadata Queries .....</b>	<b>21</b>
2.3.2.7	<b>MSG-2.2-001: HAIO Product Request.....</b>	<b>23</b>
2.3.2.8	<b>MSG-2.3-001/002: HAIO Login Request.....</b>	<b>24</b>
2.3.3	IA – HUI (I/F-3).....	25



2.3.3.1	<b>MSG-3-001</b> Starting of the HUI .....	25
2.3.3.2	<b>MSG-3-002</b> Register client with message broker .....	26
2.3.3.3	<b>MSG-3-003</b> Unregister client with message broker .....	26
2.3.3.4	<b>MSG-3-004</b> Request loading observations selected in HUI .....	26
2.3.3.5	<b>MSG-3-005</b> Request loading data from on-demand reprocessing .....	26
2.3.4	SPG On-demand reprocessing (I/F-4).....	27
2.3.4.1	<b>MSG-4-001/002</b> : Request an On-Demand Reprocessing .....	27
<b>3</b>	<b>ROLES .....</b>	<b>28</b>
<b>4</b>	<b>USE CASES .....</b>	<b>29</b>
4.1	UC-001: Product Ingestion .....	29
4.2	UC-002: Search by Metadata .....	31
4.3	UC-003: File Retrieval.....	32
4.4	UC-004: On-Demand Reprocessing Request-Response.....	34
<b>5</b>	<b>DESIGN DESCRIPTIONS .....</b>	<b>35</b>
5.1	Product Ingestion (I/F-1).....	35
5.2	Direct HAIO Access (I/F-2).....	36
5.3	IA – HUI (I/F-3).....	36
5.4	SPG On-Demand Reprocessing (I/F-4) .....	37
<b>6</b>	<b>PROCESS DESCRIPTIONS .....</b>	<b>39</b>
6.1	Product Ingestion (I/F-1).....	39
6.2	Direct HAIO Access (I/F-2).....	42
6.3	IA – HUI (I/F-3).....	44
6.3.1	Starting up .....	44
6.3.2	Logging in.....	44
6.3.3	Data retrieval.....	45
6.4	SPG On-Demand Reprocessing (I/F-4) .....	45
<b>7</b>	<b>PHYSICAL DESCRIPTIONS.....</b>	<b>47</b>
7.1	Systematic Processing.....	47
7.2	On-Demand Processing.....	48
7.3	Bulk reprocessing.....	49
7.4	System descriptions.....	49
7.4.1	HSA system.....	49
7.4.2	HCSS system.....	49
7.4.3	Grid .....	49
7.4.4	Shared areas .....	49
7.5	Requirements .....	49
7.6	Procedures.....	50
<b>8</b>	<b>ASSUMPTIONS AND OPEN ISSUES .....</b>	<b>51</b>

<b>9</b>	<b>APPENDIX A: LIST OF METADATA FOR EACH PRODUCT.....</b>	<b>52</b>
9.1	Metadata Types .....	52
9.2	Common Metadata .....	52
9.3	ObservationContext metadata .....	58
9.4	Versioning Metadata .....	59
9.5	HSA Control Metadata.....	60
<b>10</b>	<b>APPENDIX B: XML FILES.....</b>	<b>61</b>
10.1	Ingestion Request .....	61
10.1.1	File Name Rules.....	63
10.1.2	DTD .....	63
10.1.3	Hierarchy.....	67
10.2	Ingestion Request Error .....	68
10.2.1	File Name Rules.....	68
10.2.2	DTD .....	69
10.3	On-demand Request.....	69
10.3.1	File Name Rules.....	70
10.3.2	DTD .....	70
10.4	On-demand Result.....	71
10.4.1	File Name Rules.....	71
10.4.2	DTD .....	71
<b>11</b>	<b>APPENDIX C: HAIO RESPONSE/QUERY EXAMPLES .....</b>	<b>72</b>
11.1	Product Metadata Request .....	72
11.2	Product Metadata VOTable response .....	73
11.3	HAIO HCSS_CLASS_TYPE Resource Class Request.....	73
11.4	HAIO HCSS_CLASS_TYPE Resource Class VOTable response.....	74
11.5	HAIO Product Resource Class Request.....	74
11.6	HAIO Product Resource Class VOTable response.....	75
11.7	HAIO HCSS_TRACK_ID Resource Class Request .....	77
11.8	HAIO HCSS_TRACK_ID Resource Class VOTable response .....	77
11.9	HAIO PRODUCT_METADATA Resource Class Request .....	78
11.10	HAIO PRODUCT_METADATA Resource Class VOTable response .....	78
11.11	Retrieving a File from HSA using HAIO .....	79
11.12	Login in to HAIO.....	80
11.13	Observation Metadata Request .....	80
11.14	Observation Metadata VOTable response .....	81
<b>12</b>	<b>APPENDIX D: FITS KEYWORDS USED IN HERSCHEL.....</b>	<b>83</b>
<b>13</b>	<b>APPENDIX E: WHO IMPLEMENTS WHAT .....</b>	<b>83</b>
13.1	HCSS.....	83
13.2	HSA.....	84
<b>14</b>	<b>APPENDIX F: RULES FOR CREATING THE HSA TARBALL OBJECT.....</b>	<b>84</b>

---

<b>15</b>	<b>APPENDIX G: HAIO HTTP ERROR CODES.....</b>	<b>87</b>
15.1	PRODUCT RETRIEVAL.....	87
15.2	LOGIN .....	88
<b>16</b>	<b>APPENDIX H: VOTABLE QUERY RESPONSE ERRORS.....</b>	<b>89</b>
16.1	VOTABLE ERRORS .....	89
<b>17</b>	<b>APPENDIX I: VIRTUAL TAR .....</b>	<b>90</b>
17.1	STRUCTURE .....	90
<b>18</b>	<b>APPENDIX J: QUERY RESULTS LIMITS .....</b>	<b>90</b>
18.1	LIMITS .....	90
<b>19</b>	<b>APPENDIX K: EXTENDED PROCESSING INGESTION CHECK.....</b>	<b>91</b>
<b>20</b>	<b>APPENDIX L: 'USER-AGENT' HTTP HEADER FORMAT.....</b>	<b>91</b>
<b>21</b>	<b>APPENDIX M: ALLOWED QUERY OPERATORS.....</b>	<b>92</b>

# 1 INTRODUCTION

## 1.1 Purpose

The present document defines the interfaces between the HCSS (Herschel Common Science System) and the HSA (Herschel Science Archive), and identifies the requirements on both systems for the transfer of metadata and data.

## 1.2 Scope

This document describes the following interfaces (see *Chapter 2.1. Data Dictionary* for definitions):

- I/F-1, Product Ingestion
  - **I/F-1.1, Available Data Notification:** Notification from HCSS to HSA of new data available. Notification from HSA to HCSS of the ingestion results. This interface implies a definition of the HCSS data types available: Observation Products and Common Products.
  - **I/F-1.2, Data Ingestion:** Copy of new products from the HCSS store to the HSA-HSTAGE area and ingestion of products metadata into the HSA.
- I/F-2, Direct HAIO access:
  - **I/F-2.1, HAIO Metadata query:** Definition of a metadata interchange mechanism and a metadata interchange document format. (These requests are related to HAIO [product.jsp](#) HTTP requests.)
  - **I/F-2.2, HAIO File retrieval:** This interface implies the implementation of a data access to HSA products, product metadata, XML ingestion requests files or Virtual TAR through the HAIO subsystem using URNs as identifier or XML ingestion requests files names. (These requests are related to HAIO [metadata.jsp](#) HTTP requests.)
  - **I/F-2.3, HAIO login:** Definition of the interface to log-in to HSA. (These requests are related to HAIO [login.jsp](#) HTTP requests.)
- **I/F-3, IA-HUI applications intercommunication:** Definition of the data interchange mechanism between the Herschel Interactive Analysis (IA) application and the HSA User Interface (HUI).
- **I/F-4, SPG On-demand reprocessing:** Definition of request, response and processed files.

This document does not describe the following interfaces:

- FTP access to HSA for retrieving either Observations or Reprocessed Observations (on-demand requests).
- Bulk Product Transfer (see [RD-4]).

### **1.3**    *Applicable documentation*

- **AD-1:** Herschel Product Definitions Document (HERSCHEL-HSC-DOC-0959)  
<http://www.rssd.esa.int/livelink?func=ll&objId=2766178&objAction=download&viewType=1>
- **AD-2:** Herschel Product Access Layer (PAL) Design (HERSCHEL-HSC-DOC-0839):  
<http://www.rssd.esa.int/SD-general/Projects/Herschel/hscdt/releases/doc/ia/pal/doc/design/html/pal-design.html>
- **AD-3:** Herschel Product Metadata: Chapter 4 of HERSCHEL-HSC-DOC-0959  
(<http://www.rssd.esa.int/livelink?func=ll&objId=2766178&objAction=download&viewType=1>) or <http://www.herschel.be/twiki/bin/view/Hcss/ProductMetaDataConvention>
- **AD-4:** HAIO user manual:  
<http://archives.esac.esa.int/hsa/aio/doc/doc/AIO%20Users%20Manual.pdf>
- **AD-5:** VOTABLE specification: <http://www.ivoa.net/twiki/bin/view/IVOA/IvoaVOTable>
- **AD-6:** FITS specification: [http://archive.stsci.edu/fits/fits\\_standard/](http://archive.stsci.edu/fits/fits_standard/) (see also: <http://archive.stsci.edu/fits/>)
- **AD-7:** Herschel Science Archive User Requirements (HERSCHEL-HSC-DOC-1081)  
<http://www.rssd.esa.int/livelink/open/2849607>

### **1.4**    *Reference documentation*

- **RD-1:** Jython language: <http://www.jython.org>
- **RD-3:** Plastic Message Definitions proposal,  
<http://www.ivoa.net/Documents/latest/SAMP.html>
- **RD-4:** HTTP Protocol: <http://www.w3.org/Protocols/Specs.html>
- **RD-5:** Herschel Bulk Product Transfer ICD (HERSCHEL-HSC-ICD-1083):  
<http://www.rssd.esa.int/livelink/open/2807875>

### **1.5**    *Acronyms*

The acronyms list used within the Herschel project can be found at:  
<http://www.herschel.be/twiki/bin/view/Hcss/HerschelAcronyms>

## 2 DOMAIN OBJECTS

### 2.1 Data dictionary

<i>Concept</i>	<i>Description</i>
Product	An object with metadata (key, value), dataset (data) and a file representation (FITS file). See Context below. See [AD-1] and [AD-3] documents. From the point of view of HSA: a product can be part of an Observation or can be a common data used by several Observations. No common products shall have access restrictions.
Context	A product that contains references to other products or contexts (these are called 'children'). See [AD-1] and [AD-2] documents.
Product Reference	A reference to a product or context that can be handled by a Context. See [AD-2] document.
HCSS URN	Product unique identifier in HCSS storage. See [AD-2] document.
HCSS Product Version	It is identified by HCSS URN number (HCSS_URN_VERSION control metadata identifier).
SPG Software Identifier	Identifies the HCSS software used for creating products/contexts.
Observation Products	Products belonging to an observation with access restrictions.
Common Products	Products shared by several observations with no access restrictions.
Product metadata	Hash table (key-value pairs) associated to a product (see Appendix A)
Product postcard	Postcard image file associated to an observation
HSA Control metadata	HSA metadata (key, value) associated to an ingested product (see Appendix A).
HTTP-GET	HTTP GET request (see [RD-4])
VOTable	XML data following VOTable specification (see [AD-4])
Zipped Local Store	Compressed archive which contains a PAL local store pool (which works with FITS files).
HsaPoolDaemon	Daemon allowing remote connections to a local PAL HsaPool.

HsaPool	PAL pool with the basic functionality for interacting with HAIO. HsaWritePool and HsaReadPool are based on HsaPool.
HsaWritePool	PAL pool for writing data that shall be ingested into the HSA. This pool saves products in FITS format.
HCSS-HSA Writeable Pool	HsaWritePool accessible through a HsaPoolDaemon.
HCSS-HSA Writable Pool Application	Application that writes data into the pool (HsaWritePool) that shall be used to request the ingestion of products into the HSA system.
HsaReadPool	PAL pool for reading data that has been already ingested into the HSA.
HcssHuiTool	HCSS tool for intercommunication between HUI and IA.
Samp HUB	Samp Message Broker. It will be used for communicating HUI and HcssHuiTool.
Ingestion Storage Area	Shared area where products can be accessed by HSA and written by HCSS.
Ingestion Request Area	Shared area between HCSS and HSA where requests for ingestion are saved.
On-demand Storage Area	Shared area where HCSS writes the on-demand reprocessed products and HSA can access to them.
On-demand Request Area	Shared area between HCSS and HSA where requests for ingestion are saved
Ingestion Request File	XML file that specifies the products that must be ingested into HSA.
Ingestion Request Error File	XML file that specifies the error found when processing an ingestion request.
On-demand Request File	XML file that specifies the parameters for an on-demand processing.
On-demand Result File	XML file that specifies the results of an on-demand processing.
Reprocessing Profile	Observation reprocessing parameter set identifier.
HUI On-demand Monitoring Tool	HUI Tool for monitoring on-demand reprocessed observations
HAIO	HSA Inter-operability Subsystem for access from external applications to data and metadata from the archive
HSTAGE	Disk area where the products to be imported in the archive are stored temporarily during the import process
HIMPORT	HSA subsystem to copy product files from the stage area to the final repository
HINGEST	HSA subsystem that ingest observation/proposal/product metadata into the HSA database
HDB	HSA Relational Data Base

---

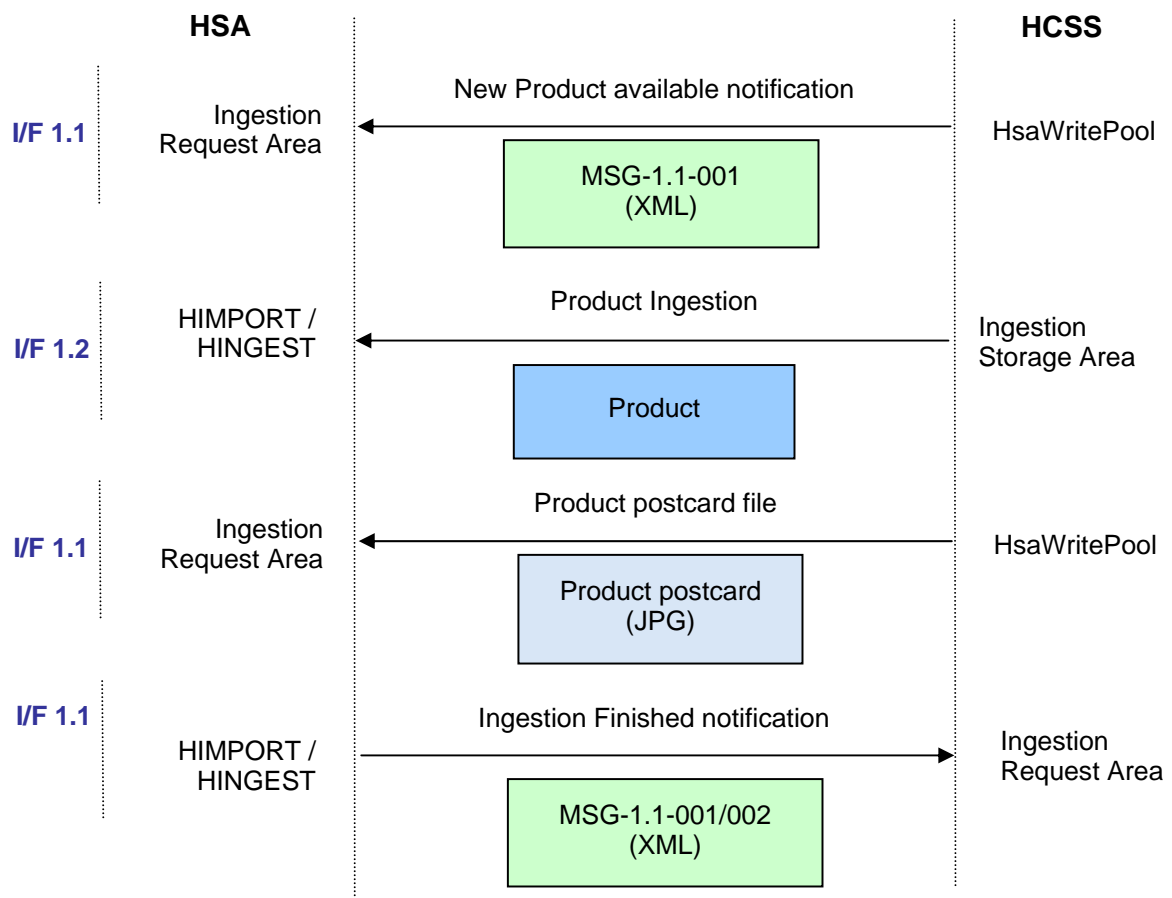
HDD	HSA subsystem on charge of distribution of product files and packages of products files to other HSA subsystems
QM	HSA middle-tier for connection to the HSA data base to access metadata
HAIO Session Identifier	HTTP cookie holding the session identifier between client (HCSS) and server (HSA) on HAIO connections.
Virtual tar	Tar file created by HAIO streamed through an HTTP connection.



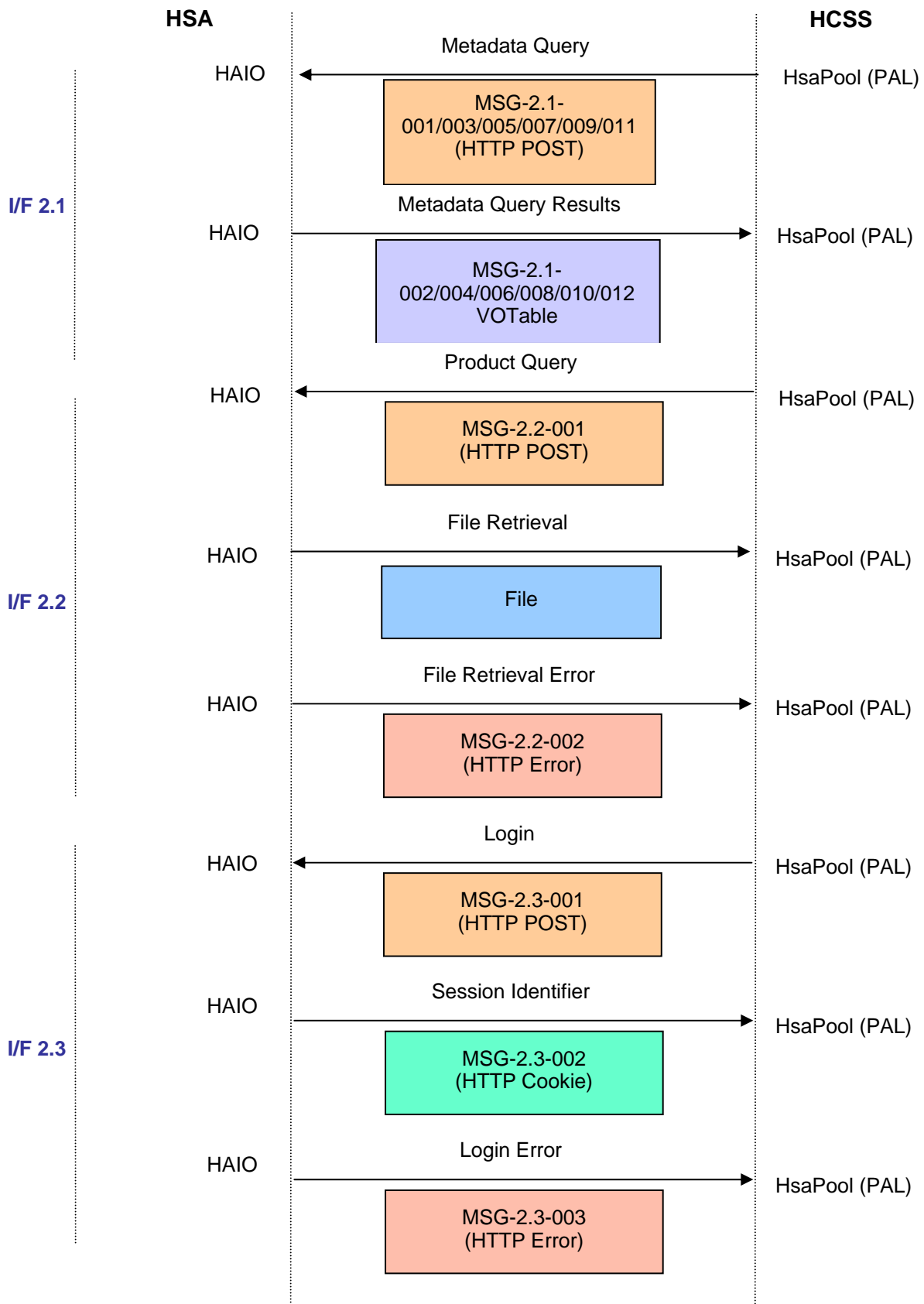
## 2.2 Objects

This section explains, for each interface, what the exchanged objects are and how they are exchanged. The following diagrams *do not show* any interchange object *sequence*.

### 2.2.1 Product Ingestion (I/F-1)

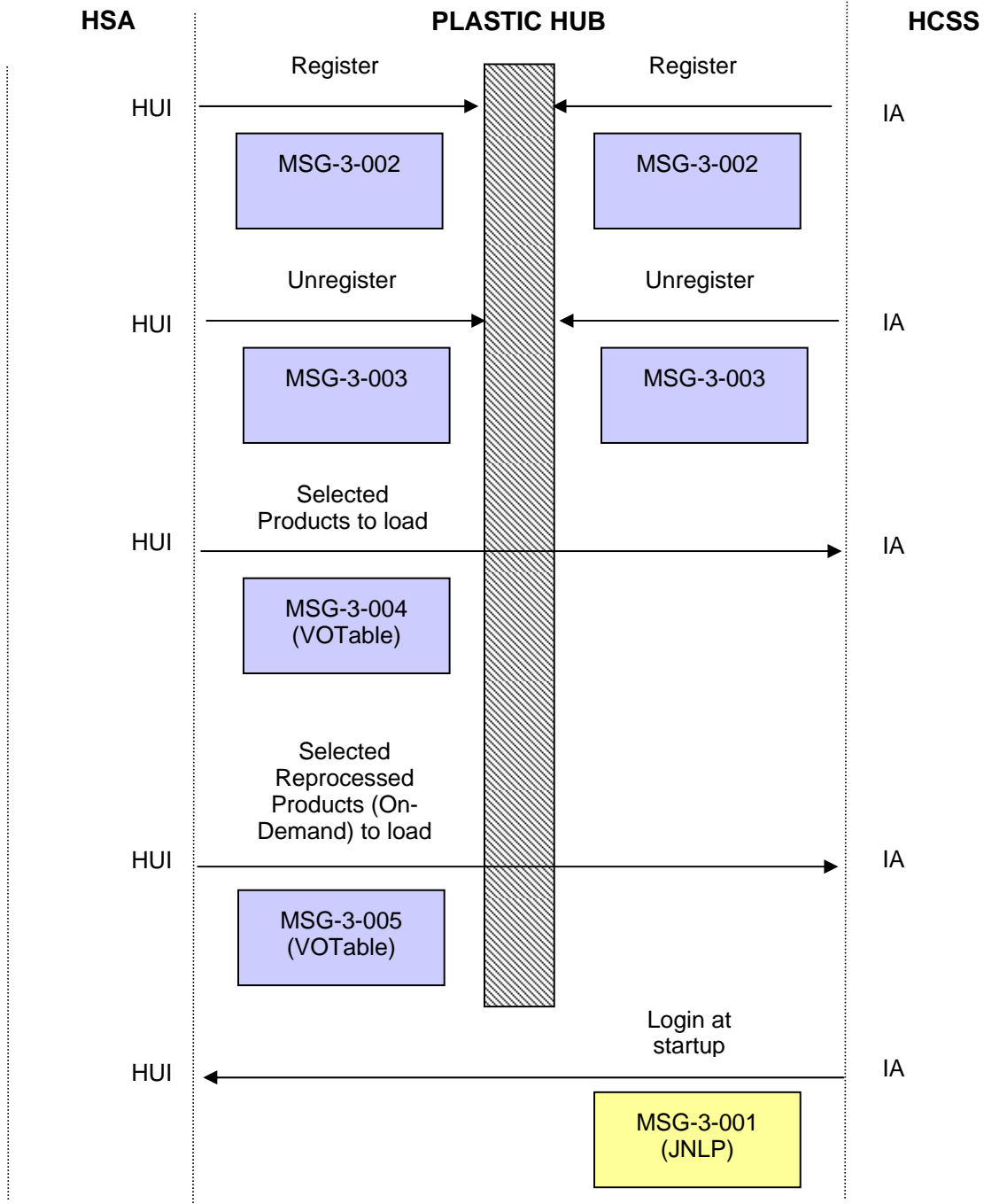


## 2.2.2 Direct HAIO access (I/F-2)

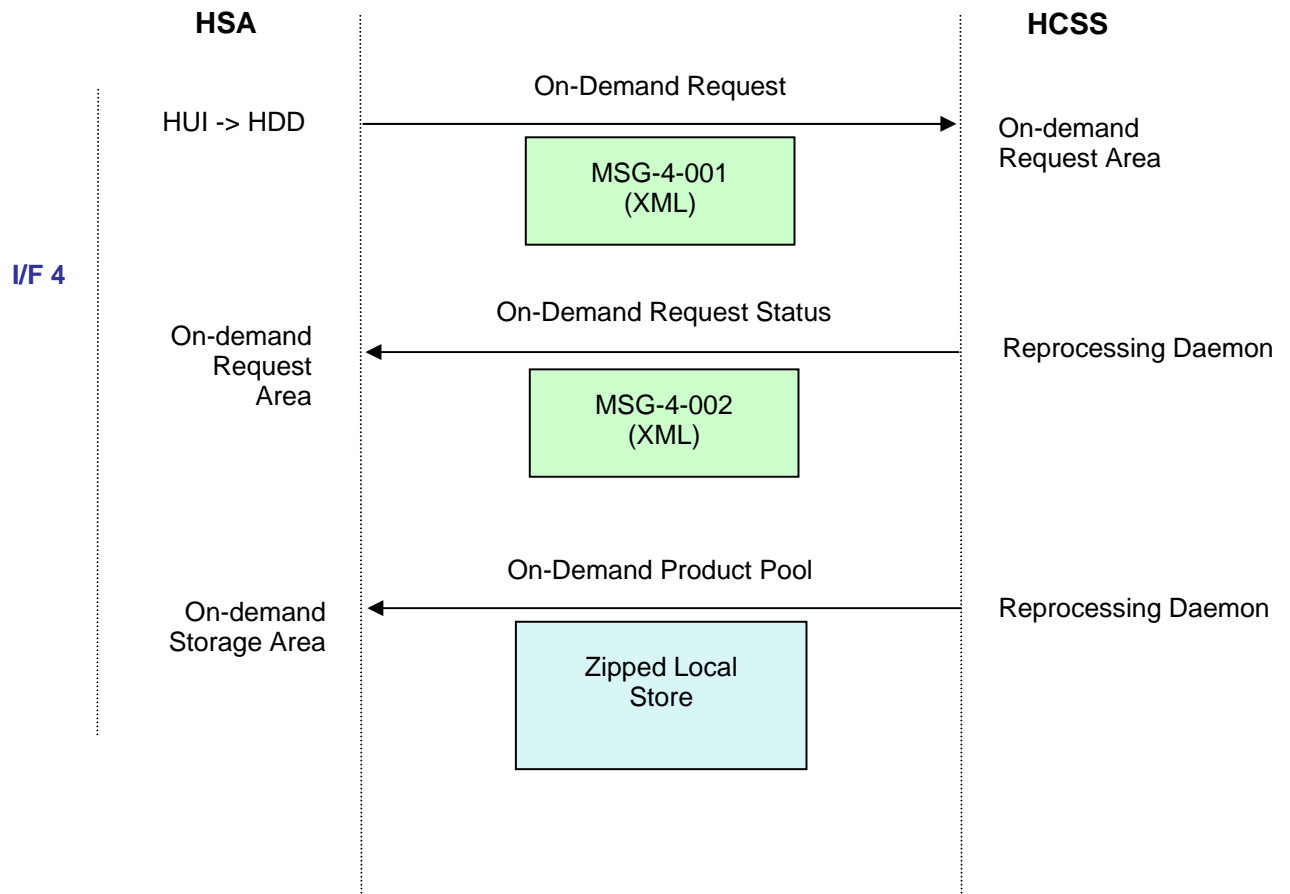


### 2.2.3 IA – HUI (I/F-3)

I/F 3



### 2.2.4 SPG On-demand reprocessing (I/F-4)



## 2.2.5 Ingestion Storage Area

Shared file system area between HCSS and HSA where HCSS products are stored and it is accessible by HSA when importing them.

The file system sub-directory for the Ingestion Storage Area is named **hsa**:

**<Ingestion Storage Area>/hsa**

## 2.2.6 Ingestion Request Area

Shared file system area between HCSS and HSA where requests for ingestion files (see MSG-1.1-001/002) are saved. It implies some other areas/directories:

- Requests: ingestion requests (MSG-1.1-001 are saved here).
- Processing: ingestion request that are being processed by the HSA ingestion process (MSG-1.1-001 are saved here.)
- Importing: on going imports into HSA.
- Processed: processed ingestion requests without errors.
- Failed: processed ingestion requests with errors (an Ingestion Request Error file is created, see MSG-1.1-002).
- Cleaned-up: processed ingestion requests whose products have been removed locally.
- Images: postcards images associated to an observation.

The file system sub-directories for the Ingestion Request area are:

<b>Area</b>	<b>Folder name</b>	<b>File System sub-directory name</b>	<b>Comments</b>
Requests	<b>requests</b>	<b>&lt;Ingestion Request Area&gt;/requests</b>	Ingestion requests
Processing	<b>processing</b>	<b>&lt;Ingestion Request Area&gt;/processing</b>	On going ingestions
Importing	<b>importing</b>	<b>&lt;Ingestion Request Area&gt;/importing</b>	On going imports
Processed	<b>processed</b>	<b>&lt;Ingestion Request Area&gt;/processed</b>	Successful ingestions
Failed	<b>failed</b>	<b>&lt;Ingestion Request Area&gt;/failed</b>	Failed ingestions
Cleaned-up	<b>cleanedup</b>	<b>&lt;Ingestion Request Area&gt;/cleanedup</b>	This area is not used by HSA.
Images	<b>images</b>	<b>&lt;Ingestion Request Area&gt;/images</b>	Postcards images in JPG format.

## 2.2.7 On-Demand Storage Area

Shared file system area between HCSS and HSA where on-demand reprocessed observations are available.

The file system sub-directory for the On-Demand Storage Area is named **products**:

<On-demand Storage Area>/products

## 2.2.8 On-Demand Request Area

Shared file system area between HCSS and HSA where requests for On-Demand Request and Results files (see MSG-4-001/002) are saved. It implies some other areas/directories:

- Requests: on-demand requests (MSG-4-001 are saved here).
- Processing: on-demand requests that are being processed by the SPG (MSG-4-001 are saved here.)
- Processed: processed on-demand requests (MSG-4-001) are saved here when the process is finished (an On-Demand Results file is created also, see MSG-4-002).

The file system sub-directories for the On-Demand Request area are:

<b>Area</b>	<b>Folder name</b>	<b>File System sub-directory name</b>	<b>Comments</b>
Requests	<b>requests</b>	<On-demand Request Area>/requests	
Processing	<b>processing</b>	<On-demand Request Area>/processing	On going requests
Processed	<b>processed</b>	<On-demand Request Area>/processed	Successful requests

## 2.2.9 Ingestion Request File

This object is an xml file. It contains the product (and child products) that should be ingested into HSA.

The file name shall follow some rules specified in Appendix B (see 10.1)

The xml file shall contain (see Appendix B, 10.1):

- The Observation Identifier for Observations. Empty for Common products.

- Path to the root storage area.
- The SPG version identifier.
- The HCSS URN of the first product to be ingested.
- The ingestion type (observation or common product).
- The parallel mode flag.
- The product and all the products that belong to the first one (a tree of products)
- For each product:
  - HCSS class type
  - Description
  - Relative path (to the root storage area)
  - HCSS URN
  - HCSS versioning metadata
    - Size
    - Total size
    - URN
    - URN version
    - Track identifier
    - Track version
    - Hash identifier
    - Class type

OI-2.2.9-1, OI-2.2.9-2

### ***2.2.10 Ingestion Request Error File***

This object is an xml file. It contains the error(s) found when HSA was processing an Ingestion Request file.

The file name shall follow some rules specified in Appendix B (see 10.2)

The xml file shall contain (see Appendix B, 10.2):

- The Observation Identifier for Observations. Empty for Common products.
- The SPG version identifier.
- The HCSS URN of the first product to be ingested.
- The ingestion type (observation or common product)
- The error message

OI-2.2.10-1

### ***2.2.11 On-demand Request File***

This object is an xml file. It contains the specification of the observation that should be reprocessed.

The file name shall follow some rules specified in Appendix B (see 10.3)

The xml file shall contain (see Appendix B, 10.3):

- HSA On-Demand Reprocessing Request Identifier.
- HCSS Reprocessing Profile Identifier.
- Observation Identifier.
- Instrument Identifier.

### ***2.2.12 On-demand Request Result File***

This object is an xml file. It contains the results of the reprocessing of an On-Demand Request file.

The file name shall follow some rules specified in Appendix B (see 10.3)

The xml file shall contain (see Appendix B, 10.3):

- HSA On-Demand Reprocessing Request Identifier.
- HCSS reprocessing profile identifier.
- Observation Identifier.
- Instrument Identifier.
- Status
- Pointer to processed data (observational products)
- Message (optional)

### ***2.2.13 HSA tarball***

This object is a tarball file which contains the selected observation (or product level, auxiliary products, etc) that has been selected through the HUI.

The structure of this file is specified in Appendix F. See Appendix B, section 10.1 also.

(See Virtual Tar object also.)



### ***2.2.14 Product Postcard File***

This object is a JPG file. It contains the product postcard associated to an observation and should be ingested into HSA. The file name shall follow some rules specified in Appendix B (see 10.1)

### ***2.2.15 Virtual Tar***

This object is a TAR file that contains the specified URNs requested in an HAIO product query. The structure is specified in Appendix I.

(See HSA tarball object also.)

## 2.3 *Message formats*

### 2.3.1 *Product Ingestion (I/F-1)*

#### 2.3.1.1 **MSG-1.1-001: Ingestion Request**

**Interface:** I/F-1.1

This message contains an Ingestion Request file object (see 2.2.9 and Appendix B, 10.1)

#### 2.3.1.2 **MSG-1.1-002: Ingestion Request error**

**Interface:** I/F-1.1

This message contains an Ingestion Request Error file object (see 2.2.10 and Appendix B, 10.2)

### 2.3.2 *Direct HAIO Access (I/F-2)*

#### 2.3.2.1 **MSG-2.1-001/002: HAIO Product Metadata Queries**

**Interface:** I/F-2.1

HAIO queries are requests for finding HSA stored products which products metadata match with an expression that involves relationships between metadata and metadata values. The expression shall be a sentence similar to a Jython boolean expression (see [RD-1]).

(See HAIO Observation Metadata Queries (section 2.3.2.6) also).

The HAIO query and its response are the following messages:

- **MSG-2.1-001 Request:** an HTTP GET Request which contains the metadata query. The value for this Query parameter should be a valid Jython Boolean expression encoded through the standard URL encoding as per RFC 2396: Uniform Resource Identifiers (URI): Generic Syntax.

<i>Parameter</i>	<i>Value</i>	<i>Comments</i>
<b>RESOURCE_CLASS</b>	PRODUCT	

<i>Parameter</i>	<i>Value</i>	<i>Comments</i>
<b>QUERY</b>	Metadata query	A valid Jython boolean expression encoded through standard URL encoding (RFC-2396)
<b>LAST_TRACK_VERSION</b>	true	Optional. True means that the results must contain only latest versions (grouped by HCSS_TRACK_ID and HCSS_TRACK_VERSION)
<b>LIMIT</b>	Maximum results allowed	Optional. If it is specified and the number of results are greater than the specified value, a VOTable containing an error is returned.
<b>SHOW_PROPRIETARY</b>	true	Optional. True means that each VOTable record will contain a field that specifies proprietary rights access.

- **MSG-2.1-002 Response:** a VOTable containing one or more records (see the format in Appendix C: Product Metadata VOTable response). A record is composed of the following parameters:

<i>UCD</i>	<i>Value</i>	<i>Comments</i>
<b>DATA_LINK</b>	HSA_DATA_LINK	URL to be invoked to download the product
<b>HCSS_URN</b>	HCSS_URN	Product metadata
<b>HCSS_FILE_NAME</b>	HCSS_FILE_NAME	Product metadata
<b>HCSS_CLASS_TYPE</b>	HCSS_CLASS_TYPE	Product metadata
<b>HCSS_URN_VERSION</b>	HCSS_URN_VERSION	Product metadata
<b>HCSS_TRACK_ID</b>	HCSS_TRACK_ID	Product metadata
<b>HCSS_TRACK_VERSION</b>	HCSS_TRACK_VERSION	Product metadata
<b>HCSS_SIZE</b>	HCSS_SIZE	Product metadata
<b>HCSS_TOTAL_SIZE</b>	HCSS_TOTAL_SIZE	Product metadata
<b>HCSS_HASH</b>	HCSS_HASH	Product metadata

<i>UCD</i>	<i>Value</i>	<i>Comments</i>
<a href="#">HAS_ACCESS</a>	true/false	Optional. It will appear only if the user requests it. 'True' means that the user can access to the product specified by <a href="#">DATA_LINK</a> .

The VOTable response contains an [info](#) item that shows the response status. See Appendix H.

The allowed query operators are specified in Appendix M.

### 2.3.2.2 **MSG-2.1-003/004: HAIO HCSS\_CLASS\_TYPE Resource Class Query**

**Interface:** I/F-2.1

HAIO HCSS\_CLASS\_TYPE queries are used to obtain information about the HCSS\_CLASS\_TYPE metadata.

These queries are composed by two messages:

- **MSG-2.1-003 Request:** it is an HTTP GET Request with the following parameters:

<i>Parameter</i>	<i>Value</i>	<i>Comments</i>
<a href="#">RESOURCE_CLASS</a>	HCSS_CLASS_TYPE	
<a href="#">HCSS_CLASS_TYPE</a>	<ul style="list-style-type: none"> <li>• Not present: all HCSS_CLASS_TYPE stored in HSA are requested.</li> <li>• None: (not useful) retrieve items with HCSS_CLASS_TYPE empty (it must not occur)</li> <li>• A value (a HCSS class type): retrieve the greatest HCSS_URN_VERSION of the specified class type.</li> </ul>	This parameter must not be present, in the HTTP-GET query, if a list of already stored HCSS_CLASS_TYPE values are requested.

- **MSG-2.1-004 Response:** VOTable response containing zero, one or more records. Each record is composed by the following parameters.

<i>UCD</i>	<i>Value</i>	<i>Comments</i>
<b>HCSS_CLASS_TYPE</b>	HCSS_CLASS_TYPE product metadata	
<b>LAST_VERSION</b>	Greatest value of VERSION_NUMBER metadata for a HCSS_CLASS_TYPE metadata	If no <b>HCSS_CLASS_TYPE</b> request parameter is specified, the value of this field is the greatest value of version number for this hcsc class type

- If the HCSS\_CLASS\_TYPE parameter was not specified in the request, the VOTable shall contain a record by each different HCSS\_CLASS\_TYPE metadata value (stored in HSA). The VOTable will be empty if no products are stored into the HSA system.
- If the HCSS\_CLASS\_TYPE parameter was specified in the request, the VOTable shall contain one record if the HCSS\_CLASS\_TYPE parameter value was found in the HSA system. The VOTable will be empty if the HCSS\_CLASS\_TYPE value was not found.

### 2.3.2.3 **MSG-2.1-005/006: HAI0 PRODUCT Resource Class Query**

**Interface:** I/F-2.1

HAI0 Product queries are used to obtain information about the Product file name or Product HCSS URN searching into the HSA system by the Product HCSS URN or Product HCSS file name.

These queries are composed by two messages:

- **MSG-2.1-005 Request:** it is an HTTP GET Request with the following parameters:

<i>Parameter</i>	<i>Value</i>	<i>Comments</i>
<b>RESOURCE_CLASS</b>	PRODUCT	
<b>urn</b>	HCSS_URN product metadata	
<b>FILENAME</b>	HCSS_FILE_NAME product metadata	
<b>HCSS_CLASS_TYPE</b>	HCSS_CLASS_TYPE product metadata	

- **MSG-2.1-006 Response:** VOTable response containing zero or one record (see Appendix C). A record is composed of the following parameters.

<b>UCD</b>	<b>Value</b>	<b>Comments</b>
<b>DATA_LINK</b>	I DATA_LINK	URL to be invoked to download the product
<b>HCSS_URN</b>	HCSS_URN	Product metadata
<b>HCSS_FILE_NAME</b>	HCSS_FILE_NAME	Product metadata
<b>HCSS_CLASS_TYPE</b>	HCSS_CLASS_TYPE	Product metadata
<b>HCSS_URN_VERSION</b>	HCSS_URN_VERSION	Product metadata
<b>HCSS_TRACK_ID</b>	HCSS_TRACK_ID	Product metadata
<b>HCSS_TRACK_VERSION</b>	HCSS_TRACK_VERSION	Product metadata
<b>HCSS_SIZE</b>	HCSS_SIZE	Product metadata
<b>HCSS_TOTAL_SIZE</b>	HCSS_TOTAL_SIZE	Product metadata
<b>HCSS_HASH</b>	HCSS_HASH	Product metadata

- VOTable shall contain a record if a product that matches with the specified request parameter values (**urn**, **FILENAME** or **HCSS\_CLASS\_TYPE**) is found in the HSA system.

#### 2.3.2.4 **MSG-2.1-007/008: HAIO HCSS\_TRACK\_ID Resource Class Query**

**Interface:** I/F-2.1

HAIO HCSS\_TRACK\_ID queries are used to obtain information about the HCSS\_TRACK\_ID metadata.

These queries are composed by two messages:

- **MSG-2.1-007 Request:** it is an HTTP GET Request with the following parameters:

<b>Parameter</b>	<b>Value</b>	<b>Comments</b>
<b>RESOURCE_CLASS</b>	HCSS_TRACK_ID	

<i>Parameter</i>	<i>Value</i>	<i>Comments</i>
<a href="#">HCSS_TRACK_ID</a>	Product HCSS Track identifier	Retrieve the greatest HCSS_TRACK_VERSION of the specified track identifier.  If this parameter is not provided, the greatest HCSS_TRACK_VERSION by each HCSS_TRACK_ID is returned.  If a HCSS_CLASS_TYPE is specified, the request is applied to the specified HCSS_CLASS_TYPE.
<a href="#">HCSS_CLASS_TYPE</a>	Product HCSS Class type.	

- **MSG-2.1-008 Response:** VOTable response containing zero or one record corresponding to the greatest HCSS\_TRACK\_VERSION that matches to the specified HCSS\_TRACK\_ID (if provided) and HCSS\_CLASS\_TYPE (if provided). The content is the same as the specified in MSG-2.1-006

### 2.3.2.5 **MSG-2.1-009/010: HAIO PRODUCT\_METADATA Resource Class Query**

**Interface:** I/F-2.1

HAIO PRODUCT\_METADATA queries are used to obtain all the ingested metadata associated to a product.

These queries are composed by two messages:

- **MSG-2.1-009 Request:** it is an HTTP GET Request with the following parameters:

<i>Parameter</i>	<i>Value</i>	<i>Comments</i>
<a href="#">RESOURCE_CLASS</a>	PRODUCT_METADATA	
<a href="#">urn</a>	Product HCSS URN	

- **MSG-2.1-010 Response:** VOTable response containing zero or one record corresponding to the metadata associated to a product (see Appendix C).

### 2.3.2.6 **MSG-2.1-011/012: HAIO Observation Metadata Queries**

**Interface:** I/F-2.1

HAIO Observation Metadata Queries are requests for finding HSA stored observation which observations metadata match with an expression that involves relationships between metadata and metadata values. The expression shall be a sentence similar to a Jython boolean expression (see [RD-1]).

These requests are similar to *HAIO Product Metadata Queries* (section 2.3.2.1). Nevertheless, the request parameters and the VOTable response contents are different.

The HAIO query and its response are the following messages:

- **MSG-2.1-011 Request:** an HTTP GET Request which contains the metadata query. The value for this Query parameter should be a valid Jython Boolean expression encoded through the standard URL encoding as per RFC 2396: Uniform Resource Identifiers (URI): Generic Syntax.

<i>Parameter</i>	<i>Value</i>	<i>Comments</i>
<b>RESOURCE_CLASS</b>	OBSERVATION	
<b>QUERY</b>	Metadata query	A valid Jython boolean expression encoded through standard URL encoding (RFC-2396)
<b>LIMIT</b>	Maximum results allowed	Optional. If it is specified and the number of results are greater than the specified value, a VOTable containing an error is returned.

- **MSG-2.1-012 Response:** a VOTable containing zero, one or more records (see the format in Appendix C: Observation Metadata VOTable response). A record is composed of the following parameters:

<i>UCD</i>	<i>Value</i>	<i>Comments</i>
<b>DATA_LINK</b>	HSA DATA_LINK	URL to be invoked to download the observation



<b>UCD</b>	<b>Value</b>	<b>Comments</b>
<b>OBS_ID</b>	ObsId	Product metadata
<b>URN_VERSION</b>	Urn_Version	Product metadata
<b>SPG_ID</b>	Spg_id	Product metadata
<b>INSTRUMENT</b>	Instrument	Product metadata
<b>OBSERVING_MODE</b>	Instrument_mode	Product metadata
<b>POS_EQ_RA_MAIN</b>	RA	Product metadata
<b>POS_EQ_DEC_MAIN</b>	DEC	Product metadata
<b>PA</b>	PA	Product metadata
<b>VOX:Image_Title</b>	Title	Product metadata
<b>VOX:OBS_START_TIME</b>	Start_Time	Product metadata
<b>VOX:OBS_END_TIME</b>	End_Time	Product metadata
<b>DURATION</b>	Duration	Product metadata
<b>PROPOSER</b>	Proposer	Product metadata
<b>VOX:EXPIRATION_DATE</b>	Expiration_Date	Product metadata
<b>OD_NUMBER</b>	OD_Number	Product metadata
<b>IS_ACTIVE_VERSION</b>	Is_Active_Version	Product metadata
<b>PROPOSAL_ID</b>	PROPOSAL_ID	Product metadata
<b>AOR</b>	Aor	Product metadata
<b>STATUS</b>	Status	Product metadata
<b>OBSSTATE</b>	Obs_State	Product metadata

The VOTable response contains an [info](#) item that shows the response status. See Appendix H.

The allowed query operators are specified in Appendix M.

### 2.3.2.7 **MSG-2.2-001: HAIO Product Request**

**Interface:** I/F-2.2

- **MSG-2.2-001 Request:** it is an HTTP GET Request with the following parameters:

<b>Parameter</b>	<b>Value</b>	<b>Comments</b>
<b>urn</b>	Product HCSS URN	Several URNs can be provided. If a single URNs is provided, the requested product is served through the HTTP connection. If more than one URN is provided, a Virtual TAR is generated and served through the HTTP connection.
<b>Protocol</b>	HTTP	If this parameter does not appear, an intermediate html page will be displayed in a human readable way
<b>metadata</b>	true	If this parameter does not appear, the already ingested product is returned. If this parameter appears and its value is 'true', a FITS file containing product metadata only is returned.
<b>requestfile_xml</b>	XML ingestion request file name.	If this parameter is present, <b>urn</b> and <b>metadata</b> parameters are discarded.
<b>OBSID</b>	Observation identifier	If <b>urn</b> parameter is present, <b>OBSID</b> parameter is discarded.
<b>PRODUCT_LEVEL</b>	The following is a list with the possible values: <ul style="list-style-type: none"><li>• All</li><li>• Auxiliary</li><li>• Calibration</li><li>• Level0</li><li>• Level0_5</li><li>• Level1</li><li>• Level2</li><li>• Level2_5</li></ul>	Used with <b>OBSID</b> parameter. If <b>PRODUCT_LEVEL</b> is All, an HSA Tarball (see Appendix F) with the complete observation is generated and served through HTTP connection. If <b>PRODUCT_LEVEL</b> is equal to any of Auxiliary, Calibration, Level0, Level0_5, etc., an HSA Tarball (see Appendix F) with the selected products is generated and served through HTTP connection.
<b>Session Identifier</b>	Identifier associated to a user generated after an authentication procedure (See MSG-2.3-001/002)	This parameter is an HTTP <i>cookie</i>

- Response:
  - If no error is found, the response is not a message but:
    - If **protocol=HTTP** is provided,
      - If a single **urn** is requested, the requested file will be included in the response in serialized binary form.
      - If several **urn's** are requested, a Virtual TAR is created and served.
    - If **protocol=HTTP** is not provided, an HTML page will be shown.
      - If a single **urn** is requested, the HTML page will show a link to a single product.
      - If several **urn's** are requested, the HTML page will show a link to a Virtual TAR file.
  - If there is an error: **MSG-2.2-002 Response:** An HTTP error with the following codes: See Appendix G: HAIO HTTP error codes, section 15.1.

### 2.3.2.8 **MSG-2.3-001/002: HAIO Login Request**

**Interface:** I/F-2.3

- **MSG-2.3-001 Request:** it is an HTTP GET Request with the following parameters:

<i>Parameter</i>	<i>Value</i>	<i>Comments</i>
<b>USERNAME</b>	User name	
<b>PASSWORD</b>	User password	
<b>Protocol</b>	HTTP	If this parameter does not appear, an intermediate html page will be displayed in a human readable way

- Response:
  - If no error is found:
    - **MSG-2.3-002 Response:** An HTTP response with a *cookie* in the HTTP header containing a **HAIO session identifier** associated to the communication between client (HCSS) and server (HSA).
  - If there is an error:
    - **MSG-2.3-003 Response:** An HTTP error with the following codes: See Appendix G: HAIO HTTP error codes, section 15.2

### 2.3.3 IA – HUI (I/F-3)

The “register/unregister client with Samp hub” messages, defined in paragraphs 2.3.3.2 and 2.3.3.3, are not exchanged between the IA application and the HUI, they are sent by IA session and by the HUI to the Samp hub and not broadcast. The Samp hub is the implementation of the message broker.

#### 2.3.3.1 MSG-3-001 Starting of the HUI

The IA application triggers the start-up of the HUI. This works by downloading the JNLP file from a previously agreed and fixed URL, and starting Java WebStart with this file.

Until version 6.x of the HCSS and version 3.6 of the HSA, the agreed URL was:

<http://archives.esac.esa.int/hsa/hsa.jnlp>

In HSA v3.7 the URL changed to:

<http://archives.esac.esa.int/hsa/ui/hui.jnlp>

However, the old URL will continue working by redirection at least until the end of 2011. The new URL was implemented in HCSS v7.0

If the log-in credentials are available in the IA session, that is, if the user has logged in, these credentials will be passed to the HUI. This happens by first writing a properties file with the following properties:

**`jnlp.esa.ldap.username=jdoe`**

**`jnlp.esa.ldap.password=passwd jdoe`**

This properties file shall be in ISO 8859-1 character encoding. Characters in the username and/or password, that cannot be directly represented in this encoding, are written using Unicode escapes; only a single 'u' character is allowed in an escape sequence. This is consistent with the behaviour of the Java methods `load(InputStream) / store(OutputStream, String)` in the `java.util.Properties` class.

Next, a property is added to the `<resources>` section of the JNLP file:

```
<resources>
  (... resources...)
  <property name="J n l p . e s a . l d a p . c r e d e n t i a l s . f i l e" v a l u e=" t h e V a l u e" />
</resources>
```

The value of this property must be the URL of the location of the properties file, e.g.

<file://localhost/tmp/credentials.properties>

Should any of these properties be missing, or contain an unexpected value, then the HUI shall start up without logging in.

### **2.3.3.2** *MSG-3-002 Register client with message broker*

**Interface:** I/F-3

Default Samp message registering the client with the message broker (the Samp hub). This message is default in the sense that it is provided in the Samp distribution already, and does not need to be implemented or specified by Herschel or the HSA. For a list of message definitions, including the ones referred to in this document, see [RD-3].

The application ID for the IA session is “HIPE” and the application ID for the HUI is “HerschelArchiveUI”, both without double quotes.

### **2.3.3.3** *MSG-3-003 Unregister client with message broker*

**Interface:** I/F-3

Default Samp message unregistering the client with the message broker. This is sent at application shutdown by both the IA session and HUI.

### **2.3.3.4** *MSG-3-004 Request loading observations selected in HUI*

**Interface:** I/F-3

The message is a default load VO table message with message ID (see [RD-3]):

<ivo://votech.org/votable/load>

The VO table will contain a list of products, in the same format as specified by the example in section 11.6 “HAIO Product Resource Class VOTable response”.

### **2.3.3.5** *MSG-3-005 Request loading data from on-demand reprocessing*

**Interface:** I/F-3

The message is a default load VO table message with message ID (see [RD-3]):

<ivo://votech.org/votable/load>

The VO table will effectively contain a (list of) URLs, pointing to data (**Zippered Local Store** file) on an HSA FTP server. This data is the output of the on-demand reprocessing job.

The fields of the VO table are similar to the ones indicated in section 11.6. They are:

<i>ID</i>	<i>UCD</i>	<i>datatype</i>	<i>arraysize</i>
<b>Reference</b>	DATA_LINK	char	*
<b>OnDemandJobIdentifier</b>	REQUEST_ID	integer	-
<b>UserName</b>	USER_NAME	char	*
<b>ExitStatus</b>	REQUEST_STATUS	char	*

## **2.3.4 SPG On-demand reprocessing (I/F-4)**

### **2.3.4.1 MSG-4-001/002: Request an On-Demand Reprocessing**

**Interface:** I/F-4

- **MSG-4-001 Request:** this message shall contain an On-Demand Request file object (see 2.2.11 and Appendix B, 10.3)
- **MSG-4-002 Response:** this message shall contain an On-Demand Request Results file object (see 2.2.12 and Appendix B, 10.4).

### 3 ROLES

The HSA-HCSS interfaces will be invoked through two different roles:

- **HSA role:** User of the HSA system. For instance, HCSS-HsaReadPool will play an HSA role when querying to the HSA (through HAIO), when a product is requested to HSA or when HCSS-SPG notifies HSA because a new observation has been processed. Another example is the regular download of all products created since the last download by ICCs (bulk product transfer, see [RD-5]).
- **HCSS-INGESTOR role:** User of the HCSS-HSA writeable pool. For instance, HSA when ingesting products from HsaWritePool.

## 4 USE CASES

### OI-4-1

#### 4.1 UC-001: Product Ingestion

**System:** HSA

**Actor:** Calibration scientist, Instrument engineer

**Interfaces:** I/F-1.1 and I/F-1.2

##### 4.1.1 Main

1. Actor saves a product into the HCSS-HSA writable pool.
2. Actor requests ingestion (**BR-008**).
3. System creates an Ingestion Request file in the Ingestion Request Area (**BR-001**).
4. System retrieves the Ingestion Request file (**BR-004**) and obtains the products to be ingested.
5. System moves the Ingestion Request file into the processing Ingestion Area (**BR-001**).
6. System retrieves the product from the Ingestion Storage Area (**BR-003, IR-001**).
7. System retrieves the product postcard from the Ingestion Storage Area (**BR-003, IR-002**).
8. If product does not exist in the system, ingest it:
  - a. System ingest METADATA product (**BR-005**)
  - b. System ingest FITS file representation of the product (**IR-001**)
  - c. System ingest control METADATA associated to the product (**BR-006**)
9. If the Ingestion Request File requests an Observation ingestion:
  - a. Update product hierarchical reference for tarball structure (see Appendix 14).
10. System repeats steps 4, 5, 6, 7, 8 and 9 for every child of the product.
11. System moves the Ingestion Request file to the final folder (**BR-001, BR-002, BR-007**).

##### 4.1.2 Extensions

- **OI-4.1.2-\*** If an error is found when processing an ingestion, a file with the error(s) must be created in the 'failed' Ingestion Area folder (**BR-007**).

##### 4.1.3 Business Rules

1. **BR-001:** Ingestion Request file: see MSG-1.1-001.
2. **BR-002:** Ingestion Request Error file: see MSG-1.1-002.
3. **BR-003:** The path is obtained from the Ingestion Request file.
4. **BR-004:** A pooling mechanism is used.
5. **BR-005:** METADATA product to ingest: Each product type (class) has it owns metadata to be ingested. See Appendix A. Each product shall have versioning METADATA (HCSS PAL versioning information). See Appendix A, 9.4.
6. **BR-006:** HSA control metadata:



- a. each product that belongs to an observation shall have the following control metadata (see Appendix A, 9.5):
    - i. obsid: observation identifier value.
    - ii. spgid: SPG software identifier.
  - b. Each product that does not belong to an observation shall have the following control metadata (see Appendix A, 9.5):
    - i. **OI-4.1.3-1**
7. **BR-007:** If an error is found, the Ingestion Request file must be moved to the ‘failed’ Ingestion Area folder and an Ingestion Request Error file must be created with the error. If no errors are found, the Ingestion Request file must be moved to the ‘processed’ Ingestion Area folder.
  8. **BR-008:** If an extended processing ingestion is requested, a verification procedure is required: See Appendix K.

#### 4.1.4 Implementation requirements

1. **IR-001:** Products in the Ingestion Storage area are saved in FITS format. See [AD-6].
2. **IR-002:** Products postcards are saved in JPG format.

## 4.2 *UC-002: Search by Metadata*

**System:** HSA

**Actor:** Astronomer, Calibration scientist, Instrument engineer, Scientific Product Analyst

**Interface:** I/F-2.1

### 4.2.1 Main

1. Actor builds the query (**BR-001**)
2. Actor sends query.
3. System processes query
4. System returns query results (**BR-002, BR-003**)

### 4.2.2 Extensions

#### OI-4.2.2-1

2. Errors sending query:
  - a. Actor receives an error if the query cannot be sent.
3. Errors processing query:
  - a. Actor receives an error if the query is not valid.
4. Errors returning query:
  - a. Connection timeout
  - b. Invalid returned data format

### 4.2.3 Business Rules

1. **BR-001:** HTTP-GET query: see messages MSG-2.1-001/003/005/007/009/011.
2. **BR-002:** VOTable specification: see messages MSG-2.1-002/004/006/008/010/012.
3. **BR-003:** HAIO will provide any results of any stored version. HUI will provide, by default, the results associated to the latest version.

### 4.2.4 Implementation requirements

1. **IR-001:** Sending queries must be implemented according to the HTTP Protocol (see [RD-3]).

## 4.3 UC-003: File Retrieval

**System:** HSA

**Actor:** Astronomer, Calibration scientist, Instrument engineer, Scientific Product Analyst

**Interface:** I/F-2.2 and I/F-2.3

### 4.3.1 Main

1. Actor sends a retrieval request (**BR-001, BR-003, BR-004**)
2. System returns the requested file (**BR-002**)

### 4.3.2 Extensions

#### OI-4.3.2-1

1. Errors sending request:
  - a. Actor receives an error if the request cannot be sent (see **BR-004**).
2. Errors processing request:
  - a. Actor receives an error if the request is not valid (see **BR-004**).

### 4.3.3 Business Rules

1. **BR-001:** Request format (see message MSG-2.2-001):
  - a. Product identifier: HCSS URN (more than one HCSS URN can be provided)
  - b. Metadata flag.
  - c. Xml file.
  - d. Session Identifier.
2. **BR-002:**
  - a. If urn parameter is present (For a FITS specification see [AD-6]):
    - i. If metadata flag parameter is not present, the returned file is the already ingested product in FITS format.
    - ii. If metadata flag parameter is also present and its value is 'true', the returned file is a FITS file containing one HDU which is a copy of the first product HDU. (The file name of this one HDU FITS file shall be the same as the original product file name.)
    - iii. If metadata flag parameter is not present and more than one HCSS URN are provided, a Virtual TAR (see Appendix I) file is returned, containing the already ingested products.
    - iv. If metadata flag parameter is present, and more than one HCSS URN are provided, the generated Virtual TAR shall contain a FITS file per product. Each file shall contain one HDU which is a copy of the first product HDU. (Each file name shall be the same as the original product file name).
  - b. If Xml file parameter is present (urn and metadata flag parameters are discarded), the returned file is an XML file.

3. **BR-003:** An authenticated user is required for accessing restricted products. If the user is already authenticated, it is not necessary to ask for login again. If the user is not authenticated, a login request must be sent. (See login request format: MSG-2.3-001).
4. **BR-004:** If there is a login error, a MSG-2.3-003 will be sent. If there is a product retrieval error, a MSG-2.2-002 will be sent.

## 4.4 *UC-004: On-Demand Reprocessing Request-Response*

**System:** HSA

**Actor:** Astronomer, Calibration scientist, Instrument engineer, Scientific Product Analyst

**Interfaces:** I/F-3 and I/F-4

### 4.4.1 Main

1. Actor sends an On-Demand Request (**BR-001**)
2. System returns the request status (**BR-002**)
3. System creates a zipped local store file with the results (**BR-003**)
4. System notifies Actor (**BR-004**)
5. Actor retrieves data (**BR-005**)

### 4.4.2 Extensions

#### OI-4.4.2-1

1. Errors sending request:
  - a. Actor receives an error if the request cannot be sent.
2. Errors processing request:
  - a. Actor receives an error if the request is not valid.
3. Errors creating zipped local store:
  - a. If an error occurs when processing the observation or when creating the zipped file, the actor receives the error in the notification.
4. Errors notifying actor:
  - a. Actor receives an error if the request is not valid.
5. Errors retrieving data:
  - a. Actor receives an error if the request is not valid.

### 4.4.3 Business Rules

1. **BR-001:** The following parameters must be specified:
  - a. HSA On-Demand Reprocessing Request Identifier.
  - b. HCSS Reprocessing Profile Identifier.
  - c. Observation Identifier.
  - d. Instrument Identifier.
2. **BR-002:** Request status: System informs to the user if the request is accepted.
3. **BR-003:** The zipped local store file only contains new Contexts and/or Products. Non modified Contexts and/or Products that are already ingested into the Archive are not part of this zipped local store file. This file is not created if the On-Demand process cannot start due to a wrong request.

4. **BR-004:** At the end of the reprocessing process, System sends an email to Actor with the current status (successful / errors). Actor can look the HUI On-demand Monitoring Tool also to verify the On-demand reprocessing status **OI-4.4.3-1**.
5. **BR-005:** Data is available through a FTP connection (the FTP access is not part of this document) or in an IA session (I/F-3).

## 5 DESIGN DESCRIPTIONS

### 5.1 *Product Ingestion (I/F-1)*

**Interfaces:** I/F-1.1 and I/F-1.2

**Messages:** MSG-1.1-001/002

A shared Ingestion Request area between HCSS and HSA must be created to handle the Ingestion Request and the Ingestion Request Error files.

The Ingestion Request Area shall contain different subdirectories (see 2.2.6).

The Ingestion Storage area must be another shared area between HCSS and HSA. In this area, the products shall be accessible by both systems.

The products and postcards that must be ingested shall be described by an xml file (Ingestion Request file).

A HCSS Product version is identified by a HCSS URN. The HCSS Software used for creating a product is provided by the SPG software identifier, which is included in the Ingestion Request file.

The HSA shall use a pooling mechanism to find new ingestion requests.

The errors found when ingesting products shall be saved in another xml file (Ingestion Request Error file).

The HSA shall move the Ingestion Request file to the suitable Ingestion Request Area subdirectory during the ingestion procedure.

The HSA shall access to the Ingestion Storage area to retrieve the product files (these files are FITS). The HSA shall access to the Ingestion Request Area to retrieve product postcards.

Usually, HSA will contain different HCSS Product versions belonging to the same observation. It means that, with the same observation identifier, there will be, for instance, several Observation Contexts (each one, will have a unique HCSS URN, even among different SPG software identifiers).

It is an error to create an Ingestion Request File observation (or common product) already ingested

unless the HCSS URN of the observation (or common product) is different. It means that there will not be an Ingestion Request File for the same HCSS URN. It does not mean that a Ingestion Request File requests an ingestion of already ingested HCSS Products/Contexts.

The HSA shall check the access restrictions when an extended processing ingestion is requested (see Appendix K).

## 5.2 *Direct HAIO Access (I/F-2)*

**Interfaces:** I/F-2.1, I/F-2.2 and I/F-2.3

**Messages:** MSG-2.1-001/002/003/004/005/006/007/008/009/010/011/012, MSG-2.2-001/002 and MSG-2.3-001/002/003

The HAIO server can handle queries to retrieve files (see MSG-2.2-001), some metadata or some functions over some metadata of products stored into the HSA system (see MSG-2.1-001 to 012).

The HAIO support different TCP/IP based protocols (up to now, raw TCP socket and HTTP). All the HAIO references in this document make reference to the http implementation, but other possible protocols (like secure ones) will be included in future versions.

(FTP access to an observation is not covered in this document.)

HAIO shall return any result of any HSA stored HCSS Product version.

NOTE: this behaviour is not the same as the HUI, which must return the results associated to the latest HCSS Product version (see AD-7).

The communication between client (HCSS) and server (HAIO) shall support cookies for creating session identifiers (requested when accessing restricted products).

## 5.3 *IA – HUI (I/F-3)*

**Interface:** I/F-3

**Message:** MSG-3-001/002/003/004/005

The interface between the IA application and the HUI exists to combine the functionality of the HUI as an interface to a data-source (the HSA) and the functionality of the IA application to manipulate the data. Using these applications together means that the results of searches done with the HUI must be made accessible by the IA application.

The HUI is an application that runs as a Java WebStart application. WebStart applications run in their own virtual machine, called *javaws*, that is different from the “normal” JVM, called *java*. The *javaws* application is started with a WebStart configuration file (*.jnlp* extension – for Java Network Launch Protocol), and includes functionality to automatically download and cache the jars required

to run the application. It also requires security features appropriate for applications that are downloaded from remote sites.

The integration with the IA application can in principle happen in one of two ways: By including the HUI jar(s) plus dependencies in the IA application, or by running the HUI as a separate application and using an inter-application communication mechanism.

The fact that the HUI has a list of several dependencies has led to the conclusion that inclusion of these dependencies in the IA application, and therefore in the DP “reference platform”, is undesirable.

The IA application is a Java SE application, and should be run in a standard JVM. The fact that both applications cannot even run in the same type JVM means that they cannot run as different threads in the same JVM. They must therefore be two completely separate processes and an inter-application communication mechanism, that interfaces two such processes, is required.

Samp [RD-3] is an implementation of such a mechanism. It consists of a “hub” that must be started separately. The applications (in this case the IA application and the HUI) register themselves as listeners. The mechanism is a simple message broadcast. Samp allows various message protocols, main ones being Java RMI and XML-RPC. We opted for XML-RPC for its slightly higher flexibility.

The most important reasons for choosing the Samp implementation are the existing experience with it in the SAT, and its wide-spread use in the astronomical community. The wide-spread usage will allow interfacing with other tools that “know” Samp (notably tools from the Virtual Observatory, such as VOSpec, Aladin, Topcat). In fact, preparations in this area are already ongoing within Herschel DP.

The contents of the message will be in VO table format.

The communication is then implemented by allowing the HUI to send message with pointers to the data, which are broadcast by the Samp hub. The IA application will listen to the message and use the pointers to retrieve the data.

In case the selection made in the HUI was of data in the HSA, the pointers will be URNs that the IA application can retrieve through the HAIO. In case the data is output from an on-demand reprocessing job, the pointer(s) will be FTP URL(s), which the IA application can use to download the data.

## **5.4 SPG On-Demand Reprocessing (I/F-4)**

**Interfaces:** I/F-4

**Message:** MSG-4-001/002

A shared On-Demand Request area between HCSS and HSA must be created to handle the On-Demand Request and the On-Demand Results Error files.

The On-Demand Request area shall contain different subdirectories (see 2.2.8).



The On-Demand Storage area must be another shared area between HCSS and HSA. In this area, the reprocessed products shall be accessible by both systems.

The on-demand request shall be described by an xml file (On-Demand Request file).

The SPG shall use a pooling mechanism to find new on-demand requests.

The reprocessing results shall be saved in another xml file (On-Demand Results file).

The SPG shall move the On-Demand Request file to the suitable On-Demand Request Area subdirectory during the reprocessing procedure.

The SPG shall create a zip file into the On-Demand Storage Area with the reprocessed products. The file path shall be specified by the On-Demand Results file.

## 6 PROCESS DESCRIPTIONS

### 6.1 Product Ingestion (I/F-1)

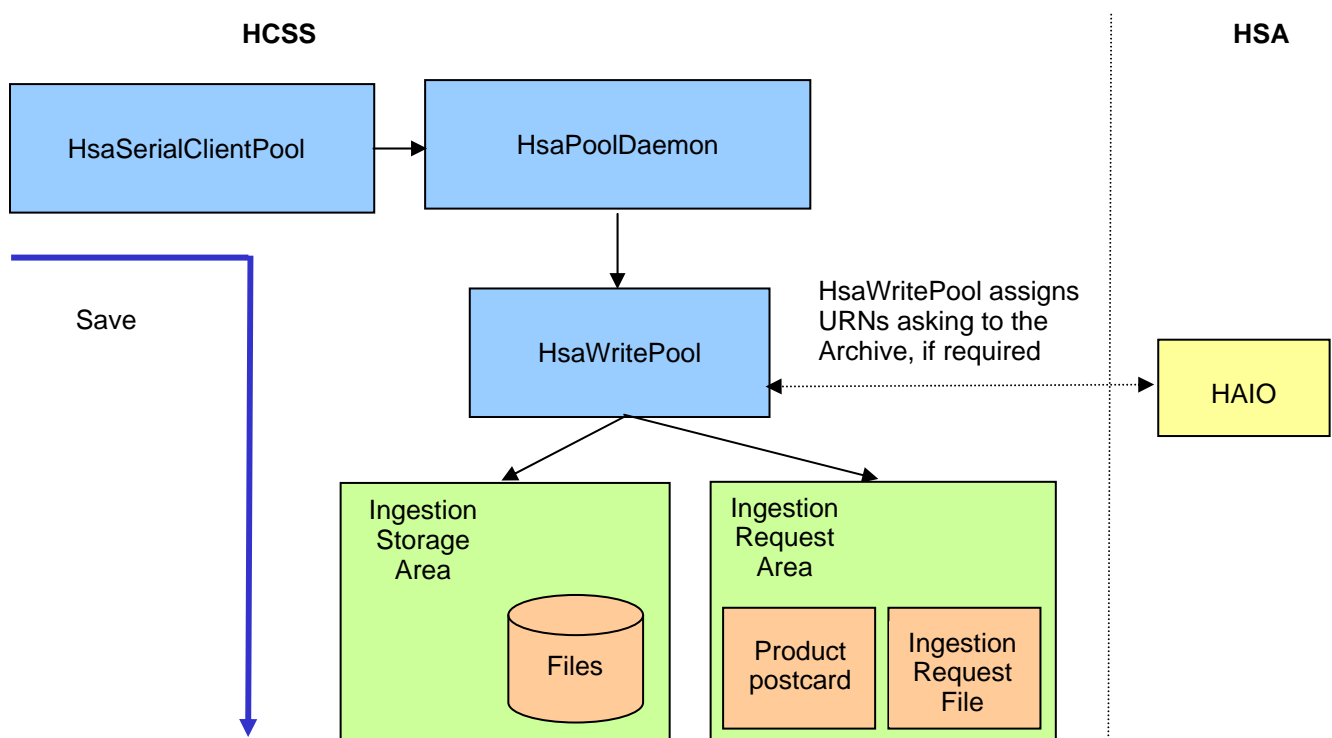
**Interfaces:** I/F-1.1 and I/F-1.2

**Use case:** UC-001

**Messages:** MSG-1.1-001/002

#### Ingestion steps performed by HCSS subsystems

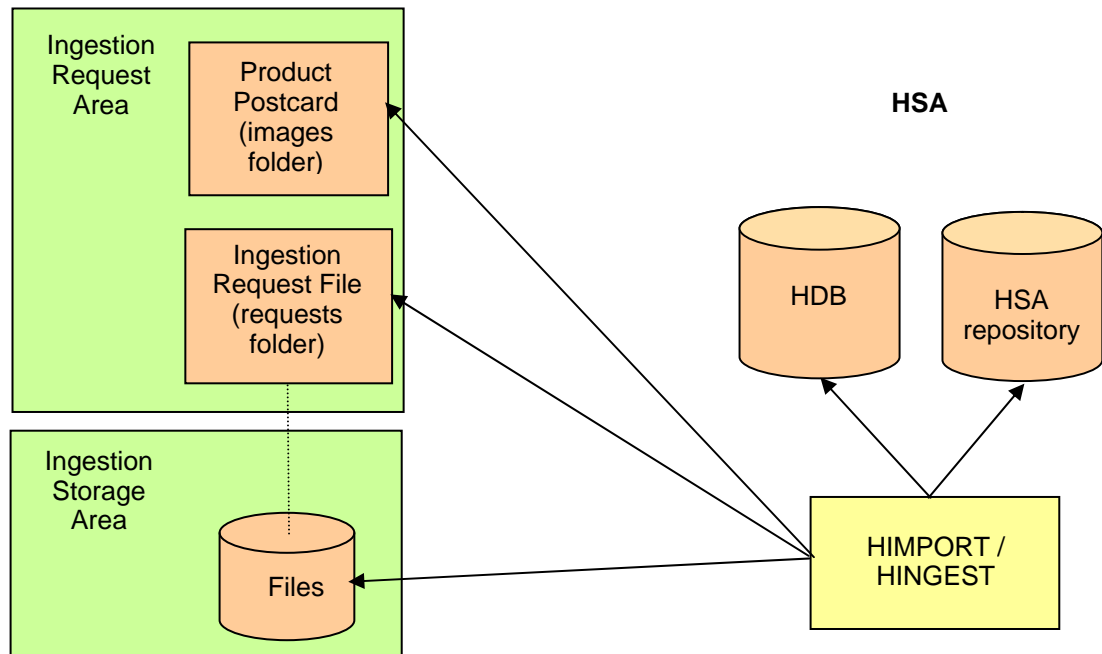
The HCSS-HSA Writable Pool Application (ie: SPG) shall save a Product into the HCSS-HSA writable pool (through a HsaPoolDaemon/HsaSerialClientPool) and it shall obtain a HCSS URN (associated to the product). The products shall be saved in FITS format. HsaWritePool could access to HSA to obtain the right HCSS URN Version number.



The HCSS User Application shall request an ingestion to the HCSS-HSA Writable Pool (through a HsaPoolDaemon/HsaSerialClientPool).

The HsaWritePool shall create an Ingestion Request file with the required information for the products to be ingested. The Ingestion Request file shall contain the product and its children (see Appendix B). Each product shall contain HCSS Versioning and Control metadata (see Appendix A). The Ingestion Request file shall be created in the **request** folder of the Ingestion Request Area.

The HsaWritePool shall create a Product Postcard file associated to an Observation (if the observation has an available one) and shall be saved as a JPG file in the **images** folder of the Ingestion Request Area.



### Ingestion steps performed by HSA subsystems

HSA shall use a pooling mechanism over the **request** folder of the Ingestion Request Area. When an Ingestion Request file is found, the HSA shall move the Ingestion Request file from the **request** folder of the Ingestion Request Area to the **processing** folder.

HSA shall read the Ingestion Request file to obtain the products and the product postcard that must be ingested. The file contains the root product and the children products. Each product is saved in FITS format in the Ingestion Storage Area. HSA shall copy the file and shall extract the metadata from the FITS file. (The metadata to extract is specified in Appendix A).

The metadata keywords ingested shall use the Herschel metadata keywords (not the FITS keywords). The FITS keywords mechanism used in Herschel is specified in Appendix D.

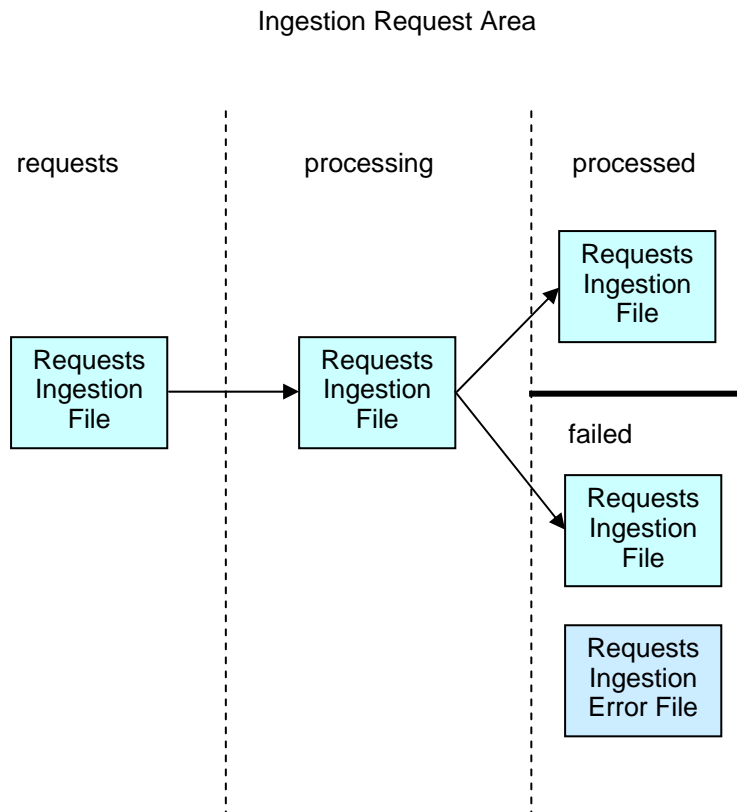
The product files can be located joining the **path\_root** request attribute and the **file\_sub\_path** item attribute (see Appendix B, example in section 10.1.2).

The product postcard file can be located joining the Ingestion Request Area **images** path and the **image** attribute (see Appendix B, example in section 10.1.2).

HSA shall use the **importing** directory for storing the products as a previous step to the products ingestion into the HSA system.

When the ingestion procedure is finished, HSA shall move the Ingestion Request file to the

**processed** folder if no errors were found. HSA shall move the Ingestion Request file to the **failed** folder if the ingestion procedure is not successful. If the ingestion procedure failed, an Ingestion Request Error file shall be created with the errors found. See Appendix B.



When an extended processing ingestion is requested, the HSA shall verify the access restrictions for each observation involved in the extended processing. See Appendix K.

## 6.2 Direct HAIO Access (I/F-2)

**Interfaces:** I/F-2.1, I/F-2.2 and I/F-2.3

**Use cases:** UC-002 and UC-003

**Messages:** MSG-2.1-001/002/003/004/005/006/007/008/009/010/011/012, MSG-2.2-001/002 and MSG-2.3-001/002/003

HCSS-PAL through the HsaPool shall connect to the HAIO server for querying product metadata and for retrieving files: products (FITS format), product metadata (FITS format) and XML ingestion request files (XML format).

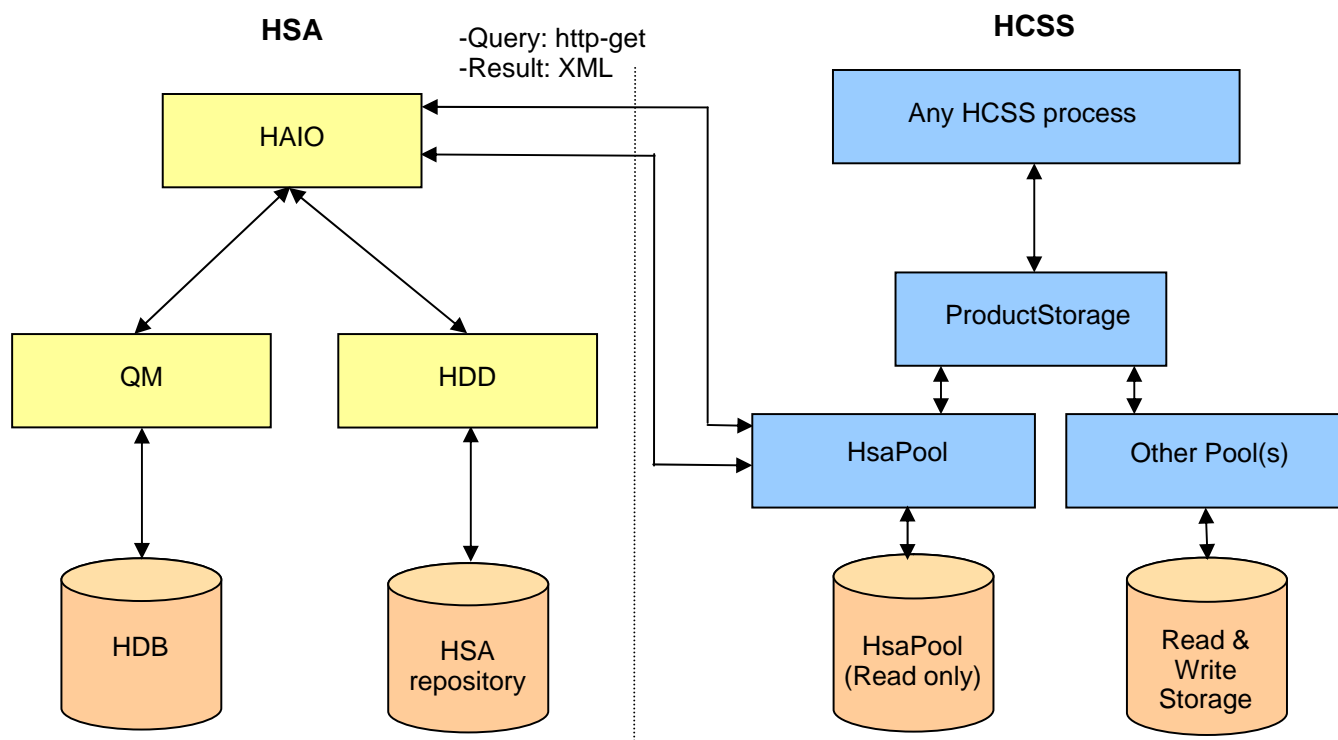


Fig. 3. SPG querying HSA.

A user authentication is required before retrieving restricted products. A login mechanism based on user name and password, and HTTP cookies shall be implemented between client (HCSS/HsaPool) and server (HSA/HAIO). The client will request a **Session Identifier** before asking for a product. The **Session Identifier** will be created and sent back to the client as an HTTP cookie. The client will use that identifier in further products requests. If there is an error when a login or a product is requested, HAIO shall send an HTTP Error code (see Appendix G).

Product metadata requests shall generate a FITS file containing one HDU containing product metadata. This FITS HDU is a copy of the first Product FITS HDU.

Virtual TAR requests shall generate a tar file (see Appendix I) that contains all the required HCSS URNs. If a user does not have access to one or more products (due to access restrictions), an access error shall be generated (HTTP proprietary error: see Appendix G).

To ensure a user has access to a set of products, a query requesting access rights (using **SHOW\_PROPRIETARY** parameter) should be executed first (see MSG-2.1-001 and 11.1 and 11.6 ).

The number of query results that the HAIO server provides shall be limited. There shall be an HAIO server limit and a client limit. These values are specified in Appendix J. The HAIO server limit shall be specified by a configuration parameter. The client limit shall be optional and it can be specified by means of the **LIMIT** query parameter (see MSG-2.1-001). The limit to be applied shall be defined as the lowest value between the HAIO server limit and the client limit. If a query generates more results than the limit to be applied, a VO Table that contains an error is generated by the HAIO server (see Appendix H).

## 6.3 IA – HUI (I/F-3)

**Interface:** I/F-3

**Use case:** UC-002, UC-003, UC-004

**See messages:** MSG-3-001/002/003/004/005

### 6.3.1 Starting up

The communication between the IA session and the HUI uses a message broker mechanism and the Samp protocol. The message broker works like a hub, that clients can send messages to. The message broker will then broadcast these messages to all registered clients.

In this case, the clients are the IA session and the HUI. The HUI is started from the IA session, so logically the IA session must be started first. The IA session is responsible for starting the message broker (the Samp hub). Once the IA session has started the broker, it will register itself as a client. When the HUI has started up, after being launched from/by the IA session, it will also register itself as a client to the message broker.

After this, the system is in a state where the IA session and the HUI can send each other messages via the message broker.

### 6.3.2 Logging in

Proprietary data is protected by username and password, associated to a Herschel LDAP account. Users can create such an account on the Herschel web-site and obtain permissions to the proprietary data from the owner of the data.

It will be possible for users to log onto their Herschel LDAP account from the IA session. If the user plans to access or search for proprietary data in the HSA, the user should log onto the IA session, before starting the HUI.

If the user is logged on in the IA session, and the HUI is launched (by clicking a button in the IA session), the user will automatically be logged on in the HUI also. This works by passing the log-in credentials (username and password) to the HUI at start-up (of the HUI).

If the HUI is started from the IA session, before the user logs on in the IA session, the user will not be automatically logged on in the HUI. In this case, to search for proprietary data in the HUI, the user will have to log on in the HUI, using the log-in functionality available in the HUI. If the user then wants to retrieve proprietary data in the IA session, he will have to log on in the IA session.

This is to make clear that logging on in both the IA session and the HUI is required, but that the log-in in the HUI will be automatic, if and only if the HUI is started from the IA session, *after* the user has logged on in the IA session.

### 6.3.3 *Data retrieval*

Taking into account what was said in the previous paragraph, in the remainder of this section we will assume that user is always appropriately logged on, in any application.

The user can start the HUI from the IA session and use it to search the HSA, or to submit on-demand reprocessing requests (this interface is described in this document and labelled I/F 4).

In case of searching the HSA, the search results will be returned to the user nearly instantaneously. The HUI offers various ways of retrieving (or downloading) the search results, one of them being “to IA session”. Selecting this option will cause the HUI to send a Samp message to the IA session with a list of product URNs. The IA session will download the products to local disk automatically, and allow them to be analyzed and manipulated as usual.

In case of requests for on-demand reprocessing, the resulting data (a **Zippered Local Store** file) will not be available immediately, due to the processing time required. After such a request, an instance of the pipeline will perform the requested processing, and in case of success, the output data will be stored in an FTP area. The processing can take in principle any amount of time, but typically durations vary from a few hours to a few days. The job status (processing, successfully completed, failed, etc.) can be seen by the user from a dedicated HUI panel. If the job status is “successfully completed” and there is output data available, then a button can be clicked in this panel to “send the data to the IA session”.

This means that the HUI will send a Samp message to the IA session, that contains the URL, indicating the location of the **Zippered Local Store** file (as produced by the SPG, see sections 2.2.4, 5.4 and 6.4). The URL must use FTP protocol and the data must be publically accessible. The IA session will then automatically download the **Zippered Local Store** file from this FTP area to local disk, unzip it into a PAL Local Store pool, and the user will be able to analyze and manipulate the data as usual.

## 6.4 *SPG On-Demand Reprocessing (I/F-4)*

**Interfaces:** I/F-4

**Use case:** UC-004

**See messages:** MSG-4-001/002

When HSA receives a new request to reprocess an observation, HSA shall assigns an unique request identifier

An On-Demand Request file must be written in On-Demand Request Area into the **requests** folder.

Before the SPG starts the on-demand procedure, the On-Demand Request file must be moved into the **processing** folder.

When the on-demand procedure is finished, the On-Demand Request file must be moved to **processed** folder. In any case an On-Demand Results file must be created reporting where the



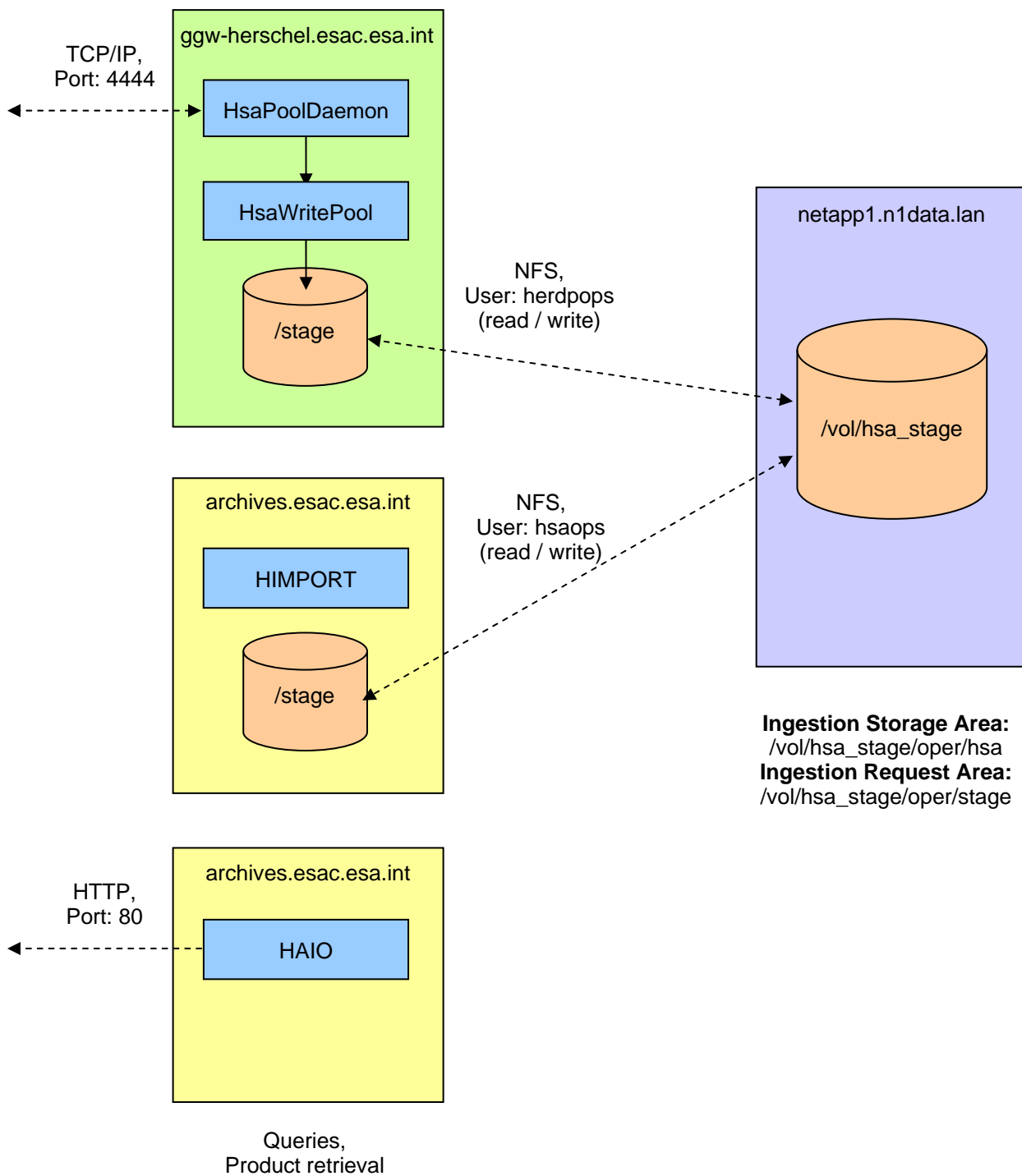
reprocessed data is available. See Appendix B.

SPG shall create a zip file into the On-Demand Storage area with the reprocessed products. This file is not created if the On-Demand process cannot start due a problem with the request file.

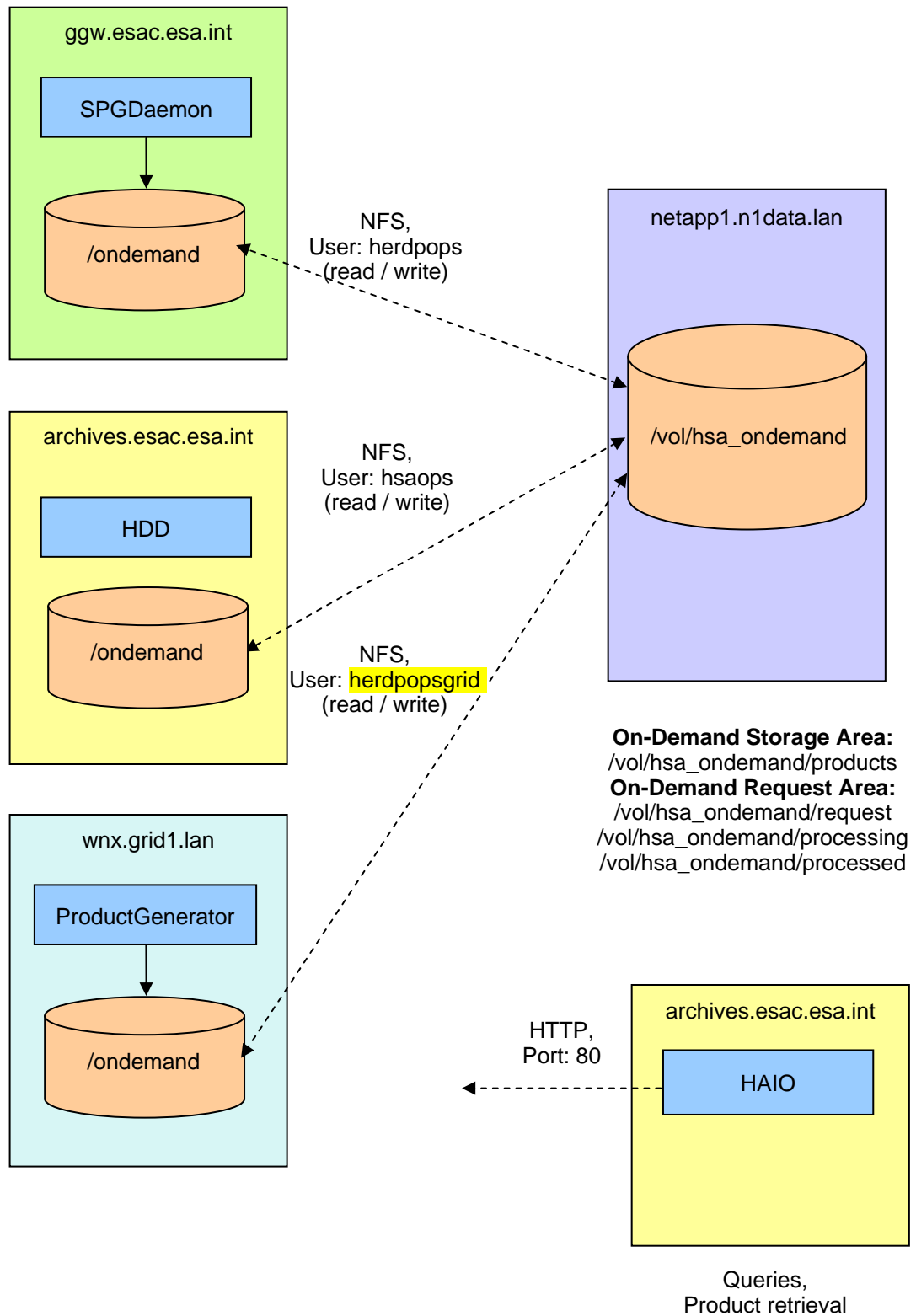
## 7 PHYSICAL DESCRIPTIONS

### OI-7-1

#### 7.1 Systematic Processing



## 7.2 On-Demand Processing



## 7.3 *Bulk reprocessing*

Same as Systematic Processing (see 7.1)

## 7.4 *System descriptions*

### 7.4.1 *HSA system*

- HAIO server: [archives.esac.esa.int](http://archives.esac.esa.int), [http, 80](http://80)
- FTP server: [hsa.esac.esa.int](http://hsa.esac.esa.int), [ftp](ftp://)
- NFS mount point: [/stage](#) on [archives.esac.esa.int](#) from [netappl.n1data.lan](#) ([/vol/hsa\\_stage](#)), user [hsaops](#) (read/write)
- NFS mount point: [/ondemand](#) on [archives.esac.esa.int](#) from [netappl.n1data.lan](#) ([/vol/hsa\\_ondemand](#)), user [hasops](#) (read/write)

### 7.4.2 *HCSS system*

- NFS mount point: [/stage](#) on [ggw-herschel.esac.esa.int](#) from [netappl.n1data.lan](#) ([/vol/hsa\\_stage](#)), user [herdpops](#) (read/write)
- NFS mount point: [/ondemand](#) on [ggw.esac.esa.int](#) from [netappl.n1data.lan](#) ([/vol/hsa\\_ondemand](#)), user [herdpops](#) (read/write)
- HCSS-HSA writable pool (PoolDaemon + HsaWritePool) located at: [ggw-herschel.esac.esa.int](#), [tcp, 4444](#)

### 7.4.3 *Grid*

- NFS mount point: [/ondemand](#) on [wnx.grid1.lan](#) from [netappl.n1data.lan](#) ([/vol/hsa\\_ondemand](#)), user [herdpopsgrid](#) (read/write)

### 7.4.4 *Shared areas*

- NFS mount point: [/vol/hsa-stage](#) on [netappl.n1data.lan](#)
  - Ingestion storage area: [/vol/hsa\\_stage/hsa](#)
  - Ingestion request area: [/vol/hsa\\_stage/hsa/.stage](#)
- NFS mount point: [/vol/hsa\\_ondemand](#) on [netappl.n1data.lan](#)
  - On-Demand storage area: [/vol/hsa\\_ondemand/products](#)
  - On-Demand request area: [/vol/hsa\\_ondemand/requests](#), [/vol/hsa\\_ondemand/processing](#) and [/vol/hsa\\_ondemand/processed](#)

## 7.5 *Requirements*

OI-7.5-1

See [AD-7]

## **7.6 Procedures**

### **OI-7.6-1**

#### **7.6.1 Product Ingestion**

HSA shall inspect the **request** folder (Ingestion Request Area) searching for Ingestion Request files. HSA shall move an Ingestion Request File to the **processing** folder when the ingestion procedure starts.

#### **7.6.2 HCSS-HSA Writable Pool data removal**

Products that have been ingested into HSA system can be removed from the HCSS-HSA Writable Pool (HsaWritePool). This procedure shall be performed by HSC.

The procedure will take the processed ingestion request files, from **processed** folder and shall remove the ingested products (from the local storage). Then, the Ingestion Request files shall be moved to the **cleanedup** folder.

#### **7.6.3 On-Demand processing**

HSC shall inspect the **requests** folder (On-Demand Request Area) searching for On-Demand Request files. HSC shall move an On-Demand Request file to the **processing** folder when the On-Demand procedure starts.

#### **7.6.4 On-Demand processed results**

On-demand reprocessed results (request, results and zipped data files) should be removed by HSA once processed.

#### **7.6.5 Metadata FITS files**

Metadata FITS files are generated using the first Product FITS HDU (that contains product metadata). These Metadata FITS files contain one HDU (which is a copy of the first product FITS HDU) and no data at all.

#### **7.6.6 Virtual TAR files**

When an HAIIO product query requests more than one product, a Virtual TAR file is generated by the HAIIO system. The Virtual TAR file contains the already ingested Products. See Virtual TAR format in Appendix I. See 6.2 also.

## 8 ASSUMPTIONS AND OPEN ISSUES

1. **OI-2.2.10-1:** Specify errors: TBW.
2. **OI-2.2.9-1:** Remove 'stored' attribute.
3. **OI-2.2.9-2:** Modify 'path\_root' attribute.
4. **OI-4-1:** Move use cases to UC document ?
5. **OI-4.1.2-1:** Add references to procedures (OI-7.6-1 related).
6. **OI-4.1.3-1:** TBW (OI-9.5-1, OI-9.5-2 related)
7. **OI-4.2.2-1:** Add references to procedures (OI-7.6-1 related).
8. **OI-4.3.2-1:** Add references to procedures (OI-7.6-1 related).
9. **OI-4.4.2-1:** Add references to procedures (OI-7.6-1 related).
10. **OI-4.4.3-1:** HUI On-demand Monitoring Tool: TBC.
11. **OI-7-1:** Physical description: TBC
12. **OI-7.5-1:** Specify constraints / other documents: TBW
13. **OI-7.6-1:** Procedures: TBW (OI-4.1.2-1, OI-4.2.2-1, OI-4.3.2-1, OI-4.4.2-1 related)
14. **OI-9.2-1:** Common metadata: TBC
15. **OI-9.2-2:** Duplicated key: PDG says that there is no problem.
16. **OI-9.2-3:** Common metadata sizes: TBC (pending PDG)
17. **OI-9.3-1:** ObservationContext metadata: TBC
18. **OI-9.5-1:** TBC.
19. **OI-9.5-2:** TBW non observational products control metadata (OI-4.1.3-1 related)

## 9 APPENDIX A: LIST OF METADATA FOR EACH PRODUCT

### 9.1 *Metadata Types*

The following table shows the string representation of each of the metadata types:

<b>Type</b>	<b>String Representation</b>
String	'string'
Boolean	'boolean'
Short	'short'
Integer	'int'
Long	'long'
Double	'double'

### 9.2 *Common Metadata*

The following table contains the product metadata that shall be ingested into the archive [OI-9.2-1](#), [OI-9.2-3](#).

<b>HCSS Metadata Name</b>	<b>HCSS Type</b>	<b>HSA Type</b>	<b>String size</b>	<b>Description</b>
acmsMode	string	string	50	ACMS mode
activeStrId	string	string	50	identification of the active STR
aorLabel	string	string	200	AOR label as entered in Hspot
aot	string	string	50	AOT Identifier
aperture	string	string	50	Instrument aperture in use

apid	short	short	n.a.	Application Programme Identifier
arrayName	string	string	50	Name of Detector Array
author	string	string	100	Author of the data
averaging	string	string	50	Averaging operator
band	string	string	200	Band
baselineModel	string	string	50	Baseline Model
baselineParams	string	string	50	Parameters of Baseline model
bbCount	long	long	n.a.	Building Block Count
bbid	long	long	n.a.	Building Block Identifier
bbType	long	long	n.a.	Building Block Type
bbTypeName	string	string	50	Building Block Type Name
biasFreq	double	float	n.a.	Bias Frequency
biasMode	string	string	50	Bias Mode
biasVoltage	double	double	n.a.	Bias voltage factor
bitPos	int	int	n.a.	Bit position of this mask
calFileId	string	string	50	Calibration file ID
calFileVersion	int	int	n.a.	Calibration file version
calThreshold	double	double	n.a.	Specified position accuracy threshold for a plateaux in calibration
camera	string	string	50	Name Camera/ detector array
cameraModel	string	string	50	Model of the camera (CQM, FM, Sixpack,...)
cd1_1	double	double	n.a.	CD1_1 element of CD matrix
cd1_2	double	double	n.a.	CD1_2 element of CD matrix
cd1_3	double	double	n.a.	CD1_3 element of CD matrix
cd2_1	double	double	n.a.	CD2_1 element of CD matrix
cd2_2	double	double	n.a.	CD2_2 element of CD matrix
cd2_3	double	double	n.a.	CD2_3 element of CD matrix
cd3_1	double	double	n.a.	CD3_1 element of CD matrix
cd3_2	double	double	n.a.	CD3_2 element of CD matrix
cd3_3	double	double	n.a.	CD3_3 element of CD matrix
cdelt1	double	double	n.a.	pixel size in axis 1



cdelt2	double	double	n.a.	pixel size in axis 2
changelog	string	string	50	Logging of changes
chopperPlateau	int	int	n.a.	Indicates the chop plateau within sequence
constVelFlag	boolean	boolean	n.a.	Constant velocity flag
conversionFactor	double	double	n.a.	conversion factor from chopper deflection (degrees) to angle on sky
creationDate	FineTime	long	n.a.	Date of product creation
creator	string	string	100	The name of the software that created the product
crota2	double	double	n.a.	rotation angle
crpix1	double	double	n.a.	CRPIX1 reference pixel of axis 1
crpix2	double	double	n.a.	CRPIX2 reference pixel of axis 2
crval1	double	double	n.a.	axis 1 coordinate at tangency
crval2	double	double	n.a.	axis 2 coordinate at tangency
ctype1	string	string	50	type of coordinate axis eg RA---TAN
ctype2	string	string	50	type of coordinate axis eg DEC—TAN
cusMode	string	string	50	CUS observation mode
dataAnalyst	string	string	50	Name of data analyst
dec	double	double	n.a.	Actual Declination of pointing
decNominal	double	double	n.a.	requested declination of pointing
decObject	double	double	n.a.	Declination of target object
deltaPix	double	double	n.a.	Correction of output angle per pixel unit offset to central pixel
description	string	string	Text	Full name of product
endDate	FineTime	long	n.a.	End date of observation
endWavelength	double	double	n.a.	End of wavelength interval
epoch	double	double	n.a.	equinox of celestial coordinate system
equinox	double	double	n.a.	equinox of celestial coordinate system
error	double	double	n.a.	Error on signal
fileName	string	string	100	name of exported file
filter	string	string	50	Filter name [SHORT/LONG/none]
fineTime	long	long	n.a.	Time of signal sampling

formatVersion	string	string	50	Version of product format
gyroPropQualIdx	double	double	n.a.	Gyro-propagated quality index
instMode	string	string	50	Instrument mode
instrument	string	string	50	Instrument name
interpMethod	string	string	50	Recommended interpolation method to be applied
jiggleId	long	long	n.a.	Jiggle Identifier
keyWavelength	double	double	n.a.	Key Wavelength
level	string	string	2	Product level
maxWavelength	double	double	n.a.	Maximum wavelength
minWavelength	double	double	n.a.	Minimum wavelength
missionConfig	string	string	50	Mission configuration
modelName	string	string	50	Instrument Model Name
naifId	string	string	50	SSO NAIF identifier
nodCycleNum	long	long	n.a.	Switching/nodding cycle number
numChopCyc	long	long	n.a.	Number of chopping cycles
numHifiSaa	long	long	n.a.	Number of HIFI reference Solar Aspect Angles
numJigglePos	long	long	n.a.	Number of jiggle positions
numNodCyc	long	long	n.a.	Number of nodding cycles
numPacsSaa	long	long	n.a.	Number of PACS reference Solar Aspect Angles
numRasterCol	long	long	n.a.	Number of raster columns
numRasterLines	long	long	n.a.	Number of raster lines
numScanLines	long	long	n.a.	Number of scan lines
numSpectra	long	long	n.a.	Number of Spectra
numSpireSaa	long	long	n.a.	Number of SPIRE reference Solar Aspect Angles
object	string	string	50	target name
objectType	string	string	50	astronomical object type
observer	string	string	100	name of observer
obsid	long	long	n.a.	Observation Identifier
obsMode	string	string	100	Observation mode name
odNumber	long	long	n.a.	operational day number

offPosFlag	boolean	boolean	n.a.	Off-position flag
onTargetFlag	boolean	boolean	n.a.	On-target flag
origin	string	string	50	site that created the product
outOfFieldFlag	boolean	boolean	n.a.	Out-of-field flag
pc1_1	double	double	n.a.	PC1_1 element of PC matrix
pc1_2	double	double	n.a.	PC1_2 element of PC matrix
pc1_3	double	double	n.a.	PC1_3 element of PC matrix
pc2_1	double	double	n.a.	PC2_1 element of PC matrix
pc2_2	double	double	n.a.	PC2_2 element of PC matrix
pc2_3	double	double	n.a.	PC2_3 element of PC matrix
pc3_1	double	double	n.a.	PC3_1 element of PC matrix
pc3_2	double	double	n.a.	PC3_2 element of PC matrix
pc3_3	double	double	n.a.	PC3_3 element of PC matrix
pixelRow	int	int	n.a.	Pixel row index
plwBiasAmpl	double	float	n.a.	PLW Bias Amplitude
pmwBiasAmpl	double	float	n.a.	PMW Bias Amplitude
pointingMode	string	string	50	Pointing mode identifier
posAngle	double	double	n.a.	Position Angle of pointing
proposal	string	string	50	proposal name
pswBiasAmpl	double	float	n.a.	PSW Bias Amplitude
ptcBiasAmpl	double	float	n.a.	PTC Bias Amplitude
qualityFlag	int	int	n.a.	Quality flag
ra	double	double	n.a.	Actual Right Ascension of pointing
raDeSys	string	string	50	Coordinate reference frame for the RA and DEC
raErr	double	double	n.a.	Error on Right Ascension of actual pointing
raNominal	double	double	n.a.	requested RA of pointing
raObject	double	double	n.a.	RA of target object
rasterColumnNum	short	short	n.a.	Raster column number
rasterLineNum	short	short	n.a.	Raster line number
readouts	double	double	n.a.	sample readouts for one ramp

references	string	string	50	References
refPixel	long	long	n.a.	Reference Pixel
roll	double	double	n.a.	Spacecraft roll angle
saa	double	double	n.a.	Reference SAA value in the range 0-180 degrees
saturation	double	double	n.a.	Fraction of saturated samples
satValuesSigned	int	int	n.a.	Saturation values signed modes
satValuesUnsigned	int	int	n.a.	Saturation values unsigned modes
scanLineNum	short	short	n.a.	Scan line number
scope	string	string	10	Scope
sedVersion	string	string	50	Version of the SED
serendipityFlag	boolean	boolean	n.a.	SPIRE serendipity mode flag
siamId	string	string	50	Reference to the applicable SIAM
skyResolution	double	double	n.a.	Spatial resolution
slewFlag	boolean	boolean	n.a.	Slew flag
slwBiasAmpl	double	float	n.a.	SLW Bias Amplitude
sswBiasAmpl	double	float	n.a.	SSW Bias Amplitude
source	string	string	50	Source packet
sourceDetector	string	string	50	Detector Source Packet
sourceSmec	string	string	50	SMEC Source Packet
specNum	long	long	n.a.	Spectrum Number
spectralResolution	double	double	n.a.	Spectral resolution of data
startDate	FineTime	long	n.a.	Start date of observation
startWavelength	double	double	n.a.	Begin of wavelength interval
status (OI-9.2-2)	string	string	50	Pixel Status or Channel Status
strInterlacingStatus	boolean	boolean	n.a.	STR interlacing status
strQualIdx	double	double	n.a.	STR quality index
subinstrumentId	string	string	50	Sub-instrument identifier
subsystem	string	string	50	Instrument Subsystem
telescope	string	string	50	Name of telescope
temperature	double	double	n.a.	

type	string	string	50	Product type identification
variability	string	string	50	Information on object variability
version	string	string	50	version of product
wavelengthId	int	int	n.a.	Key Wavelength ID
wcsReference	string	string	50	Reference of Coordinate System
wcsType	string	string	50	Type of Coordinate System
wheelPos	int	int	n.a.	Wheel position
zeroPointOffset	double	double	n.a.	Zero point offset

(Latest Common Product metadata can be found at:

[http://www.herschel.be/twiki/bin/view/Hcss/ProductMetaDataConvention#Commonly\\_used\\_meta\\_data](http://www.herschel.be/twiki/bin/view/Hcss/ProductMetaDataConvention#Commonly_used_meta_data))

### 9.3 *ObservationContext metadata*

#### OI-9.3-1

Field	Metadata	Comments
Observation ID	ObservationContext.observid	
Observation ID list	ObservationContext.observid	
Observation proprietary rights status (proprietary, public, when it is public)	DERIVED PARAMETER	
Observation proprietary rights expire date	DERIVED PARAMETER	
Target name. Name resolution should be allowed using SIMBAD or NED, to be selected by the user.	ObservationContext.object	HSPOT in Versant
Target position (including radius around a central position or box), given in equatorial, ecliptic or galactic coordinates.	ObservationContext.raNominal ObservationContext.decNominal ObservationContext.ra / .dec	HSPOT in Versant HSPOT in Versant Depends on pointing mode: (1) Fixed target maps => barycenter (2) For moving targets, RA, DEC of starting point
Target List	DERIVED PARAMETER	
Astronomical object type	ObservationContext.objectType	HSPOT in Versant (TBD)
Observation duration	ObservationContext.duration	HSPOT in Versant
Observation date and time	ObservationContext.startDate, ObservationContext.endDate	HSPOT in Versant
Observation OD	ObservationContext.odNumber	HSPOT in Versant
Instrument	ObservationContext.instrument	HSPOT in Versant
Instrument subsystem	ObservationContext.obsMode	HSPOT in Versant (TBD: mapping. Maybe, it depends on two fields in Versant)

Field	Metadata	Comments
AOT ID	ObservationContext.aot	HSPOT in Versant
AOR label	ObservationContext.aorLabel	HSPOT in Versant
Wavelength range	ObservationContext.wavelength	(1) Only one field in ObservationContext: 'wavelength', pipeline shall transform the value used by the instrument into 'wavelength'. (2) <b>TBD</b> : type of this field: +Single value +List +Range
Frequency range	DERIVED PARAMETER (wavelength)	
V LSR	DERIVED PARAMETER (wavelength)	
Proposal ID	ObservationContext.proposal	HSPOT in Versant
Proposal category	ObservationContext.category	HSPOT in Versant
Proposal abstract contents	ObservationContext.abstract	HSPOT in Versant
Observer name	ObservationContext.observer	HSPOT in Versant
Quality flag	<b>TBD</b>	<b>TBD</b>
Pointing mode (line scan, raster, fix pointing, others)	ObservationContext.pointingMode	Pointing Product
Entries in catalogue products (e.g., Pointing source list, Spectral line list, observer's generated catalogues or atlas)	<b>TBD</b>	<b>TBD</b>
Building Block ID	<b>TBD</b>	<b>TBD</b>
Types of sub-measurements of Observations (to be specified)	<b>TBD</b>	<b>TBD</b>
Instrument specific data contents (e.g. Fluxes, glitch rates, to be further specified)	<b>TBD - Instrument specific</b>	<b>TBD</b>
Image stamp showing the Quick-look observation product	ObservationContext.Level1.quickLook	It is a product, not Metadata.

Blue: Parameter derived from other/s parameter/s

Red: To be defined

Green: Proposed parameter name

## 9.4 Versioning Metadata

Versioning Metadata identifies a HCSS Product version. The main keyword is HCSS\_URN, which is unique (even among different SPG Software versions).

Versioning Metadata must be associated to each product. They are required for the PAL versioning mechanism.

Versioning Metadata is not part of the Common Metadata, and the Ingestion Request File shall provide the suitable versioning metadata to associate to each product (to be ingested into HSA).

There could be several HCSS Product versions belonging to the same observation at the same time. (See 5.2 Direct HAIIO Access ).

<b>KEY</b>	<b>TYPE</b>
HCSS_URN	string
HCSS_FILE_NAME	string
HCSS_CLASS_TYPE	string
HCSS_CLASS_PATH	string
HCSS_URN_VERSION	long
HCSS_TRACK_ID	string
HCSS_TRACK_VERSION	long
HCSS_SIZE	long
HCSS_TOTAL_SIZE	long
HCSS_HASH	string

## 9.5 HSA Control Metadata

### OI-9.5-1

Control Metadata is used by the HSA system to identify all the products/context that belong to an Observation (Control Metadata shall be provide by the Ingestion Request File).

<b>KEY</b>	<b>TYPE</b>	<b>DESCRIPTION</b>
obsid	string	Observation Identifier
spgid	string	SPG software identifier

**OI-9.5-2**

The following table contains the metadata used by the HSA system for common products (these metadata shall be provide by the Ingestion Request File).

<b>KEY</b>	<b>TYPE</b>	<b>DESCRIPTION</b>
spgid	string	SPG software identifier

## 10 APPENDIX B: XML FILES

### 10.1 Ingestion Request

This section specifies the Ingestion Request File object.

This object is an xml file with a tree of the products (and/or contexts) that must be ingested.

The root element is **request**, with the following attributes:

<b>Attribute</b>	<b>DESCRIPTION</b>
id	Observation Identifier, or nothing (empty string) for common products
path_root	Absolute path to the HSA write pool
spgid	SPG identifier: <b>SPG vX.Y.Z</b>
type	Request type: <b>observati on</b> or <b>common</b>
urn	HCSS URN to the root (first) element to be ingested
parallel_mode	Specifies if the Observation belongs to a SPIRE PACS parallel mode
image	Product postcard image file name. It is composed by: <b>&lt;Observati on I denti fi er&gt;–&lt;HCSS cl ass type&gt;–&lt;HCSS URN Versi on&gt;. j pg</b>
cus_mode	Specifies observation cusMode



<b>Attribute</b>	<b>DESCRIPTION</b>
proc	Processing type
generic_subpath	Processing path
instrument	Observation instrument
extended	It contains all the observation identifiers involved in an extended processing

Inside **request** element, a tree of **item** elements are written. Each item has the following attributes:

<b>Attribute</b>	<b>DESCRIPTION</b>
class	HCSS class product
description	Product description
file_sub_path	Relative path (to the <b>request</b> path_root attribute) to the product
file_absolute_path	Absolute path to the product
id	Observation identifier. For Common Products it is empty.
type	<p>Product identifier. For the first item, it is:</p> <ul style="list-style-type: none"> <li>For Observational Products: The identifier shall start with the observation identifier, minus, the HCSS class type, minus and the HCSS URN version. <b>&lt;Observation Identifier&gt;--&lt;HCSS class type&gt;--&lt;HCSS URN Version&gt;</b></li> <li>For Common Products: The file name shall start with 'C', minus, the HCSS class type, minus and the HCSS URN version. <b>C--&lt;HCSS class type&gt;--&lt;HCSS URN Version&gt;</b></li> </ul> <p>For the following items, it is the parent item keyword (HCSS reference) by which is referenced.</p>
stored	It is 'true' if the item is already stored into HSA. 'false' means that in the moment that the ingestion was requested, the item was not found in HSA.
urn	Product HCSS URN
distribution_path	Product file location inside the tar file (see Appendix F).

<b>Attribute</b>	<b>DESCRIPTION</b>
distribution_filename	Product file name inside the tar file (see Appendix F).

Each item can have metadata associated. Metadata items, **meta**, are located inside a **metadata** item (see example). Each **meta** item has the following attributes:

<b>Attribute</b>	<b>DESCRIPTION</b>
id	Metadata Identifier
type	Metadata type
value	Metadata value

### 10.1.1 File Name Rules

It is based on the 'id' **item** attribute:

- For Observational Products:

The file name shall start with the observation identifier, minus, the HCSS class type, minus and the HCSS URN version. The extension shall be 'xml'.

**<Observation Identifier>--<HCSS class type>--<HCSS URN Version>.xml**

- For Common Products:

The file name shall start with 'C', minus, the HCSS class type, minus and the HCSS URN version. The extension shall be 'xml'.

**C--<HCSS class type>--<HCSS URN Version>.xml**

### 10.1.2 DTD

```
<!DOCTYPE REQUEST [
<!ELEMENT request (item)>
<!ATTLIST request id CDATA #IMPLIED>
<!ATTLIST request path_root CDATA #REQUIRED>
<!ATTLIST request spgid CDATA #REQUIRED>
<!ATTLIST request type (observation|common) #REQUIRED>
```

```
<!ATTLIST request urn CDATA #REQUIRED>
<!ATTLIST request parallel_mode (true|false) #REQUIRED>
<!ATTLIST request image CDATA #REQUIRED>
<!ATTLIST request cus_mode CDATA #REQUIRED>
<!ATTLIST request proc CDATA #IMPLIED>
<!ATTLIST request generic_subpath CDATA #REQUIRED>
<!ATTLIST request instrument CDATA #REQUIRED>
<!ATTLIST request extended CDATA #IMPLIED>
```

```
<!ELEMENT item (metadata?, item*)>
<!ATTLIST item class CDATA #REQUIRED>
<!ATTLIST item description CDATA #IMPLIED>
<!ATTLIST item file_sub_path CDATA #REQUIRED>
<!ATTLIST item file_absolute_path CDATA #IMPLIED>
<!ATTLIST item stored (true|false) #REQUIRED>
<!ATTLIST item urn CDATA #REQUIRED>
<!ATTLIST item distribution_path CDATA #REQUIRED>
<!ATTLIST item distribution_filename CDATA #REQUIRED>
<!ATTLIST item id CDATA #REQUIRED>
```

```
<!ELEMENT metadata (meta*)>
```

```
<!ELEMENT meta>
<!ATTLIST meta id CDATA #REQUIRED>
<!ATTLIST meta type (%TYPES;) #REQUIRED>
<!ATTLIST meta value CDATA #IMPLIED>
```

```
<!ENTITY % TYPES (string|boolean|short|int|long|double)>
]>
```

## Example

File name: 1-herschel . i a. Observati onContext-0. xml

(Postcard file name: 1-herschel . i a. Observati onContext-0. jpg)

Content:

```
<?xml version="1.0" encoding="UTF-8"?>
<request id="1" path_root="/stage/hsa" spgid="SPG v1.0.1" cus_mode="" type="1-
herchel . i a. Observati onContext-0" urn="urn: hsa: herchel . i a. obs. Observati onContext: 0"
parallel_mode="false" image="1-herschel . i a. obs. Observati onContext-0. jpg" proc="systematic"
generic_subpath="systematic/client_products/172. 25. 4. 42_25836@wn32. n1grid. lan_1302159579918-
default" instrument="HIFI">

<!--root (Observati onContext)-->
<item class="herchel . i a. obs. Observati onContext" description="Unknown"
```

```
file_sub_path="herschel . i a. obs. Observati onContext/hhi fi 0000001obs. fi ts"
id="1- herschel . i a. obs. Observati onContext-0" stored="fal se"
urn="urn: hsa: herschel . i a. obs. Observati onContext: 0" di stri buti on_path="/1"
di stri buti on_fi lename="hhi fi 0000001obs. fi ts">
  <metadata>
    <meta id="HCSS_SI ZE" type="l ong" val ue="161111"/>
    <meta id="HCSS_URN_VERSI ON" type="l ong" val ue="0"/>
    <meta id="HCSS_TRACK_ID" type="stri ng" val ue="OBS_1207830806113281000"/>
    <meta id="HCSS_HASH" type="stri ng" val ue="206522a3a1f7ff43d0b82fa40ff409bd"/>
    <meta id="HCSS_URN" type="stri ng" val ue="urn: hsa: herschel . i a. obs. Observati onContext: 0"/>
    <meta id="HCSS_FI LE_NAME" type="stri ng" val ue="hhi fi 0000001obs. fi ts"/>
    <meta id="HCSS_TRACK_VERSI ON" type="l ong" val ue="0"/>
    <meta id="HCSS_TOTAL_SI ZE" type="l ong" val ue="161111"/>
    <meta id="HCSS_CLASS_TYPE" type="stri ng" val ue="herschel . i a. obs. Observati onContext"/>
  </metadata>

  <!--Observati onContext chi ld -->
  <i tem cl ass="herschel . i a. pal . MapContext" descri pti on="Unknown"
    file_sub_path="herschel . i a. pal . MapContext/hmapcontext. fi ts" id="ref_1" stored="fal se"
    urn="urn: hsa: herschel . i a. pal . MapContext: 0" di stri buti on_path="/1/ref_1"
    di stri buti on_fi lename="hmapcontext. fi ts">
    <metadata>
      <meta id="HCSS_SI ZE" type="" val ue="90496"/>
      <meta id="HCSS_URN_VERSI ON" type="" val ue="0"/>
      <meta id="HCSS_TRACK_ID" type="" val ue="Unknown_1207830805689592000"/>
      <meta id="HCSS_HASH" type="" val ue="0edd55ecd61b01ebfaa87c62c34a49cc"/>
      <meta id="HCSS_URN" type="" val ue="urn: hsa: herschel . i a. pal . MapContext: 0"/>
      <meta id="HCSS_FI LE_NAME" type="" val ue="hmapcontext. fi ts"/>
      <meta id="HCSS_TRACK_VERSI ON" type="" val ue="0"/>
      <meta id="HCSS_TOTAL_SI ZE" type="" val ue="90496"/>
      <meta id="HCSS_CLASS_TYPE" type="" val ue="herschel . i a. pal . MapContext"/>
    </metadata>

    <!--MapContext chi ld (Product)-->
    <i tem cl ass="herschel . i a. dataset. Product" descri pti on="p1"
      file_sub_path="herschel . i a. dataset. Product/hproduct_20080410T143325423Z. fi ts"
      id="ref_1. 1" stored="fal se" urn="urn: hsa: herschel . i a. dataset. Product: 1"
      di stri buti on_path="/1/ref_1/ref_1. 1"
      di stri buti on_fi lename="hproduct_20080410T143325423Z. fi ts">
      <metadata>
        <meta id="HCSS_SI ZE" type="" val ue="3892"/>
        <meta id="HCSS_URN_VERSI ON" type="" val ue="1"/>
        <meta id="HCSS_TRACK_ID" type="" val ue="Unknown_1207830805381514000"/>
        <meta id="HCSS_HASH" type="" val ue="7c1acd5732011a3db8f659e6fd46c759"/>
        <meta id="HCSS_URN" type="" val ue="urn: hsa: herschel . i a. dataset. Product: 1"/>
        <meta id="HCSS_FI LE_NAME" type="" val ue="hproduct_20080410T143325423Z. fi ts"/>
      </metadata>
    </i tem>
  </i tem>
</di stri buti on_fi lename="hmapcontext. fi ts">
```

```

    <meta id="HCSS_TRACK_VERSION" type="" value="0" />
    <meta id="HCSS_TOTAL_SIZE" type="" value="3892" />
    <meta id="HCSS_CLASS_TYPE" type="" value="herchel . i a. dataset. Product" />
  </metadata>
</item> <!--Product-->

  <!--More Products or Contexts-->

</item> <!--MapContext-->

  <!--More Products or Contexts-->

</item> <!--ObservationContext-->
</request>

```

File name: 1-herchel . i a. pal . versi oni ng. TagsProduct. xml

Content:

```

<?xml version="1.0" encoding="UTF-8"?>
<request id="" path_root="/stage/hsa" spgid="SPG v1.0.1" cus_mode="" type="C-
herchel . i a. pal . versi oni ng. TagsProduct-0" urn="urn: hsa: herchel . i a. pal . versi oni ng. TagsProduct: 0"
parallel_mode="false" image="" instrument="Unknown"
generic_subpath="systematic/c/ident_products/172.25.4.42_25836@wn32.n1grid.lan_1302159579918-
default">
  <!--root (TagsPrduct)-->
  <item class="herchel . i a. pal . versi oni ng. TagsProduct" description="Unknown"
file_sub_path="herchel . i a. pal . versi oni ng. TagsProduct/htagsproduct_22148499406150087. fts"
id="1-herchel . i a. pal . versi oni ng. TagsProduct-0" stored="false"
  urn="urn: hsa: herchel . i a. pal . versi oni ng. TagsProduct: 0" distribution_path="/null"
distribution_filename="htagsproduct_22148499406150087. fts">
    <metadata>
      <meta id="HCSS_SIZE" type="long" value="11182" />
      <meta id="HCSS_URN_VERSION" type="long" value="0" />
      <meta id="HCSS_TRACK_ID" type="string" value="" />
      <meta id="HCSS_HASH" type="string" value="8f415b043983f14480d1e14187272ac9" />
      <meta id="HCSS_URN" type="string" value="urn: hsa: herchel . i a. pal . versi oni ng. TagsProduct: 0" />
      <meta id="HCSS_FILE_NAME" type="string" value="htagsproduct_22148499406150087. fts" />
      <meta id="HCSS_TRACK_VERSION" type="long" value="-1" />
      <meta id="HCSS_TOTAL_SIZE" type="long" value="11182" />
      <meta id="HCSS_CLASS_TYPE" type="string" value="herchel . i a. pal . versi oni ng. TagsProduct" />
    </metadata>

  </item> <!--TagsProduct-->
</request>

```

### 10.1.3 Hierarchy

The first element is `request`. Inside a `request` element, there is ONE `item` element. This `item` is the `ObservationContext`. Its `id` attribute is the observation identifier.

Inside the `ObservationContext` item, there could be the following items:

```
id=level 0
id=level 0_5
id=level 1
id=level 2
id=quality
id=auxiliary
id=calibration
```

The identifiers are unique at this level.

For instance:

```
<request id="12345678" path_root="/stage/hsa" spgid="SPG v1.0.1" type="12345678-herchel . i a . obs . observati onContext-274" parallel_mode="false" image="12345678-herchel . i a . obs . observati onContext-274 . jpg">
```

```
<!-- First item: observationContext -->
```

```
<item class="herchel . i a . obs . Observati onContext" id="12345678-herchel . i a . obs . Observati onContext-691" ...>
```

```
<!-- level 0 id="level 0" -->
<item class="xxxx" id="level 0" ...>
  <item class="cccc" id="[id]" ...>
  ...
</item>

<!-- level 0_5 id="level 0_5" -->
<item class="xxxx" id="level 0_5" ...>
  <item ...>
  ...
</item>

<!-- level 1 id="level 1" -->
<item class="xxxx" id="level 1" ...>
```

```
<i tem ...>
  ...
</i tem>

<!-- level 2 id="level 2" -->
<i tem class="xxxx" id="level 2" ...>
  <i tem ...>
    ...
  </i tem>

<!-- quality id="quality" -->
<i tem class="xxxx" id="quality" ...>
  <i tem ...>
    ...
  </i tem>

<!-- auxiliary id="auxiliary" -->
<i tem class="xxxx" id="auxiliary" ...>
  <i tem ...>
    ...
  </i tem>

<!-- calibration id="calibration" -->
<i tem class="xxxx" id="calibration" ...>
  <i tem ...>
    ...
  </i tem>

</i tem>
<!-- end ObservationContext ---->
</request>
```

## 10.2 Ingestion Request Error

This section specifies the Ingestion Request Error File object.

### 10.2.1 File Name Rules

- For Observational Products:

The file name shall start with the observation identifier, minus, the HCSS class type, minus and the HCSS URN version. The extension shall be 'result.xml'.

<Observation Identifier>--<HCSS class type>--<HCSS URN Version>.result.xml

- For Common Products:

The file name shall start with 'C', minus, the HCSS class type, minus and the HCSS URN version. The extension shall be 'result.xml'.

C--<HCSS class type>--<HCSS URN Version>.result.xml

### 10.2.2 DTD

```
<!DOCTYPE RESULT [  
<!ELEMENT result (MESSAGE)>  
<!ATTLIST result id CDATA #IMPLIED>  
<!ATTLIST result spgid CDATA #REQUIRED>  
<!ATTLIST result type (observation|common) #REQUIRED>  
<!ATTLIST result urn CDATA #REQUIRED>  
  
<!ELEMENT MESSAGE (#PCDATA)>  
>
```

#### Example

File name: 1-herschel.la.ObservationContext-0.result.xml

Content:

```
<?xml version="1.0" encoding="UTF-8"?>  
<result id="1" spgid="spg_1" type="observation" urn="  
urn:hsa:herschel.la.obs.ObservationContext:0">  
  <message>  
    Error message  
  </message>  
</result>
```

### 10.3 On-demand Request

This section specifies the On-Demand Request File object.

This object is an xml file with one item named **request** that contains the following items:



<i>Item</i>	<i>Description</i>	<i>VALUE Attribute</i>
<b>REQI D</b>	Request identifier	<Request identifier>
<b>OBSI D</b>	Observation identifier	<Observation identifier>
<b>I NSTRUMENT</b>	Herschel Instrument	SPIRE   PACS   HIFI
<b>PROFI LE</b>	Profile	Predefined profile identifier. Currently only 'systematic' is defined.

### 10.3.1 File Name Rules

The file name shall start with 'request', underscore, and the request identifier. The extension shall be 'xml'.

`request_<request i denti fi er>. xml`

### 10.3.2 DTD

```
<!DOCTYPE REQUEST [  
<!ELEMENT REQUEST (REQI D, OBSI D, I NSTRUMENT, PROFI LE)>  
  
<!ELEMENT REQI D>  
<!ATTLIST REQI D VALUE CDATA #REQUIRED>  
  
<!ELEMENT OBSI D>  
<!ATTLIST REQI D VALUE CDATA #REQUIRED>  
  
<!ELEMENT I NSTRUMENT>  
<!ATTLIST I NSTRUMENT VALUE (hi fi | pacs | spi re) #REQUIRED>  
  
<!ELEMENT PROFI LE>  
<!ATTLIST PROFI LE VALUE CDATA #REQUIRED>  

```

#### Example

File name: `request_1.xml`

Content:

```
<?xml versi on="1.0" encodi ng="UTF-8"?><!DOCTYPE REQUEST SYSTEM  
'http://archi ves. esac. esa. i nt/hsa/ai o/conf/ondemand_request. dtd' >
```

```
<REQUEST>
  <REQID VALUE="14293011-0" />
  <OBSID VALUE="1342186278" />
  <INSTRUMENT VALUE="SPIRE" />
  <PROFILE VALUE="systematic" />
</REQUEST>
```

## 10.4 On-demand Result

This section specifies the On-Demand Results File object.

### 10.4.1 File Name Rules

The file name shall start with 'request', underscore, and the request identifier. The extension shall be 'result.xml'.

```
request_<request identifier>.result.xml
```

### 10.4.2 DTD

```
<!DOCTYPE RESULT [
  <!ELEMENT RESULT (REQID, OBSID, INSTRUMENT, PROFILE, STATE, DATA, MESSAGE)>

  <!ELEMENT REQID>
  <!ATTLIST REQID VALUE CDATA #REQUIRED>

  <!ELEMENT OBSID>
  <!ATTLIST REQID VALUE CDATA #REQUIRED>

  <!ELEMENT INSTRUMENT>
  <!ATTLIST INSTRUMENT VALUE (hifi|pacs|spire) #REQUIRED>

  <!ELEMENT PROFILE>
  <!ATTLIST PROFILE VALUE CDATA #REQUIRED>

  <!ELEMENT STATE>
  <!ATTLIST STATE VALUE (success|failed) #REQUIRED>

  <!ELEMENT DATA>
  <!ATTLIST DATA VALUE CDATA #IMPLIED>
```

```
<!ELEMENT MESSAGE (#PCDATA)>
```

```
]>
```

### Example

File name: `request_1.result.xml`

Content:

```
<RESULT>
```

```
  <REQID VALUE="request id"/>
```

```
  <OBSID VALUE="obsid"/>
```

```
  <INSTRUMENT VALUE="hifi|pacs|spire"/>
```

```
  <PROFILE VALUE="profile"/>
```

```
  <STAGE VALUE="success|failed"/>
```

```
  <DATA VALUE="null|file or folder name"/>
```

```
  <MESSAGE>
```

```
    optional field containing, for example, an error message if the request was flagged as failed
```

```
  </MESSAGE>
```

```
</RESULT>
```

## 11 APPENDIX C: HAIO RESPONSE/QUERY EXAMPLES

### 11.1 Product Metadata Request

- Get HCSS\_URN by query (get the HCSS\_URN metadata value that matches with an specified query). The query shall be a Jython boolean expression.
  - ◆ command: `HCSS_CLASS_TYPE=PRODUCT&QUERY`  
example: `instrument == 'HIFI' and type == 'TYPE1'`
  - ◆ http equivalent:  
`http://archives.esac.esa.int/hsa/aio/jsp/metadata.jsp?RESOURCE_CLASS=PRODUCT&QUERY=instrument%3D%3D'HIFI'%20and%20type%3D%3D'TYPE1'`
  - ◆ SQL equivalent:  
`SELECT HCSS_URN FROM DATA WHERE instrument = 'HIFI' and type = 'TYPE1'`
- Get HCSS\_URN by query (get the HCSS\_URN metadata value that matches with an specified query) grouped by HCSS\_TRACK\_ID, returning the greatest HCSS\_TRACK\_VERSION only. The query shall be a Jython boolean expression.
  - ◆ command: `HCSS_CLASS_TYPE=PRODUCT&QUERY`  
example: `instrument == 'HIFI' and type == 'TYPE1'`

## ◆ http equivalent:

```
http://archives.esac.esa.int/hsa/aio/jsp/metadata.jsp?RESOURCE_CLASS=PRODUCT&
QUERY=instrument%3D%3D'HIFI'%20and%20type%3D%3D'TYPE1'&LAST_TRACK_VERSION=true
```

## ◆ SQL equivalent (pseudo syntax):

```
SELECT HCSS_URN FROM DATA WHERE instrument = 'HIFI' and type = 'TYPE1' GROUP BY
(HCSS_TRACK_ID AND MAX(HCSS_TRACK_VERSION))
```

## • Specifying a query results limit:

◆ command: `HCSS_CLASS_TYPE=PRODUCT&QUERY=[query]&LIMIT=nnn`

## ◆ Example:

```
http://archives.esac.esa.int/hsa/aio/jsp/metadata.jsp?RESOURCE_CLASS=PRODUCT&
QUERY=instrument%3D%3D'HIFI'%20and%20type%3D%3D'TYPE1'&LIMIT=5000
```

## • Showing access rights:

## ◆ command:

```
HCSS_CLASS_TYPE=PRODUCT&QUERY=[query]&SHOW_PROPRIETARY=true
```

## ◆ Example:

```
http://archives.esac.esa.int/hsa/aio/jsp/metadata.jsp?RESOURCE_CLASS=PRODUCT&
QUERY=instrument%3D%3D'HIFI'%20and%20type%3D%3D'TYPE1'&SHOW_PROPRIETARY=true
```

## 11.2 Product Metadata VOTable response

Same as the VOTable specified in *HAIO Product Resource Class VOTable response*.

## 11.3 HAIO HCSS\_CLASS\_TYPE Resource Class Request

Using the `RESOURCE_CLASS=HCSS_CLASS_TYPE` of the HAIO metadata query, the following information can be obtained:

## • Get the greatest HCSS\_URN\_VERSION of an HCSS\_CLASS\_TYPE

## ◆ http equivalent:

```
http://archives.esac.esa.int/hsa/aio/jsp/metadata.jsp?RESOURCE_CLASS=HCSS_CLASS_TYP
E&HCSS_CLASS_TYPE=herschel.ia.dataset.Product
```

## ◆ Command Result: VOTable response containing one or none records as in the example 11.4.

In case, the requested HCSS\_CLASS\_TYPE is found, the LAST\_VERSION field should be parsed.

## • Get all the different values of the metadata HCSS\_CLASS\_TYPE

## ◆ http equivalent:

[http://archives.esac.esa.int/hsa/aio/jsp/metadata.jsp?RESOURCE\\_CLASS=HCSS\\_CLASS\\_TYPE](http://archives.esac.esa.int/hsa/aio/jsp/metadata.jsp?RESOURCE_CLASS=HCSS_CLASS_TYPE)

- ◆ Command Result: VOTable response containing one record per HCSS\_CLASS\_TYPE present in the HSA database (as in the example 11.4)

- Check if an HCSS\_CLASS\_TYPE is stored

- ◆ http equivalent:

[http://archives.esac.esa.int/hsa/aio/jsp/metadata.jsp?RESOURCE\\_CLASS=HCSS\\_CLASS\\_TYPE&HCSS\\_CLASS\\_TYPE=herschel.ia.dataset.Product](http://archives.esac.esa.int/hsa/aio/jsp/metadata.jsp?RESOURCE_CLASS=HCSS_CLASS_TYPE&HCSS_CLASS_TYPE=herschel.ia.dataset.Product)

- ◆ Command Result: VOTable response containing one or none records as in the example 11.4

## 11.4 HAIIO HCSS\_CLASS\_TYPE Resource Class VOTable response

Using the **RESOURCE\_CLASS=HCSS\_CLASS\_TYPE** of the HAIIO metadata query, the following response can be obtained:

```
<VOTABLE version="1.1">
<RESOURCE type="results">
  <DESCRIPTION>HSA Metadata Query Service</DESCRIPTION>
  <INFO name="QUERY_STATUS" value="OK" />
  <TABLE>
    <FIELD ID="hcss_class_type" ucd="HCSS_CLASS_TYPE" datatype="char" arraysize="*" />
    <FIELD ID="last_version" ucd="HCSS_URN_VERSION" datatype="int" />
    <DATA>
      <TABLEDATA>
        <TR>
          <TD>herschel.ia.obs.quality.QualityContext</TD>
          <TD>7</TD>
        </TR>
      </TABLEDATA>
    </DATA>
  </TABLE>
</RESOURCE>
</VOTABLE>
```

## 11.5 HAIIO Product Resource Class Request

Using the **RESOURCE\_CLASS=PRODUCT** of the HAIIO metadata query, the following information can be obtained:

- Check if an HCSS\_URN is stored

- ◆ http equivalent:  

```
http://archives.esac.esa.int/hsa/aio/jsp/metadata.jsp?RESOURCE_CLASS=PRODUCT&urn=urn:hsa:herschel.ia.dataset.Product:0
```
- ◆ Command Result: VOTable response containing one or none records as in the example 11.6
- Get the HCSS\_FILE\_NAME metadata value searching by an HCSS\_URN
  - ◆ http equivalent:  

```
http://archives.esac.esa.int/hsa/aio/jsp/metadata.jsp?RESOURCE_CLASS=PRODUCT&urn=urn:hsa:herschel.ia.dataset.Product:0
```
  - ◆ Command Result: VOTable response containing one or none records as in the example 11.6
- Check if an HCSS\_FILE\_NAME is stored
  - ◆ http equivalent:  

```
http://archives.esac.esa.int/hsa/aio/jsp/metadata.jsp?RESOURCE_CLASS=PRODUCT&FILENAME=Product_v1.fits
```
  - ◆ Command Result: VOTable response containing one or none records as in the example 11.6

## 11.6 HAI0 Product Resource Class VOTable response

Using the **RESOURCE\_CLASS=PRODUCT** of the HAI0 metadata query, the following response can be obtained:

```
<VOTABLE version="1.1">
<RESOURCE type="results">
  <DESCRIPTION>HSA Metadata Query Service</DESCRIPTION>
  <INFO name="QUERY_STATUS" value="OK" />

  <TABLE>
    <FIELD ID="Reference"          ucd="DATA_LINK"          datatype="char" arraysize="*" />
    <FIELD ID="URN"                ucd="HCSS_URN"          datatype="char" arraysize="*" />
    <FIELD ID="HCSSFileName"       ucd="HCSS_FILE_NAME"    datatype="char" arraysize="*" />
    <FIELD ID="HCSSClassType"       ucd="HCSS_CLASS_TYPE"   datatype="char" arraysize="*" />
    <FIELD ID="HCSSURNVersion"      ucd="HCSS_URN_VERSION"  datatype="int" />
    <FIELD ID="HCSSTrackID"         ucd="HCSS_TRACK_ID"     datatype="char" arraysize="*" />
    <FIELD ID="HCSSTrackVersion"    ucd="HCSS_TRACK_VERSION" datatype="int" />
    <FIELD ID="HCSSSize"            ucd="HCSS_SIZE"         datatype="int" />
    <FIELD ID="HCSSTotalSize"       ucd="HCSS_TOTAL_SIZE"   datatype="int" />
    <FIELD ID="HCSSHash"           ucd="HCSS_HASH"         datatype="char" arraysize="*" />
  <DATA>
  <TABLEDATA>
  <TR>
```

```

        <TD>http://archives.esac.esa.int/hsa/ai o/j sp/product. j sp?urn=urn: hsa: herschel . i a. o
        bs. Observati onContext: 1&protocol =HTTP</TD>
        <TD>urn: hsa: herschel . i a. obs. Observati onContext: 1</TD>
        <TD>hhi fi 00000001obs. fi ts </TD>
        <TD>herschel . i a. obs. Observati onContext</TD>
        <TD>1</TD>
        <TD>OBS_1207830806113281000</TD>
        <TD>0</TD>
        <TD>161111</TD>
        <TD>161111</TD>
        <TD>206522a3a1f7ff43d0b82fa40ff409bd</TD>
    </TR>
</TABLEDATA>
</DATA>
</TABLE>
</RESOURCE>
</VOTABLE>

```

If **SHOW\_PROPRIETARY=true** is added to **RESOURCE\_CLASS=PRODUCT** request, the following response can be obtained (**HAS\_ACCESS** is added):

```

<VOTABLE version="1.1">
<RESOURCE type="results">
    <DESCRIPTION>HSA Metadata Query Service</DESCRIPTION>
    <INFO name="QUERY_STATUS" value="OK" />

    <TABLE>
        <FIELD ID="Reference"          ucd="DATA_LINK"          datatype="char" arraysize="*" />
        <FIELD ID="URN"                 ucd="HCSS_URN"          datatype="char" arraysize="*" />
        <FIELD ID="HCSSFile name"       ucd="HCSS_FILE_NAME"    datatype="char" arraysize="*" />
        <FIELD ID="HCSSClassification"  ucd="HCSS_CLASS_TYPE"   datatype="char" arraysize="*" />
        <FIELD ID="HCSSUrnVersion"      ucd="HCSS_URN_VERSION"  datatype="int" />
        <FIELD ID="HCSSTrackID"         ucd="HCSS_TRACK_ID"     datatype="char" arraysize="*" />
        <FIELD ID="HCSSTrackVersion"    ucd="HCSS_TRACK_VERSION" datatype="int" />
        <FIELD ID="HCSSSize"            ucd="HCSS_SIZE"         datatype="int" />
        <FIELD ID="HCSSTotalSize"       ucd="HCSS_TOTAL_SIZE"   datatype="int" />
        <FIELD ID="HCSSHash"            ucd="HCSS_HASH"        datatype="char" arraysize="*" />
        <FIELD ID="HAS_ACCESS"          ucd="HAS_ACCESS"        datatype="char" arraysize="*" />
    </TABLE>
    <TABLEDATA>
    <TR>
        <TD>http://archives.esac.esa.int/hsa/ai o/j sp/product. j sp?urn=urn: hsa: herschel . i a. o
        bs. Observati onContext: 1&protocol =HTTP</TD>
        <TD>urn: hsa: herschel . i a. obs. Observati onContext: 1</TD>
        <TD>hhi fi 00000001obs. fi ts </TD>

```

```
<TD>herschel . i a. obs. Observati onContext</TD>
<TD>1</TD>
<TD>OBS_1207830806113281000</TD>
<TD>0</TD>
<TD>161111</TD>
<TD>161111</TD>
<TD>206522a3a1f7ff43d0b82fa40ff409bd</TD>
<TD>>true</TD>
</TR>
</TABLEDATA>
</DATA>
</TABLE>
</RESOURCE>
</VOTABLE>
```

## 11.7 *HAIO HCSS\_TRACK\_ID Resource Class Request*

Using the **RESOURCE\_CLASS =HCSS\_TRACK\_ID** of the HAIO metadata query, the following information can be obtained:

- Get the greatest HCSS\_TRACK\_VERSION of an HCSS\_TRACK\_ID
  - ◆ http equivalent:  
`http://archives.esac.esa.int/hsa/aio/jsp/metadata.jsp?RESOURCE_CLASS=HCSS_TRACK_ID&HCSS_TRACK_ID=OBS_1207830806113281000`
  - ◆ Command Result: VOTable response containing one or none records as in the example 11.8.  
In case, the requested HCSS\_TRACK\_ID is found, the HCSS\_TRACK\_VERSION field should be parsed.
- Get all the greatest HCSS\_TRACK\_VERSION of each HCSS\_TRACK\_ID searching by HCSS\_CLASS\_TYPE
  - ◆ http equivalent:  
`http://archives.esac.esa.int/hsa/aio/jsp/metadata.jsp?RESOURCE_CLASS=HCSS_TRACK_ID&HCSS_CLASS_TYPE=herschel.ia.obs.ObservationContext`
  - ◆ Command Result: VOTable response containing one or more records as in example 11.8.

## 11.8 *HAIO HCSS\_TRACK\_ID Resource Class VOTable response*

Same as the VOTable specified in *HAIO Product Resource Class VOTable response*.



## 11.9 HAI0 PRODUCT\_METADATA Resource Class Request

Using the **RESOURCE\_CLASS=PRODUCT\_METADATA** of the HAI0 metadata query, the following information can be obtained:

- Get all the ingested metadata associated to a product by HCSS\_URN

- ◆ http equivalent:

```
http://archives.esac.esa.int/hsa/aio/jsp/metadata.jsp?RESOURCE_CLASS=PRODUCT_METADATA&urn=urn:hsa:herschel.ia.dataset.Product:0
```

- ◆ Command Result: VOTable response containing one or none records as in the example

11.10.

## 11.10 HAI0 PRODUCT\_METADATA Resource Class VOTable response

Using the **RESOURCE\_CLASS=PRODUCT\_METADATA** of the HAI0 metadata query, the following response can be obtained:

```
<VOTABLE version="1.1">
<RESOURCE type="results">
  <DESCRIPTION>HSA Metadata Query Service</DESCRIPTION>
  <INFO name="QUERY_STATUS" value="OK" />

  <TABLE>
    <FIELD ID="KEY"          ucd="KEY"          datatype="char" arraysize="*" />
    <FIELD ID="TYPE"         ucd="TYPE"         datatype="char" arraysize="*" />
    <FIELD ID="VALUE"        ucd="VALUE"        datatype="char" arraysize="*" />
    <DATA>
      <TABLEDATA>
        <TR>
          <TD>description</TD>
          <TD>string</TD>
          <TD></TD>
        </TR>
        <TR>
          <TD>instrument</TD>
          <TD>string</TD>
          <TD>HIFI</TD>
        </TR>
        ...
      </TABLEDATA>
    </DATA>
  </TABLE>
```

&lt;/RESOURCE&gt;

&lt;/VOTABLE&gt;

## 11.11 Retrieving a File from HSA using HAIO

The **protocol=HTTP** http parameter, allows HAIO to stream the file to the client. If this parameter is removed, an intermediate html page will be displayed in a human readable way.

- Product retrieval
  - ◆ http equivalent:

```
http://archives.esac.esa.int/hsa/aio/jsp/product.jsp?urn=urn:hsa:herschel.ia.dataset.Product:0&protocol=HTTP
```
  - ◆ Result: the already ingested product in FITS format through the http stream (due to **protocol=HTTP** http parameter). If there is a problem/error the HAIO server will raise an HTTP Error (see Appendix G).
  
- Metadata retrieval
  - ◆ http equivalent:

```
http://archives.esac.esa.int/hsa/aio/jsp/product.jsp?urn=urn:hsa:herschel.ia.dataset.Product:0&metadata=true&protocol=HTTP
```
  - ◆ Result: a FITS file containing the metadata of an already ingested product through the http stream (due to **protocol=HTTP** http parameter). If there is a problem/error the HAIO server will raise an HTTP Error (see Appendix G).
  
- XML request ingestion file retrieval
  - ◆ http equivalent:

```
http://archives.esac.esa.int/hsa/aio/jsp/product.jsp?requestfile_xml=1234-herschel.ia.dataset.Product-0.xml&protocol=HTTP
```
  - ◆ Result: the XML request ingested file through the http stream (due to **protocol=HTTP** http parameter). If there is a problem/error the HAIO server will raise an HTTP Error (see Appendix G).

## 11.12 Login in to HAIO

A client can do login into the HAIO by invoking the next URL

- ◆ http equivalent:

```
http://archives.esac.esa.int/hsa/aio/jsp/login.jsp?username=<username>&password=<password>&protocol=HTTP
```

As explained in point 2.3.2.7, the server will return a HTTP cookie that should be used in the following product retrieval to access proprietary data.

In case of login problem, an HTTP error (login error) will be raised (see Appendix G).

## 11.13 Observation Metadata Request

- Check if a specified observation is stored.

- ◆ http equivalent:

```
http://archives.esac.esa.int/hsa/aio/jsp/metadata.jsp?RESOURCE_CLASS=OBSERVATION&ObsId=1342187465
```

- ◆ SQL equivalent:

```
SELECT * FROM OBSERVATION WHERE OBSERVATION_ID = 1342187465
```

- Get all metadata values per observation which has been made with and specific instrument and observing mode, by query. The query shall be a Jython boolean expression.

- ◆ http equivalent:

```
http://archives.esac.esa.int/hsa/aio/jsp/metadata.jsp?RESOURCE_CLASS=OBSERVATION&QUERY=%28INSTRUMENT%3D%3D%27HIFI%27%20AND%20INSTRUMENT_OBS_MODE==%27HifiSScanModeDBS%27%29&SELECTED_FIELDS=OBSERVATION&RETURN_TYPE=VOTABLE
```

- ◆ SQL equivalent (pseudo syntax):

```
select * from hsa.observation
where observation_id in (
  select o.observation_id
  from hsa.observation o, hsa.instrument i, hsa.observing_mode om
  where o.instrument_oid = i.instrument_oid
  and i.instrument_name like 'HIFI'
  and o.observing_mode_oid = om.observing_mode_oid
  and om.observing_mode_name like 'HifiSScanModeDBS'
  group by o.observation_id)
```

- Specifying a query results limit:

## ◆ http equivalent:

```
http://archives.esac.esa.int/hsa/aio/jsp/metadata.jsp?RESOURCE_CLASS=OBSERVATION&
QUERY=%28INSTRUMENT%3D%3D%27HIFI%27%20AND%20INSTRUMENT_OBS_MODE%3D%3D%27HifiSScanMo
deDBS%27%29&SELECTED_FIELDS=OBSERVATION&LIMIT=5000&RETURN_TYPE=VOTABLE
```

## 11.14 Observation Metadata VOTable response

Using the **RESOURCE\_CLASS=OBSERVATION** of the HAIIO metadata query, the following response can be obtained:

```
<VOTABLE version="1.1">
<RESOURCE type="results">
  <DESCRIPTION>
    HERSCHEL Metadata Query Service:
    Total number of records available: 1
  </DESCRIPTION>
  <INFO name="QUERY_STATUS" value="OK"/>
  <TABLE>
    <FIELD ID="Reference"          ucd="DATA_LINK"          datatype="char" arraysize="1"/>
    <FIELD ID="ObsId"              ucd="OBS_ID"             datatype="char" arraysize="1"/>
    <FIELD ID="Urn_Version"        ucd="URN_VERSION"        datatype="char" arraysize="1"/>
    <FIELD ID="Spg_Id"             ucd="SPG_ID"             datatype="char" arraysize="1"/>
    <FIELD ID="Instrument"         ucd="INSTRUMENT"         datatype="char" arraysize="1"/>
    <FIELD ID="Instrument_mode"    ucd="OBSERVING_MODE"     datatype="char" arraysize="1"/>
    <FIELD ID="RA"                 ucd="POS_EQ_RA_MAIN"     datatype="double"/>
    <FIELD ID="RA"                 ucd="POS_EQ_DEC_MAIN"    datatype="double"/>
    <FIELD ID="PA"                 ucd="PA"                 datatype="double"/>
    <FIELD ID="Title"              ucd="VOX: Image_Title"   datatype="char" arraysize="1"/>
    <FIELD ID="Start_Time"         ucd="VOX: OBS_START_TIME" datatype="char" arraysize="1"/>
    <FIELD ID="End_Time"           ucd="VOX: OBS_END_TIME"  datatype="char" arraysize="1"/>
    <FIELD ID="Duration"           ucd="DURATION"           datatype="char" arraysize="1"/>
    <FIELD ID="Proposer"           ucd="PROPOSER"           datatype="char" arraysize="1"/>
    <FIELD ID="Explanation_Date"   ucd="VOX: EXPLANATION_DATE" datatype="char" arraysize="1"/>
    <FIELD ID="OD_Number"          ucd="OD_NUMBER"          datatype="double"/>
    <FIELD ID="Is_Active_Version"  ucd="IS_ACTIVE_VERSION"  datatype="char" arraysize="1"/>
    <FIELD ID="PROPOSAL_ID"        ucd="PROPOSAL_ID"        datatype="char" arraysize="1"/>
    <FIELD ID="Aor"                ucd="AOR"                datatype="char" arraysize="1"/>
    <FIELD ID="Status"             ucd="STATUS"             datatype="char" arraysize="1"/>
    <FIELD ID="Obs_State"          ucd="OBSSTATE"           datatype="char" arraysize="1"/>
  <DATA>
  <TABLEDATA>
  <TR>
    <TD>
```

---

```
http://archives.esac.esa.int/ehsdev/ai o/j sp/product.jsp?PROTOCOL=HTTP&OBSER
VATION_OID=81345
</TD>
<TD>1342187465</TD>
<TD>78070</TD>
<TD>SPG v6.1.0</TD>
<TD>SPIRE</TD>
<TD>Spi rePhotoLargeScan</TD>
<TD>323.7904106072874</TD>
<TD>-1.066043611273894</TD>
<TD>247.74167380662294</TD>
<TD>SMMJ212312</TD>
<TD>2009-12-01 00:34:34.0</TD>
<TD>2009-12-01 00:46:48.0</TD>
<TD>734000.0</TD>
<TD>pvspre</TD>
<TD>2009-12-01 00:34:34.0</TD>
<TD>201</TD>
<TD>>true</TD>
<TD>Calibration_pvspre_56</TD>
<TD>
Calibration_phot_2-Spi rePhotoLargeScanGen-1scanx1scanRep4
</TD>
<TD>null value</TD>
<TD>LEVEL2_PROCESSED</TD>
</TR>
</TABLEDATA>
</DATA>
</TABLE>
</RESOURCE>
</VOTABLE>
```

## 12 APPENDIX D: FITS KEYWORDS USED IN HERSCHEL

Metadata keywords used in Herschel are written in FITS files using a translation mechanism in order to adapt the Herschel metadata keywords to the FITS keywords (sizes).

The mechanism creates a 'HIERARCH' FITS keyword that contains the Herschel metadata keyword and the FITS keyword.

For instance, the Herschel metadata 'instrument' is translated into 'INSTRUME' FITS keyword.

The FITS file will contain the following lines:

```
INSTRUME= 'Unknown ' / Instrument attached to this product
HIERARCH key.INSTRUME='Instrument'
```

The Herschel metadata keyword is 'instrument'.

The FITS keyword is 'INSTRUME'. The value is 'Unknown'.

HIERARCH keyword links the FITS key 'INSTRUME' with the Herschel keyword 'instrument'

## 13 APPENDIX E: WHO IMPLEMENTS WHAT

### 13.1 HCSS

- IA session (Jide/Hipe)
- SPG
- HCSS-HSA Writable Pool (HsaPoolDaemon + HsaWritePool)
- HsaPoolDaemon
- Ingestion Request File
- Ingestion Storage/Request Area
- HsaWritePool
- HsaReadPool
- On-Demand server
- On-Demand Results file
- Zipped on-demand reprocessed observation
- HcssHuiTool
  - HUI launcher
  - Interaction between IA and the Message Broker for intercommunication between

## HUI and IA

## 13.2 HSA

- HINGEST
- HIMPORT
- HAIO
- HUI
- HUI On-demand Monitoring Tool
- FTP area for downloading
- On-Demand Storage/Request Area: shared NFS Directory for reprocessed Observations (On-demand Observations Reprocessing)
- On-Demand Request file
- Ingestion Request Error file
- Extended processing access restrictions verification

## 14 APPENDIX F: RULES FOR CREATING THE HSA TARBALL OBJECT

The XML Ingestion Request File used during ingestion contains the hierarchical relationships between products (see Ingestion Request File hierarchical structure in section 10.1.3).

Each product is an `Item` element. For instance, an `ObservationContext`, which contains a Level 0 Context will be written as:

```
<request id="12345678" ... >
  <item id="12345678-herschel . i a. obs. Observati onContext-nnn"
    di stri buti on_path="/12345678"
    di stri buti on_fi lename="observati on_fi l e_name. fi ts"
    cl ass="herchel . i a. obs. Observati onContext" ... >
    <item id="level 0"
      di stri buti on_path="/12345678/level 0"
      di stri buti on_fi lename="map_context_fi l e. fi ts"
      cl ass="herchel . i a. pal . MapContext" ... >
    </item>
  </item>
```

Each `Item` contains a `di stri buti on_path` and a `di stri buti on_fi lename` attributes. The path where a

product must be saved is the composition of these two attributes: `di stri buti on_path` and a `di stri buti on_fi lename`

- General rule: `di stri buti on_path / di stri buti on_fi lename`

For instance, an Observation context, which contains a Level 0 context (with a product) and an Auxiliary context (with another product), written as:

```
<request id="12345678" ... >
  <i tem id="12345678-herschel . i a. obs. Observati onContext-nnn"
    di stri buti on_path="/12345678"
    di stri buti on_fi lename="observati on_context. fi ts"
    cl ass="herchel . i a. obs. Observati onContext"
    fi l e_sub_path="[path]/observati on_context. fi ts" ... >
    <i tem id="level 0"
      di stri buti on_path="/12345678/level 0"
      di stri buti on_fi lename="level 0. fi ts"
      cl ass="herchel . i a. pal . MapContext"
      fi l e_sub_path="[path]/level 0. fi ts" ... >
      <i tem id="mml x"
        di stri buti on_path="/12345678/level 0/mml x"
        di stri buti on_l i st="mml x. fi ts"
        cl ass="herchel . i a. dataset. Product"
        fi l e_sub_path="[path]/mml x. fi ts" ... >
      </i tem>
    </i tem>
    ...
    <i tem id="auxil i ary"
      di stri buti on_path="/auxil i ary"
      di stri buti on_fi lename="auxil i ary_context. fi ts"
      cl ass="herchel . i a. obs. auxil i ary. Auxil i aryContext"
      fi l e_sub_path="[path]/auxil i ary_context. fi ts" ... >
      <i tem id="sRem"
        di stri buti on_path="/auxil i ary/sRem"
        di stri buti on_fi lename="srem. fi ts"
        cl ass="herchel . i a. obs. auxil i ary. srem. SremProduct"
        fi l e_sub_path="[path]/srem. fi ts" ... >
      </i tem>
    </i tem>
  </i tem>
</request>
```

requires the following paths:

- for the observation Context:



[/12345678/observation\\_context.fits](#)

- for the level 0 context:

[/12345678/level 0/level 0.fits](#)

- for the level 0 product:

[/12345678/level 0/mml x/mml x.fits](#)

- for the auxiliary context:

[/auxiliary/auxiliary\\_context.fits](#)

- for the auxiliary product:

[/auxiliary/sRem/srem.fits](#)

The XML Ingestion Request File must be provided inside a directory named **.exported** located at tarball top level.

## 15 APPENDIX G: HAIO HTTP ERROR CODES

### 15.1 PRODUCT RETRIEVAL

The following table shows the HTTP error codes when retrieving products through HAIO server.

<b>Error code</b>	<b>Format</b>	<b>Comments</b>
600	Process Errors: <extra information>	General unexpected error.
601	Proprietary Error: <extra information>	User is not allowed to download this product. Product has proprietary information.
602	Session expired	Inactive time exceeded.
603	Malformed request	The product retrieval pattern is not a valid one or not understood by the server.

According with HCSS 8.0, HAIO server will also shows the following HTTP error codes when the client is HIPE v8.0 or higher (see appendix L):

<b>Error code</b>	<b>Format</b>	<b>Comments</b>
600	Process Errors: <extra information>	General unexpected error.
461	Proprietary Error: <extra information>	User is not allowed to download this product. Product has proprietary information.
462	Session expired	Inactive time exceeded.
400	Malformed request	The product retrieval pattern is not a valid one or not understood by the server.

## 15.2 LOGIN

The following table shows the HTTP error codes when requesting login to HAIO server.

<b>Error code</b>	<b>Format</b>	<b>Comments</b>
604	Login problems: <extra information>	A list of possible reasons for this error: <ul style="list-style-type: none"><li>• User does not exist</li><li>• Password is not valid</li><li>• LDAP connection problems</li></ul>

According with HCSS 8.0, HAIO server will also shows the following HTTP error codes when the client is HIPE v8.0 or higher (see appendix L):

<b>Error code</b>	<b>Format</b>	<b>Comments</b>
464	Login problems: <extra information>	A list of possible reasons for this error: <ul style="list-style-type: none"><li>• User does not exist</li><li>• Password is not valid</li><li>• LDAP connection problems</li></ul>

## 16 APPENDIX H: VOTABLE QUERY RESPONSE ERRORS

### 16.1 VOTABLE ERRORS

The following table shows the values for the VOTable (that contains a query result) response **info** item:

<b>Value</b>	<b>Comments</b>
<b>OK</b>	The query was successfully executed.
<b>ERROR 605</b>	The number of query results exceed the maximum number of results that the HAIO system can serve.
<b>ERROR 606</b>	The number of query results exceed the maximum number of results specified by the client through the <b>LIMIT</b> parameter (see MSG-2.1-002).
<b>ERROR <i>msg</i></b>	Another error. <i>msg</i> explains the problem.

According with HCSS 8.0, HAIO server will also shows the following HTTP error codes when the client is HIPE v8.0 or higher (see appendix L):

<b>Value</b>	<b>Comments</b>
<b>OK</b>	The query was successfully executed.
<b>ERROR 465</b>	The number of query results exceed the maximum number of results that the HAIO system can serve.
<b>ERROR 466</b>	The number of query results exceed the maximum number of results specified by the client through the <b>LIMIT</b> parameter (see MSG-2.1-002).
<b>ERROR <i>msg</i></b>	Another error. <i>msg</i> explains the problem.

## 17 APPENDIX I: VIRTUAL TAR

### 17.1 STRUCTURE

The Virtual TAR file is composed of entries with the following structure:

`I dentI fi er / HCSS_CLASS_TYPE / FI LE_NAME`

Where:

<i>Item</i>	<i>Value</i>
<code>I dentI fi er</code>	A random string.
<code>HCSS_CLASS_TYPE</code>	A Herschel product class type.
<code>FI LE_NAME</code>	A Herschel product file name.

## 18 APPENDIX J: QUERY RESULTS LIMITS

### 18.1 LIMITS

The following table shows the values that specify the query results limits:

<i>Limit by</i>	<i>Example value</i>	<i>Comment</i>
Server	25000	Defined by configuration.
Client	5000 (default value)	Optional, specified by using the <b>LIMIT</b> query parameter (see MSG-2.1-001).

## 19 APPENDIX K: EXTENDED PROCESSING INGESTION CHECK

When an extended processing ingestion is requested, the XML ingestion request file contains the **extended** attribute. This attribute contains all the observation identifiers involved in the extended processing (a list of comma separated values), for instance:

```
<request id="12345678" ... extended="12345678, 12345679, 12345680" ...>
```

The ingestion check procedure must verify the access restrictions for each observation involved:

1. if ALL observation identifiers are PUBLIC:  
No checks are required.
2. if one or more observation identifiers is/are PRIVATE:  
CHECK: ALL expiration dates are the same.  
OK: process  
NOK: error

## 20 APPENDIX L: 'USER-AGENT' HTTP HEADER FORMAT

The HIPE client shall be specified using 'User Agent' HTTP header with the following format:

```
User-Agent: <msg> : v.n.m
```

Where:

- <msg> is a string that does not contain ':' (colon)
- v, n and m are numbers.

Example:

```
User-Agent: HIPE (Herschel Interactive Processing Environment): 8.0.3145
```

## 21 APPENDIX M: ALLOWED QUERY OPERATORS

The allowed query operators (in the query expression) are:

<b>Precedence</b>	<b>Type</b>	<b>Operator</b>	<b>Symbol</b>
1	Grouping	Parenthesis	()
2	Relational	Equals	==
2	Relational	Not equals	!=
2	Relational	Greater than	>
2	Relational	Lower than	<
2	Relational	Greater or equals than	>=
2	Relational	Lower or equals than	<=
2	Relational	Like: '%' => multiple characters '_' => one character	LIKE
3	Logic	Not	not
4	Logic	And	and
5	Logic	Exclusive or	xor
6	Logic	Or	or

For string metadata both, single and double quotes shall be supported.