Customer	:	**ESRIN**	**Document Ref**	:	**SST_CCI-SSDv1-BC-001**
WP No	:		**Issue Date**	:	**17 July 2013**
			Issue	:	**1.2**

Project : **CCI Phase 1 (SST)**

Title : **CCI-SST System Specification Document**

Abstract : This document contains the SSD for the ESA SST_cci project.

Author :

R. Quast, N. Fomferra,
M. Boettcher
Brockmann Consult GmbH

Checked :

C. Merchant, Science Leader
University of Reading

P. Spinks, Project Manager
Space ConneXions Ltd.

Accepted :

Digitally signed by Craig Donlon
DN: cn=Craig Donlon, o=EOP-SME, ou=ESA/ESTEC, email=craig.donlon@esa.int, c=NL
Date: 2013.07.18 13:13:45 +02'00'

C. Donlon
ESA Technical Officer

Distribution : SST_cci team members

ESA (Craig Donlon)

sst
cci

AMENDMENT RECORD

This document shall be amended by releasing a new edition of the document in its entirety. The Amendment Record Sheet below records the history and issue status of this document.

AMENDMENT RECORD SHEET

ISSUE	DATE	REASON FOR CHANGE
0.2	25 Jul 2012	Document structure, TOC, table of figures
0.8	30 Oct 2012	Version for internal review
0.9	02 Jan 2013	Integration of team comments
1.0	29 Jan 2013	Initial version
1.1	27 Jun 2013	RID conclusions integrated
1.2	17 Jul 2013	Updated sizing and performance analysis and cost estimates

RECORD OF CHANGES IN THIS ISSUE

Issue	Page/Sec.	Reason	Change
1.2	Sec. 7.3, 7.4		Updated sizing and performance analysis and cost estimates
1.1	Sec. 1.1	RID CD-1	Reference to funding removed
1.1	Sec 2.1	RID CD-2	Time series plot inserted
1.1	Fig. 2-2	RID CD-3 RID CD-11	Inserted new introductory section (Section 2) that links the activities in Phase 1 of SST CCI to its system elements
1.1	Sec. 2.1	RID CD-4	Section 1.4 "Document conventions" added
1.1	Sec. 2.2	RID CD-5	Table of external interfaces added to section 2.2.
1.1	Sec. 5.2 Table 5-2	RID CD-6	Table of main internal interfaces between subsystems added to section 2.4; tables of internal interfaces added to section 5.1.1, interfaces table in section 5.2.1 updated, interface identifiers and endpoints added
1.1	Sec. 2.2	RID CD-7	Wording changed to avoid misunderstanding that exchange of products with other ECVs is only an option. The system supports it, and it shall be used.
1.1	Sec. 2.3	RID CD-8	Wording changed from "acceptance" to "use" in order to avoid the association with "requirements".
1.1	Sec. 3.1	RID CD-9	Sentence added to explain the interaction with data providers. Figure 2-3 updated, feedback to data providers added

Issue	Page/Sec.	Reason	Change
1.1	Sec. 4.1 Figure 4-1	RID CD-12	Figure split into two, "prototype" avoided in figure to stress the alternative between distributed or centralised production.
1.1	Sec. 4.1.2 Sec. 4.1.1	RID CD-13 RID CD-14	Sentence added in the first paragraph of section 4.1 that user services are not touched by this trade-off.
1.1	Sec. 4.1.3	RID CD-15	Hybrid solution added as an option, same code base for stand-alone OSTIA added to section 4.1.2
1.1	Sec. 4.2.4	RID CD-16	6th bullet merged with 7th bullet, network infrastructure considered as facility of an existing environment
1.1	Sec. 4.3	RID CD-17	Question marks added
1.1	Sec. 4.3.2	RID CD-18	First sentence changed
1.1	Sec 4.3.3	RID CD-19	Common user services and common archive added with arguments for sharing
1.1	Sec. 4.3.3	RID CD-20	Rewording of first "arguments against ..." bullet
1.1	Sec. 4.4.1	RID CD-21 RID CD-23 (1)	Paragraph added to 4.4 that user services would be the same in both alternatives.
1.1	Sec. 4.4.2 Sec. 4.4.3	RID CD-22 RID CD-23 (2)	Paragraph added to section 4.4.3 that environments that provide the SLSTR data are an advantage
1.1	Sec. 4.6	RID CD-24 RID CD-25	Aspect of environment with SLSTR data available now features the "conclusion".
1.1	Sec. 5 Sec. 5.1.5	FID CD-33 RID CD-26	Forward reference to section 5.1 added to section 2.4
1.1	Sec. 5.1	RID CD-27	Function for bulk access with media circulation or cloud storage added to section 5.1, including forward reference to section 5.1.3.4 that handles bulk data access.
1.1	Sec. 5.1.2	RID CD-28	OPeNDAP and WMS explicitly named in the description of the figure, forward references to subsections added
1.1	Sec. 5.1.3.4	RID CD-30	Text updated, reference to requirement SST-SR-6220 added
1.1	Sec. 5.1.3.4	RID CD-31	Statement added to sections 4.4.2 and 5.1.3.4that analysis is based on cloud solutions available at the time of investigation.
1.1	Sec. 5.1.4	RID CD-32	Clarification added that catalogue data access is meant to be used for single products. Reference to bulk download added.
1.1	Sec. 5.1.6	RID CD-34	"optional" dropped from the description of the DMS, sentence added that the project benefits from shared managed documents.
1.1	Sec 5.1.7, Figure 5-8	RID CD-36	Figure updated, "optional" dropped to avoid misunderstanding that OPeNDAP and WMS servers themselves are optional, and not only the authentication, that will be necessary only if load turns out to be high.
1.1	Sec. 5.2.3, Figure 5-12	RID CD-37	"optionally" dropped from figure and text

Issue	Page/Sec.	Reason	Change
1.1	Sec. 5.2.4	RID CD-38	Forward reference to MMS section added
1.1	Sec. 5.2.5	RID CD-39	"logical sequence of steps" replaced by "logical dependency graph of steps"
1.1	Sec. 5.2.5	RID CD-40	Text below table of request fields modified to link requests and the resource management done for their processing.
1.1	Sec. 5.2.5	RID CD-41	"optionally" dropped for storage space management
1.1	Sec. 5.2.6	RID CD-43	Section 5.2.6 defines that data quality indicators are to be stored in the inventory for analysis and progress monitoring. Section 5.1.3 defines the collection of performance indicators for user data access.
1.1	Sec. 5.3.5	RID CD-45	New section 5.3.5.4 on the necessary development work for MMS added
1.1	Sec. 5.3.5.1	RID CD-46	Explanation added to make clear the difference between the pre-computed single-sensor matchups used in the prototype and the subscene and NWP extracts pre-computed by (!) the MMS as an optimisation for processing cycles and repeated MMD generation.
1.1	Sec. 5.3.5.3	RID CD-47	Table 5-4 with MMS interface identification information added
1.1	End of doc.	RID CD-48	Summary section added

EXECUTIVE SUMMARY

This Sea Surface Temperature (SST) System Specification Document proposes a design for an operational system for SST for the ESA Climate Change Initiative (CCI). The design is based on a few basic operational use cases and a trade-off analysis of design alternatives. The result is a functional design with components optimised for tasks as determined by user requirements for the SST climate data record: responsive bulk reprocessing of full records; support of cycles of improvement of algorithms in response to user feedback; and routine extension of CDRs with new data without disruption by reprocessing requirements.

TABLE OF CONTENTS

1. INTRODUCTION 10
1.1 Purpose and scope 10
1.2 References 10
1.3 Acronyms 12
1.4 Document conventions 13
1.5 Document structure 13
2. THE SST CCI PROJECT AND ITS ACTIVITIES 14
2.1 Cardinal requirements and system overview 14
2.2 Input data access 17
2.3 Multi sensor matchup system (MMS) 18
2.4 Level 2/3 processor 19
2.5 Level 4 processor 21
2.6 Distribution system and user feedback management 21
2.7 Activities in Phase 1 of SST CCI 23
2.7.1 Algorithm development for SST retrieval 23
2.7.2 Round-robin algorithm comparison and selection 24
2.7.3 Other algorithm developments 26
2.7.4 System verification 26
2.7.5 Product validation and climate assessment 27
2.7.6 Summary of project activities 27
3. OVERVIEW OF THE SST CCI SYSTEM 29
3.1 Purpose and intended use 29
3.2 Main functions 35
3.3 High level decomposition and main internal interfaces 36
4. SST CCI OPERATIONAL SCENARIOS 39
4.1 User roles 39
4.2 User information and data access 40
4.3 Processing and validation 41
4.4 Algorithm improvement 44
5. INFRASTRUCTURE ALTERNATIVES AND THEIR IMPLICATIONS 47
5.1 Enhancement of the distributed prototype, or central SST infrastructure? 47
5.1.1 Enhancement of the distributed prototype 49
5.1.2 New deployment in a central SST infrastructure 49
5.1.3 Pros and cons 50
5.2 Use of existing earth observation processing data centres 51
5.2.1 ESA MMFI for SST CCI? 51
5.2.2 Embedding of an SST CCI system into an operational environment 52
5.2.3 Operations instructed by the SST CCI development team 52

5.2.4 Pros and cons ... 53
5.3 Degrees of sharing with other ECVs ... 53
5.3.1 Sharing user services and long term archiving ... 54
5.3.2 Sharing a production and development environment ... 54
5.3.3 Pros and cons ... 55
5.4 Cluster or cloud? ... 55
5.4.1 Running the SST CCI system in a cluster ... 56
5.4.2 Running the SST CCI system in a cloud ... 57
5.4.3 Pros and cons ... 57
5.5 Processing middleware alternatives ... 58
5.5.1 Grid Engine ... 58
5.5.2 Calvalus/Hadoop ... 58
5.5.3 Pros and cons ... 60
5.6 Conclusion for SST CCI ... 61
6. FUNCTIONAL DESIGN ... 63
6.1 Services for the users ... 63
6.1.1 Components and interfaces ... 63
6.1.2 Structured output data storage ... 67
6.1.3 Data access to ECV products ... 68
6.1.4 Online catalogue ... 71
6.1.5 Web presentation and community interaction ... 73
6.1.6 Access to tools, documentation and algorithm implementations ... 77
6.1.7 User management ... 78
6.2 Production and reprocessing for consistent versions ... 81
6.2.1 Components and interfaces ... 81
6.2.2 Structured processing storage ... 88
6.2.3 Input ingestion ... 89
6.2.4 Data processors ... 90
6.2.5 Production control ... 91
6.2.6 Data product quality checks ... 99
6.3 Continuous algorithm improvement ... 100
6.3.1 Processor interfaces ... 100
6.3.2 Processor version concept ... 104
6.3.3 Virtualised test environment ... 106
6.3.4 Multi-sensor matchup datasets ... 107
6.3.5 Multi-sensor matchup system ... 109
6.4 System documentation ... 116
7. DEVELOPMENT, LIFE CYCLE, COST AND PERFORMANCE ... 118
7.1 Re-use and development ... 118
7.2 System life cycle drivers and considerations ... 119

7.3 Sizing and performance analysis 120
7.4 Cost analysis 124
8. REQUIREMENTS TRACEABILITY 126
9. SUMMARY AND CONCLUSION..130

LIST OF FIGURES

Figure 2-1: Algorithm improvement triggered by user feedback 15
Figure 2-2: High-level decomposition of the future system.................................17
Figure 2-3. Schematic of IR processor, showing components carried over from ARC and those (blue boxes labelled SST CCI) developed in this project 20
Figure 2-4: Entry page to SST CCI website 22
Figure 2-5. Development of in-situ independent OE2 retrieval for AVHRR and SEVIRI with by radiance bias correction to ATSR 24
Figure 2-6: Objective procedure for comparing algorithms in round-robin exercise 25
Figure 3-1: GMPE ensemble median SST analysis [RD 4] and sea ice data, which includes operational OSTIA. A satellite-only independent version of OSTIA provides analysis for SST CCI 29
Figure 3-2: Time series of SST anomaly 30
Figure 3-3: Improvement cycles with user feedback (large cycle including red boxes) and by development team (green short cycle), and stable climate data records (blue) in the SST CCI system 30
Figure 3-4: Context of the SST CCI system with data providers and users.........32
Figure 3-5: Subsystems for production, user services and long term archiving ..36
Figure 3-6: Components of the SST CCI system for production, user services and long term archiving 37
Figure 4-1: User information, data access, and user feedback handling activities 40
Figure 4-2: Reprocessing and validation activities 42
Figure 4-3: Bulk-processing activities...44
Figure 4-4: Algorithm improvement activities 45
Figure 5-1: Distributed system in a configuration known from the SST CCI prototype 48
Figure 5-2: New deployment into a central SST CCI system 48
Figure 5-3: Implementing the SST CCI system on top of an existing ground segment? 51
Figure 5-4: Sharing user services and data stewardship with other ECVs..........54
Figure 5-5: Cluster or cloud..56
Figure 5-6: Network access versus data-local processing 59

Figure 6-1: Portal and backend services for SST CCI users............................... 64
Figure 6-2: Structure of the online archive for SST CCI users 67
Figure 6-3: OPeNDAP user interface 69
Figure 6-4: MERCI catalogue search interface 72
Figure 6-5: MERCI product details page 73
Figure 6-6: BEAM forum as example of a managed forum 75
Figure 6-7: Issue tracking system 76
Figure 6-8: Central user management for web-resources.................................. 80
Figure 6-9: Components of the production and development subsystem involved in a workflow for SST CCI L3 and L4 processing... 81
Figure 6-10: Main interfaces of the production and development subsystem 84
Figure 6-11: File system structure for SST CCI 88
Figure 6-12: Ingestion function to pull input data from a pickup point of a data provider.. 90
Figure 6-13: Service hierarchy and granularity of control at the different levels.. 91
Figure 6-14: SST processing workflow from level 1 and level 2 satellite data to level 4 analysis products 92
Figure 6-15: Quality check of inputs, intermediates and outputs......................... 99
Figure 6-16: Quality check and its interfaces 100
Figure 6-17: Wrapper script example (ARC processor call) for Grid Engine 102
Figure 6-18: wrapper script example (product cataloguing) for Calvalus 103
Figure 6-19: Operator interface for BEAM processors 104
Figure 6-20: Mapper interface with for other Java processors in Hadoop......... 104
Figure 6-21: Processors and versioned processor installation packages for modularity 105
Figure 6-22: Processor and configuration versioning with freeze and release.. 105
Figure 6-23: Local test environment with full data access................................ 107
Figure 6-24: Multi-sensor matchups with different sensor combinations of one day 108
Figure 6-25: NetCDF header of MMD file with dimensions and prefixed variables 109
Figure 6-26: Matchup database schema with observations and matchups and their relations 111
Figure 6-27: Referencing input files from observations.................................... 112
Figure 6-28: Test and validation cycles with the Multi-sensor Matchup System 112
Figure 7-1: Online data volumes needed for SST CCI Phase 2 (estimates) 122
Figure 7-2: Offline data volumes needed for SST CCI Phase 2 (estimates) 123

1. INTRODUCTION

1.1 Purpose and scope

This System Specification Document (SSD) of the ESA Climate Change Initiative (CCI) for Sea Surface Temperature (SST) proposes a design for an operational system for SST for CCI phase 2 and beyond. SST is one of 13 Essential Climate Variables (ECV) currently studied by CCI. The system covers production, user services, and long term stewardship for the SST contribution to the Climate Data Record (CDR) to be generated and continuously updated and improved by CCI.

The SSD is a response to the SST System Requirements Document (SRD) [AD 7], both the scenarios and requirements therein. It is a deliverable of the SST CCI project requested in the Statement of Work (SoW) [AD 1]. In addition to a structured description of the system, the SSD analyses several trade-offs, among them the choice of an infrastructure (SST cluster, grid, or deployment in a cloud) and the sharing with other ECVs (a system per ECV, a system per group of ECVs, one CCI system for all ECVs). Important aspects are the system life cycle and growth over time, and the cost-to-performance relation.

The system design is based on experience with a prototype developed for SST CCI phase 1. The prototype is documented in a System Prototype Description (SPD), an Input Output Data Definition (IODD), and a Detailed Processing Model (DPM). While updates of the IODD and DPM documents will be applicable to Phase 2 with their main contents, the SPD is replaced by this SSD for phase 2. The degree of re-use from the prototype is one of the trade-off questions discussed in this SSD.

1.2 References

The following documents are applicable to this document:

ID	Title	Issue	Date
[AD 1]	ESA Climate Change Initiative Phase I - Scientific User Consultation and Detailed Specification Statement of Work (SoW), including Annex G: Sea Surface Temperature ECV	1.4	09.11.2009
[AD 2]	Sea Surface Temperature ECV Proposal		16.07.2010
[AD 3]	Sea Surface Temperature CCI User Requirements Document, SST_CCI-URD-UKMO-001 (URD)	2	30.11.2010
[AD 4]	Sea Surface Temperature Data Access Requirements Document, SST_CCI-DARD-UOL-001 (DARD)	1.0	27.01.2012
[AD 5]	Sea Surface Temperature Product Specification Document, SST_CCI-PSD-UKMO-002 (PSD)	2	11.11.2011

ID	Title	Issue	Date
[AD 6]	Sea Surface Temperature MMD Content Specification, SST_CCI-REP-UOL-001	C	22.07.2011
[AD 7]	Sea Surface Temperature System Requirements Document, SST_CCI-SRD-BC-001 (SRD)	1.2	30.04.2012
[AD 8]	Sea Surface Temperature Detailed Processing Model, SST_CCI-DPM-BC-001 (DPM)	1.0	04.10.2012
[AD 9]	Sea Surface Temperature System Prototype Description, SST_CCI-SPD-BC-001 (SPD)	1.1	
[AD 10]	Sea Surface Temperature Product Validation Plan, SST_CCI-PVP-UoL-001 (PVP)	1.0	30.01.2012
[AD 11]	Sea Surface Temperature Algorithm Selection Report, SST_CCI-ASR-UOE-001 (ASR)	1.0	30.06.2012
[AD 12]	Sea Surface Temperature Algorithm Theoretical Basis Document, SST_CCI-ATBDv0-UOE-001 (ATBD)	1.0	08.11.2011

The following documents are referenced in this document:

ID	Title	Issue	Date
[RD 1]	ESA CCI Project Guidelines, EOP-DTEX-EOPS-TN-10-0002	1.0	05.11.2010
[RD 2]	Reference Documents List, SST_CCI-REP-UOE-001	1	27.09.2011
[RD 3]	Acronyms List, SST_CCI-REP-UOE-002	1	27.09.2011
[RD 4]	Web site of the Group for High Resolution Sea Surface Temperature at ghrsst-pp.metoffice.com, see also www.ghrsst.org		06.01.2011
[RD 5]	Edinburgh Compute and Data Facility web site. 1 August 2007. U of Edinburgh. 7 Jan. 2011 at www.ecdf.ed.ac.uk		07.01.2011
[RD 6]	European Committee for Space Standardisation - Software, ECSS-E40	C	06.03.2009
[RD 7]	Systematic observation requirements for satellite-based data products for climate, GCOS Satellite Supplement,	Update 2011	Dec 2011

Additional reference documents are listed in [RD 2].

1.3 Acronyms

The following SST-specific acronyms are used in this report.

Acronym	Definition
ARC	ATSR Reprocessing for Climate
(A)ATSR	(Advanced) Along-Track Scanning Radiometer
AVHRR	Advanced Very High Resolution Radiometer
BADC	British Atmospheric Data Centre
BEAM	Earth observation toolbox and development platform
CCI	Climate Change Initiative
CDR	Climate Data Record
CF	Climate Forecast
CMIP5	Coupled Model Intercomparison Project Phase 5
DARD	Data Access Requirements Document
DPM	Detailed Processing Model
ECDF	Edinburgh Compute and Data Facility
ECMWF	European Centre for Medium-Range Weather Forecasts
ECSS	European Cooperation for Space Standardisation
ECV	Essential Climate Variable
ESA	European Space Agency
GBCS	Generalised Bayesian Cloud Screening
GDS	GHRSST Data Processing Specification
GHRSST	Group for High-Resolution SST
GMPE	GHRSST Multi Product Ensemble
ICD	Interface Control Document
IR	Infrared
MetOp	Meteorological Operational (EUMETSAT)
MD	Match-up Dataset (single-sensor)
MMD	Multi-sensor Match-up Dataset
MMS	Multi-sensor Match-up System
NOAA	National Oceanic and Atmospheric Administration
NEODC	NERC Earth Observation Data Centre
NERC	Natural Environment Research Council
NWP	Numerical weather prediction
OSI-SAF	Ocean & Sea Ice Satellite Application Facility (EUMETSAT)
OSTIA	Operational Sea Surface Temperature and Sea Ice Analysis

Acronym	Definition
PMW	Passive Microwave
SDI	Saharan Dust Index
SEVIRI	Spinning Enhanced Visible and Infrared Imager
SGE	Sun Grid Engine
SST	Sea Surface Temperature
UoE	University of Edinburgh

Additional acronyms are listed in [RD 3].

1.4 Document conventions

As a design document this document is a response to the requirements document. Requirements are traced to subsections of the document in a large table in Section 8. The backward references from sections to requirements are interspersed within the text. The notation used for this is an arrow followed by the system requirement identifier and the requirement title, usually in parenthesis, e.g. (→ SST-SR-1240 Output versions). The paragraph or sentence describes the design for the corresponding requirement.

1.5 Document structure

After this formal introduction

Section 2	describes the activities in the SST CCI project and their relation to components of the SST CCI system
Section 3	provides an overview of the SST CCI system, describing its purpose and intended use, its context, its main functions and components
Section 4	describes main operational scenarios and use cases of the system
Section 5	is a trade-off analysis looking at the alternatives for infrastructure, middleware, sharing with other ECVs, and enhancement of the prototype and its constellation
Section 6	provides a functional architecture with components and interfaces, ordered by the three aspects of user's views to the system, the system operator's view for reliable production, and the algorithm developers' view for continuous improvements
Section 7	is a collection of further analyses regarding re-use of components, system life cycle, and cost and performance
Section 8	traces system requirements to sections of this document

2. THE SST CCI PROJECT AND ITS ACTIVITIES

2.1 Cardinal requirements and system overview

The Climate Change Initiative (CCI) Sea Surface Temperature (SST) project of the European Space Agency (ESA) aims at providing SST satellite data records to meet the requirements of the climate research community [AD 3].

Three of ESA's five cardinal requirements (CR) name the main outputs that are expected from the CCI project: "climate-quality" algorithms (CR-1), "world-class" time series of ECV products (CR-2), and complete specifications for an operational production system (CR-4).

In Phase 1 of the SST CCI project a prototype production system has been built, which generates world-class SST products [AD 5] (fulfils CR-2) and has been used for a range of activities (see below) for climate quality algorithm development and testing (satisfies CR-1).

For addressing CR-4, the SST CCI team has used ESA's cardinal requirements and SST technical requirements, the user community's user requirements, and the inherent knowledge of the SST CCI team to specify the system requirements for the SST CCI Phase 2 system (hereafter referred to as "the future system") in the System Requirements Document [AD 7]. The technical specification of this future system is described in this document.

The SST CCI prototype processor has been implemented, hosted and maintained at University of Edinburgh (a cluster known as ECDF), Centre de Meteorologie Spatiale (CMS, using some developments within the Ocean and Sea Ice-Satellite Applications Facility, OSI-SAF), and the MetOffice (OSTIA, the operational surface temperature and ice analysis system). The ECDF provides a high-performance cluster of servers connected to a large amount of high-performance fibre channel disk storage, so that each cluster node sees the same data in the same way.

The prototype processing system is capable of processing about 50 TB of input data into about 5 TB of output data in a highly parallel fashion. The ECDF offers a Sun Grid Engine (SGE) processing grid, which is able to distribute workload across the ECDF resource pool by an efficient scheduling of jobs, and can monitor and manage jobs, CPU cores and system memory used.

The SST CCI user products comprise L2P data (Level 2), uncollated and daily-collated Level 3 products, and daily-analysed SST from OSTIA (Level 4). At the end of Phase 1 of the project, SST products will have been produced and be available for the long-term (1991–2010) and demonstration (June & August 2007 and January to March 2012) periods. The products have been fully specified in the PSD [AD 5].

Though several elements of the prototype production system can be used in the future system, not all elements of the future system have been fully prototyped. In particular, the elements for supporting a continuous user-driven improvement of retrieval algorithms have not been implemented in a sustainable manner. The elements and activities needed for implementing the continuous algorithm improvement become clear by illustrating the mode of operation for the future system as specified by the team (Figure 2-1, trapezoidal shapes indicate manual activities).

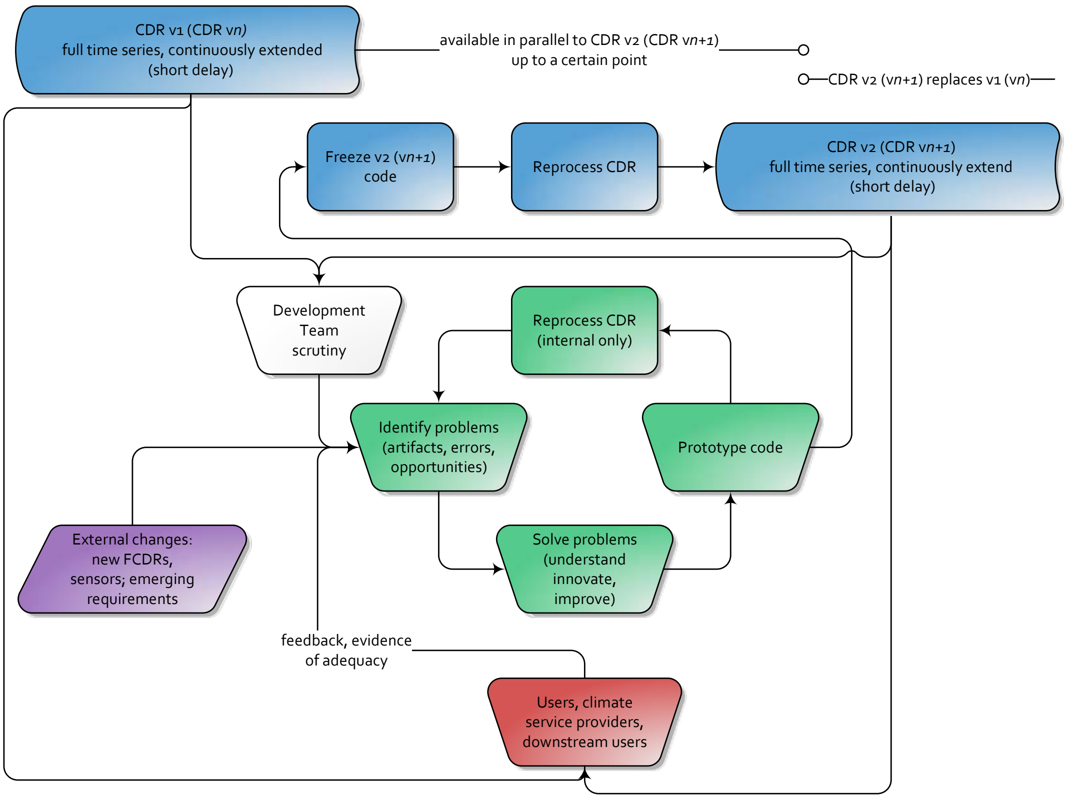


Figure 2-1: Algorithm improvement triggered by user feedback

Improvement of retrieval algorithms used in the future system will be essentially user-driven. The model of improvement involves the following elements and steps:

1. There is a stable climate data record (CDR) of a certain version. This stable CDR is continuously extended by newly acquired data in short-delay processing mode (blue cylindrical shape at the top of Figure 2-1; the initial CDR v1 is generated by the prototype system).
2. Climate users and SST community users as well as the SST CCI Development Team itself assess the CDR data and provide feedback. This feedback can trigger a need for an internal algorithm improvement cycle (green boxes and connecting arrows). External changes like new versions of input data, new sensors, and new emerging requirements can trigger the need for an improvement cycle as well.
3. The expectation is that the improvement cycle will be conducted repeatedly and rapidly in reaction to different identified problems. The cycle involves data ranging from multi-sensor match-up datasets to the full FCDR. The individual activities in the internal improvement cycle are:
    a. Using evidence from users and the development team's scrutiny to analyse and identify problems

   b. Problem solving by problem understanding and suggesting algorithm improvements and innovations
   c. Prototyping retrieval algorithm changes
   d. Extending multi-sensor match-up datasets with recomputed retrieval results and, if applicable, internal reprocessing of the CDR
4. Consolidation of retrieval algorithms after internal improvements leads to a ‘freezing’ of the source code. Reprocessing of the full FCDR with the ‘frozen’ algorithms gives the next version of the CDR (users have made clear that this shall not occur more often than once in a year). Both the older and the newer version of the CDR (v2, second blue cylindrical shape at the top of Figure 2-1) are available in parallel for a certain period of time.
5. The newer version of the CDR replaces the older version; it is extended and improved by the activities described in the previous Steps 1 to 4.

Key elements of the algorithm improvement are the implementation of the Development Team and the management of user feedback and requirements. From the technical point of view, the activities of the Development Team during the internal algorithm improvement cycles (green boxes and connecting arrows in Figure 2-1) need to be supported by a largely automated system component capable of computing match-ups of multiple *in-situ*, satellite sensor and ancillary data sources. The rules and methods for this multi-sensor match-up system (MMS) have been implemented in Phase 1 of SST CCI; implementing the automated operation within the future system is foreseen for Phase 2. Similarly, the management of user feedback has only been marginally implemented in Phase 1 through an electronic mail interface. The purpose and scope of the MMS is further explained in Section 2.3 and specified in Section 6.3.

Figure 2-2 illustrates a high-level decomposition of the future system into its essential functional elements and activities. Rectangular shapes indicate technical elements (Level 2/3 and 4 processors, MMS, distribution system for documents, data, and tools). Cylindrical shapes indicate data storage elements (satellite and ancillary input data, validation and other ancillary data, SST CCI products and MMD files) while trapezoidal shapes indicate manual activities (user feedback management, algorithm development, data ingestion for verification and validation). Note that the Development Team does not appear in the diagram explicitly, though its members conduct the manual activities and operate the system as a whole. The colouring of shapes is used to discern elements receiving external input (orange), elements distributing output to external users (purple) and core elements (blue). Boldface letters denote elements, activities or functions that have been prototyped in Phase 1; plain letters denote elements that have been prototyped for the demonstration but not for the long-term production in Phase 1; italic letters indicate functions that have not been prototyped in Phase 1, but will be addressed for implementing the future system in Phase 2.

The remainder of this overview briefly describes and discusses the system elements and functions in turn. Where these elements have supported major activities within Phase 1, these activities and the role they have fulfil are also described, with references to where in this document these elements are specified.

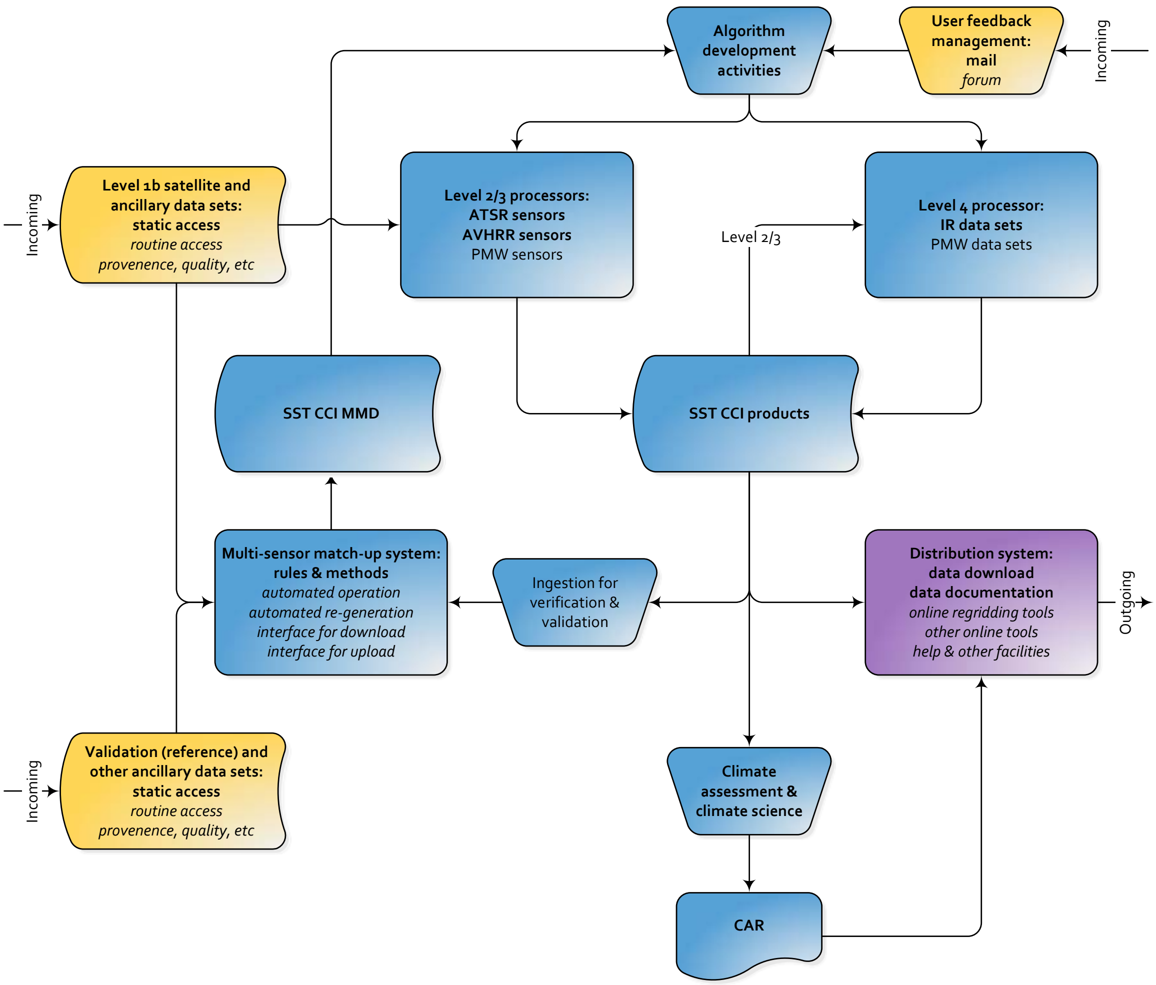


Figure 2-2: High-level decomposition of the future system

2.2 Input data access

The access requirements and procedures for all data that have been needed as input to perform the SST CCI project are listed and described in the Data Access Requirements Document [AD 4]. The data include:

- Satellite products from ESA and third party missions (e.g. Level 1b from ATSR-1, ATSR-2, and AATSR; NOAA AVHRR GAC Level 1, MetOp internal Level 1b, SEVIRI Level 3C, AMSR-E GHRSST L2P, TMI GHRSST L2P)
- Ancillary data (e.g. ECMWF)
- In situ observation data sources as well as higher-level products needed for product inter-comparison
- Historical archives and currently operational sources

All data have been available for the SST CCI project via FTP, SFTP or HTTP to obtain from source and have been downloaded in manual activity.

For the prototype, the data are accessible statically. For the future system routine (i.e. automated) access including provenance and quality checking will be required for all needed input data, which, in addition to those listed above, will include satellite data from IASI and SLSTR. Input data access for the future system is specified in Section 6.2.

2.3 Multi sensor matchup system (MMS)

The MMS is a component that has been started as a novel development for the prototype. It is capable of computing matchups between in-situ and satellite data from different sensors, and of generating multi-sensor matchup datasets (MMD) of matchups that can include satellite sub-scenes, in-situ records, NWP ancillary data, and processed SST.

One challenge for the MMS has been the heterogeneous input with respect to data content, format, and temporal and spatial coverage. In the prototype MMS the matchups have been based on pre-matched single sensor matchups. The data types and sensors that have been included in Phase 1 are:

- Single-sensor matchups from ATSR1, ATSR2, AATSR, MetOp AVHRR, SEVIRI, and NOAA AVHRR
- Level 1 satellite images from ATSR1, ATSR2, AATSR
- Level 1 satellite images from MetOp AVHRR
- Level 1 satellite images from AVHRR from NOAA-10 to NOAA-19
- Level 2 satellite images from AMSR-E
- Level 2 satellite images from TMI
- Level 3 Aerosol and sea ice concentration
- *In-situ* observation trajectories
- ECMWF era-interim ancillary data
- Results of SST retrieval from ARC CCI processing in MMD format

In an early period of Phase 1 of the SST CCI project the MMS was used to generate a complete set of MMDs for the years 1991–2010 containing more than 6,000,000 matchups. These MMDs have been the basis for doing further work. The initial complete run of the MMS has revealed several types of errors in input data and processing, which have required a semi-automated approach to analyse the cause of errors and to handle errors. The MMS has been verified by comparing its results to the outputs of an independent implementation of the corresponding matchup and extraction algorithms. In addition, all types of outputs have been manually inspected and compared with corresponding input data values on a sample basis.

The MMS database contains all information necessary for doing queries and extracts on spatial and temporal criteria, and criteria of satellite combinations contained in matchups. The prototype MMS already provides the infrastructure to compute matchups on the basis of *in-situ* data and satellite data without pre-computed single sensor matchups, and contains functions to extend the matchups into the future with newly ingested inputs. These functions, however, have not been exploited in Phase 1.

In Phase 1 the MMS has supported the following activities:

- Algorithm development by the SST_cci team for SST retrieval
- Round robin exercise for SST retrieval algorithm decision
- Algorithm development for classification in high latitudes

- Algorithm development for ocean thermal skin model
- Algorithm development for near-surface ocean turbulence model
- Algorithm development for uncertainty estimation
- Product verification
- Product validation (including stability assessment and uncertainty validation)
- Climate assessment

The MMS is therefore a key component in the problem identification and problem solving components of Figure 2-1, as well as in verification and validation of products.

For the future system the MMS needs to be further developed in order to support the algorithm improvement activities illustrated in Figure 2-1 in an automated and sustainable manner. In addition, the capability of ingesting and extracting data from IASI (from a EUMETSAT archive), SEVIRI (from a CMS archive) and SLSTR is required. IASI data are needed for assessing and monitoring stability of MetOp AVHRR that will bridge the gap between AATSR and SLSTR. The future system's MMS is specified in Section 6.3.

2.4 Level 2/3 processor

The SST CCI system prototype has been based on software from an earlier project, ARC [RD 296, Merchant et al., 2012]. Functionally, it performs pre-processing (including co-location adjustments), Bayesian cloud detection (which involves associating NWP fields with satellite data and fast RT modelling), coefficient-based and optimal estimation retrieval (first version, denoted OE1) of SST, and creation of 0.05° resolution daily outputs. Another project under UK national funding created additional processing modules for OE1 SST retrieval using AVHRR-GAC inputs. This work was completed prior to the start of the CCI project.

The SST CCI processor development has involved several modifications of the ARC basis including modification of auxiliary files (e.g., updated coefficient files) and development of new software modules. In particular, new modules have been developed for using prior SST gradients for cloud detection, for optimal estimation retrieval (second version, denoted OE2) for ATSRs and AVHRR-GAC, and for conversion of outputs to L2P. In addition, the cost parameter from the optimal estimation retrievals has been used to improve cloud detection. The ARC-based prototype Level 2/3 processor is described in further detail in the System Prototype Description [AD 9].

The creation of MetOp AVHRR SSTs for SST CCI has built on the MetOp operational production at OSI-SAF. Internal to the OSI-SAF processor, work files at 0.05° spatial resolutions are created. These work files are an intermediate product representing considerable additional value compared to the MetOp satellite observations, since cloud detection and confidence level processing have been conducted, and, in addition, useful auxiliary information has been associated with the satellite data. The work files have been made available to the CCI project on an as-is basis. They are property of the EUMETSAT OSI-SAF, and their use within the CCI project is formally acknowledged.

Within the SST CCI project, these work files have been augmented with outputs of fast radiative transfer simulation (using RTTOV9), in order to support OE1 and OE2 retrieval of SST. These new work files contain all fields necessary to support OE1 and OE2 retrieval of SST. The system demonstrated under the CCI therefore consists of a new processor to generate OE2 SSTs for MetOp, which has been demonstrated on the ECDF cluster.

SSTs from SEVIRI are provided by the OSI-SAF operational and experimental systems. SEVIRI OSI-SAF products are copyright of EUMETSAT and their use within the CCI project is formally acknowledged. The OSI-SAF prototype system for SST CCI is described in further detail in Section SPD [AD 9].

A summary of the structure of the infrared (IR) processor, showing the origin of different elements, is illustrated in Figure 2-3. Note that the radiative transfer modelling (RTM) call is not relevant to the MetOp and SEVIRI choice of cloud detection (Bayesian or OSI-SAF). For AVHRR-GAC, CLAVR-X cloud detection has been embedded into the ARC system under the NCEO project.

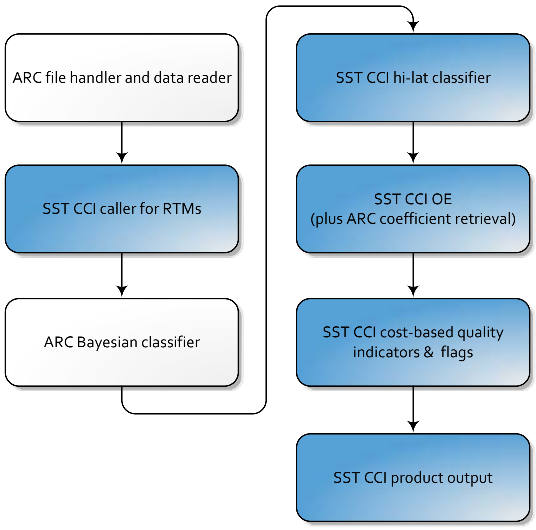


Figure 2-3. Schematic of IR processor, showing components carried over from ARC and those (blue boxes labelled SST CCI) developed in this project

Passive microwave SSTs are as obtained from Remote Sensing Systems (RSS). Useful and practicable improvements in PMW SST uncertainty characterisation are expected, and so an MW post-processor module has been developed to update PMW SSTs with improved uncertainty estimates and output in SST CCI L2P format.

For the future system, several infrastructure alternatives are contrasted in this document: the possible use of existing earth observation processing data centres or migrating into the cloud are discussed in Sections 5.2 and 5.4 of this document, processing middleware alternatives in Section 5.5. The future Level 2/3 processing and production is specified in Section 6.2.

2.5 Level 4 processor

The L2P SSTs produced by the Level 2/3 processing chains have been used for Level 4 SST analysis in a specific version of the Met Office's OSTIA system. The Level 4 analysis is based on the operational OSTIA configuration, but improved via exploitation of both the improved SSTs and uncertainty estimates contained new products. The long-term SST CCI Level 4 product is a daily mean SST depth analysis, whereas the operational OSTIA is foundation SST analysis.

Since the Level 2/3 SST retrievals for ATSR and AVHRR have aimed to be independent of in situ observations the possibility of OSTIA providing a truly independent satellite-only analysis has been raised. The feasibility and benefit of this possibility has been assessed and realised. The prototype Level 4 analysis system is described in further detail in the SPD [AD 9].

For the future system, the pros and cons for enhancing the distributed prototype into a distributed future system, or deploying the future system into a central infrastructure are discussed in Section 5.1. The future Level 4 processing and production is specified in Section 6.2.

2.6 Distribution system and user feedback management

There has been no dissemination system within the prototype SST CCI system – users have access to the Climate Data Research Package from the UK Centre for Environmental Data Archive, using Met Office work space. An SST CCI project website has been created that is available at http://www.esa-sst-cci.org (see Figure 2-4).

The website provides a project overview, information about the project team, the project plan, information about the round-robin procedure, access to public project documents, contact points, and a collection of frequently asked questions. An elaborate SST retrieval system is scoped for Phase 2 of the project, as outlined in the systems requirement document [AD 9]

For the lifetime of the prototype, public access to the SST CCI products is provided via HTTP and FTP to the ECDF NAS. Afterwards the data will be sent to the GHRSST Long Term Stewardship and Reanalysis Facility (LTSRF) for archiving.

The Climate Data research package (CDRP) is linked from the SST CCI website, but physically served from a remote FTP service hosted by CEDA. A simple registration on the SST CCI website will provide the access information to the user by email, and the user can download the data packages.

Tools for re-gridding and regional averaging have been developed in the project to generate subset products and composites with uncertainty propagation. These tools will be available for download when the development and verification is completed. Integration of these tools into a web service is required for the future system to be developed in Phase 2 of the project [AD 9].

Figure 2-4: Entry page to SST CCI website

Access to MMD datasets for the round-robin exercise has been provided by FTP. A registration by email provides the access information to download the MMD files and to upload MMD result files for participation in the round-robin comparison.

There have been no automated or tracked user feedback mechanisms within the prototype. However, all SST CCI product files contain metadata referring users to the project website at http://www.esa-sst-cci.org and a contact email science.leader@esa-sst-cci.org.

An elaborated system for managing user feedback is scoped for Phase 2 of the project, as outlined in the systems requirement document [AD 7] and specified in Section 6.1 of this document.

2.7 Activities in Phase 1 of SST CCI

2.7.1 Algorithm development for SST retrieval

Algorithm development activities have focused on improving heritage development that were brought to (ARC software) or made available (OSI-SAF processor work files, Met Office's OSTIA system) for the project. The main algorithmic improvement has been the further development of optimal estimation SST retrieval in order to achieve independence of *in-situ* observations.

Earlier optimal estimation (OE) retrieval was by design tied to the calibration of *in-situ* observations, mostly from drifting buoy observations because of their availability throughout the 1991 to 2010 period. Being an optimal estimation method, it requires use of a fast radiative transfer model (RTM), and only gives low-bias (GCOS compliant) SSTs, if the fast RTM has low bias when simulating brightness temperatures (BTs) relative to the observed sensor BTs. The bias correction is implemented on BTs (radiances), not on SST. This is preferable, since then the optimal estimator minimises the risk of introducing patterns of bias in SST, which may happen if bias correction is attempted directly on SSTs. An overview of the design for the new OE retrieval is illustrated in Figure 2-5 (blue boxes indicate new SST CCI developments).

Independence has been achieved by fully exploiting the ATSR series (the only series whose accuracy is adequate for the purpose) as a reference sensor. The bias correction is undertaken on BTs (radiances) for all IR sensors, using ATSR SSTs based on fully updated radiative transfer modelling. This approach has been intended to ensure that the retrieved SSTs ultimately exhibit low bias relative to ATSR SSTs, and independence.

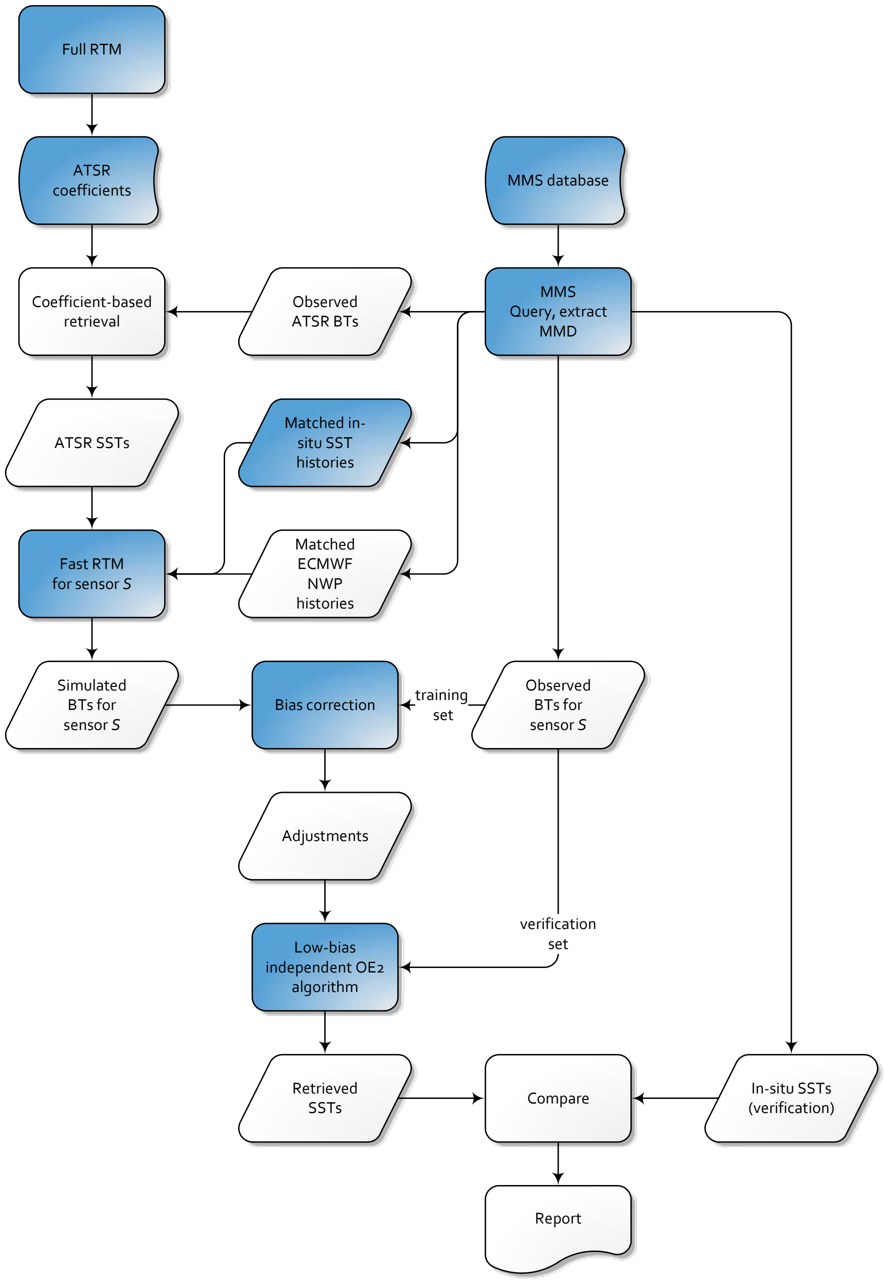


Figure 2-5. Development of in-situ independent OE2 retrieval for AVHRR and SEVIRI with by radiance bias correction to ATSR

2.7.2 Round-robin algorithm comparison and selection

In order to identify the best performing retrieval algorithm or combination of algorithms, the SST CCI project ran an open algorithm selection exercise. This consisted of algorithm intercomparison (described in ESA documents as the "Round Robin") followed by selection of algorithms following criteria defined in the Product Validation Plan [AD 10].

The exercise was open over a four month period ending 31st January 2012 and involved both the project team and external participants. Ten external teams expressed interest in participation, of which two were able in practice to submit algorithm selection results in time for consideration. The submitted external algorithms were cutting edge algorithms of significant interest. Relevant to the ATSR series was the Oxford-RAL Aerosol and Cloud retrieval (ORAC, submitted by RAL), an advanced optimal estimator recently extended to include SST, although only applicable to daytime scenes. Relevant to the AVHRR series was Incremental Regression (IR, submitted by NOAA), which is a powerful fusion of model-based and empirical regression approaches. The internal algorithms included existing coefficient based retrievals for the ATSRs, and a day-and-night (infra-red only) optimal estimator tuned (for both ATSRs and AVHRRs) to ATSR SSTs. ORAC as currently formulated is not sufficiently general because it doesn't apply to night-time scenes and gave out-of-target biases. Optimal estimation was selected as the best available, most consistent and independent algorithm for use by SST CCI for ATSR and AVHRR sensors. The whole selection procedure is described in the Algorithm Selection Report [AD 11].

To participate in the SST CCI round-robin algorithm selection exercise, participants obtained information about accessing the multi-sensor (ATSR-2, AATSR, MetOp AVHRR, NOAA-17 AVHRR, NOAA-18 AVHRR and NOAA-19 AVHRR) matchup data set provided as the common data set for the exercise, the document explaining the data contents, and the round-robin protocol, which sets out the procedures for involvement.

An important aspect of the exercise has been the approach taken to ensure an objective comparison of algorithms. Most importantly, subsets of the provided information have been earmarked for particular uses: as training data (for algorithm development, including and empirical tuning), testing data (for internal testing of results by participants) and selection data (reserved for calculation of selection metrics, not used in algorithm development). This concept is illustrated schematically in Figure 2-6.

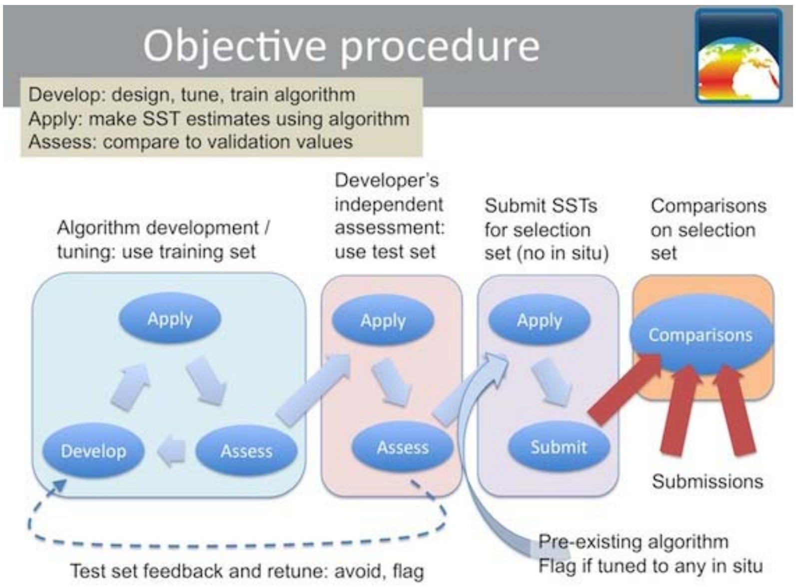


Figure 2-6: Objective procedure for comparing algorithms in round-robin exercise

The selection criteria for the SST CCI round-robin algorithm selection exercise were pre-defined before the start of the activity. All assessments were carried out with reference to drifting buoys. Further details on each criterion can be found in Section 4 of the Product Validation Plan [AD 10].

To support the development and validation activities, the SST CCI project has created a multi-sensor match-up dataset (MMD) of temporal and spatial coincidences between multiple satellite datasets of both brightness temperatures and SST retrievals and time series of SST from in situ sensors (such as a drifting buoy). The round-robin data package (RRDP) essentially has been an MMD, which includes multi-sensor matchup data records for training, test, and algorithm selection. The MMD Content Specification [AD 6] provides a detailed description of contents and format.

The round-robin exercise has been a major driver for the requirements on the system for producing MMD files, the MMS, which are described in the System Requirements Document System Requirements Document [AD 9].

2.7.3 Other algorithm developments

Besides the algorithm developments for SST retrieval summarised in Section 2.7.1, work has been carried for developing methods for SST uncertainty estimation, SST depth adjustments, SST diurnal adjustments, and further developing methods for detecting clouds and classifying sea ice at high latitudes. The development results are described in the Algorithm Theoretical Basis Document [AD 12].

Again, similar to the procedures illustrated in Figure 2-5, using MMD files created by the MMS has facilitated the algorithm development and validation tasks.

2.7.4 System verification

The verification activities in SST CCI Phase 1 will be fully described in the System Verification Report (SVR). The activities address four areas of functionality: prototype processors (create SST CCI prototype products), multi-sensor match-up system (MMS, supports CCI science behind products), tools applicable to SST CCI products (for aggregating SST data), and data provision (user access to the Climate Data Research Package).

Verifying the MMS by spot-checking and manually inspecting MMS database records and MMD files created for the round-robin exercise (see Section 2.7.2) has been an early activity within the project.

Verifying the prototype processors with focus on verification of the products they produce are the key activities within the still ongoing verification tasks. Besides checking product files for completeness of content and consistency with the product specification, all files are checked for content ensuring that all variables have values within the specified limits or a fill value. In particular, retrieved SST values and uncertainties are further spot-checked by verifying SSTs extracted from the outputs to be equal to (within a certain tolerance within required accuracy limits) to SSTs calculated independently from multi-sensor matchup dataset (MMD) files.

2.7.5 Product validation and climate assessment

The validation and climate assessment of SST CCI products is a work still to come at the writing of this document. In brief, the SSTCCI Level 2, Level 3 and Level 4 products will be independently validated using high-quality SST measurements made *in-situ* from a number of sources. In addition, the SST CCI Level 4 products will be compared to other Level 4 products as part of the Group for High Resolution SST (GHRSST) Multi Product Ensemble (GMPE) and other inter-comparisons carried out as part of the Climate Assessment Report (CAR). The CAR will also include other kinds of assessment, as detailed in the Product Validation Plan [AD 10].

2.7.6 Summary of project activities

A summary of the activities of the SST CCI project carried out in Phase 1 and how these activities are related to the prototype and the future systems is given in Table 2-1.

Table 2-1: Summary of project activities

Activity in Phase 1	Prototype system	Future system
Defining user requirements	None	User requirements have been a source for the future system's System Requirements Document. New user requirements can trigger algorithm improvement activities as illustrated in Figure 2-1
Data access	Static datasets (historical data) have been obtained	In addition need routine (short-delay) access to ongoing missions. Extend datasets used to IASI and SLSTR. Update with new Level 1b where relevant.
Input data provenance, quality control, etc.	Has been carried out ad hoc as required, e.g., by noting and reporting corrupted data when found in processing	Automated checking of data; provenance control, e.g., track version of data used in different outputs
Development and creation of multi-sensor match-up rules and datasets	Has been applied in a single MMS run, using in part pre-calculated match-up datasets for individual sensors	Same rules and methods, expanded to new datasets. Automated operation and generation of matches (in-situ, high-latitude, and clear-sky satellite-satellite)

Algorithm development activities	MMS has supported use of controlled subsets of data for algorithm development activities. Retrieval development, classification, skin-depth adjustment, and uncertainty algorithms	Interface needs to be defined and implemented to configure subsets of MMS output for automated extraction and delivery, tailored to particular investigations
Upgrading and generalising prototype processor	Improved and newly developed algorithms have been implemented. ECDF prototype has been extended in order to be able to process MMD input	Need for capability of processing MetOp FRAC and SLSTR. Assess migration of the ECDF part (including MMS) to CEMS
Round-robin exercise	MMS supported controlled, blind Round Robin protocol, with receipt of external contributions	Interfaces for external RR participants to download and upload MMD and automated ingestion of contributions need to be defined and implemented
Product generation	Has been carried out once, for the long-term (1991—2010) and demonstration (June and August 2007, and January to March 2012) periods	Algorithm improvement will be triggered by user feedback and external changes, followed be reprocessing of the full CDR as decided by the Development Team
Product verification	Verification procedures have not been integrated into the prototype	Verification will be integrated into the future system: verification tools, automated ingestion from products into MMS
Product validation	To be carried out using MMS outputs in line with the PVP	Requires more standard tools for routine validation on generation of new products
Climate assessment	To be carried out using products and MMS outputs	Requires automatic generation of standard assessment metrics on generation of new products

3. OVERVIEW OF THE SST CCI SYSTEM

Note that in this and the subsequent sections backward references from sections and paragraphs to requirements are interspersed within the text. The notation used for this is an arrow followed by the system requirement identifier and the requirement title, usually in parenthesis, e.g. (→ SST-SR-1240 Output versions). The paragraph or sentence where such an arrow appears describes the design for the corresponding system requirement. All requirements are listed in Section 8.

3.1 Purpose and intended use

The SST CCI system aims at providing a long-term global SST climate data record derived from the best satellite and auxiliary data and retrieval algorithms available. Figure 3-1 and Figure 3-2 show one day SST analysis and a time series of 30 years SST anomaly, respectively, as an example of what is generated by SST CCI.

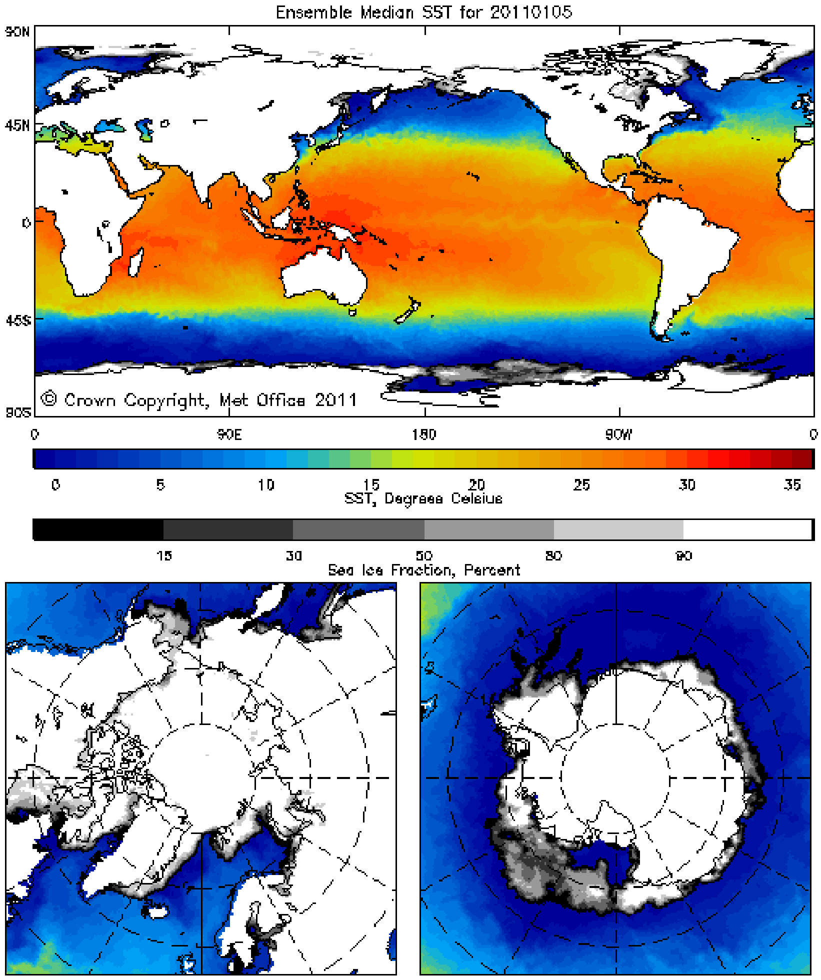


Figure 3-1: GMPE ensemble median SST analysis [RD 4] and sea ice data, which includes operational OSTIA. A satellite-only independent version of OSTIA provides analysis for SST CCI

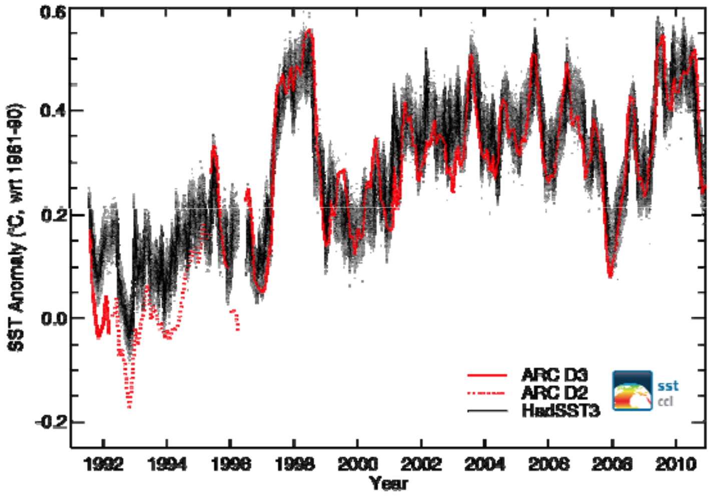


Figure 3-2: Time series of SST anomaly

As already mentioned in Section 2.1, the SST CCI system aims at being a platform for cycles of improvement of the algorithms and methods of SST retrieval. It supports a model of how it is going to be used as illustrated in Figure 3-3 and explained in the paragraphs below.

At least one consistent climate data record (CDR) version is available to users and is extended by short delay processing at any time (→ SST-SR-1240 Output versions).

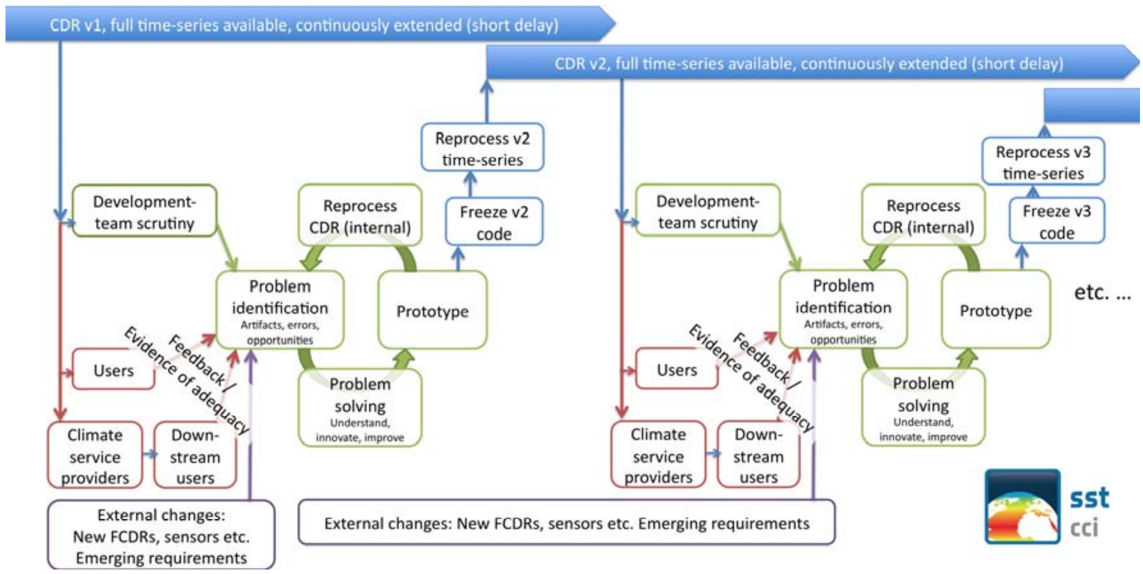


Figure 3-3: Improvement cycles with user feedback (large cycle including red boxes) and by development team (green short cycle), and stable climate data records (blue) in the SST CCI system

Different groups of users provide feedback. They and the development team contribute to problem identification and options for improvements. Problems are selected for solving. Also new versions of inputs, the availability of data from new sensors, or emerging new requirements initiate the short cycle (→ SST-SR-5230 Agile requirements selection).

In the short cycle the development team finds new solutions to identified problems by analysis, prototyping, internal reprocessing and validation. This may already require reprocessing of the full time series (→ SST-SR-5270 Short development cycle).

When the development team decides to release a new version the code is frozen and the time series is reprocessed if necessary. The new version is made available for validation to users and external validators as release candidate (→ SST-SR-5250 Version decisions, SST-SR-5260 Version release process).

For a certain period of time the release candidate exists concurrently to the previous version to allow for feedback and optional improvement, and for users to switch over (→ SST-SR-5255 Overlapping versions, SST-SR-5265 Version management).System context and external interfaces

The SST CCI system has interfaces to its data providers and its users. Figure 3-4 shows the SST CCI system in its context, which is explained in the paragraphs below.

Climate researchers and users from the SST community receive products (and information about them). The SST CCI development team (see below) actively requests feedback from climate users and the SST community. The system shall support the development team in gathering this evidence. The interface to satellite data providers usually is the Level 1 product. Ancillary and validation datasets are provided from different external sources and projects.

Other ECVs provide comparison data to SST, in particular sea level, sea ice, ocean colour, aerosol products in order to improve consistency among CCI products. Other ECVs receive SST products for consistency checks.

An SST CCI development team shall be established. It consists of experts in Sea Surface Temperature that deliver scientific leadership within the project. It supports, directs the sequence of and prioritizes operations within their agreed scope, and helps develop Sea Surface Temperature to meet climate users' needs. The development team has a mandate defined by terms of reference to support evidence-driven improvements in the development cycle. It is considered part of the SST CCI system with functions assigned to it and therefore does not show up in this diagram (→ SST-SR-6340 Science issues, SST-SR-5110 Community process, SST-SR-5220 development team, SST-SR-5230 Agile requirements selection, SST-SR-5240 Development and evaluation).

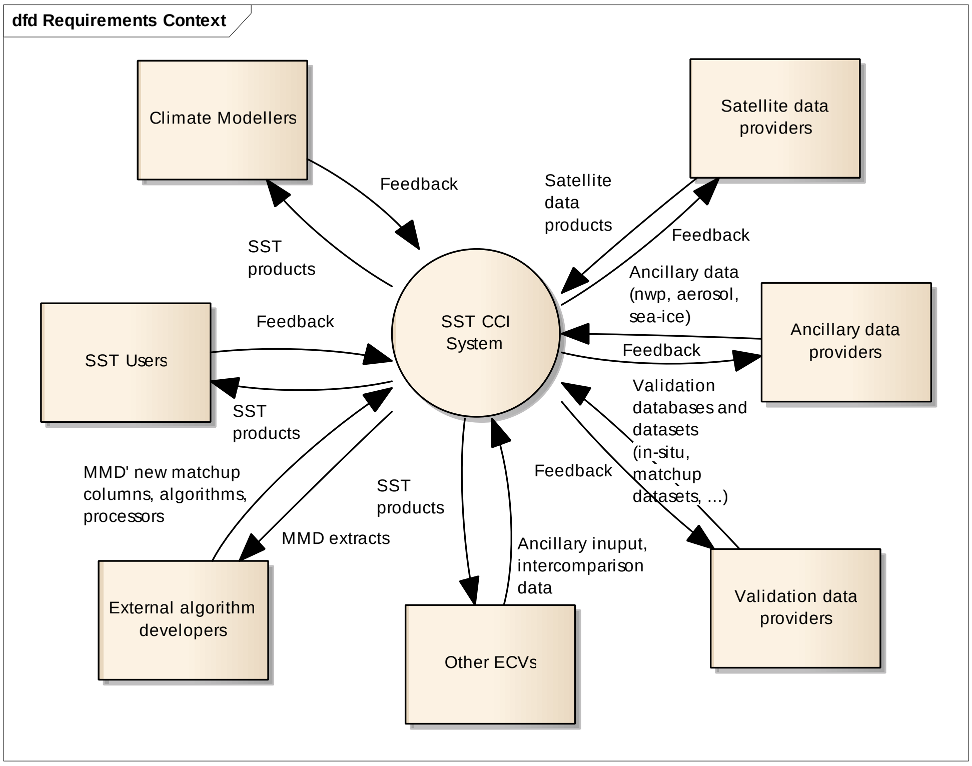


Figure 3-4: Context of the SST CCI system with data providers and users

Validation is a two step activity: Firstly, validation is part of the process. It is performed as a QA step by the development team before a new CDR is provided to users. It is also performed to identify anomalies. Secondly, users independently validate the CDR and provide feedback.

Validation and development may also be undertaken by external algorithm developers, and this is foreseen to take place as an interaction using multi-sensor matchup data (MMD) managed by a multi-sensor matchup system (MMS).

Table 3-1 lists the external interfaces of the SST CCI system. There are six main interfaces with the satellite input data interfaces on one end and the SST output data on the other end. Some of the interfaces have different endpoints. Therefore, they are split into corresponding sub-interfaces. Interfaces for historical data are listed because of possible reprocessings. In order to avoid too much duplication with the DARD [AD 4] the list of sources for the validation data are not repeated here. DARD section 6 and 7 provide a comprehensive list including access information. The interfaces to data providers may change during the project due to new data sources available or different needs by the algorithms.

Table 3-1: External interfaces of the SST CCI system

Ifc ID	Interface Name	Source	Location/Protocol	Interface item description
Ifc-1	**Satellite input data interface**	various	FTP pull	DARD 4
Ifc-1.1	(A)ATSR input data interface	ESA	FTP from NEODC	DARD 4.1
Ifc-1.2	AVHRR GAC L1 input data interface	NOAA	NOAA CLASS / University of Maryland	DARD 4.3
Ifc-1.3	MetOp A and B AVHRR L1B input data interface MetOp A and B IASI input data interface	Eumetsat	ftp pull, optionally EUMETCAST	DARD 4.4
Ifc-1.4	Sentinel 3 SLSTR	ESA	Media circulation from Collaborative Ground Segment (CGR) or provision in a shared processing environment	
Ifc-1.5	VIIRS	NASA	FTP from NASA	
Ifc-2	**Ancillary input data interface**	Various	FTP pull	DARD 5
Ifc-2.1	ERA-Interim	ECMWF	FTP from BADC	DARD 5.1
Ifc-I2.2	CLAVR-X	NOAA	FTP from University of Wisconsin	DARD 5.2
Ifc-2.3	NCEP/NCAR Reanalysis	NOAA NCEP	FTP from NOAA	DARD 5.3
Ifc-2.4	OSI-SAF Maximum Gradient Atlas interface	CMS	CMS	DARD 5.4
Ifc-2.5	AOML Ocean Current Climatology Atlas interface	AOML	AOML	DARD 5.5
Ifc-2.6	SSM/I Sea ice concentration maps OSI-409 Sea ice concentration reprocessing	OSI SAF	FTP from OSI SAF High Latitude processing centre	DARD 5.6, 5.7
Ifc-2.7	Absorbing aerosol index	NASA GSFC, TEMIS	NASA GSFC TEMIS	DARD 5.8

Ifc ID	Interface Name	Source	Location/Protocol	Interface item description
Ifc-2.8	SAGE II Aerosol	NASA	NASA	DARD 5.9
Ifc-3	**Validation input data interface**	Various	FTP pull	DARD 4
Ifc-3.1	In-situ data interface	various	FTP pull	DARD 6
Ifc-3.2	Inter-comparison data interface	various	FTP pull	DARD 7
Ifc-4	**ECV consistency check data interface** Data from OC, SL, sea ice thickness, sea ice concentration, aerosol optical depth, clouds	ESA CCI	FTP pull	DARD 8 NetCDF-4 files of complete data records
Ifc-5	**MMS data exchange interface**	SST CCI	Email, FTP	MMD content specification [AD 6]
Ifc-5.1	MMD retrieval interface for algorithm developers	SST CCI	Web form/email for access information, FTP pull from SST CCI	MMD content specification [AD 6]
Ifc-5.2	MMD transfer interface for algorithm developers	External algorithm developers	FTP push to SST CCI	MMD content specification [AD 6]
Ifc-6	**Climate user and SST user interface**	SST CCI	HTTP, FTP, email	see below
Ifc-6.1	SST CCI Web interface	SST CCI	HTTP	Section 6.1 below
Ifc-6.2	SST CCI feedback interface	SST CCI	Email, Web forms (issue tracking)	Section 6.1 below
Ifc-6.3	SST CCI data retrieval interface	SST CCI	Web form/email for access information, FTP pull from SST CCI, optionally provision of SST CCI outputs in a shared processing environment	PSD [AD 5] SST CCI Level 2, Level 3, Level 4 products

3.2 Main functions

To fulfil its purpose in such a context the SST CCI system provides three high level functions:

- production in broad sense, i.e. the generation of the CDR contribution
- dissemination in broad sense i.e. serving the user with data, documentation, services
- improvement, i.e. the chain from feedback handling via updates to versions of the CDR

The Statement of Work [AD 1] asked for the "fundamental operations" of the system. For production the focus is on offline reprocessing of complete missions, repeated as necessary to implement requirements/requests for improvements from users. Functions for this are:

- storage to make available and hold inputs, intermediates, outputs, auxiliary data
- data processors for the processing steps to transform inputs into outputs
- control of processing workflows, and of massive parallel bulk production in particular
- quality checks, automated and visual
- validation, matchups, comparison with reference data
- documentation, metadata to distinguish versions
- ingestion of new data and corresponding auxiliary data for short delay processing

The way bulk production is managed is of highest importance here, as it is crucial for the agility of the cycles and thus the quality of the outputs.

For dissemination the focus is on the service for climate users. Functions for this are:

- project information, data discovery, catalogue service
- data access, online and optionally on media, bulk access to complete data records
- data customisation, tools available as service and optionally as installable software
- validation support, access to reference data, reports
- documentation, access to documents on products and algorithms, example products
- feedback handling, issue tracking, forum and email communication
- long-term preservation of the data and its representation information

Whether the climate users are served rapidly and with the data and formats they need is a key aspect. Interacting with and serving the SST community is essential, too, to get support for the program and feedback on data and methods.

For algorithm improvement and evolution, the focus is on a suitable environment with a small effort to exchange processor versions and configurations. Besides the feedback function of user services the main functions are:

- test of new prototype processor versions. For this the SST system provides a processor interface where new processor versions are plugged in for easy exchange.

- validation of results, multi-sensor matchup system
- access to full mission data, reprocessing capability for large sets of inputs when required
- version management for processors, configurations, and data, with documentation of what has been tested, updated, released

The overall agility of the SST CCI process depends on such a accessible, easy to use and powerful infrastructure and its day to day use by the development team.

3.3 High level decomposition and main internal interfaces

The functions listed in the previous subsection are implemented by functional components. These functional components can be grouped in subsystems. This subsection provides an overview of the components of the SST CCI system as starting point for the operational scenarios in section 4 and design trade-offs in section 5. The detailed specification with all components, functions and interfaces follows in section 6, user services in section 6.1 and production and development in sections 6.2 and 6.3.

Starting with a very coarse structure, Figure 3-5 distinguishes three subsystems: production, user services, and data stewardship. Besides the different functions and interfaces these subsystems also encapsulate the earth observation data in a different way.

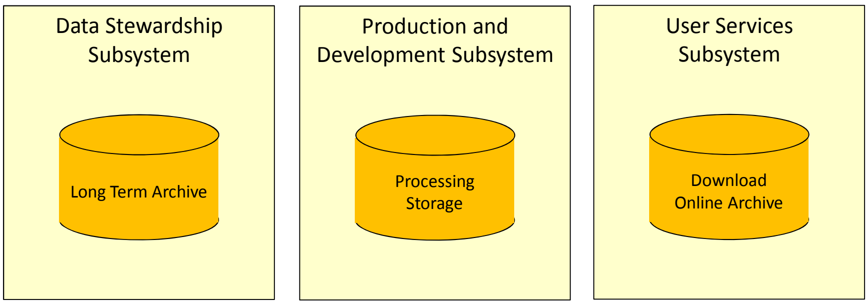


Figure 3-5: Subsystems for production, user services and long term archiving

- Processing storage of the production subsystem in the middle is accessed by processors and is optimised for high data throughput. This shall be protected against direct and concurrent access by users.
- Direct and concurrent access is provided by a second archive accessible online for data download. It is part of the user services subsystem on the right.
- While the online archive is optimised for external access, the long term archive is optimised for reliable long term storage of the data, and all representation information required using it. Typical access to the long term archive is in large chunks of e.g. a complete product set of a certain version. The long term archive is part of a subsystem for data stewardship on the left.

Different uses imply different features and optimisations. This does not exclude that e.g. two of the subsystems are integrated into one or hosted by one organisation. But the interface between them also allows separating them. It is mainly a data interface allowing the exchange of earth observation datasets.

Stepping down to the component level of detail, Figure 3-6 shows components of all three subsystems.

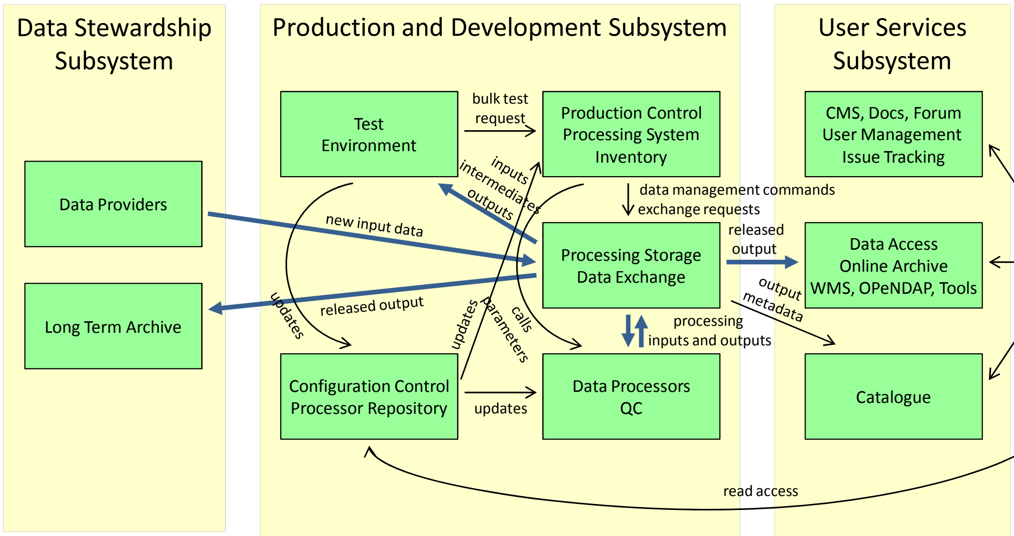


Figure 3-6: Components of the SST CCI system for production, user services and long term archiving

- The components of the two columns in the middle constitute the production and development subsystem. Production control, processing storage and the processors provide the basic infrastructure for processing. A test environment with read access to all data and the option to use the production infrastructure for bulk tests serves the development needs.

  A middleware shall implement main functions of the processing system and of processing storage to support bulk production. A middleware is a software service layer above the operating system that makes it easier for developers to develop applications. Examples are generic batch queuing and scheduling services and cluster management software. Depending on the middleware selected, the two functions of processing control and data storage can be tightly coupled.
- The user services subsystem consists of at least three components: web presentation, data access by users, and a catalogue. The web presentation includes user forum and issue tracking. Data access offers different protocols and supports online regridding and aggregation. The catalogue is used for product search and metadata access. In addition, users can get read access to the processor repository for documentation purposes (→ SST-SR-1295 Processor documentation) and optionally for external use and validation.
- The input data of SST CCI and the outputs of it are hosted in a data stewardship subsystem. It provides long term data preservation and bulk data

provision on request. Depending on whether this function is provided externally, components for the preservation of inputs and outputs may have to be foreseen within the system. (→ SST-SR-1200 Output preservation, SST-SR-1241 Long-term stewardship, SST-SR-1500 Backup archive, SST-SR-1501 Bulk archive retrieval)

The main internal interfaces of the prototype are those for data exchange between production and user services and between production and a long-term stewardship system (if they are different systems). Table 3-2 lists them with references to the description of the exchanged data items. Interfaces between the components of a subsystem are described in the respective sections 6.1.1, 6.2.1 and 6.3.1 of the functional design.

Table 3-2: Internal interfaces of the SST CCI system

Ifc ID	Interface Name	Source	Location/Protocol	Interface item description
Ifc-7	**Data stewardship subsystem data interface** SST CCI Level 2, Level 3 and Level 4 products	SST CCI production subsystem	FTP push to data stewardship subsystem (GHRSST)	PSD [AD 5]
Ifc-8	**Online archive feeding data interface** SST CCI Level 2, Level 3 and Level 4 products	SST CCI production subsystem	rsync/sftp pull by SST CCI user services	PSD [AD 5]

4. SST CCI OPERATIONAL SCENARIOS

The following sections describe the main SST CCI operational scenarios, in particular: user data access and information, processing and validation of the full or partial CDR, and algorithm improvement by means of the MMS. The specification of the elements implementing these functions is elaborated in detail in Section 6.

4.1 User roles

Two major groups of users who interact with the SST CCI system can be distinguished: the (internal) development team and the (external) users, including climate users and the SST community. They are actors in the operational scenarios. The users are grouped in roles depending on how they use the system.

- climate users
    - interested in consistent stable datasets, with occasional version upgrades
    - data format compatible with their models and other ECVs (NetCDF-4, projection)
    - provide feedback originating from the use of SST CCI data in climate modelling
- SST community users
    - international community
    - interested in best sea surface temperature products
    - skilled in SST retrieval, provide feedback from external validation activities
    - provide proposals for alternative methods and improvements
    - is invited to perform (external) algorithm development, comparison with multi-sensor datasets (MMD)
- development team with scientists, operators, and system integrators
    - has mandate to push forward SST CCI, to decide about requirements to analyse and implement, and to decide which algorithms to test, all in agreement with oversight arrangements representing ESA
    - maintains and improves the system, manages the data archives, and initiates and monitors production
    - is in dialog with users to collect feedback and requirements
    - cyclically improves the system and issues CDR versions
- ESA
    - supervises the project and/or makes oversight arrangements, decides about overall direction of the project

Other actors not considered as users of the system are e.g. data providers. Nevertheless, the development team will interact with data providers and give feedback on input data quality in order to improve the inputs.

4.2 User information and data access

A typical scenario for information and data access by a user is illustrated in Figure 4-1: a new user visits the SST CCI portal, informs herself/himself, and eventually requests SST CDR data for use in her/his project.

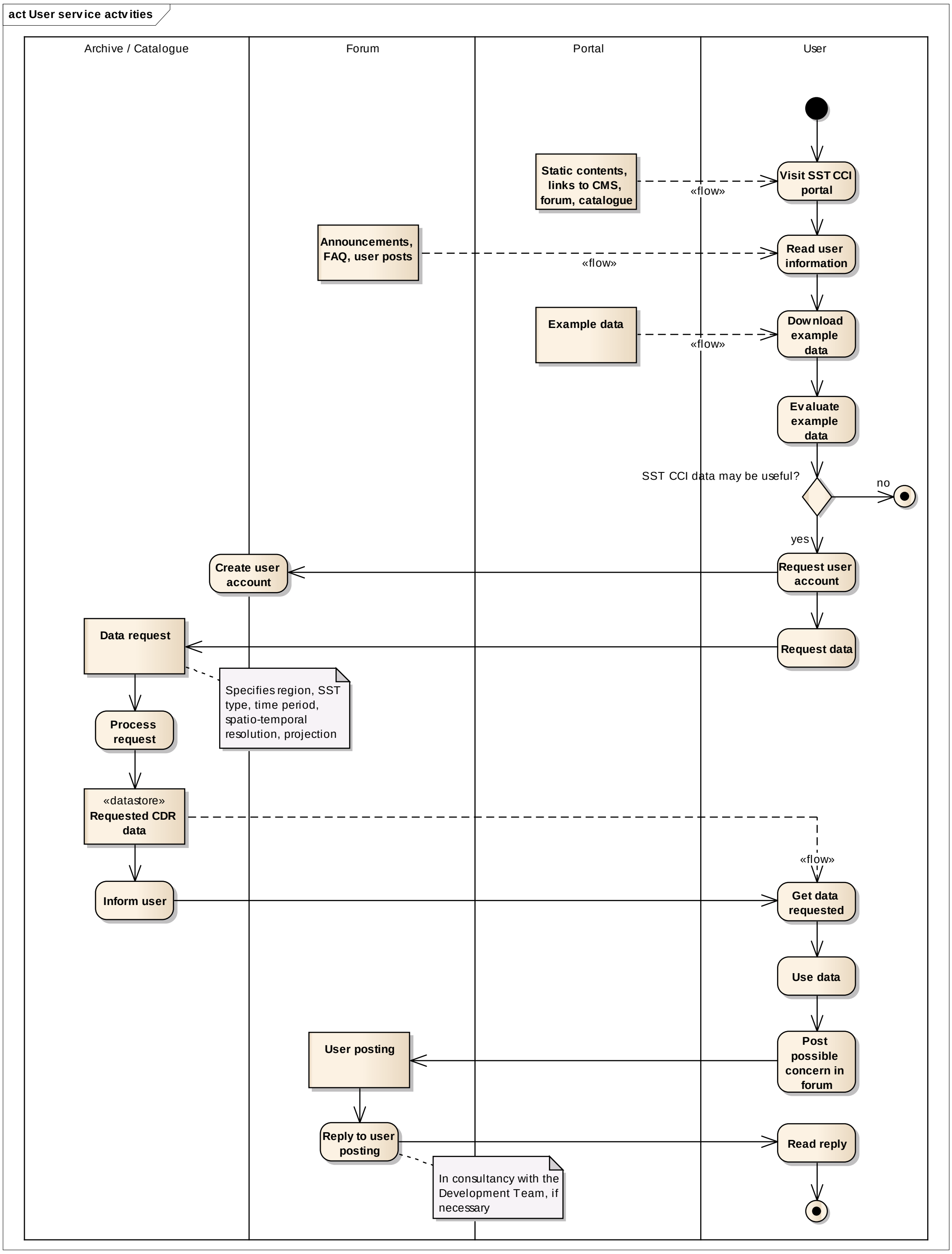


Figure 4-1: User information, data access, and user feedback handling activities

Static web pages provide links to dynamic contents in the CMS, the user forum, and a catalogue service. After reading general information on the project and its data products, the user consults the forum for latest news, frequently asked questions, issues or more specific topics. After reading all necessary information, the user decides to download a static set of example data to eventually conclude whether the SST CDR is useful for his/her specific application or not.

If the SST CDR is considered useful, the user requests an account, which is valid for both the data archive and the forum. After the account has been created the account details have been communicated, the user visits the catalogue service to create and submit a request for obtaining CDR data. The request may specify the geographic region of interest, the SST type (e.g. skin or depth), the time period of interest, the required spatial and temporal resolutions, and the map projection.

The data request is processed, and when completed, the user is informed. After the user has obtained the requested data, these are used.

The user may report possible concerns or issues with the data to the forum. The user posting then is replied, in consultancy with the development team if necessary. The user reads the reply and is satisfied. If concerns or issues still exist, the user may continue and start a discussion with the forum. Eventually the user feedback might point out an issue of the CDR processing algorithm, which then are recorded and maintained in the project's issue tracker.

Individual users, of course, may conduct activities, which deviate from the typical scenario. For example, expert users involved in the validation will not need to download example data or consult the forum. On the other hand, the activities shown in Figure 4-1 are not complete. For example, a user posting reporting inconsistencies between the actual and the documented data format or contents may lead to an announcement in the forum or an update of the project documentation offered by the CMS.

4.3 Processing and validation

The typical scenario for the activities carried out during the processing and the internal and independent, external validation of the SST CDR data is illustrated in Figure 4-2.

Note that there is no technical difference between the terms *processing* and *reprocessing*, because processing of the full or part of the CDR is foreseen as an activity in the algorithm improvement cycle (Section 4.4). Reprocessing and processing are used as synonyms. The integration of reprocessing into the algorithm improvement cycle is reflected in Figure 4-2, where the starting point represents the activity of algorithm improvement within the cycle, which precedes the reprocessing activity itself.

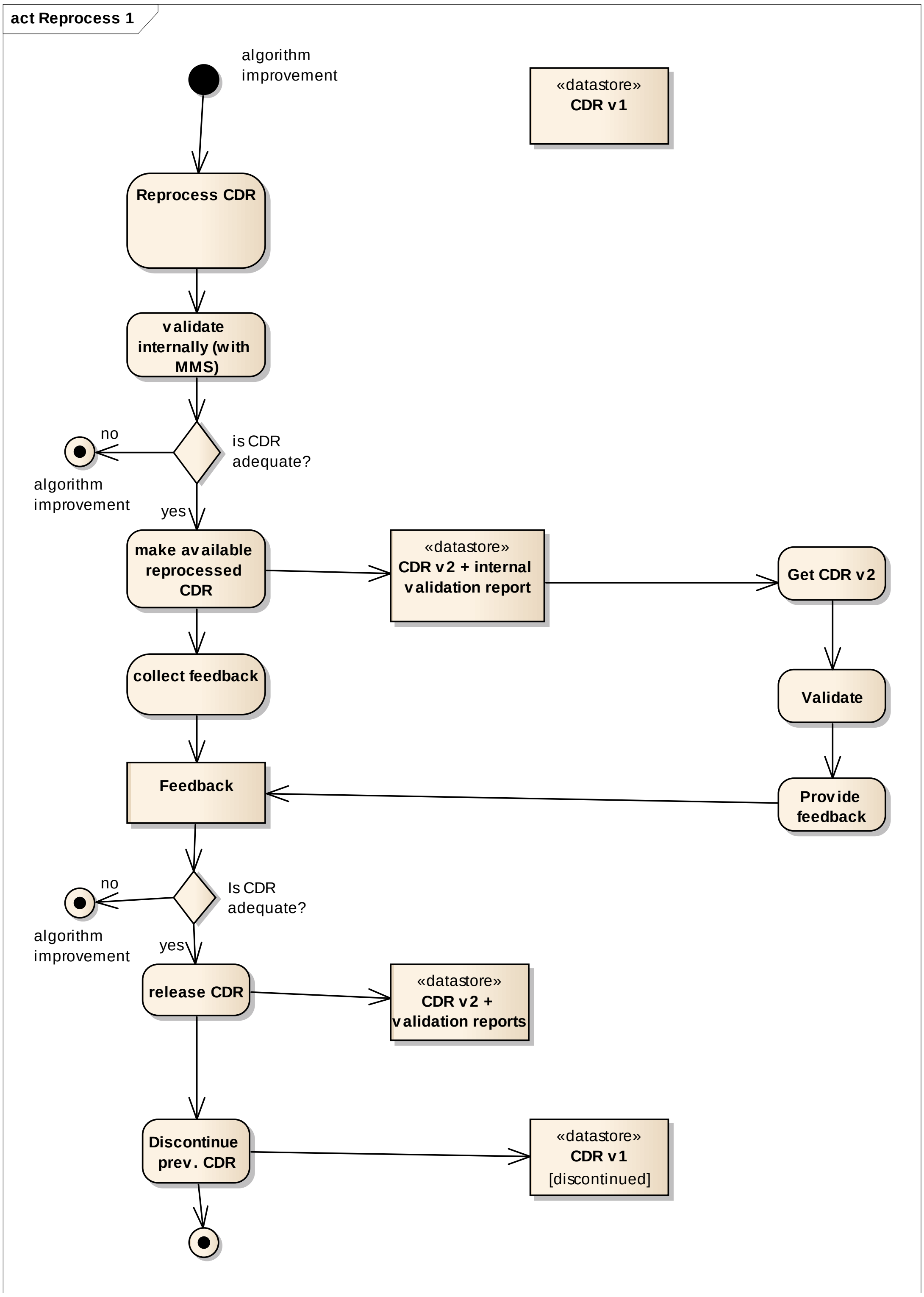


Figure 4-2: Reprocessing and validation activities

Processing and validation involves the development team and expert users. Data exchange is facilitated by user services (see Section 4.2). The series of activities carried out during processing and validation starts with the reprocessing of the full CDR. Reprocessing is a bulk processing activity, which is detailed in Figure 4-3.

After reprocessing, the resulting CDR is validated internally using the MMS before it is made available. . The CDR is published (CDR v2 in Figure 4-2) and made available through the user data catalogue and data request services as explained in the previous section.

Not every new CDR necessarily replaces its predecessor. Feedback from external experts and the internal decision may lead to another improvement cycle or the release of the CDR as new version. Users can validate the full CDR data and provide feedback to the development team. Feedback is to be reported in form of validation reports and may include match-up data sets in MMD format ([AD 6]). After scrutinising and evaluating the user feedback the development team decides whether the reprocessed CDR is adequate compared to user requirements and/or the degree to which it has improved in the respects that initiated the improvement cycle. If it is not adequate or sufficiently improved, a decision must be taken about a future cycle in which, the algorithm is addressed by the improvement-cycle activities explained in the next section.

If the reprocessed CDR is adequate, the CDR is released together with its validation reports. For a period of six months both the new and the previous CDR (CDR v1 in Figure 4-2) are available and continued. After this period the previous CDR is discontinued.

Reprocessing is a development team internal activity, which is detailed in Figure 4-3. However, there are actions to be conducted by either scientists or operators of the development team. The first action is the definition of the bulk-processing task. The definition specifies the software bundles and versions to be used, the processor configuration parameters, the product types, and the time period to be processed. (→ SST-SR-1220 Input versions, SST-SR-1230 Reprocessing input versions)

The definition of the processing task is given to the operators in the development team, which stage the data needed for the bulk task into fast storage and schedule the production. The operators monitor the production. If necessary, the operators handle exceptions trying to repeat the affected part of the processing, or by investigation and delegation to the scientists in the development team. (→ SST-SR-1390 Operator).

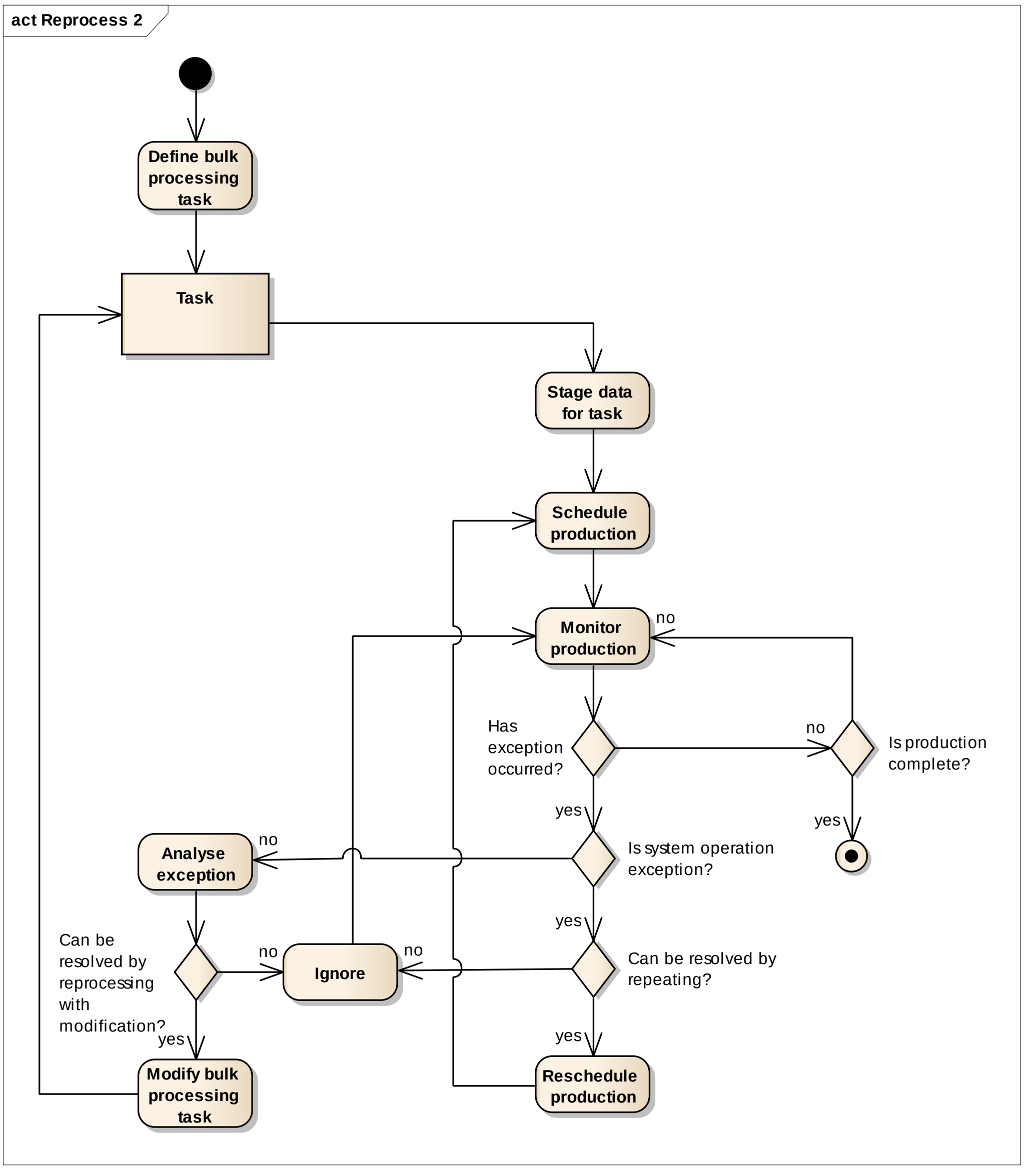


Figure 4-3: Bulk-processing activities

4.4 Algorithm improvement

The activities carried out for the purpose of algorithm improvement are illustrated in Figure 4-4. Both the development team and expert users are involved, with data exchange facilitated by user services.

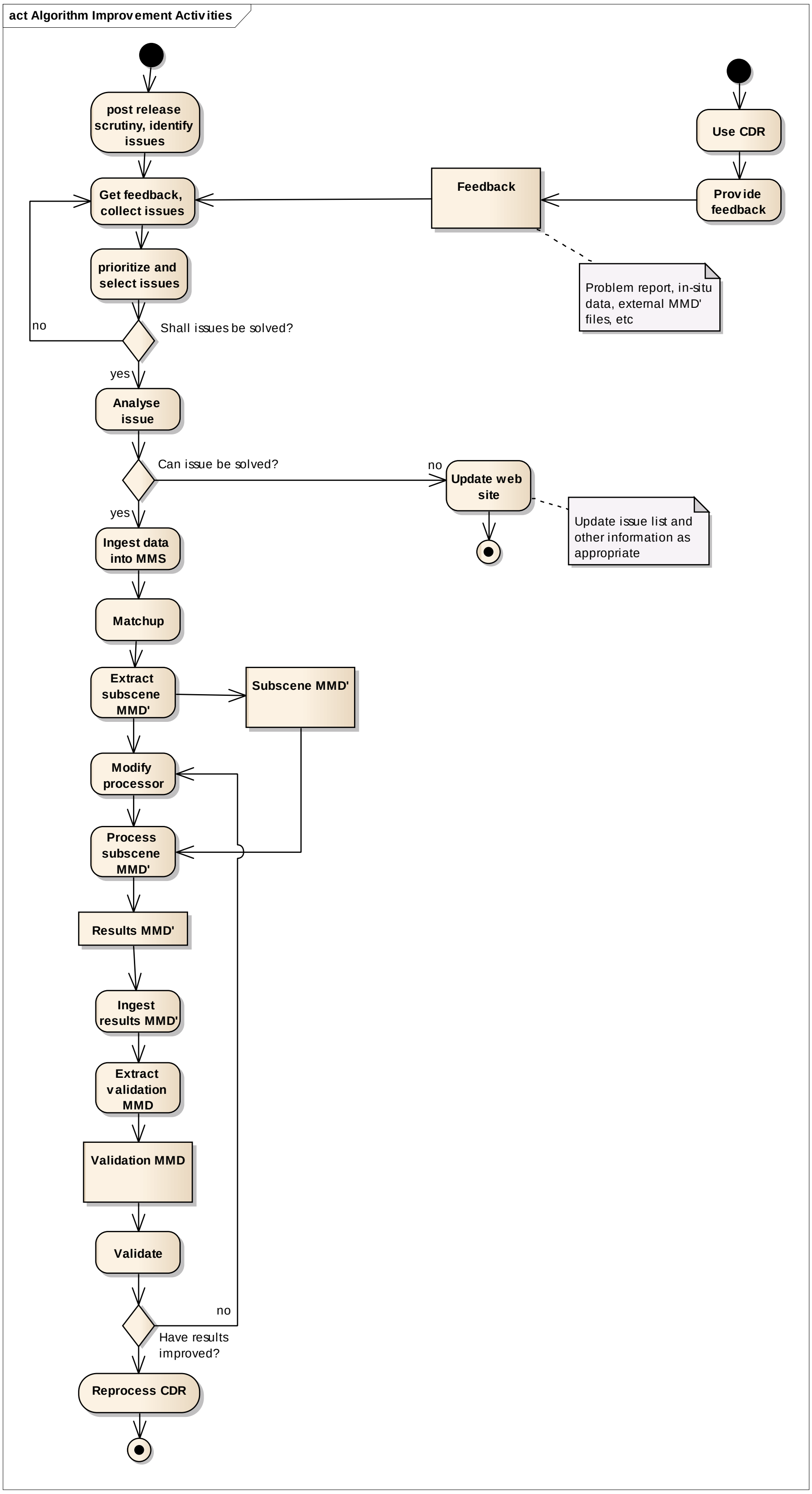


Figure 4-4: Algorithm improvement activities

Experts using the SST CDR report problems (possibly attached with in-situ data or MMD' files) to the SST CCI team by means of the user forum. The development team further investigates the data during post-release activities and scrutinises issues and decides whether these shall be resolved. If an issue shall be resolved, it is further analysed in order to decide if it can be resolved. If it cannot be resolved, the development team creates a new issue in the issue tracker and provides an updated list of known issues and other issue-related user documentation to the user services in order to update the web site.

If the development team is convinced that the issue can be resolved, any in-situ data and MMD files attached to the problem report are ingested into the MMS. If necessary, matchup data records are calculated and sub-scene MMD files are extracted.

In a new branch of the processor software source code, the development team makes necessary modifications in order to resolve the problem. Then the sub-scene MMD files are processed with the modified processor. The development team ingests result MMD' files into the MMS and then extracts an MMD file containing all data required for validation. (This generic description covers modifications to the full range of algorithms: SST retrieval, classification, skin-subskin adjustments, uncertainty estimations, etc.)

The development team conducts the validation of the sub-scene processing results of the modified algorithms; if the results have not improved the issue is re-analysed, starting another iteration of subsequent activities. If results have improved, the full CDR is reprocessed and validated more broadly to make sure that results have not deteriorated in other areas or respects.

5. INFRASTRUCTURE ALTERNATIVES AND THEIR IMPLICATIONS

There are some fundamental decisions that determine how the SST CCI system will look.

- To what extend shall the system use and build on the prototype?
- Shall the system be implemented in an existing infrastructure of an existing data centre?
- Which functions or subsystems are candidates for sharing with other ECVs?
- Shall the system run completely virtualised in a cloud?
- Shall a middleware be used, and which one?
- What shall be the balance between routine running and improvement cycles?

This section describes the **trade-off** for these questions. It may not be complete in its alternatives, but they are considered the most relevant for SST CCI.

The questions are not completely independent of each other. The existing infrastructure of a dedicated data centre will exclude the deployment in a cloud. Also the middleware question may be decided already in this case.

The following sections describe for each alternative how it will look, what has to be done in order to realise it, and the pros and cons.

5.1 Enhancement of the distributed prototype, or central SST infrastructure?

Figure 5-2 and Figure 5-2 depict the two alternatives of a distributed system and a centralised system. The distributed system uses an existing, specialised centre for Level 4 analysis while the centralised system has all processing in one place. Other aspects like user services are not touched by the alternatives.

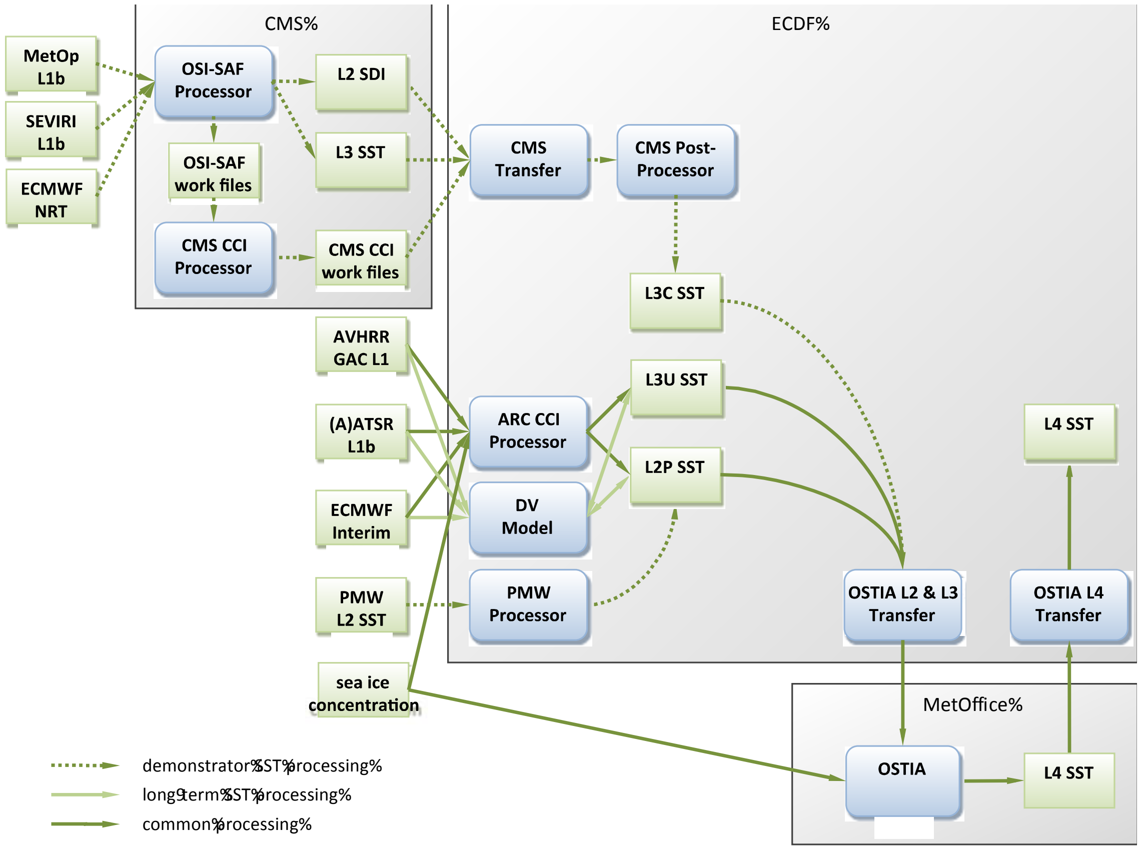
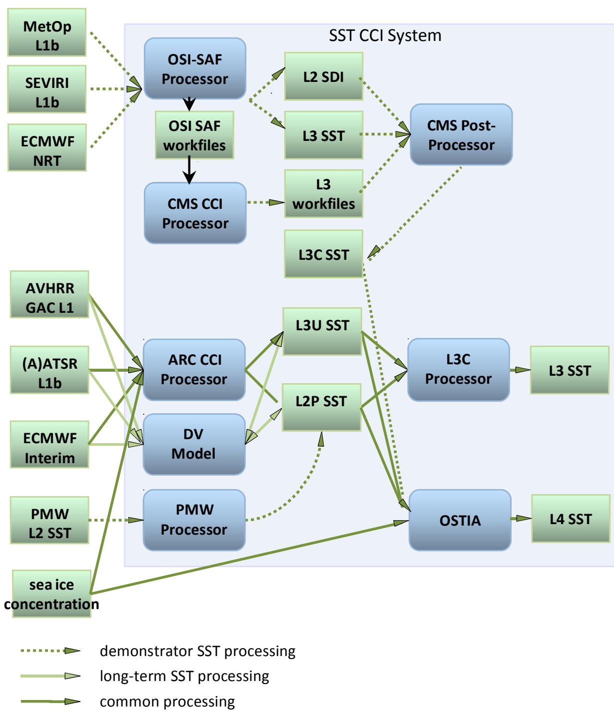


Figure 5-1: Distributed system in a configuration known from the SST CCI prototype



new deployment / central system

Figure 5-2: New deployment into a central SST CCI system

5.1.1 Enhancement of the distributed prototype

The SST CCI prototype mainly covers development, production and validation functions. Production in the prototype is distributed over three sites. The ARC processor ((A)ATSR Re-analysis for Climate) runs on the ECDF cluster at Edinburgh University, the OSI SAF system runs at Météo France CMS, and the OSTIA system runs at UK Met Office. The multi-sensor matchup system (MMS) is hosted on the ECDF cluster, too.

The ECDF cluster is a multi-purpose grid running Oracle Grid Engine. The project gets a certain amount of resources regarding processing time and storage. The storage is hierarchically split into a larger NAS and faster high performance disks accessible from all nodes of the cluster. The OSI SAF system is run operationally in a 24h fashion. The CCI sub-chain is a by-product of this system. The OSTIA reanalysis system is controlled by the Met Office Suite Control System. It is an offline variant of the operational NRT OSTIA system run daily. If the OSTIA reanalysis system was run daily at the Met Office to produce the short delay L4 product as part of a distributed CCI system, the reanalysis system would make use of several services already used by the NRT OSTIA. These services include functions for data extraction and QC using Met Office's Observational Processing System OPS, monitoring systems, observational database, data retrieval for short-delay mode, data delivery to users.

The distributed subsystems interact with each other using data interfaces. Product files are exchanged online using ssh based protocols.

This configuration is one option for the CCI phase 2 system for SST. As it only covers functions for production it must be supplemented by functions for long term data stewardship and by functions for user services.

To use this for SST CCI phase 2 the following actions are required (→ SST-SR-5215 L4 analysis system update):

- agreement with the involved external operational entities with respect to maintenance, sustainability, conformity and reliability and data and services
- software development for improved operations, automated data exchange, processor versioning, concurrent algorithm development and test, further development of the Multi-sensor Matchup System prototypes in Phase 1 in order to achieve automisation robust and routine usage of tasks
- extension of data volume and processing power for Sentinel data requirements
- either extension by or interfacing to user services with user access to SST CCI data
- either extension by or interfacing to long term archiving facility

5.1.2 New deployment in a central SST infrastructure

In order to make this alternative comparable with the distributed one, only the production and development subsystem is considered. In the central SST infrastructure the processors for Level 2 processing, Level 3 processing and Level 4 analysis are deployed and integrated. The data are available, and data streams for new input data are routed to this system.

The critical point is how this centralised system is kept up-to-date and whether it is the primary development platform for new algorithms. If not, one of the purposes of the system is not fulfilled, and the migration of new processor versions will require additional steps. To make it the development platform, the centralised system provides a virtual private network (VPN) access without barriers for the different contributors of the development team.

To use this for SST CCI phase 2 the following actions are required:

- find and decide about a central SST infrastructure provider
- agreement with the contributing entities
- VPN setup for distributed development team
- software development for improved operations, processor versioning, fully featured Multi-sensor Matchup System
- software development to make the OSTIA reanalysis system a stand-alone one without loosing the transfer of improvements from the NRT OSTIA (shared codebase), and with replacements or interfaces to services used (e.g. OPS)
- extension for Sentinel data volume and processing power requirements
- either integration of or interfacing to user services with user access to SST CCI data
- either integration of or interfacing to long term archiving facility

5.1.3 Pros and cons

Arguments for the decentralised approach are

- no integration effort as it is already done in the prototype
- short cycle may be optimised as production is close to respective experts for the processing level, processing system is kept up-to-date with minimal effort
- data stream for new data and auxiliary data is already established
- duplication of services shared between operational NRT OSTIA and OSTIA reanalysis system is avoided, update of software system is simplified, no risk of failure of stand-alone version

Arguments for the central approach are

- independence from respective distributed operational entities and the allowed update cycles of their operational systems, may increase stability of the CDR and flexibility of update cycles suitable for CCI
- no data transfer required from site to site for intermediate products, both in case of bulk reprocessing and in case of continuation of the CDR
- sharing infrastructure with other ECVs is possible with this configuration
- the infrastructure and middleware can be selected optimal for SST CCI

There also is the option of a hybrid solution with a migration of parts of the CMS subsystem but a decentralised Level 4 analysis system (leaving OSTIA in UKMO). This is very similar to the decentralised approach with most features of it (data transfer, subsystems close to respective experts).

5.2 Use of existing earth observation processing data centres

There are organisations that operate earth observation processing facilities since many years. And they do this using dedicated and well-elaborated infrastructures. Two scenarios are considered here:

- Re-use: The implementation of the SST CCI system as an instance of an existing infrastructure
- Outsourcing: The embedding of an SST CCI system into an operational environment and context

Both of them may be considered options where operations has a loosely coupled interface to development while the alternative is that

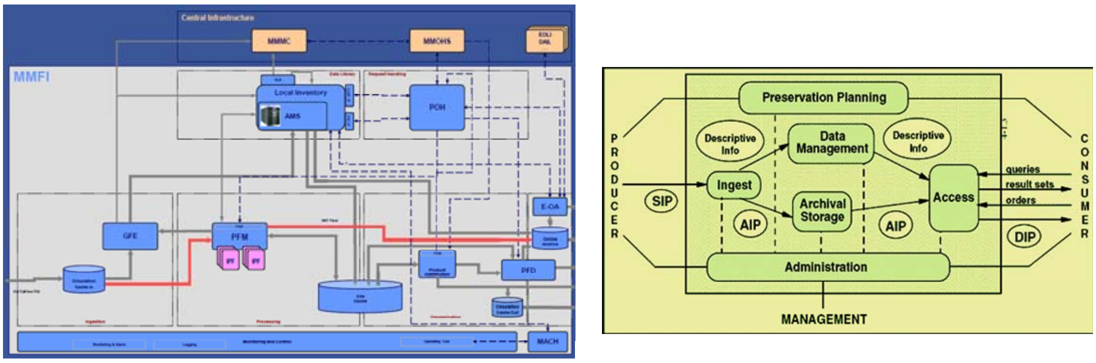
- operations are instructed by the development team, with tight coupling of development and operations.

These three alternatives are considered in the following subsections.

5.2.1 ESA MMFI for SST CCI?

To explain how an SST CCI system may look using existing EO infrastructure, this section considers the ESA Multi Mission Facility Infrastructure (MMFI) (Figure 5-3) as an example. MMFI has been developed as payload data ground segment seven years ago to replace the initial ENVISAT processing facility, and it is operated since then also for ERS and third party mission data.

MMFI implements the Open Archival Information System (OAIS) model. It is an archive-centric approach. Processing is understood as consumer of its inputs from archive and producer of its outputs to archive. The archive is the buffer in processing chains except for NRT workflows (red line). MMFI further is optimised for systematic data-driven processing and for processing user orders on demand.



MMFI architecture
[http://www.sciops.esa.int/SD/CSG/EGW08/GianMariaPinna.pdf]

OAIS Model
[figure taken from http://en.wikipedia.org/wiki/OAIS]

Figure 5-3: Implementing the SST CCI system on top of an existing ground segment?

In such a multi-mission facility SST specific parts would be the integrated processing facilities(IPF), the configurations of each of the components, and the workflows to be implemented in Product Order Handling. The MMFI in this configuration does not foresee bulk re-processing as a use case. For example, the concept of product identifiers (instead of sets of products) is used with requests. As a second example the product dissemination element is optimised for customer-specific delivery packages. Questions of how bulk reprocessing is ordered, massively parallelised, and how to avoid that the archive becomes a bottleneck are open.

To use an infrastructure like the MMFI for SST CCI the following actions are required:

- develop and configure product models of the SST CCI products for the Local Inventory
- configure ingestion and dissemination
- implement processors as IPF using the Generic IPF interface
- implement workflows for the SST CCI processing chains in the MMFI production control element to use the on-request processing for bulk reprocessing
- add a generator and monitor for requests to implement bulk processing of complete missions
- optionally interface the MMFI processing control element with a processing middleware for bulk processing (a combination that is not an optimal solution)

5.2.2 Embedding of an SST CCI system into an operational environment

As a second alternative, the operations of a specific system for (SST) CCI could be outsourced to some processing centre that provides the environment and personnel to operate the system. The difference to the next alternative is that in this case production control is operated by the processing centre, and the interface between development and operations are requests for bulk production, software/processor installation packages, and result datasets.

To use the outsourcing approach the following actions are required:

- development, integration and configuration of SST CCI system
- agreement with processing centre, definition of operations and of system updates
- deployment and transfer to operations
- processor update deliveries, definition of production tasks

5.2.3 Operations instructed by the SST CCI development team

Both alternatives above include the use of facilities encapsulated into operational environments. The access to such operational facilities by the development team for tests, validation and development may be restricted. This third alternative is a facility that whose routine running can be operated by an “operations team” and at the same time be used by the development team for test, also bulk tests.

To use the development-focussed approach the following actions are required:

- optionally decide about hosting to the level of hardware, network, operating system
- development, integration, configuration, deployment of the SST CCI system
- definition of roles and corresponding rights

- VPN setup for distributed development team

5.2.4 Pros and cons

Arguments for the MMFI approach are

- software development is small

Arguments against the MMFI approach are

- MMFI is not optimised for the CCI use case of repeated full-mission re-processing
- The archive-centric approach is a build-in I/O bottleneck because of the data transfer from and to the archive. In this case, streaming data between processing steps is much more efficient.
- For other existing multi-mission solutions it has to be evaluated carefully whether they natively support massive parallel bulk reprocessing

Arguments for embedding SST CCI into an operational environment are

- experience with operations (more stringent than short-delay provision for SST CCI)
- other facilities of the environment may be used (user services, long term archiving, network infrastructure)

Arguments for the development-focussed approach are:

- development team has the knowledge to assess and decide what to process and when to use new algorithms
- mode of operations with frequent reprocessing to test and validate new algorithms/new parameterisation is supported best
- short cycle supported best
- minimal effort for transfer to operations for new versions
- data is controlled by the SST CCI teams (data license for inputs relevant for other ECVs)
- Robust reliable production is less important for bulk reprocessing and short-delay provision than in an NRT environment, because small delays due to disturbances can be recovered with relatively smaller margin of additional resources.

5.3 Degrees of sharing with other ECVs

Considering the subsystems for development and production, for user services, and for long term data storage, different combinations of sharing are possible, among them

- a central infrastructure instance with all subsystems shared by all / a group of ECVs
- separate production subsystems but shared user services and shared long term data storage (Figure 5-4)
- separate collaborative system instances with harmonised interfaces
-

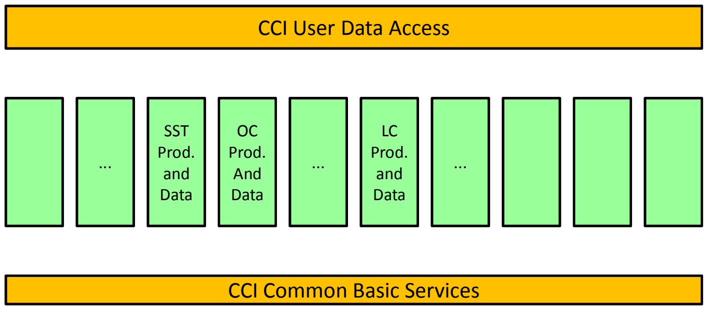


Figure 5-4: Sharing user services and data stewardship with other ECVs

The questions that distinguish these configurations are

- Shall user services and long term archiving/data stewardship be shared?
- Shall the production and development subsystem be shared among ECVs?

5.3.1 Sharing user services and long term archiving

With a single user services there is a common web interface where grouped ECVs can be accessed. They are presented in a common way, considering their specifics to a certain degree. Functions like user management, document server and code repository are shared. There is a common catalogue that supports cross-collection search to find products from grouped ECVs in a common response. Users access products using the same service for grouped ECVs. Online data services for sub-setting, re-gridding and aggregation are equipped with ECV-specific plug-ins to implement e.g. the specifics of error propagation in SST.

In case of non-shared user services the minimum requirement is interoperability. Search via one catalogue interface allows to search in the catalogues of several ECVs with the same criteria. One user account is sufficient to download data from other ECVs. Several ECVs implement the same protocols and accept the respective credentials of their users.

The long term archive stores output product versions together with representation information, metadata and documentation for long term stewardship. Long term stewardship can be expected to extend beyond the CCI project runtime the long-term archive. In addition long term archiving keeps a backup of all input data which releases production systems from data reliability requirements (→ SST-SR-1500 Backup archive). If SST CCI shares the long term archive with several ECVs it simply stores its outputs and optionally retrieves products from the same instance as other ECVs.

5.3.2 Sharing a production and development environment

A production and development environment shared by several ECVs, among them SST CCI, could provide common resources on different levels usable by all ECVs:

- production storage
- processing hardware (virtualisation)
- processing middleware (cluster)

- production control (multi-ECV production system)

If the same inputs are used sharing production storage saves space. Every ECV can immediately use the results of ECVs it shares a production storage with. If sharing is on the level of the hardware SST will get virtual machines in such an environment to deploy the SST CCI system.

If sharing is on the level of the processing middleware then there will be e.g. a Grid Engine instance or a Calvalus/Hadoop instance (see section 5.5 for the middleware choice) with quotas and fair scheduling for SST and other ECVs. If one ECV does not use its share the resources can be used by other ECVs temporarily. Depending on the policy for scheduling a reprocessing campaign for one ECV may restrict the use of the infrastructure by other ECVs.

Sharing or not sharing the production subsystem is in one direction obviously related to the decision whether to continue the prototype configuration from section 5.1. To use sharing means to migrate the prototype to the shared platform.

Even if sharing requires too much harmonisation among ECVs there may be common multi-ECV elements provided by CCI that can be configured for SST. They are then completed by SST-specific elements and plug-ins to compose the SST CCI system.

5.3.3 Pros and cons

Arguments for sharing are

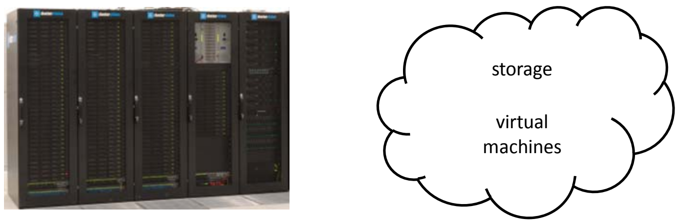
- Operations and maintenance required only once for the shared functions
- Common user services can support functions like cross collection search in a catalogue an harmonised access to data and subsetting services
- A common data archive can serve for several ECVs without additional effort. It can be used for outputs, but also for inputs to release the processing systems of single ECVs from high reliability data storage requirements.
- Peak power for SST is higher if it can use the resources of other ECVs temporally
- Synergy by immediate availability of outputs and data exchange among ECVs, e.g. sea ice for SST without data transfer.
- Sharing provides functions to ECVs with less infrastructure or an infrastructure the single ECV does not have in its prototype

Arguments against sharing are

- Harmonisation is work, some prototypes are already developed on the basis of existing elaborated systems
- Dependencies increase the risk of failure (→ SST-SR-1520 Autonomous system)
- User communities and the interaction with users are different for different ECVs. A common user service may be not flexible enough

5.4 Cluster or cloud?

How will the SST CCI system look and what has to be done to run the SST CCI system on a cluster or in "the cloud"? There are only a few experiences with the cloud approach, but there are experiences with different clusters, among them the shared ECDF cluster used for the SST CCI prototype and a Calvalus/Hadoop cluster used for other CCI projects. The middleware question is treated in the next subsection. We concentrate on the differences between cluster and cloud here.



cluster
(https://www.wiki.ed.ac.uk/download/attachments/14254618/eddie-final.png)

cloud

Figure 5-5: Cluster or cloud

Note that user services and data access looks the same from the user's perspective in both cases. There is no necessity to move the user services themselves into the cloud. User services in the cluster case would serve data from the cluster, and in the cloud case they would serve data stored in "the cloud".

5.4.1 Running the SST CCI system in a cluster

A cluster is a composition of computers, storage and network that can be used together for concurrent processing in a coordinated way. Clusters provide:

- computing power, physical or virtual machines, homogeneous or different classes, same operating system
- storage, optionally hierarchical with slower background storage and fast disks, or distributed to the processing nodes
- network, between machines, to storage, optionally with different bandwidths
- front end / master, dedicated machine(s) to control jobs on the cluster
- deployment, a way to install software on the cluster to be used from all machines
- staging, a way to make accessible large data volumes in case they are stored in background storage
- I/O to/from cluster, a way to add input data to the cluster and to retrieve output data, either via a network connection or offline on media
- quota regarding space and processing time

As an example ECDF consists of 286 machines with together 2912 CPU cores, high performance disks accessible from all nodes, and NAS background storage, Gigabit connection of nodes with 10 Gigabit backbone, and Infiniband to a subset of nodes.

A cluster may be dedicated to a project or it may be shared. Depending on which form of middleware is foreseen a configuration of storage and network should be selected: network storage with a fast network for Grid Engine, or preferably local disks at computing nodes for Calvalus/Hadoop.

To run SST CCI on a cluster the following actions are required:

- decide about middleware
- deploy processor software and install production control, integrate with middleware

- implement data management for I/O and optionally for staging
- ingest data into storage

This has already been done for the prototype and the ECDF cluster.

5.4.2 Running the SST CCI system in a cloud

A cloud infrastructure abstracts from the physical hardware and provides services like storage and virtual machines. Cloud infrastructures are available commercially from different providers (Amazon as an example). They provide:

- computing power in form of virtual machines, selectable classes regarding CPU power and RAM (Amazon EC2)
- storage as virtual disks, mountable on virtual machines (Amazon EBS)
- network storage accessible via the Web (Amazon S3)
- a Hadoop MapReduce implementation (Amazon EMR) based on network storage (S3)

Virtual disks are more expensive (magnitude of 100 Euros per TB and month) than network storage (magnitude of 50 Euros per TB and month) such that network storage will be used for the processing archive. Computing power is either paid on an hourly basis or with an advanced payment and a smaller hourly rate (magnitude of 2500 Euros per year for a 16GB/8 core machine). (All numbers are very rough estimations to understand the relative magnitudes!) We have considered a cloud solution (Amazon EC2/S3) available at the time of investigation. Future solutions may provide a cloud specialised to earth observation that may provide the input data as a service and other features attractive for SST CCI.

To run SST CCI in a cloud the following actions are required:

- cost estimation
- set up virtual machines and storage
- optionally decide about middleware
- deploy processor software and install production control, integrate with middleware
- ingest input data into network storage

In order to use the flexibility of a cloud the ready-to-use configuration of a virtual machine template for SST CCI can be stored in network storage and can be launched on demand.

5.4.3 Pros and cons

Arguments for a cluster solution are

- Structure can be selected suitable for the relatively large data volumes of EO data processing, e.g. by considering data-locality for processing
- Degree of reliability/availability can be adjusted to the offline processing task
- Both approaches are designed for large data volumes and high degree of concurrency, but price of a cluster of the same power and capacity can be expected to be lower than commercial cloud

Arguments for a cloud solution are

- Flexible use of processing power possible

- Output data can be distributed to users via cloud network storage without copying
- No hardware maintenance

Independent of the decision between cluster or cloud an environment that already provides large amounts of input data is an advantage. If the Sentinel 2 SLSTR data are available in either a cluster or a cloud environment to several projects the SST CCI project will be able to follow the approach to transfer the algorithms to the data.

5.5 Processing middleware alternatives

There are practical experiences with two processing middleware approaches: Oracle Grid Engine and Calvalus/Hadoop. How they can be used for an SST CCI system is described in this section.

5.5.1 Grid Engine

Oracle Grid Engine, previously Sun Grid Engine, is a batch queuing system for computer clusters. It schedules and manages the execution of large numbers of concurrent jobs. A typical grid engine cluster consists of a master node and execution nodes.

The main command provided by the grid engine middleware is qsub. Example job submission:

> **qsub** -l h_rt=24:00:00,sages_1ppn=1 -j y -cwd -o log/nwparc3-2009-05-01-n15.out \
>
> -N ar20090501n15 **bin/nwparc3-run.sh 2009 05 01 n15**

It submits a job to the cluster. The command line argument of the qsub command will be executed on some node in an environment identical to the one of the qsub execution. The submitted job gets attributes that match node capabilities and queue features (-l).

To run the SST CCI system using a Grid Engine cluster the following activities are required:

- write pairs of start and run scripts for each processing step. The start script splits the processing request into qsub calls to be handled on a single host each. The run script is executed on some execution node of the cluster. Both the start and the run script may contain loops to iterate over the inputs. The border between these two iterations can be moved towards more concurrency or towards larger jobs on a single node.
- integrate production workflows for the SST CCI processing chain and the iteration over the input time period in production control
- install processors
- ingest input data

Parts of this have already been done for the SST CCI prototype.

5.5.2 Calvalus/Hadoop

Apache Hadoop is an open-source cluster management software framework for data-intensive distributed computing. It is based on ideas from Google for a distributed file system and the map-reduce programming model. Hadoop is implemented in Java and can be extended with this programming language. There are several contributors to the framework, among them Yahoo. In a short form:

Hadoop = HDFS + scheduler + map-reduce

The Hadoop Distributed File System manages data on local disks of processing nodes, cares for automated replication, is used for automated software deployment to the distributed cache at nodes, and makes accessible all files homogeneously and transparently from all nodes.

Hadoop introduces the concepts of *job*, *task* and *attempt* to support the robust parallel processing of large sets of inputs. A job with a set of inputs is split into tasks for a single input each. A task is executed by a node in an attempt that is automatically repeated on a different node in case of failure. Hadoop provides job queues, different scheduling strategies, priorities, and fairness. Scheduling is aware of where the data is stored to schedule tasks to nodes that have (a replica of) the data.

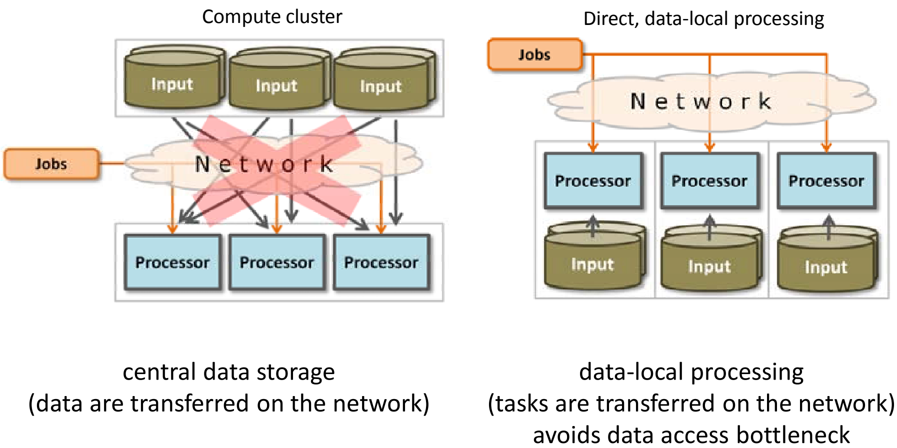


Figure 5-6: Network access versus data-local processing

Hadoop map-reduce is a programming framework that allows concurrent aggregation of data. Map-reduce supports such jobs by sorting, partitioning and streaming of intermediate data between nodes.

Calvalus is a software system that extends Hadoop by

- processor adapters for Unix executables in any programming language, processor adapters for the ESA BEAM Graph Processing Framework
- EO data processors, all BEAM operators, Level 3 binning and mosaicing algorithms, matchup with in-situ data, and trend analysis. Matchup extraction is done on the fly on full-mission input datasets with processing of extracts with the selected algorithm(s).
- a data directory structure for inputs and outputs, auxiliary data, processor software bundles, in-situ data, and user areas
- a Hadoop configuration, a job submission client, a portal for job submission
- production monitoring and control for processing workflow management and bulk production

As a formula:

Calvalus = EO data processing with Hadoop

To run SST CCI on Calvalus/Hadoop the following activities are required:

- integrate processors, build processor bundles as .tar.gz of processor installations of the different SST CCI processors (or integrate them as BEAM operators)
- integrate the SST CCI Level 3 as map-reduce job, optionally do the same for the Level 4 analysis step
- prepare request templates and configure production control for the SST CCI bulk workflows
- ingest the input data into HDFS

5.5.3 Pros and cons

Arguments for Grid Engine as middleware are:

- simple integration of executables and shell scripts, simple interface with qsub
- NFS for software access (no deployment necessary) and data access from all nodes
- scheduling that considers job attributes like CPU and memory demand

Disadvantages of Grid Engine as middleware are:

- shared storage is generally a bottleneck for large EO data sets
- staging is required if storage is hierarchical, adding complexity

Arguments for Calvalus/Hadoop as middleware are:

- distributed file system HDFS with standard disks on standard machines, automated replication for data security and continued service in case of node or disk failure
- scheduling that considers data-locality, data-local processing avoids network bottleneck,
- higher level of service in job scheduling that cares for parallelisation on product sets (or file splits if required), with jobs, tasks, attempts, automated failover and repetition (the Grid Engine job corresponds to a Hadoop task)
- map-reduce implementation with automated partitioning and streaming, avoids file I/O, supports concurrent distributed aggregation, used for EO Level 3 and Level 4 processing, matchup computation, trend analysis
- distributed cache for automated software deployment on nodes

Disadvantages of Calvalus/Hadoop as middleware are:

- HDFS does not fully support a random-access read/write file interface. A local data copy is therefore necessary for NetCDF files which can only be accessed through a dedicated (random-access requiring) library. (Note that this data copy usually is not a copy over the network, but just a constant overhead.)

(→ SST-SR-5280 Storage extensions, SST-SR-5290 Processing extension, SST-SR-5300 Performance scalability)

5.6 Conclusion for SST CCI

The first trade-off (between distributed and centralised SST retrieval and L4 analysis) is finely balanced. The two strong arguments are "close to the experts" (favouring distributed) and "avoiding data transfers" (favouring centralised). If "close to the experts" can be implemented by a central SST infrastructure with low barriers to remote usage then this will be the better choice. The criticality of avoiding data transfers depends on transfer rates and the required timeliness of provision of SST CCI products in short delay mode. The centralised approach requires a stand-alone Level 4 reanalysis system (OSTIA, with shared code base with the operational OSTIA at UKMO).

The conclusion regarding outsourcing is not to transfer operations to an external team of an existing data centre. Arguments for this are that the development team shall have a strong role and that outsourcing would make the improvement cycle more complicated. This does not exclude the use of a data centre as a (lower-) service, however.

In particular, if a data centre provides shared access to Sentinel 3 SLSTR data, its use as a cluster or cloud environment as described in section 5.4.3 is highly desirable. Although SLSTR data can be expected to be available only in the second half of CCI phase 2, and although the establishment of Collaborative Ground Segments that will provide the data is not yet finalised, SST CCI should investigate and aim at using such an environment since the focus of SST CCI is on using SLSTR data. The criteria for selecting such an environment are

- performant access to full mission data, also with increasing volumes of SLSTR, keeping up with reprocessing times
- sufficient space for the output data
- sufficient processing capacity for frequent partial reprocessing, and for full mission reprocessing about two times a year
- open access to the infrastructure by the development team (VPN access), SST CCI software updates without restrictions
- capabilities for data exchange, network capacity for dissemination to shared user services or directly to users
- moderate price affordable by the SST CCI project

This has to be investigated and compared with an SST CCI stand-alone solution before a selection or a final conclusion can be drawn. The CEMS infrastructure is a candidate to be investigated at the time of writing.

The two arguments for sharing with other ECVs are possible synergies from the exchange of results and cost reduction by using the same function or infrastructure. The two arguments against it are "harmonisation is work" and "independence". The recommendation is to share on a level where independence is ensured for all that is required for the small cycle, e.g. by providing dedicated processing resources for SST CCI available all the time. For other functions to be shared the harmonisation effort should be compared to the gain. User services are good candidates for sharing, either with other ECVs or with existing SST data providers, e.g. to meet availability requirements (→ SST-SR-6370 User services availability). Another candidate is input data provision (→ SST-SR-1510 Archive reliability), in particular for the larger amount of Sentinel 3 SLSTR data during the first years of the mission.

Regarding cluster or cloud and the middleware to be used, two cluster alternatives are acceptable with different benefits. To continue with Grid Engine from the prototype will be the smallest change, maximising scope for other improvements. To use Calvalus/Hadoop for SST CCI allows use of Calvalus' bulk processing capabilities and of on-the-fly matchup processing. Both these Calvalus functions are interesting for SST CCI. So a Calvalus/Hadoop cluster is the recommendation in the absence of external constraints (e.g. on where a cluster should be located). If an existing infrastructure is used that provides the SLSTR data then SST CCI has to adopt to the given middleware solution in such an environment.

Using a generic commercial cloud service (Amazon EC2) is still out of scope for earth observation data processing for the foreseeable timescale, because the ingestion and storage of large amounts of input data is relatively too expensive.

6. FUNCTIONAL DESIGN

The functions for user services, production management and for the continuous improvement cycle are to be implemented by functional components. This section defines these components by their purpose and function and by the data they manage. The section further defines the main interfaces of the system by the interface items, their content and format, and the protocol used.

6.1 Services for the users

User services are the functions and interfaces that an external user perceives and uses to interact with the system, and that the SST CCI project uses to present itself to the users and interact with them. A set of public resources allows users to gather information about the project, the goals, the data provided and algorithms used to produce the results. In addition to this, of course, the resulting dataset has to be made available using access mechanisms so that users can choose their preferred way of obtaining the data.

The same set of web-resources shall allow registered users to easily up- and download data, tools, and documentation. In addition, anonymous and registered users shall have access to information exchange tools like a managed forum, news feeds and other. Users access information and catalogue services through a central web-portal that bundles links to web-resources implemented independently from the central portal.

The SST CCI user services mainly comprise two functional aspects:

- Access to the data generated by the processing system, to matchup datasets and online subsetting and processing resources
- A community interface for general information exchange, document management, user registration and interaction.

6.1.1 Components and interfaces

These two high-level functional aspects are implemented using a number of dedicated software packages that implement specific web functionalities (Figure 6-1).

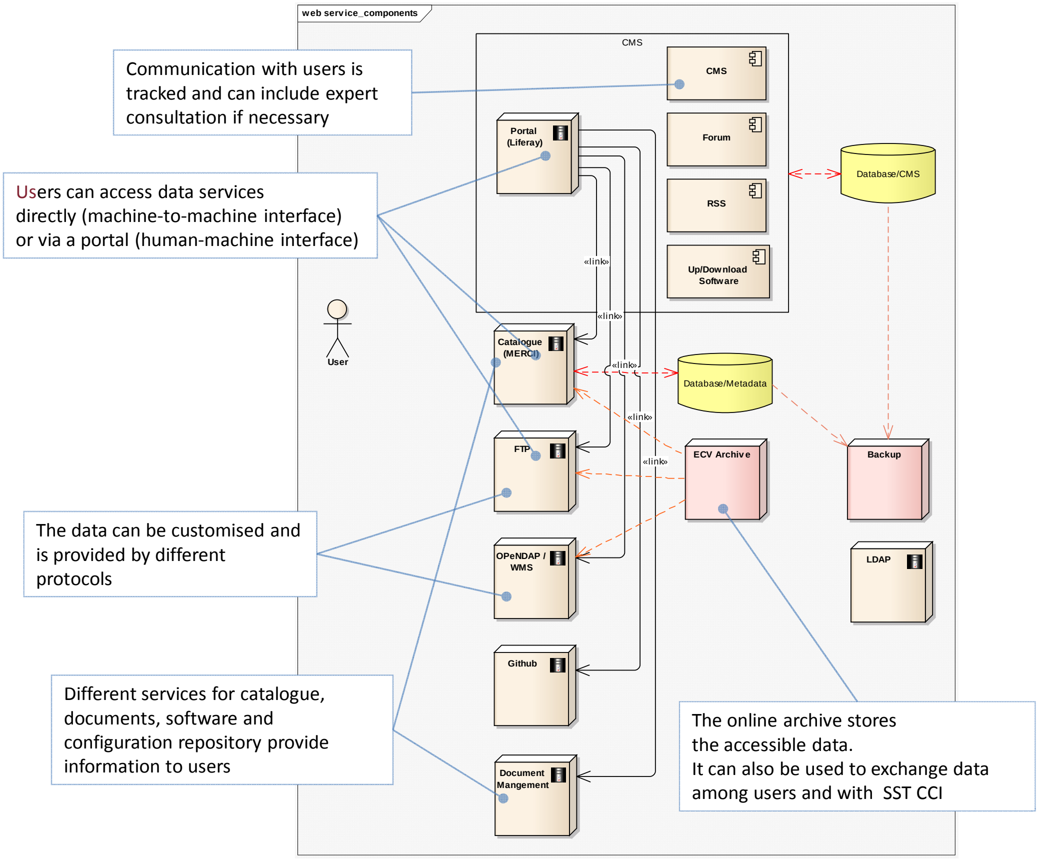


Figure 6-1: Portal and backend services for SST CCI users

A central SST CCI web-portal bundles all the resources available to the users (→ SST-SR-6240 Web site). The portal is preferably implemented using a content management system (CMS). Some of the desired functionality is directly available in the CMS portal software (user forum etc.) while other components act as stand-alone or even remote web-resources. All of the distributed functional components are connected using links from the central portal. The background components have standard interfaces that in principle could also be integrated into other portals.

The other user-accessible services are catalogue (section 6.1.4 below), data access via FTP (section 6.1.3.1), OPeNDAP and WMS (section 6.1.3.2), and functions for customised data by subsetting, regridding, regional averaging (section 6.1.3.5), bulk product access using media circulation or cloud storage for large amounts of output data for climate users (section 6.1.3.4), a document server and access to the software repository to publish a subset of the project documents and algorithm implementations (section 6.1.6). (→ SST-SR-1295 Processor documentation, SST-SR-6180 Output provision, SST-SR-6190 FTP access, SST-SR-6230 Online subsetting and regridding)

For functions that allow write access to resources like contributions to the forum or provision of inter-comparison data it is desirable to identify users. For this purpose users can register in a user management service based on LDAP. It ensures that users have the same credentials for all the services.

Table 6-1: Main interfaces of the production and development subsystem

Ifc ID	Interface Name	Endpoints (provider, user)	Interface items content and format	Data exchange protocol
Ifc-9.1	Web portal interface	Web portal External user's browser	Web pages, forms, forum entries, issues	HTTP(S)
Ifc-9.2	Catalogue interface	Catalogue External user's browser, OGC CSW client	Catalogue queries, result sets, metadata entries, browse images, product data packages	HTTP(S), OGC CSW
Ifc-9.3	Online archive interface	FTP server External user's FTP client	Data products	FTP, FTPS
Ifc-9.4	OPeNDAP and WMS interface	OPeNDAP and WMS server External user's OPeNDAP client, WMS	Requests, data subsets, images	HTTP, OPeNDAP, OGC WMS

Ifc ID	Interface Name	Endpoints (provider, user)	Interface items content and format	Data exchange protocol
		client, browser		
Ifc-9.5	Processor repository interface	Repository server External user's version control client, browser	Software source packages, versioning information	git, HTTPS
Ifc-9.6	Document exchange interface	Document management system User's web browser	Documents, versioning information	HTTPS
Ifc-9.7	User management interface	LDAP server Users: portal, FTP server, document management system	Authentication and authorisation information, user profile information	LDAP

6.1.2 Structured output data storage

The online archive is the backbone of all data services of the SST CCI system. Users access the online archive directly via FTP and HTTP. Other data services use the data and customise it according to the user's request. The production subsystem feeds new data and new versions into it for publication.

The online archive is structured by a directory tree where files are organised according to archiving rules (→ SST-SR-1210 Structured storage, SST-SR-6160 Output format and naming). The archiving rules for SST CCI use the following pattern:

<categ>/<type>[/<sensor>]/<version>/year/month[/<day>]/

which generates paths like

eodata/l2p-sstskin/avhrr-noaa16/v1/2010/05/12/...

An example archive structure following this rule is depicted in Figure 6-2.

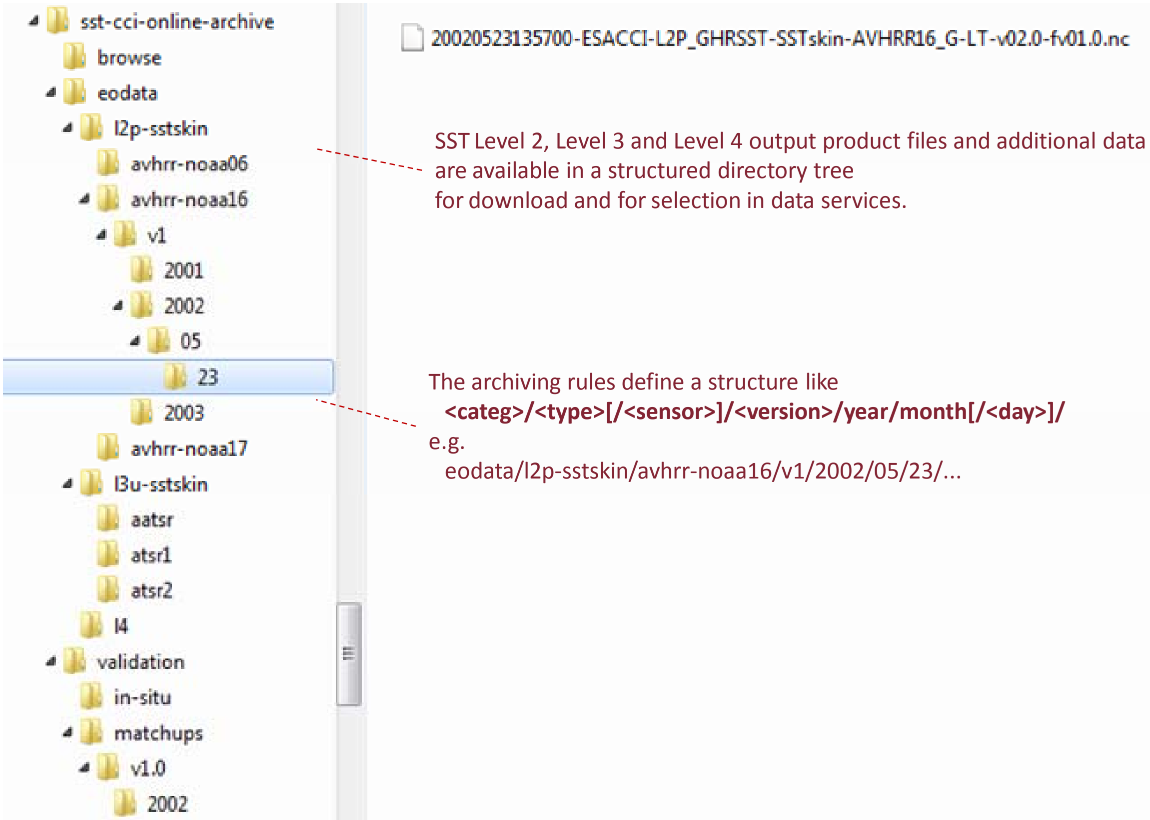


Figure 6-2: Structure of the online archive for SST CCI users

The archiving rules and thus the structure of the tree distinguishes type information, version, and time information. As long as all SST products are global, it is not required to add spatial coverage in the rules.

Besides the different output products, the directory tree also contains data for external validators: in-situ reference data, and pre-computed matchups in the MMD format. Additional categories, types and sensors are added as needed.

6.1.3 Data access to ECV products

Based on the online archive, users access data using one of the interfaces provided by the system. Besides plain FTP and HTTP, an OGC Web Map Service serves images of the data, an OPeNDAP server serves subsets, and a catalogue service (see section 6.1.4) allows search and retrieval via metadata. Finally, bulk dissemination makes available the complete CDR contribution. (→ SST-SR-6180 Output provision, SST-SR-6190 FTP access, SST-SR-6200 Web access, SST-SR-6210 OPeNDAP access, SST-SR-6220 Bulk access)

Depending on the data volume distributed and the available resources, for some of the protocols user access constraints and load-balancing mechanisms are taken into account. For the essential ways of data access, in particular FTP download and bulk data access, performance indicators for data volume and time required are collected by the respective services. The indicators can be analysed for performance monitoring.

6.1.3.1 FTP

The complete data archive of output products is accessible to the users via FTP or one of the "secure" variants like SFTP or FTPS (→ SST-SR-6190 FTP access). The user has read access to the archive of output products, structured as depicted in Figure 4-2. The output archive root-directory is a symbolic link to each user's home directory so that all users share the same directory structure.

Depending on the bandwidth available and the download traffic actually observed it may be necessary to introduce access restrictions. These can be realized by introducing user accounts, i.e. connecting the FTP server to the LDAP-based user management system, and by introducing bandwidth restrictions on anonymous access.

A common, up to date FTP server implementation is vsftpd.

6.1.3.2 WMS and OPeNDAP

Access to the data shall also be implemented using protocols that concentrate on the exploitation of the datasets, in contrast to the file-based mechanisms described in the previous subsection. Protocols implemented for this purpose are OGC WMS and OPeNDAP.

Both access protocols are supported by the open source THREDDS server, which is suggested to be employed for the SST product dissemination. This data server implementation works with the output file format of the SST CCI products directly as it builds on top of the NetCDF library. The THREDDS server can be configured to use the output data archive file structure as catalogue base structure. It follows the suggested structure described in chapter 6.1.2.

The product metadata is derived from the NetCDF files by an automated archive scanning and parsing mechanism. The archive scanning mechanism also detects new product files or the removal of files from the archive and reflects these changes immediately in the web-interface.

The user can navigate through the archive using the directory structure of the archive files the same way used for browsing through the FTP archive. The THREDDS server software supports the user by offering a simple HTTP based user interface that implements simple directory browser functionality as shown in Figure 6-3. This way, specific files in the archive can be located in a very intuitive and natural way.

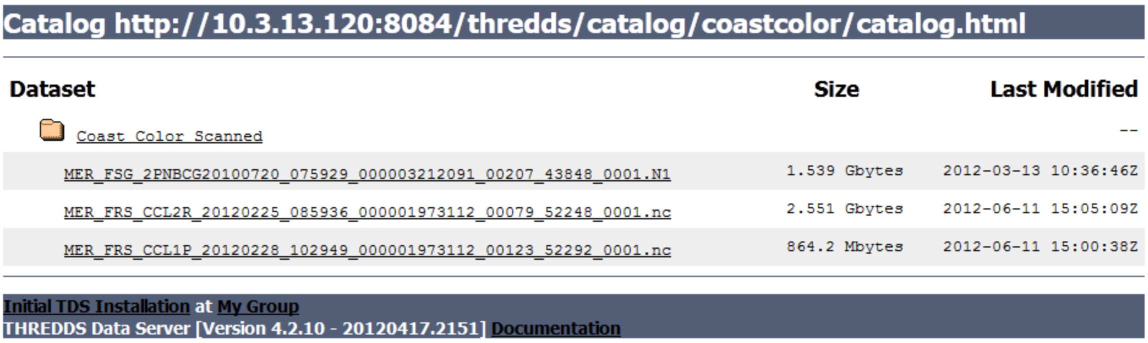


Figure 6-3: OPeNDAP user interface

Having located a file of interest using the graphical web interface the user can access this specific file using the complete set of OPeNDAP operations as defined by the specification ([RD-1]).

The same input path also serves as target for the WMS interface. This interface serves images according to the OGC specification using HTTP as protocol.

As there are no restrictions planned (yet) on the availability of these protocols, no specific login to the services is required. If it is necessary to restrict the access to these services during the operations phase, this can be implemented by using an Apache HTTP server that bridges the access to the resources. The Apache HTTP server then uses the central user management facilities to grant/deny the access to the OPeNDAP and WMS services.

6.1.3.3 Matchup data

As a service for interacting with the SST and user communities on algorithm development and validation, the user services portal also publishes matchup datasets in the MMD format (→ SST-SR-6290 External algorithm development, SST-SR-6300 Matchup datasets, SST-SR-5140 Matchup dataset processing). These pre-generated MDs contain in-situ measurements, Level 1 satellite data subscenes, auxiliary data, and optionally processed SST data.

Besides the access via FTP the Multi-sensor Matchup System (MMS) provides an interface via the portal to generate custom-made multi-sensor subsets of the MMDB, defined by spatial and temporal match-up windows, criteria for multiple matches of sensors and selection of required output variables.

Users can upload datasets in the MMD format with additional data related to matchups. The FTP-server contains a specific folder that is open for uploaded matchup data. The uploaded MMD may include additional satellite data or processed output from some algorithm for inter-comparison. There is an online check for the data provided to check the MMD format and the proper reference to existing matchups, which informs the user if format or reference problems prevent upload. In addition to this, a mandatory web/e-mail form is available to be used to inform the SST CCI science team about newly uploaded data. (→ SST-SR-3180 Ingest external MMD inputs, SST-SR-6310 Matchup inputs)

6.1.3.4 Bulk access

Special considerations need to be taken into account when a complete reprocessed SST dataset shall be made available to the climate modelling community (→ SST-SR-6220). The output dataset for the existing sensors comprises approximately 60 TB, and some reprocessing cycles may required that updates to all of this output shall be delivered to climate users. The amount of data will increase with new sensors including SLSTR, and it is expected that the online bandwidth required for this task cannot be supplied online within the user services system described above. Two alternative approaches are dissemination via the cloud and dissemination by circulating media.

Dissemination via cloud storage

The dataset can be uploaded to a commercial cloud-storage supplier that also provides the download bandwidth necessary. The benefit of this approach is that the complicated tasks of traffic balancing is delegated to a provider and that no further interaction is necessary after the data has been uploaded.

However, this approach generates considerable additional costs which are outlined in section 7.4, Cost analysis. A commercial cloud solution (Amazon S3) available at the time of investigation has been considered in this analysis. Future cloud solutions dedicated to earth observation may have different features and costs.

If the SST CCI is hosted in a shared environment that preferably also provides the input data as described in the trade-off in section 5.4.3 and 5.6 then dissemination can be avoided if the application using the data is hosted in the same shared environment. The SST CCI output data is simply made available to the project using the SST CCI data in this shared environment by proper access rights.

Hardware based dissemination

As an alternative to the approach described above, the data is copied to suitable physical media that are shipped to the users using standard parcel services. Each user receiving the set of media copies the data to local storage and, when finished, ships the media to the next user. This reduces the dissemination costs significantly, as the set of media just needs to be procured once. They can be re-used for subsequent dissemination tasks. In contrast, this dissemination method increases significantly the amount of interaction with the users, copying the data, packaging and shipping.

Another aspect to be considered for this approach is the time needed for data copying. In contrast to a distributed cloud-based solution that allows parallel data access by many users, the set of media can only be copied by one user at a time. A realistic estimation is that a modern 3 TB USB3 device can be read/written within 8 hours. If users cannot parallelize the copy operation, the raw data transfer time would then sum up to approximately 7 days.

To increase the dissemination rate, the data can be copied to more than one set of media so that a set of parallel distribution chains can be formed.

Hardware costs are assumed to be approx. 8000,- € per set of media, assuming USB hard drives with a capacity of 3 TB each, shipping costs being neglected. Again, the cost estimate is valid for the time of investigation and may change in the future.

6.1.3.5 Online tools

There are tools made available as an online service to extract data. In particular the regridding tool can be used online to create versions of the data on the fly to cope with variable resolution requirements. It extracts data on different grids on the server side avoiding unnecessary transfer of data. For more tools see section 6.1.6 below.

6.1.4 Online catalogue

The catalogue service is the metadata interface to the SST CCI system with a web-based graphical user interface for search and retrieval (→ SST-SR-6250 Catalogue). Functions of the catalogue are

- discovery of available datasets
- search using product name, spatio-temporal search criteria, version, quality and statistics search criteria, presentation of results as list, map footprints on a zoomable map, thumbnail images
- inspection of metadata and quicklook images
- collection of products in a shopping basket
- download of collected products via HTTP streaming or as package via FTP
- subsetting of outputs by the search region if the product format supports this

Note that it is essentially required that the SST CCI data holdings are discoverable by other metadata catalogs, which implies compliancy of SST CCI metadata with international metadata standards (e.g. ISO 19115 or INSPIRE) and a catalogue service capable of handling these standards.

In addition to the access functions the catalogue service registers new products. Product registration comprises input format and quality check, metadata extraction, calculation of statistical parameters, quicklook and thumbnail generation.

The following figures show examples for these functions on the basis of the MERCI catalogue system. MERCI is a basic modular catalogue implementation that covers the catalogue functions required for SST CCI. Figure 6-4 shows the catalogue search interface.

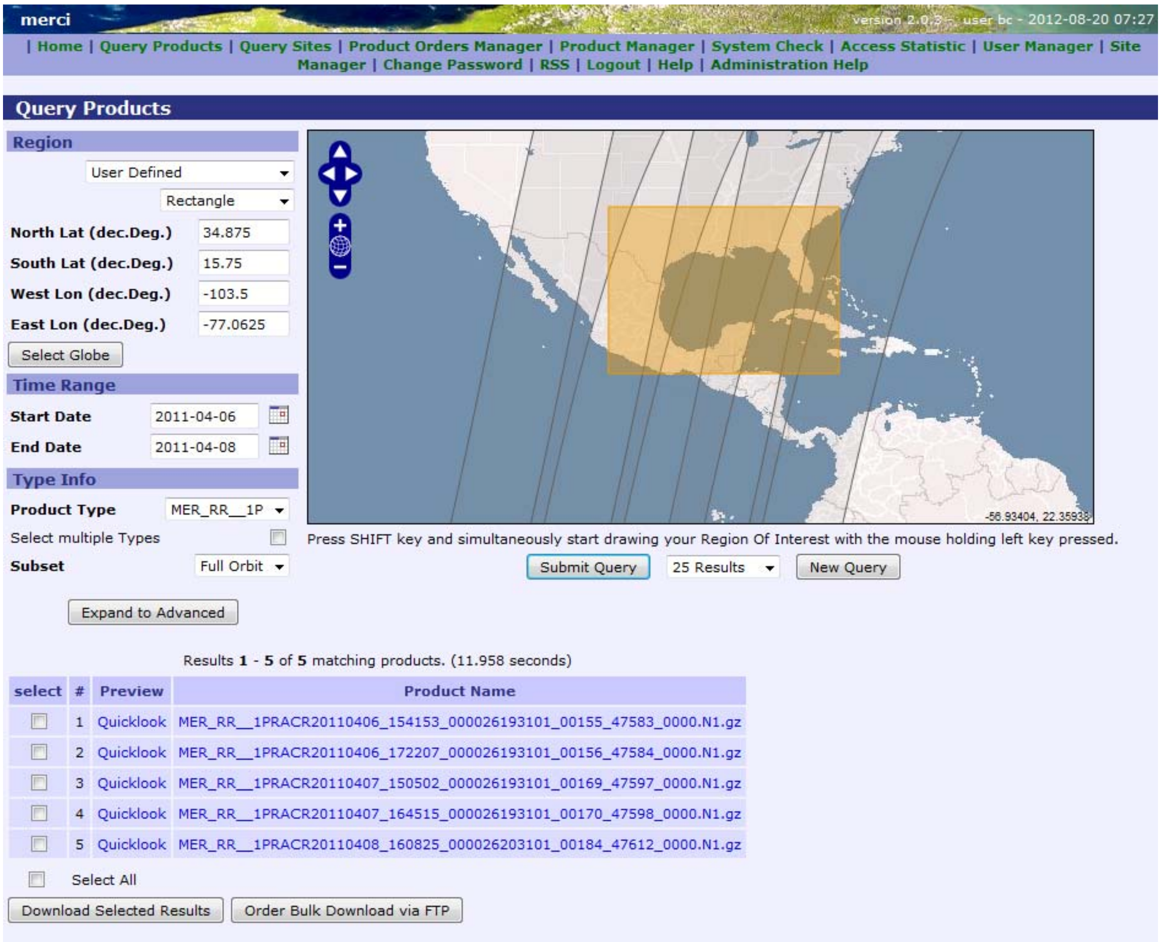


Figure 6-4: MERCI catalogue search interface

Each product registered in the system can be viewed in detail on the product detail page (Figure 6-5). This details web-page lists the metadata of the product, the quicklook image and the geo-boundary of the product on top of a map. The page also gives access to further details like the statistics, the results of the quality checks and a facility that allows generating images of each band of the product (Quicklook–on-the-fly functionality).

Products can be further inspected and downloaded using the links provided in search results and details pages. This is meant to be used for single products. Bulk data access is described in section 6.1.3.4.

The product registration function is an application that is applied to new products. Configurations and product-type-specific plug-ins identify the product, generate statistics and quicklooks, and enters the product into the catalogue database.

The catalogue supports product versions. Products are identified by name and version. The name is a logical identifier of the product usually composed of type and time information. Together with the version information it is unique. The catalogue supports search by a configured "current" version per type. Users can select a different version in the search mask.

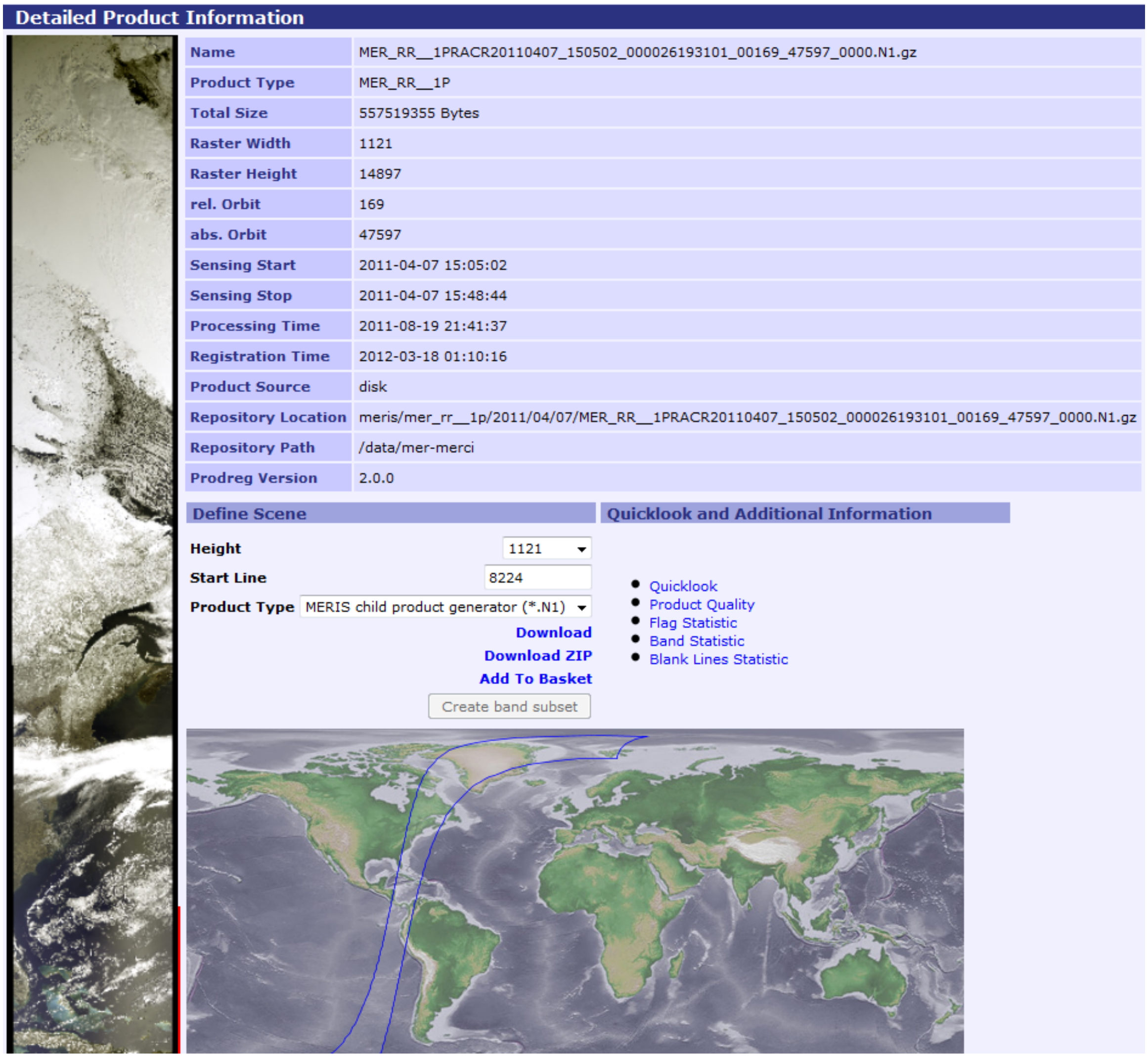


Figure 6-5: MERCI product details page

6.1.5 Web presentation and community interaction

In addition to data access user services provide information on different levels. The web site shall act as a central starting point to explore achievements, data and other resources available to the users (→ SST-SR-6240 Web site). It shall provide basic information for users not familiar with the project, in depth access to resources for the users of the data (→ SST-SR-6170 Product features) with static and dynamic content, and finally access to administrative pages and tools for the system operators.

The approach for the SST CCI system is to configure a content management system (CMS) as information service front end. The CMS proposed for SST CCI is the open-source solution Liferay. Liferay is used successfully within ESA, e.g. for the CalvalPortal (http://calvalportal.ceos.org). Liferay provides

- Separation of content and layout, corporate web site layout using CSS

- Support for authoring by separated creation and publishing, dedicated approval step
- Management of links independent from web pages, links to services and data access
- User rights management, LDAP interface (see user management below)
- Document database
- News feeds
- Forum

Functionality not provided by the CMS shall be implemented using separate, dedicated tools that are integrated into the CMS (as pluggable "portlets"). These are especially the issue tracking and the extended document management functionality.

The basic functionality of any CMS is to separate the design of the web-site from the content. This allows keeping a consistent look-and-feel that is automatically applied to all new pages added to the web-site. In Liferay, the page design can be customised, using a specific set of CSS that is integrated into the system and applied to all web-content belonging to a specific domain.

The content creation is supported from any remote computer for any user that has the necessary access rights. The CMS separates the editing and formatting process from the publishing. This simplifies the editing process as the new web-content can be created in its final external form (style, layout, color-scheme), reviewed by any other administrative user and finally published. (→ SST-SR-6330 Web site)

Beyond this functionality, Liferay includes a large number of web-applications out of the box. For SST CCI this provides a toolbox of building blocks that allow the creation of a modern web-site.

Figure 6-6 shows the BEAM forum as an example. Fully fledged forum functionality is contained in the standard Liferay distribution. It is successfully used to implement the ESA-BEAM user forum (http://www.brockmann-consult.de/cms/web/beam/forum) and the CalValPortal discussion forum (http://calvalportal.ceos.org/cvp/web/guest/forum).

Questions and issues discussed in a public forum help to spread the information to the community (→ SST-SR-6170 Product features). The forum as a web resource allows users to follow and participate to discussions about specific issues or questions. This helps in understanding in more depth the evolution of an idea, a solution of a problem. To operate a forum that is valuable to the users, it is required that the forum is managed by a skilled administrator that can both directly answer basic questions and get in contact with specialists to request answers to complex problems. A skilled administrator is the key to an informative and valuable forum. (→ SST-SR-6270 Forum or help desk, SST-SR-6360 Forum maintainer)

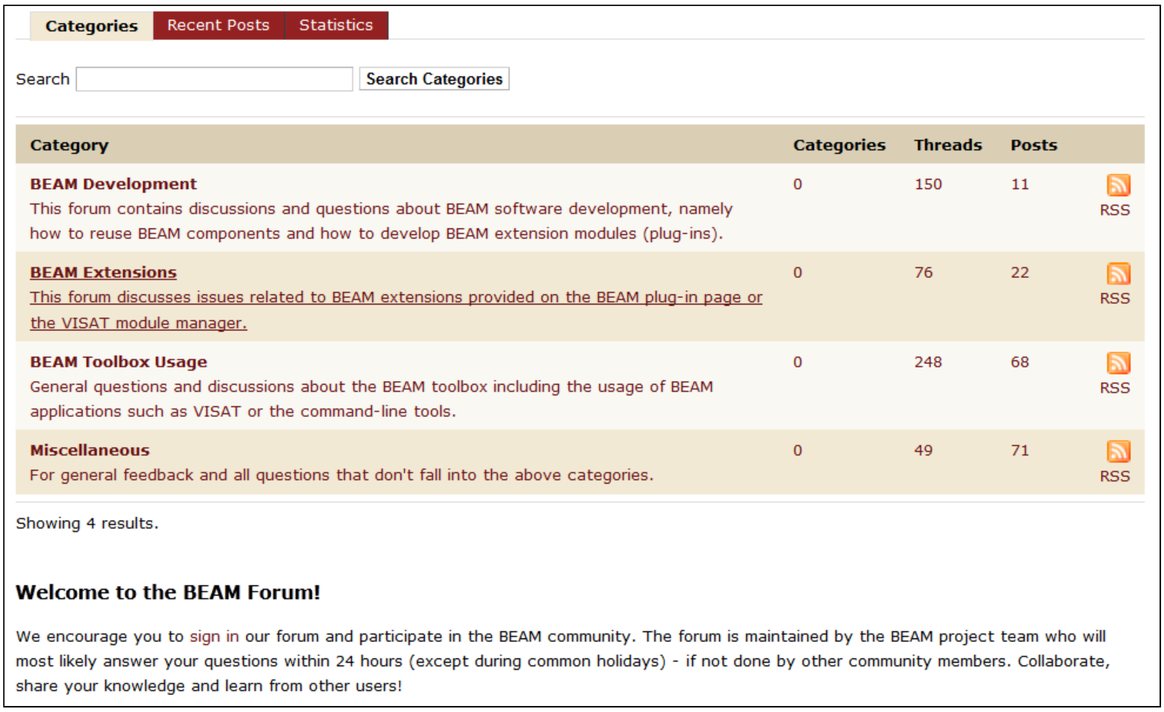


Figure 6-6: BEAM forum as example of a managed forum

In addition to the forum, the web-portal also includes an e-mail based access mechanism to the administrators. This opens a more private feedback channel, in contrast to the forum where every comment or question posted is per definition publicly visible.

Liferay includes a RSS implementation that supports ATOM 1.0, RSS 1.0 and RSS 2.0 protocols. Any announcement published in Liferay can be fed to a RSS channel. Despite this standard functionality, RSS channels can also be attached to Message Board Threads, Blog Entries and Wiki Pages. The user is offered a pre-defined RSS subscription button at several places in the portal. (→ SST-SR-6260 News feed)

Issue tracking

To track the interactions with a user that raised a problem, problems with products or software issues, the web-portal contains an issue-tracking module (see Figure 6-7) (→ SST-SR-6170 Product features, SST-SR-6340 Science issues). This approach ensures that issues and follow-up actions are tracked and assigned to a developer.

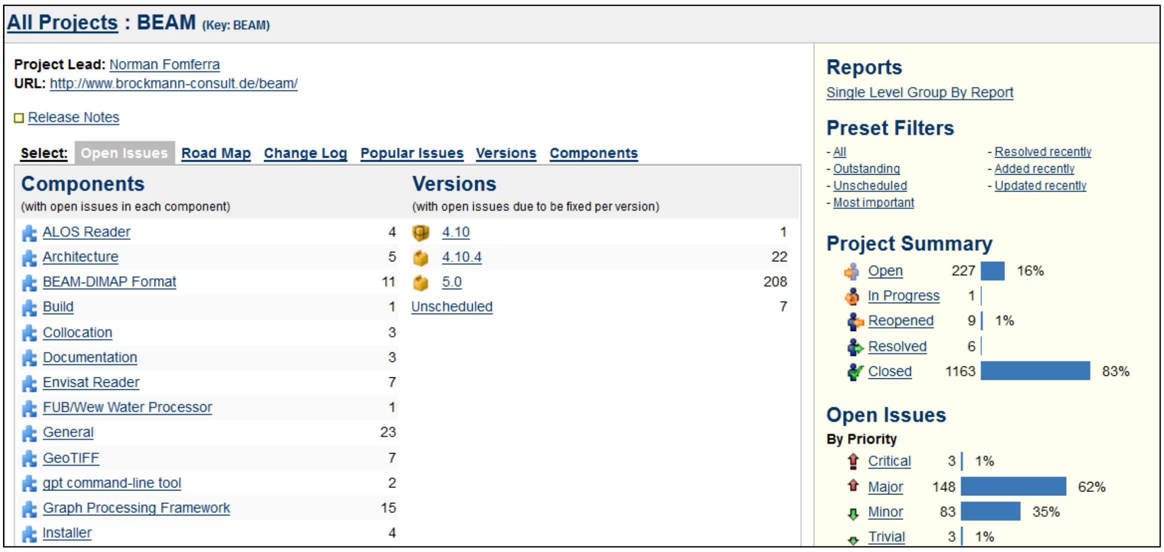


Figure 6-7: Issue tracking system

For each issue entered into the system, the issue tracker records at least

- A problem area
- Title and description
- Submitter and date of entry
- Responsible person and status information
- A history of comments

A system of scaled user rights allows editing, creation and deletion of all issues or restrictions on specific problem areas or sub-projects. The reading access to the issue tracking system shall be open to the public without any restrictions. This approach creates confidence and clarity about the SST CCI project and software status.

For the use within the SST CCI web portal, it is proposed to use Atlassian Jira as software implementing the functionality required. Although this is commercial software, Atlassian provides free software licenses for Open Source Projects. Jira fits into the context of the web portal as it allows customisation to implement a corporate design and it supports LDAP user authorisation.

Community interaction

Despite the responsibility for interaction with the users based on the services available in the web-representation (see Figure 6-1), the development team is responsible for a community process.(→ SST-SR-5110 Community process)

The development team actively contributes to the international scientific dialogue about SST variables, initiating and contributing to discussions about

- accuracy of retrieval
- strengths and weaknesses of algorithms
- calibration and validation methods
- product formats and metadata
- exploitation of the ECV dataset

For this purpose the SST CCI user services provide a platform to promote the use of the SST variables, to announce updated datasets and to obtain feedback on limitations or possible or required improvements. (→ SST-SR-6170 Product features).

Community interaction may additionally consider involving modern means of communication such as Twitter, Facebook, and Google+ as already offered on the ESA web site.

6.1.6 Access to tools, documentation and algorithm implementations

In addition to data and metadata the SST CCI project provides tools, documentation and processor implementation source code to interested users. Tools are software that can be downloaded and used locally to work with the SST CCI data. Documentation intended for users are the descriptions of the data (product user guide, validation reports including degree of compliance with URD) and of the algorithms (algorithm theoretical basis document) (→ SST-SR-6170 Product features, SST-SR-1366 Product User Guide). The product user guide also includes sample code in different programming languages of how to read SST CCI data (→ SST-SR-4130 Data access software).

Access to processor implementations is provided to meet requirements for transparency. Commented public code can be reviewed, and that operational code is available for possible improvements. This avoids the disadvantages of the previous ESA approach for IPF development with the time-consuming and inflexible detour via detailed processing model document, and full re-implementation.

There are three ways how this is made available:

- via CMS by offering issued versions of the documents and of tools for download
- in an optional document management system for the exchange, collaborative authoring and versioning of documents
- in a software repository with read access to processor and tool implementations that are public code

The tools that may be provided are (→ SST-SR-4110 Subsetting and regridding, SST-SR-4120 Visualisation and data analysis, SST-SR-1320 Trend analysis):

- Re-gridding tool to extract data on different grids (to meet a wide range of requirements) and formats (among them Obs4MIPs to facilitate model data comparisons). This tool is also available as an online service.
- Visualisation tool
- Data reading and subsetting tools
- Trend and step change analysis tools
- Regional and temporal averaging tool

These tools are offered as desktop applications suitable to execute on all major operation systems. A platform-specific standard installer software installs the tools at the user's computer.

Document management system

The document management system (DMS) supports collaborative editing and exchange of intermediate document versions for the development and operations team. This system shall allow a structured view to a complete document tree, with all the version history included. The DMS is integrated with the CMS and with the LDAP user management. Candidate implementations that allow for this are KnowledgeTree (https://www.knowledgetree.com/) or other DMS solutions.

The alternative to a DMS is a directory tree with versioned documents like the one for the data products and a rule how to add or update documents and issue new versions. What is missing in such an approach is document metadata and efficient cross-document search capabilities.

In any case, the SST CCI project and the development team benefits from shared managed documents.

Processor software repository

The software of the SST CCI processing system and the processing algorithm code is under configuration control. Section 6.3.2 Processor version concept below describes the project-internal version control approach. Part of the version control system with open source processor and tools implementations are publicly readable (→ SST-SR-4140 Open source tools). This enables confidence in the final data products as each step of the processing chain can be reproduced by any user. Additional to the raw source code, the repository should also contain the information required to build the software from the source-code. (→ SST-SR-6290 External algorithm development, SST-SR-6280 Processor repository)

The software repository contains the actual processing code and all prior versions. Whether the current development branch is made accessible is decided by the development team. All of these versions can be accessed using a simple mechanisms to choose a tag and download the source-code of the selected version.

The write access to the processor repository is restricted to the development team including or extended by the algorithm developers (if external), and the operations put configurations under version management. As all software changes are updated directly in the repository, the software changes are published almost immediately and are made available for review in a short time. (→ SST-SR-6290 External algorithm development)

The version control system proposed for the code is git. There are public code repositories like GitHub that host open and closed projects.

6.1.7 User management

User management of the SST CCI system is a service that stores information about users and their roles. Roles are related to access rights. This is useful even though the data policy of CCI is free and open access:

- to restrict administrative access to web resources, e.g. to the CMS
- to distribute the available bandwidth in a fair way to concurrent users
- to identify an author of a contribution in the forum
- to provide a drop-down area to a provider of data

Roles

The role-based access control (RBAC) of the SST CCI system is based on user roles with different responsibilities and use cases:

- Climate users access the complete dataset and updated versions of it. They are one of the main user groups. Feedback is expected from them.
- SST science community users are also interested in parts of or the complete dataset. They may be interested in the algorithms, inter-comparison with other sensors, matchups for validation. Feedback is expected from them, too.
- The interested public may be interested in the goals and achievements of SST CCI, example data products, visualisation, explanation.
- The development team comprises algorithm and system developers and validators. This role has intersecting interests with the science community, but has the mandate to push forward the system. This team uses the system for the algorithm improvement cycle. It is responsible for the data being generated, its quality, the scientific and technical documentation, and the web content.
- The operations team comprises system operators that are responsible to maintain the system and to run production and reprocessing (→ SST-SR-1390 Operator). The user forum is managed by a member of either the development team or the operations team. Questions are delegated to the experts from either of the teams.

There are several tasks related to user services and web resource that make use of user management. These tasks include

- User management itself
- Web content creation and update
- Email support
- Management of the forum
- Update of software, tools and data provided by the user services
- Issue tracking
- Delegation of in-depth questions to the appropriate specialists

While the system supports named individual users also anonymous access shall be possible. Technically, 'anonymous' is a common external user that can be used by anyone to access the system and to communicate. The only restrictions is, that if e.g. bandwidth restrictions apply, all anonymous users share a contingent. In case of many concurrent anonymous users named users have an advantage.

Registry

To allow a convenient user experience with the web-interfaces exposed by the SST CCI system, each user shall be able to use the same access credentials to log in to any of the services. This functionality is implemented using an LDAP authentication mechanism.

A central LDAP server hosts all user data required by the several system components i.e.

- user name
- password
- user role(s)
- optionally a way to contact the user, on the user's choice

Figure 6-8 shows how different front end user services access the LDAP server for authentication.

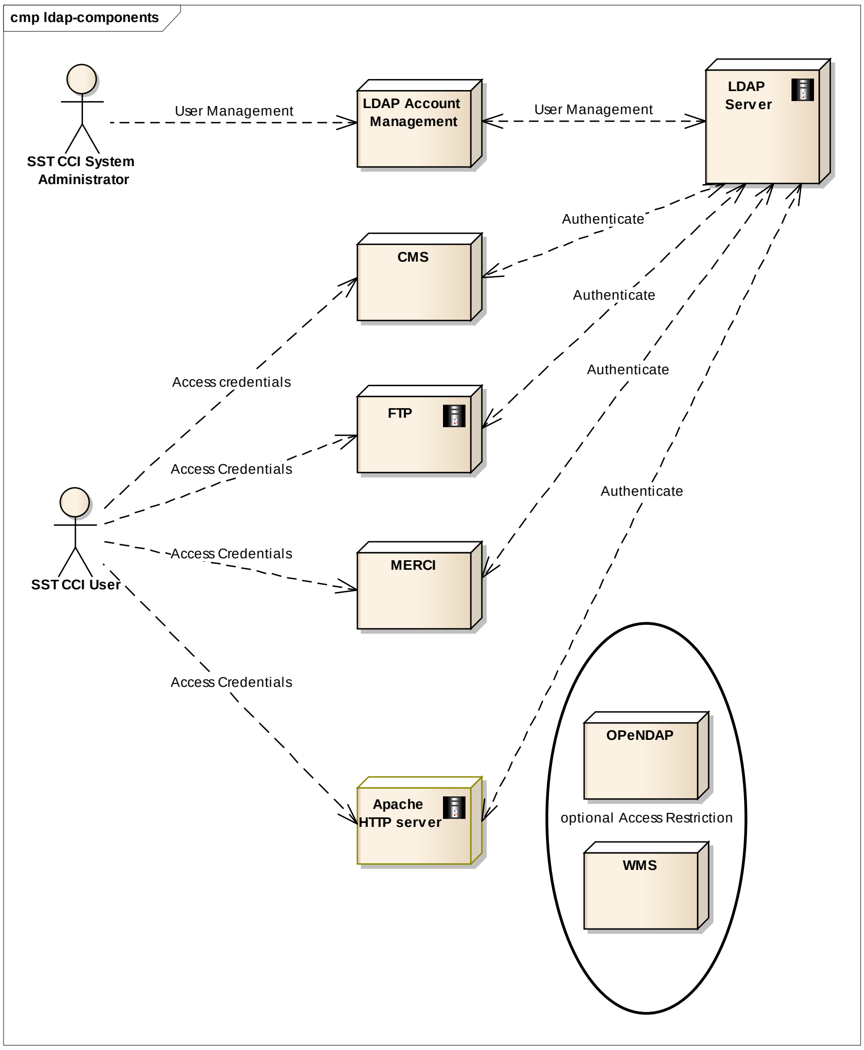


Figure 6-8: Central user management for web-resources

In addition, the resources that should initially be accessible without restrictions (WMS, OPeNDAP) can be integrated into the centralized user access rights system. In the case of extensive use of these resources, the web access can be piped through an Apache HTTP server that forwards the incoming request to the WMS/OPeNDAP implementing resources. The Apache HTTP server also connects to the LDAP server and verifies the user credentials and grants access.

The user management administration functions are realized using a LDAP Account Management utility for the operations team. This functionality is provided using a GUI based account management systems for LDAP, e.g. LDAP Account Manager (http://www.ldap-account-manager.org/) or Webmin (http://www.webmin.com/).

6.2 Production and reprocessing for consistent versions

This section defines the structures and functions that implement the operational systematic production and reprocessing in the SST CCI system. Focus of this is consistency, completeness, traceability, and efficiency of the processes. Some of the functions and structures depend on the middleware selected (see section 5.5 Processing middleware alternatives). In these cases, two variants are described, one for a Grid Engine middleware and one for a Calvalus/Hadoop middleware.

6.2.1 Components and interfaces

The decomposition shown in Figure 6-9 results from further breaking down the production and development subsystem.

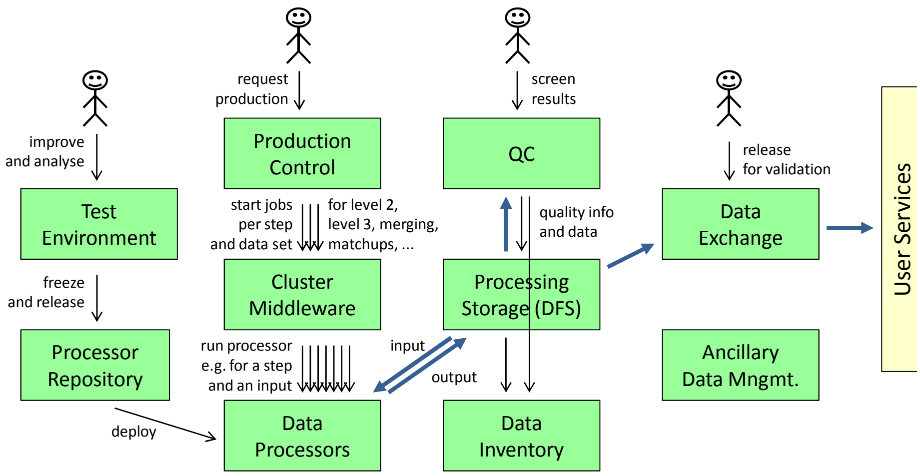


Figure 6-9: Components of the production and development subsystem involved in a workflow for SST CCI L3 and L4 processing

Table 6-2 describes the components of the production and development subsystem by their name, their purpose and function, the local data stored and managed, and the implementation approach.

Table 6-2: Components of the production and development subsystem

Component	Purpose and Function	Data	Implementation
Data processors	Generates level 2, level 3 and level 4 CCI SST products	Auxiliary data	Extensions of existing processors

Component	Purpose and Function	Data	Implementation
Processing storage	Stores input data products, intermediates and outputs as well as auxiliary data, validation data and processor software bundles in a structured directory tree, makes them available to processors (→ SST-SR-1450 Input storage size, SST-SR-1460 Output storage size)	Data product files directory tree	HDFS file system managing local disks of nodes for Calvalus/Hadoop, NFS file system and centralised fast storage for Grid Engine (→ SST-SR-1470 Online storage)
Cluster middleware	Handles processing jobs and tasks (Calvalus/Hadoop) or jobs (Grid Engine), uses configuration and plug-ins to generate tasks and to call processors (→ SST-SR-1480 Parallel processing)	job queue status of each job	Hadoop job tracker or Grid Engine job control
Production control	Handles production requests, manages workflows, manages resources processing capacity and storage space	Workflow definitions request queue status of requests	Calvalus production monitor
Data inventory	Handles product entries and collections, attributes of products like QA information, extensional collections (lists) of product entries and intentional collections (logical selection criteria like type and time)	Product entries collections	HTTP REST service, PostgreSQL database, PostGIS

Component	Purpose and Function	Data	Implementation
Ancillary data management	Systematically ingests auxiliary data from external sources, stores aux data in processing storage, triggers production waiting for consolidated aux data, implements strategies of auxiliary data selection for processors (temporal coverage, proximity)	Aux data in processing storage Ingestion configuration Trigger rules Aux data selection rules	Combination of data exchange modules, processing storage, processor wrapper plug-ins
Quality check	Checks product integrity and content with specialised data processors (with quality flags/report as output), adds quality attributes to inventory entries, computes data de-duplication in case of overlapping inputs, generates quicklooks, provides tools for systematic visual screening and for product inspection	QA working area in processing storage inventory attributes of product entries	Data product readers, tests for missing data and for geo-coding issues, ESA BEAM framework
Data exchange	Systematically ingests data from different sources, transfers outputs on release of a version or systematically, re-formats data if required, registers inputs in inventory, triggers QC and production	Ingestion configuration, transfer configuration trigger rules	Ingestion modules for different protocols

Component	Purpose and Function	Data	Implementation
Test environment	Provides sandboxes with full (read) access to input data, processor installations, tools installed, deployment tool and request client to run bulk tests via production control, and local storage for outputs	Software installations	VirtualBox or VMware virtual machines
Processor repository	Stores code of processor implementations, versioning with branches, authorship, simplified user interface to store and to retrieve versions, supports automated deployment of processor bundles	Code repository	GitHub with public repository, git software package, wrapper tools for simplified access

The components of the production subsystem interact with each other, with other subsystems and with system users and operators. For this the components provide and use interfaces. All of the components have interfaces, but some of the interfaces characterise the SST CCI system most. Figure 6-10 shows these interfaces and Table 6-3 below describes them by the interface exchange items, their content and format, and the protocol used.

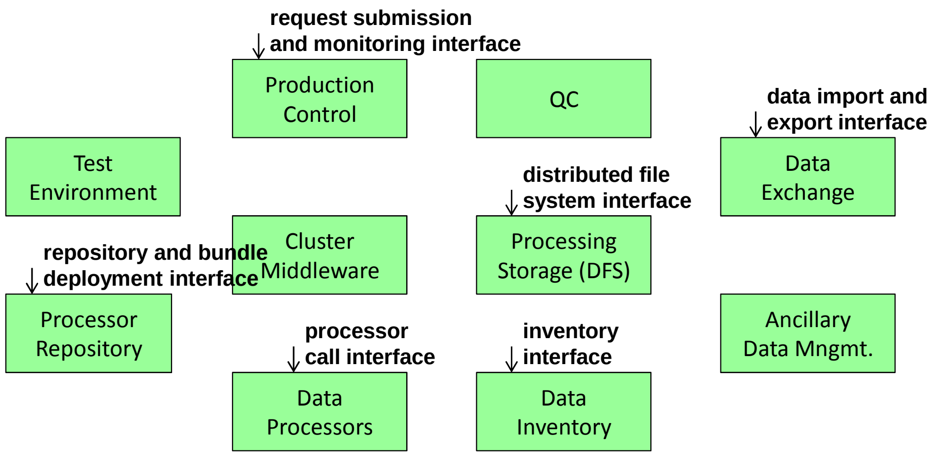


Figure 6-10: Main interfaces of the production and development subsystem

Table 6-3: Main interfaces of the production and development subsystem

Ifc ID	Interface Name	Endpoints (provider, user)	Interface items content and format	Data exchange protocol
Ifc-10.1	processor call interface(s) (→ SST-SR-1350 Processor framework, SST-SR-1360 Executable integration)	Data processors user: Production control via cluster middleware	Calls with parameters (command line), commands (interrupt) for control input data as files or streams working directory output data as files or streams progress and error messages for monitoring, return code, result specification (output product file names) log files	Executables, Unix process, inter-process communication, wrapper scripts, working directory, file interface for inputs, outputs, parameters, reports Plug-in software libraries, Unix threads, ESA BEAM Graph Processing Framework, function interface, in-memory operator chaining, tile cache, product readers and writers, can be extended
Ifc-10.2	distributed file system interface	Processing storage users: Production control, cluster middleware, QC, data exchange, ancillary data management, test environment	File system of directories and files, accessible from all nodes of the cluster File objects and data streams for Hadoop Unix file descriptors and file handles for Grid Engine	HDFS protocol for Hadoop, hadoop client for file system functions (mkdir, cd, ls, put, get, rm, cp), Java interface for data streams NFS for Grid Engine, Unix commands (mkdir, cd, ls, put, get, rm, cp), Unix file open method

Ifc ID	Interface Name	Endpoints (provider, user)	Interface items content and format	Data exchange protocol
Ifc-10.3	repository and bundle deployment interface	Processor repository users: test environment, production control	Version lists, version specification Processor source code tree, source packages Processor bundle for deployment including software libraries, bundle descriptor Auxiliary data packages associated with processors	git protocol for software repository command line and web client to upload and manage processor versions sftp (and hdfs for Hadoop) for bundle deployment
Ifc-10.4	request submission and monitoring interface (→ SST-SR-1410 System status, SST-SR-1420 Operating tool, SST-SR-1440 Cancel and resume)	Production control users: operators, test environment	Production requests in text file representation (OGC WPS XML) or as HTTP request/response HTTP URLs for status retrieval Web pages with lists of production jobs and their status, control functions	Human-machine interface: Web pages Request submission, monitoring and control command line client Machine to machine interface: (RESTful) HTTP service, OGC WPS

Ifc ID	Interface Name	Endpoints (provider, user)	Interface items content and format	Data exchange protocol
Ifc-10.5	inventory interface (within production subsystem, used by production service and by processors in wrapper scripts)	Inventory users: operators, production control, QC	Queries (for collections, with constraints on attributes (QA) and time) Collection lists Product entries with metadata	Human-machine interface: Web pages Machine-to-machine: (RESTful) HTTP service, command line client
Ifc-10.6	data import and export interface (→ SST-SR-1190 Input data interface)	Data exchange users: operators, production control	Retrieval commands Product files Transfer commands Delivery packages	sftp, scp, ftp, http media (disks, tapes)

Other interfaces exist that are used internally, e.g. the one used by the data exchange component to notify production control about new products, or the one between production control and cluster middleware to start and monitor processing jobs.

6.2.2 Structured processing storage

The structure definition for the processing storage file system is the basis for all processing functions described below. The processing file system hosts earth observation inputs, outputs, ancillary data, reference data, and also processor software bundles (see section 6.3.2 Processor version concept). This section defines the directory structure for it. The hierarchy, versioning, and naming schema - called archiving rules - identifies each item in this structure. There is exactly one place for each existing or newly created item or file. Knowing the archiving rules a function can identify its inputs and determine and distinguish locations for its outputs. (→ SST-SR-1210 Structured storage, SST-SR-1240 Output versions, SST-SR-1180 Validation datasets, SST-SR-5275 Internal version storage, SST-SR-5310 Measures for stability)

(Note that the structure proposed here is just a template and that concrete implementations may deviate from it, especially by introducing more layers or different orderings of the directory hierarchy. The principles of archiving rules and how functions use them remain the same.)

Figure 6-11 shows the top level directories of the processing storage of SST with earth observation data, browse images, ancillary data, processor software bundles, reference data, and a user space.

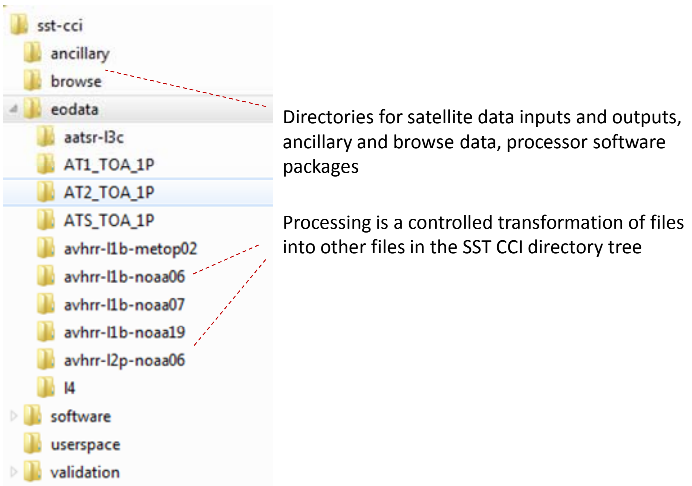


Figure 6-11: File system structure for SST CCI

The archiving rule for earth observation data is

<archive-root>/**eodata**/<type>/<version>[/<region>]/<year>[/<month>[/<day>]]/<file>

Example:

/sst-cci/eodata/
ATS_TOA_1P/r03/2009/06/01/ATS_TOA_1PRUPA20090601_004927_000065272079_00288_37917_0115.N1

There are types for all sensor and level combinations used in SST CCI. This way, a processor may read from /sst-cci/eodata/avhrr-l1b-noaa06/ and write to /sst-cci/eodata/avhrr-l2p-noaa06/ . For new inputs or new types of outputs directories can be added below /sst-cci/eodata/ .

The archiving rules for the other categories are

<archive-root>/**browse**/<type>/<version>[/<region>]/<year>[/<month>[/<day>]]/<file>

<archive-root>/**auxiliary**/<type>/<version>/<year>[/<month>]/<file>

<archive-root>/**validation**/<type>/<version>[/<region>]/<year>[/<month>[/<day>]]/<file>

<archive-root>/**software**/<type>/<package>**-**<version>.tar.gz

<archive-root>/**userspace**/<user>/

Examples:

/sst-cci/browse/ATS_TOA_1P/r03/2009/06/01/ATS_TOA_1PRUPA20090601_004927_000065272079_00288_37917_0115.jpeg

/sst-cci/auxiliary/clavrs-cld-noaa18/1.0/2009/06/NSS.GHRR.NN.D09180.S0236.E0425.B2116364.SV.cmr.h5

/sst-cci/reference/drifters/1.0/2007/insitu_WMOID_11931_20070820_20080402.nc

/sst-cci/software/arc/arc-1.0.tar.gz

Implementation hints:

- In case components of the processing environment are not collocated several instances of the storage structure exist. Each of them may contain only parts of the data types. Replication between them transfers the outputs of one component to the storage instance accessible by components that use these outputs. Replication can be implemented by rsync using ssh public key authentication.
- For Calvalus/Hadoop with distributed local storage the structure is instantiated in HDFS with optional replication.
- For Grid Engine middleware environments with hierarchical storage the structure is instantiated in background storage, with parts actually used staged to fast network storage accessible by processing nodes.

6.2.3 Input ingestion

This function is required to initially stage satellite input data and ancillary data from the long-term archive or an external source to the processing storage, and also to systematically retrieve and ingest newly acquired data and corresponding ancillary data and in-situ reference data from an external provider for the continuous extension of the CDR (→ SST-SR-1140 Sentinel inputs, SST-SR-1150 VIIRS inputs, SST-SR-1170 Auxiliary inputs, SST-SR-1190 Input data interface, SST-SR-2110 Input ingestion, SST-SR-2120 Auxiliary update, SST-SR-2125 Validation data update, SST-SR-5160 Other sensor's inputs, SST-SR-5170 New input type, SST-SR-5180 Other ECVs as ancillary, SST-SR-5190 New ancillary type). Figure 6-12 shows the function and the elements involved.

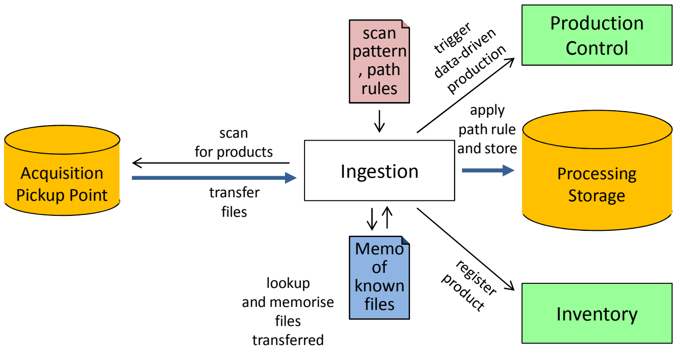


Figure 6-12: Ingestion function to pull input data from a pickup point of a data provider

- The ingestion function accesses a local or remote pickup point and scans it for new files.
- A local pickup point can be used to scan media for ingestion.
- The scan descends a directory tree and uses filters to detect data products.
- The function repeatedly scans the pickup point using a memory of known files to detect new ones for data-driven processing. Errors are handled by retry and reporting (→ SST-SR-2170 Online processing error handling).
- The function uses path rules and version information to store the products in the processing storage directory structure. (→ SST-SR-1220 Input versions, SST-SR-1470 Online storage)
- The function initiates registration of the product in the inventory. (→ SST-SR-2160 Online processing status)
- The function triggers data-driven production of the newly ingested product. Also quality checks for new products are initiated this way. If desired, input data is reformatted, e.g. to transform Sentinel or VIIRS inputs into the format and bands of their predecessors for data reduction. (→ SST-SR-2180 Short-delay processing, SST-SR-1220 Input versions, SST-SR-1230 Reprocessing input versions, SST-SR-1160 Sentinel data reduction, SST-SR-1165 VIIRS data reduction, SST-SR-2170 Online processing error handling)

Implementation hints:

- For Calvalus/Hadoop with distributed local storage ingestion uses the HDFS interface to store files in the processing storage. The interface controls replication and block size of the files.

The ingestion function shall be implemented by the Data Exchange component.

6.2.4 Data processors

The data processors of the CCI SST system convert input products into higher level products. They are versioned modules plugged into processing chains. The processors included are:

- ARC-CCI SST retrieval processor

- DV model processor (also does depth adjustments)
- PMW conversion processor
- Level 3 aggregation processor
- Level 4 analysis processor

The processors implement the methods described in the ATBD and generate the outputs described in the PSD. (→ SST-SR-1250 SST retrieval, SST-SR-1260 Error characterisation, SST-SR-1270 Projection and compositing, SST-SR-1280 Sensor merging, SST-SR-1290 Output products) The MMS is not considered a data processor but a component on its own. It is described in section 6.3.5.

6.2.5 Production control

Production control is the function that initiates and controls data processing activities of the system. The approach for production control is to use some generic cluster middleware but to augment it with the

- handling of processing workflows
- bulk production and managed data-driven processing
- constraint handling like the availability of auxiliary data to be waited for
- integration of manual activities like quality checks
- resource management

The production control function shall be implemented by services that the operator interacts with (→ SST-SR-1390 Operator, SST-SR-1400 Automation) and that allows for adaptations of workflows (→ SST-SR-5170 New input type, SST-SR-5200 New processing chain. Figure 6-13 shows the hierarchy of services of production control including the cluster middleware and the data processors. The granularity of tasks is an example of how a bulk request is broken down into smaller jobs and processing tasks that together generate the desired outputs.

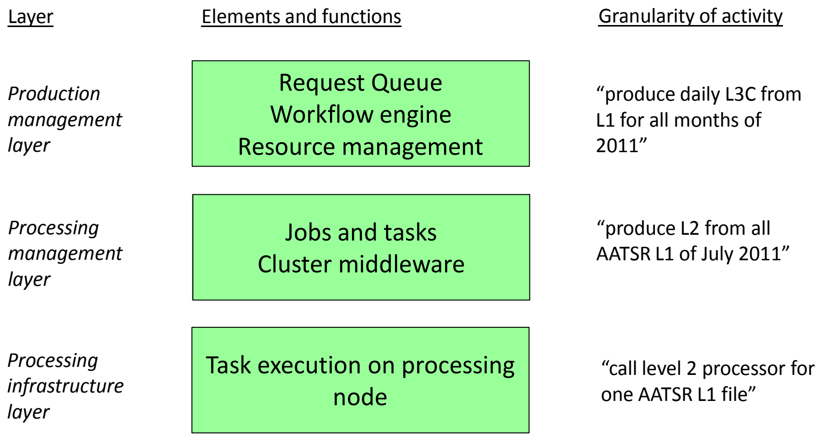


Figure 6-13: Service hierarchy and granularity of control at the different levels

Figure 6-14 shows the SST CCI processing workflow with the logical dependency graph of steps and the corresponding data flow (→ SST-SR-1370 Processing chain).

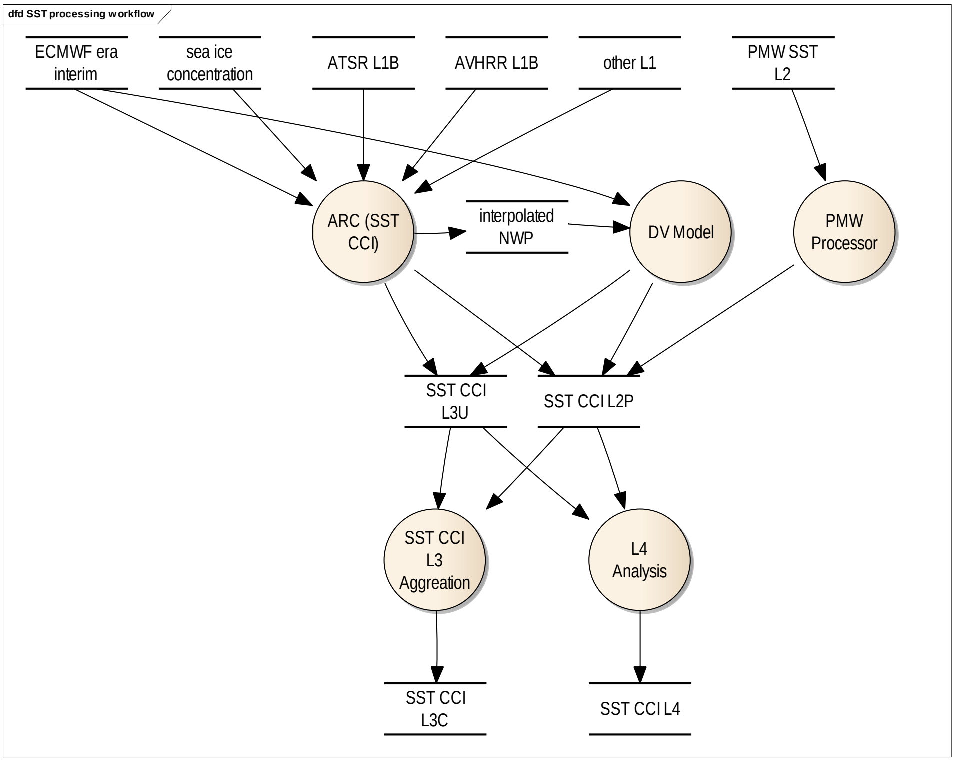


Figure 6-14: SST processing workflow from level 1 and level 2 satellite data to level 4 analysis products

- The ARC processor in its SST CCI configuration retrieves SST from AVHRR (METOP and NOAA AVHRR GAC) and ATSR Level 1 products and generates L2P and L3U products according to the PSD [AD 5]. It is foreseen to extend ARC to new level 1 inputs from SLSTR and VIIRS. (→ SST-SR-6110 L2P outputs, SST-SR-6130 L3U outputs, SST-SR-1110 ATSR Inputs, SST-SR-1120 AVHRR GAC inputs, SST-SR-1130 AVHRR METOP inputs, SST-SR-6160 Output format and naming, SST-SR-1140 Sentinel inputs, SST-SR-1150 VIIRS inputs, SST-SR-1250 SST retrieval, SST-SR-1260 Error characterisation, SST-SR-1290 Output products)
- The Diurnal Variability Model adds corresponding information to the L2P and L3U products. (→ SST-SR-6150 L4 diurnal outputs). It may also be used for PMW in the future.
- The PMW processor converts microwave products into the SST format, optionally adding bias adjustments and uncertainty/quality information (→ SST-SR-1135 PMW inputs)
- The SST CCI L3 aggregation tool generates daily composites. (→ SST-SR-6120 L3C outputs, SST-SR-1270 Projection and compositing, SST-SR-1290 Output products)
- The L4 Analysis system generates the SST CCI level 4 products according to the PSD [AD 5]. (→ SST-SR-6140 L4 outputs, SST-SR-6150 L4 diurnal outputs, SST-SR-6160 Output format and naming, SST-SR-1280 Sensor merging, SST-SR-1290 Output products)

Step definitions and constraints

Production control manages this workflow by step definitions for sets of products, and by constraints in form of preconditions and postconditions. The constraints ensure that dependencies between steps are obeyed and that all steps are performed for a product file.

Step	**ARC step**
Description	NWP interpolation, classification and SST retrieval from IR imagery
Parameters	Time period to be processed, e.g. a month Sensor to be processed, AVHRR or ATSR
Preconditions	L1B sensor products of time period are available/staged
Postconditions	L2P products for sensor and time period are available L3U products for sensor and time period are available
Auxiliary	ECMWF era interim adjacent to acquisition time of product file sea ice concentration of day of acquisition time

Step	**DV Model step**
Description	Refinement of SST (time and depth standardisation)
Parameters	Time period to be processed, e.g. a month Sensor to be processed, any IR imagers, potentially PMW also
Preconditions	L2P products for sensor and time period are available L3U products for sensor and time period are available
Postconditions	L2P products with DV for sensor and time period are available L3U products with DV for sensor and time period are available
Auxiliary	-

Step	**PMW step**
Description	Format conversion of PMW data into CCI format, adding adjustments and uncertainty information
Parameters	Time period to be processed, e.g. a month Sensor to be processed
Preconditions	PMW L2 products of sensor and time period are available/staged

Postconditions	L2P products for PMW sensor and time period are available
Auxiliary	-

Step	**Level 3 aggregation step**
Description	Generation of daily aggregates with uncertainty propagation
Parameters	Time period to be aggregated, e.g. 24 hours, day and night separated Sensor to be processed, AVHRR or ATSR
Preconditions	L2P products (with DV) for ATSR and time period are available L3U products (with DV) for ATSR and time period are available
Postconditions	L3C products for sensor and time period are available
Auxiliary	-

Step	**Level 4 analysis step**
Description	Generation of daily analysis products
Parameters	Time period to be aggregated, e.g. a day
Preconditions	L2P products (with DV) for AVHRR and time period are available L2P products (with DV) for ATSR and time period are available L3U products (with DV) for AVHRR and time period are available L3U products (with DV) for ATSR and time period are available
Postconditions	L4 products for time period are available
Auxiliary	-

Note that for aggregation steps there is an input set and usually a single output product while for the level 2 processing steps there is a one-to-one relation between inputs and outputs.

Bulk production and reprocessing

For *bulk production* of a period of several years processing is controlled in chunks of months (→ SST-SR-1380 Bulk reprocessing). For each month of the period:

- L1B products of AVHRR, ATSR and the PMW sensors are made available/staged if necessary in order to fulfil initial preconditions.

- The steps for all months of the periods and for the different sensors are generated.

After step generation, steps are executed. Step execution manages still relatively complex tasks. A step is applied to a set of inputs, executing processors for every single product of the set/month. In case of aggregation a step is applied for an aggregation period and a corresponding group of products.

- All steps with preconditions fulfilled are executed. Successful execution leads to fulfilled postconditions that trigger the execution of subsequent steps.
- Concurrency is limited by the available resources and by the constraints of pre- and postconditions.

Production on request

For *production on request*, the set of inputs, the period, and the sensors are defined in a production request. The request may also select a subset of the workflow to be executed. Else, the process is similar to bulk production. For all inputs to be processed or for the respective period (or region):

- L1B products of AVHRR, ATSR or the PMW sensors are made available/staged if necessary in order to fulfil initial preconditions.
- The steps for all months of the period and for the different sensors are generated.
- Intermediate and output products are optionally stored in the user space of the product directory tree.

Data-driven production

For *data driven production* an ingestion process is configured that ingests new input data and triggers production (→ SST-SR-2130 Data-driven processing, SST-SR-2150 CDR extension, SST-SR-2180 Short-delay processing). For each new input product

- the steps for this product are generated
- the precondition for this product is fulfilled by providing the product in the processing directory structure
- the steps are executed obeying their pre-conditions
- If the production depends on certain auxiliary data that may be available only after a while then the required auxiliary data is formalised as precondition, and auxiliary ingestion triggers fulfilment of corresponding preconditions for data-driven processing steps. (→ SST-SR-2140 Wait for ancillary, SST-SR-2180 Short-delay processing)
- Aggregating steps (level 3 and level 4) are generated separately on a delayed execution basis, e.g. controlled by a timer.

Production resource management and queuing

As described in the previous paragraphs, configuration, a request or an emerging new product initiates production. For the production control function we generalise the concept of the production request in such a way that there is a (set of) request(s) for bulk production, and that also data-driven production generates requests with every new product to accompany and control its workflow through the system. The production request specifies what shall be processed with respect to (input) product set and workflow. (Often, the product set is intentionally specified by a time interval to be processed.)

Requests are persistent representations of what the system shall do, is doing, and has done. They have a status. Once generated, they are queued, prioritised, planned, and processed by the derivation of steps according to the workflow a request refers to. Production control manages the queue of requests, controls processing of the steps that have been derived for each of them, and updates the state of requests.

There are different representations of requests that are implementation dependent. Among them are an internal format defined by a database schema and an external format defined by an XML schema. As a minimum, a request comprises the following fields:

Field	Description (or by the ingestion system in case of data-driven production)
identifier	a unique ID and a name provided by the user/operator (or by the ingestion system in case of data-driven production)
workflow	type of request, name of workflow (which defines the steps to be executed and the granularity of chunks of the input product set)
input specification	Intentional or extensional description of the data to be processed, a set of years in the simplest case of a bulk request (if the workflow defines the types of input products), or a path to an input product (in case of data-driven production)
status	The complex status of the steps derived for the request, the progress of processing, and a summary status describing whether the request is being processed, failed, or succeeded.

When processing such requests the processing system of SST CCI has resources that are limited. The infrastructure resources considered are

- processing time, CPUs
- main memory
- storage space, disk usage for inputs and outputs
- queue entries in middleware systems

Production control monitors the use of some of these resources while using the middleware to monitor and control the rest.

- For Calvalus/Hadoop Hadoop dispatches tasks to hosts and thus manages the use of CPUs. Main memory currently is not explicitly managed by Hadoop. Restricting the number of tasks per host leads to a similar effect.

- For Grid Engine the number of slots per host as well as the size of jobs regarding to their main memory demand can be criteria to restrict concurrency.

Production control restricts step execution according to availability of resources it monitors. These are at least the number of steps of a certain type to be executed concurrently on the middleware. Example: The restriction may be to run at most 4 concurrent ARC processing steps. Note that each step may process a complete month of inputs, resulting in the concurrent processing of four months of input data. The restriction leads to an almost ordered sequence of processing large requests. This simplifies progress monitoring.

In addition, the management of storage space is performed by the production control function. For a request the space required is estimated on the basis of the input product set and the workflow selected. Intermediate results and optionally inputs not used are deleted from processing storage in order to re-use the space for the current processing. (→ SST-SR-5265 Version management)

Auxiliary data handling

Auxiliary data are stored in the processing directory structure to make it available to processing steps. For a certain processing step and a type of auxiliary data required there is a selection rule to determine the auxiliary data product required. Often, the rule is based on acquisition time. Rules required for SST are:

- Sea ice data are available in a granularity of daily product files. The ARC processing step selects sea ice data using the acquisition start time of the product to be processed. The sea ice product is selected by **temporal coverage**. It temporarily covers the acquisition start time of the product.
- NWP data are available in a resolution of 6 hours. The ARC processing step selects NWP data using the acquisition start time of the product to be processed. The two NWP data products are selected by **proximity**. The NWP data product closest before and closest after the acquisition start time is selected for interpolation. The interpolation itself is done as part of the ARC processing step. Implementation hint: CDO operators interpolate ECMWF NWP data.

Note that considering NWP interpolation as part of the ARC processing step is a design decision. It means that the interpolated NWP is not stored and not reused. To change this, NWP interpolation would have to be modelled as a step on its own.

Exception handling

The production control function detects failure by monitoring and because processors report success or failure at the end of processing (→ SST-SR-1430 Error handling, SST-SR-2170 Online processing error handling). Two main categories of failure can be distinguished: system failure and processing failure. Processing failure may be caused by input data considered invalid or by unexpected exceptions causing the processor to fail.

- Resuming production after the system is repaired can in many cases cure failure caused by the system. The steps that ran at the time of the failure are repeated. To enable this the system keeps track of all steps for a bulk request that are successfully executed. The resume function skips the steps already done and restarts those that had been interrupted or that are next in the workflow. Here, the granularity of steps and optionally a means within a step to detect partial results determine the overhead caused by a failure.

In case of Calvalus/Hadoop partial system failure of a single node results in automated retry of the task on a different processing node. In case of success no further recovery is needed and the failure is transparent to production control. Similar to this if a disk fails in the HDFS of Calvalus/Hadoop a different node transparently serves the product. The under-replication of the products on this disk is repaired automatically, and the erroneous disk should be replaced.

- Failure to process a certain input product with a certain processor results in a more or less qualified exception report by the processor. Production control records the failure and continues production only of those steps that do not depend on the failed step. The input conditions of subsequent steps ensure this as they do not get fulfilled. Without modifications repetition usually is not helpful in case of processing failure. An operator may provide a correct input or update parameters before repeating. The operator may also decide to skip the product that has caused the failure such that subsequent aggregation steps do no longer depend on this input. This is handled by production control by updating the set of products in the pre-condition of the aggregation step.

Retry may be initiated automatically with a limitation of the number of cycles. Or it may be operator-driven by a command.

Calvalus/Hadoop comes with a feature to allow a percentage of failures without interrupting the workflow. This automation helps in cases with infrequent processing errors caused by single erroneous input products.

Manual steps and dynamic decisions

The degree of automation of a workflow is an operational decision (→ SST-SR-1400 Automation). Operators may want to automate to the maximum extent, or they may control certain parts of the workflow on their own. Typical manual steps in a workflow are quality checks by visual screening of input or output data. The continuation of the workflow may depend on the result of such a step.

Operators use tools to perform manual steps. The operator gets informed about what shall be "processed" manually and provides feedback to the system regarding the result. One form of implementation of this is to separate the workflow into the part before the manual step and the parts after it. When the first part terminates the manual step (quality check) is performed. In this step the operator modifies the data, e.g. by sorting into good and bad products. Then the operator initiates the second part of the workflow by a new request. The production control function immediately supports this.

Another form of implementation is to provide the data to be manually handled in a certain area for an operator and to accept feedback via a tool. The implementation of this feedback may use the constraints and set pre-conditions in order to initiate continuation of processing. To sort products into good and bad can be implemented the same way as exceptions are handled. Failed quality checks leads to a failed step or a step failed for certain inputs. Production for the "good" products can be continued.

If this is not sufficient because the workflow depends on the result of a (manual) step then step generation for the continuation of the workflow has to be delayed. The production control function supports this by dynamic generation of steps.

6.2.6 Data product quality checks

The quality of the SST output products obviously depends on the quality of the input data. Because the SST workflow contains an aggregation of inputs a single corrupted input that remains undetected can compromise a daily composite. The quality check function supports automated and operator-performed quality checks and the integration of their results into the SST workflow. (→ SST-SR-1300 Quality check, SST-SR-6180 Output provision).

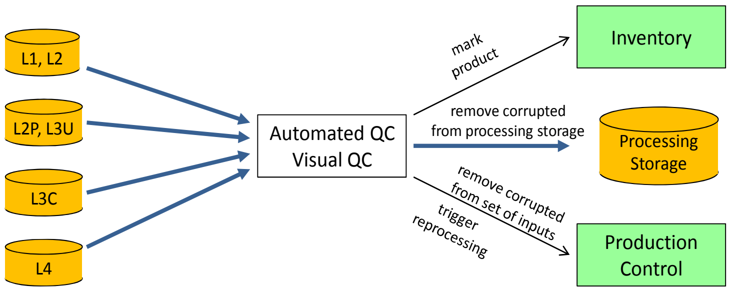


Figure 6-15: Quality check of inputs, intermediates and outputs

- First, all input products are screened automatically for product consistency (format, file size, geo-coding, data content). Optionally, all input products are converted to quicklooks for visual screening. Corrupted products are marked in the inventory and they are removed from processing storage.
- Second, the L2P and L3U products are optionally screened before they are used for L3 or L4 aggregation. Again, corrupted products are removed from processing storage and from the set of inputs for L3 and L4 processing in the production workflow.
- Third, the L3 and L4 output products are quality checked and validated before release.

For data-driven short-delay processing quality checks can be postponed to after generation of the products. In case QC detects issues products are retracted, reprocessed, and replaced if necessary.

The collaboration diagram in Figure 6-16 finally shows how Quality Check is integrated into its context by data and control flows.

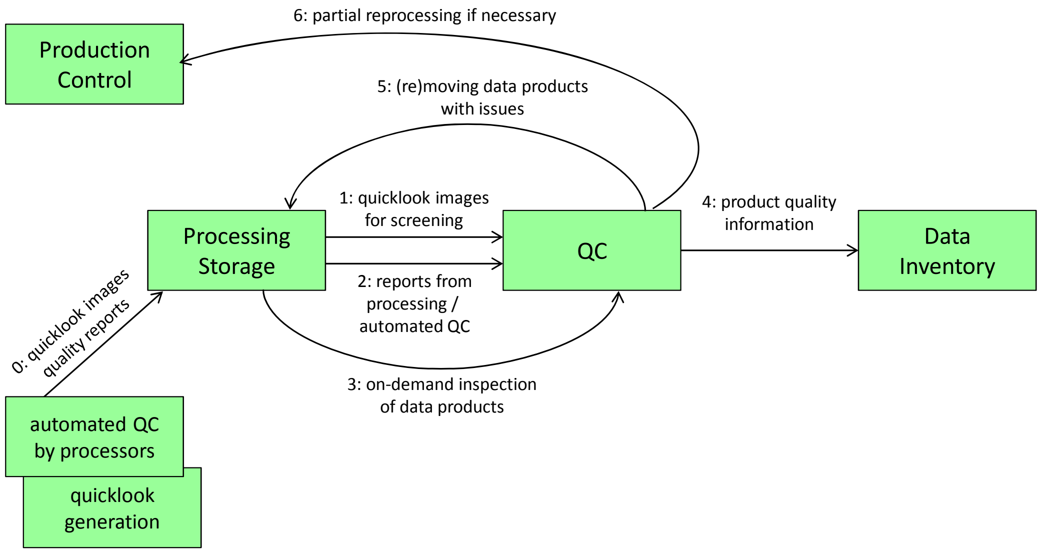


Figure 6-16: Quality check and its interfaces

- QC retrieves new quicklook images and quality reports from automated QC processors via processing storage.
- Operators inspect data products from processing storage if necessary. Product quality information is updated in the inventory. This includes data quality metrics that can be used as performance indicators by analysing inventory content.
- In case of quality issues partial reprocessing may be triggered or the products with quality issues are removed from the collection.

6.3 Continuous algorithm improvement

This section defines the structures and functions that extend the production and reprocessing environment for continuous improvement. Focus is on flexibility, rapid testing and prototyping, and an interface to full mission data and reprocessing capabilities also for test and validation. The concepts described are processors, versioning, a test environment, and the multi-sensor matchup system MMS. The concept of processors and versioning contribute to the modularity of the system. Within SST CCI they are most relevant for algorithm developers and validators.

6.3.1 Processor interfaces

Data processors are one of the means for modularisation in earth observation processing infrastructures. A processor is a software component that can be parameterised and that generates usually one higher level output product of a certain type from one or several input products of certain types in one call of the processor.

The SST DPM [AD 8] defines three processors for data transfer and seven processors for data transformation, among them the CCI ARC processor, the CMS OSI-SAF processor, and the OSTIA L4 analysis processor. They come with individual interfaces. As it is one of the goals of the project to simplify the transfer to operations of science code, an extensive standardisation of processor interfaces like the ESA IPF interface is not required in preference for the approach to wrap the existing processors into adapters. For this to work, processors have to provide as a minimum:

- parameterisation - parameters to specify inputs and other parameters
- data access - read access to inputs and auxiliary data, write access to outputs and optional runtime space (working directory),
- feedback - intermediate status, success or failure, identification of the result(s)
- packaging and versioning - structured package of software, identified by name and version
- robustness - science processors have to pass certain verification tests in order to ensure that they are robust enough to be wrapped into adapters (e.g. science code shall exit gracefully on error, shall be multi-processor capable, shall not produce memory leaks, etc.)

Depending on the processing middleware used the processor wrapper is an adapter between the middleware task/job interface and the respective processor (→ SST-SR-1350 Processor framework, SST-SR-1360 Executable integration).

- In a Grid Engine environment 'start' and 'run' scripts care for job generation and job execution with processor calls. Processors are executables installed in the shared network file system.
- In a Calvalus/Hadoop environment and in case of a processor provided as Unix executable a 'call' script cares for input provision and output archiving. Processors are software packages automatically deployed on demand.
- A processor implemented in Java may either implement the BEAM operator interface or the Hadoop Mapper interface to be executed without additional adapter in a Calvalus/Hadoop environment. Processors are Java software library packages automatically deployed on demand.

Unix executables use a file interface to access inputs and write outputs. Parameters are transmitted as command line arguments, as environment variables, or in a parameter file. Unix executables can provide feedback with the return code and with messages on stdout/stderr and in log files. Figure 6-17 and Figure 6-18 show processor wrapper scripts for a Grid Engine environment and for a Calvalus/Hadoop environment respectively. Both of them provide the environment for the processor call and call it with the parameters it requires.

```
#!/bin/bash

# nwparc-run.sh
# call pattern: nwparc-run.sh <year> <month> <day> <sensor> <input>

. mymms
. $MMS_HOME/bin/mms-env.sh

year=$1
month=$2
day=$3
sensor=$4
input=$5

echo "`date -u +%Y%m%d-%H%M%S` nwparc $sensor $input..."

export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:$MMS_CDO/lib
export PATH=${PATH}:$MMS_CDO/bin

startTime=${year${month}${day}000000
stopTime=${year${month}${day}235959

# provide working dir

wd=$MMS_TEMP/arc3-$year-$month-$day-$sensor
mkdir -p $wd
cd $wd

# run CDOs for NWP interpolation

if ! $MMS_HOME/bin/nwp-tool.sh false $sensor $pattern \
    $input \
    $MMS_ARCHIVE/ecmwf-era-interim/v01 \
    $sensor-nwp-$startTime-$stopTime.nc

then
    echo "`date -u +%Y%m%d-%H%M%S` nwp $sensor $input ... failed"
    exit 1
fi

# run ARC

arc3auxfiles=`find $MMS_ARC3 -type f | grep -v 'dat/'`
ln -f $arc3auxfiles .
mkdir -p dat
ln -f $MMS_ARC3/dat/* dat

case $sensor in
    atsr.1) inp=MMD_ATSR1.inp ;;
    atsr.2) inp=MMD_ATSR2.inp ;;
    atsr.3) inp=MMD_AATSR.inp ;;
    avhrr.n10) inp=MMD_NOAA10.inp ;;
    avhrr.n11) inp=MMD_NOAA11.inp ;;
    avhrr.n12) inp=MMD_NOAA12.inp ;;
    avhrr.n14) inp=MMD_NOAA14.inp ;;
    avhrr.n15) inp=MMD_NOAA15.inp ;;
    avhrr.n16) inp=MMD_NOAA16.inp ;;
    avhrr.n17) inp=MMD_NOAA17.inp ;;
    avhrr.n18) inp=MMD_NOAA18.inp ;;
    avhrr.n19) inp=MMD_NOAA19.inp ;;
    avhrr.m02) inp=MMD_METOP02.inp ;;
    *) exit 1 ;;
esac

if ! ./CCI_ARC_Linux $inp $sensor-nwp-$startTime-$stopTime.nc $input \
     ${input%.nc}-arc3.nc
then
    echo "`date -u +%Y%m%d-%H%M%S` nwparc $sensor $input ... failed"
    exit 1
fi

echo "`date -u +%Y%m%d-%H%M%S` nwparc $sensor $input ... done"
```

Figure 6-17: Wrapper script example (ARC processor call) for Grid Engine

```
#!/bin/bash

# export repositoryRoot=/archive/sst/eodata
# merci-product-reg-2.1.1-SNAPSHOT-prodreg-call.sh \
#   hdfs://master00:9000/calvalus/projects/sst/atsr.3-l2/2005/03/11/20100701000000-
ESACCI-L3U_GHRSST-SSTskin-AATSR-LT-v02.0-fv01.0.nc \
#   hdfs://master00:9000/calvalus/projects/sst/atsr.3-l2/2005/03/11

set -e

inputURL=$1
outputURL=$2
inputFilename=$(basename $inputURL)
day=$(basename $outputURL)
month=$(basename $(dirname $outputURL))
year=$(basename $(dirname $(dirname $outputURL)))
type=$(basename $(dirname $(dirname $(dirname $outputURL))))

echo "`date -u +%Y%m%d-%H%M%S` cataloguing $inputFilename ..."

if hadoop fs -ls ${outputURL}/ql/${inputFilename%.nc}.png 2> /dev/null; then
  echo "skipping ${inputFilename}, ql exists"
  exit 0
fi

echo "`date -u +%Y%m%d-%H%M%S` retrieve input ..."

hadoop fs -get $inputURL .

# set up parameters and environment

cat > prodreg.properties <<EOF
RepositoryDir = ${repositoryRoot}
GenerationDir = .
GeoBoundaryStep = 128
GeoSegmentStep = 2000
EOF

mkdir -p $type/$year/$month/$day
mkdir -p tmpdir

echo "`date -u +%Y%m%d-%H%M%S` register in catalogue ..."

export JAVA_TMP=$(pwd)/tmpdir
/home/hadoop/opt/merci/merci-product-reg-2.1.1-SNAPSHOT/prodreg.sh -c
config.properties -q -t -s cc-2.0 -g -o $type/$year/$month/$day $inputFilename

echo "`date -u +%Y%m%d-%H%M%S` store quicklooks ..."

hadoop fs -put $type/$year/$month/$day/ql/${inputFilename%.nc}.png
${outputURL}/ql/${inputFilename%.nc}.png
hadoop fs -put $type/$year/$month/$day/tn/${inputFilename%.nc}.png
${outputURL}/tn/${inputFilename%.nc}.png

echo "`date -u +%Y%m%d-%H%M%S` cataloguing $inputFilename ... done"
```

Figure 6-18: wrapper script example (product cataloguing) for Calvalus

Java processors in BEAM can consist of a single operator or of a graph of operators applied to input products to produce an output. Figure 6-19 shows the interface of operator implementations in BEAM. One of the compute... methods are to be implemented.

```
package org.esa.beam.framework.gpf;
public class Operator {
    final OperatorContext context;
    protected MyOperator() {
        context = new OperatorContext(this);
        context.injectParameterDefaultValues();
    }
    public abstract void initialize() throws OperatorException;
```

```
    public void computeTile(Band targetBand, Tile targetTile, ProgressMonitor pm)
        throws OperatorException {
        ...
    }
    public void computeTileStack(Map<Band, Tile> targetTiles,
                                 Rectangle targetRectangle, ProgressMonitor pm)
        throws OperatorException {
        ...
    }
    public void dispose() {}
    ...
}
```

Figure 6-19: Operator interface for BEAM processors

There are objects for products, bands (variables), and tiles with access functions via the operator context for inputs and outputs. The objects abstract from the respective file formats by readers and writers. Processing parameters are provided as function parameters via the operator context, and a processor gives feedback by callback functions and - if necessary - exceptions. BEAM processors can be used as executables (gpf command), interactively (VISAT tool), and in Calvalus.

```
package org.apache.hadoop.mapreduce;
public class Mapper<KEYIN,VALUEIN,KEYOUT,VALUEOUT> {
    public Mapper() { ... }
    protected void setup(Context context)
        throws java.io.IOException, java.lang.InterruptedException { ... }
    protected void map(KEYIN key, VALUEIN value, Context context)
        throws java.io.IOException, java.lang.InterruptedException { ... }
    public void run(Context context)
        throws java.io.IOException, java.lang.InterruptedException { ... }
    protected void cleanup(Context context)
        throws java.io.IOException, java.lang.InterruptedException { ... }
}
```

Figure 6-20: Mapper interface with for other Java processors in Hadoop

Java processors in Calvalus/Hadoop may choose to implement the Hadoop Mapper interface instead of the BEAM Operator interface. This interface requires to implement either a function map() or a function run() (Figure 6-20). Inputs are accessed as streams or as sets of generic key-value pairs that are supported by an efficient Hadoop-internal file format. Also outputs are streams or sets of generic key-value pairs.

While both Java interfaces can be more efficient because they can use streaming and operator chaining in BEAM, the use of the wrapper scripts allows adaptation to Unix executables without changing them. Both alternatives are available in the SST CCI system.

6.3.2 Processor version concept

Processors, or more precisely processor bundles including software and configuration are the units that are under configuration control in the SST CCI system (→ SST-SR-1330 Processor configuration control, SST-SR-6280 Processor repository, SST-SR-5250 Version decisions, SST-SR-5260 Version release process, SST-SR-5310 Measures for stability). Each processor bundle has a version. A bundle may include one or several processors. There is a runtime structure of the processor software, and optionally there is a different development structure. The runtime structure is packed into an installation package for deployment. The packing procedure labels this installation package with the proper version in its file name. (→ SST-SR-5210 Transfer to operations).

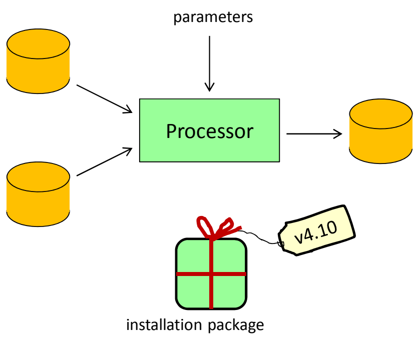


Figure 6-21: Processors and versioned processor installation packages for modularity

Processing jobs specify a bundle by name and version, and a processor to be run (→ SST-SR-1340 Concurrent processor versions). A combination of bundles and their respective versions used for the generation of a dataset form an assembly. The SST CCI assemblies comprise a certain version of the CCI ARC processor, a certain version of the CMS OSI-SAF processor, and a certain version of the OSTIA L4 analysis processor. With the same assembly the same output is reproducible from the same input if required.

In order to simplify development, the schema for development versions is less strict. It follows the snapshot approach sometimes used with the 'git' version control system. The same version number can be used repeatedly to tag the development version as long as it is not frozen. Figure 6-22 shows the versioning schema and its principles. (→ SST-SR-5130 Development versions).

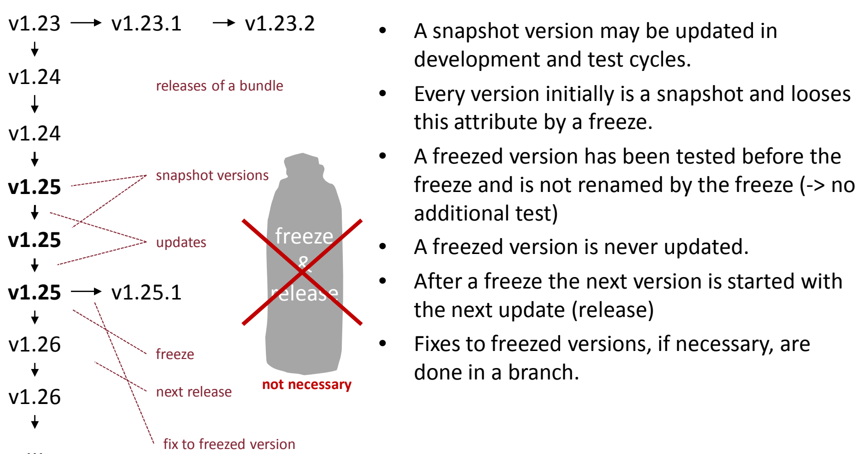


Figure 6-22: Processor and configuration versioning with freeze and release

The git version control system and a git server (e.g. GitHub) is proposed for the processor bundles of the SST CCI system. If there are parts that are not open source the corresponding bundles are kept in a non-public repository of the git server. All other bundles are in a public repository (public for reading) in order to allow interested users to review the software and configurations used and to allow developers to contribute.

The SST CCI version control provides the following convenience functions for processor developers:

- convenient commit: stores the current status of the directory tree of the development structure of the processor bundle, a convenience command; includes adding/deleting files from the directory tree, commit and push to the repository, tagging with the current version number, or a new branch version number in case of fixes
- convenient checkout: restores the directory tree of the development structure of the processor bundle; checks out main trunk or specified version. If a version is specified a simple commit starts or continues a branch.
- freeze and release: commits and freezes the current version and ensures that convenient commits will use the next version.
- deploy: generates and optionally installs the installation package of the current version. If the version is frozen the installation package contains this feature as a marker file. Only frozen versions shall be used for production and for bulk tests.

Concurrent development by experienced developers can still use the commands of the version control system (git) for merging and branching if required. The convenient functions can be used without detailed knowledge of version control systems when conflicts caused by concurrent changes of the same module are avoided.

6.3.3 Virtualised test environment

The SST CCI system provides a component made of virtual machines to be used for processor development, problem analysis, and test runs on single products (→ SST-SR-5120 Development environment). To run tests or to analyse a problem, a template virtual machine can be copied, used for tests, and deleted after it is no longer required. No long-term state is maintained in such a machine. State is instead kept in the software repository.

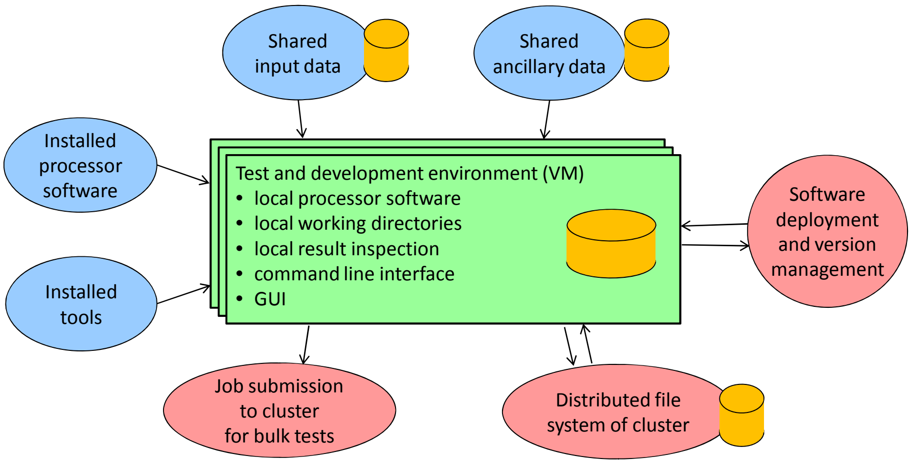


Figure 6-23: Local test environment with full data access

Figure 6-23 depicts the virtual machines and their environment. A virtual machine comes with a recent version of each processor pre-installed. The machine has access to the software repository to update an installed processor bundle to a particular version, to commit changes, and deploy them in the production environment (→ SST-SR-5210 Transfer to operations, SST-SR-5270 Short development cycle). Tools are installed as well, and there is access to test data and to auxiliary data. Access to the full processing storage of the cluster production environment archive is used for the analysis of problems. And for bulk tests the interface for job submission to the processing cluster can be used from the test and development environment. (→ SST-SR-5150 Full timeline access). By limiting resource shares for test production operational reprocessing is not compromised (→ SST-SR-5310 Measures for stability).

6.3.4 Multi-sensor matchup datasets

Matchups are pairs or multiples of observations that coincide in space and time. Often one of the observations is in-situ data, and the matchups allow to compare outputs of processing of satellite data with the in-situ as reference. In SST there are floating buoys, moored buoys and ships that collect large amounts of in-situ data. In a multi-sensor matchup dataset this in-situ data is related to satellite observations.

Figure 6-24 shows a plot of multi-sensor matchups of one day (02.06.2010). It shows matchups of

- ATSR + Metop + SEVIRI + AVHRR (2 to 4 platforms) + in-situ (A+M+S, large red dots)
- ATSR + Metop + AVHRR + in-situ (A+M, large blue dots)
- ATSR + SEVIRI + AVHRR + in-situ (A+S, turquoise dots)
- Metop + SEVIRI + AVHRR + in-situ (M+S, yellow dots)
- ATSR + AVHRR + in-situ (A, small red dots)
- Metop + AVHRR + in-situ (M, small purple dots)

SEVIRI + AVHRR + in-situ has not been plotted as there are many. The in-situ are from drifting buoys. If there are two dots at the same location they differ in time substantially (one orbit or more).

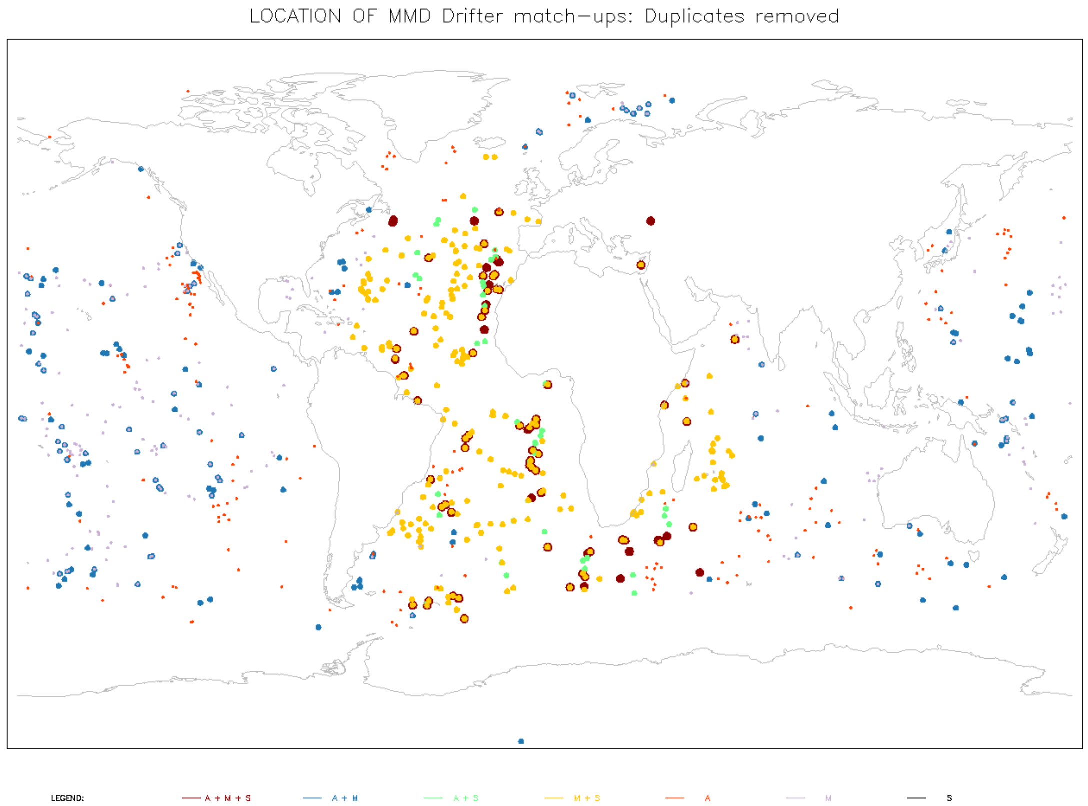


Figure 6-24: Multi-sensor matchups with different sensor combinations of one day

Sets of such matchups are provided as multi-sensor matchup datasets, MMD for short ([AD 6]) (→ SST-SR-1310 Matchup analysis). The MMD is organised in records, one for each matchup. A record comprises fields for each sensor that constitutes the matchup (ATSR, Metop, ...), in-situ, additional sensors (PMW data), and auxiliary data (sea ice, aerosol). Several fields per sensor provide time and location information, payload data as small subscenes of unprocessed and optionally processed data of about 10 km times 10 km or other configurable sizes, and auxiliary data, in particular NWP. Defined fill values are used where a sensor is not part of a matchup. (→ SST-SR-3140 Extract satellite subscenes, SST-SR-3150 interpolate NWP, SST-SR-3160 Process SST)

MMDs are presented in NetCDF. The record fields become NetCDF variables. Figure 6-25 shows an excerpt of an MMD file header.

```
netcdf mmd_201006 {
dimensions:
  match_up = 9706 ;
  atsr_md.cs_length = 8 ;
  atsr_md.length = 65 ;
  atsr_md.ui_length = 30 ;
  metop.len_filename = 65 ;
  metop.len_id = 11 ;
  metop.ni = 21 ;
  metop.nj = 21 ;
  seviri.len_filename = 65 ;
  seviri.len_id = 11 ;
  seviri.ni = 5 ;
  seviri.nj = 5 ;
  atsr.ni = 101 ;
  atsr.nj = 101 ;
  avhrr.ni = 25 ;
  avhrr.nj = 31 ;
  amsre.ni = 11 ;
  amsre.nj = 11 ;
  tmi.ni = 11 ;
  tmi.nj = 11 ;
  seaice.ni = 15 ;
  seaice.nj = 15 ;
  aai.ni = 1 ;
  aai.nj = 1 ;
  history.time = 24 ;
  history.qc_length = 2 ;
```

```
variables:
int matchup_id(match_up) ;

  char atsr_md.insitu.callsign(match_up, atsr_md.cs_length) ;
  byte atsr_md.insitu.dataset(match_up) ;
  double atsr_md.insitu.time.julian(match_up) ;
  float atsr_md.insitu.longitude(match_up) ;
  float atsr_md.insitu.latitude(match_up) ;
  short atsr_md.insitu.sea_surface_temperature(match_up) ;

  float aatsr.longitude(match_up, atsr.ni, atsr.nj) ;
  float aatsr.latitude(match_up, atsr.ni, atsr.nj) ;
  short aatsr.reflec_nadir_0550(match_up, atsr.ni, atsr.nj) ;
  ...

  double metop.msr_time(match_up) ;
  double metop.dtime(match_up, metop.ni) ;
  short metop.lon(match_up, metop.ni, metop.nj) ;
  short metop.lat(match_up, metop.ni, metop.nj) ;
  short metop.IR037(match_up, metop.ni, metop.nj) ;
  ...
  short metop.sst(match_up, metop.ni, metop.nj) ;
  byte metop.sst_confidence_level(match_up, metop.ni, metop.nj) ;

  float seaice.sea_ice_concentration(match_up, seaice.ni, seaice.nj) ;

  double history.insitu.time(match_up, history.time) ;
  float history.insitu.latitude(match_up, history.time) ;
  float history.insitu.longitude(match_up, history.time) ;
  float history.insitu.sea_surface_temperature(match_up, history.time) ;
  char history.insitu.position.qc(match_up, history.time, history.qc_length
}
```

Figure 6-25: NetCDF header of MMD file with dimensions and prefixed variables

MMDs are flexible regarding the fields included (→ SST-SR-3130 Update flag fields). An MMD may include all fields available (about 300 for SST, about 1 MB storage per record, about 14 GB storage per day, 1.5 GB compressed). Or it comprises a subset of fields. MMDs can be created externally with new variables to provide new information related to matchups, e.g. the output of an algorithm applied to a sensor, or the data of a new sensor. In order to relate different MMDs to each other each matchup and record is uniquely identified by a matchup_id value. This value remains the same also across different MMDs.

6.3.5 Multi-sensor matchup system

The SST multi-sensor matchup system (MMS) generates and processes MMDs (→ SST-SR-1310 Matchup analysis). A prototype of the MMS has been developed in the first phase of SST CCI. It had been optimised for one-time MMD generation for nineteen years of input data based on pre-matched single-sensor matchup files (MD). Three extensions to this prototype are to generate matchups from in-situ and satellite data on the fly (→ SST-SR-3120 Compute multi-sensor matchups), to optimise subscene extraction, and to simplify extension by new columns. (→ SST-SR-3130 Update flag fields, SST-SR-3260 Optimised for re-processing cycles)

This section defines EO data organisation, database schema, and the algorithm improvement cycle with the MMS.

6.3.5.1 File System Layout

One principle of the MMS is that the original data is stored in its original input format and that the MMS references earth observation data to generate matchups instead of copying earth observation data into a database. For this reason the file organisation is managed by the MMS.

The input files of the MMS are organised in a directory structure following an archiving rule that distinguishes types or sensors (→ SST-SR-3250 Structured storage of original inputs):

/<archive-area>/eodata/<type>/<revision>/<year>[/<month>[/<day>]]/<file>

It may be rooted at any path that further may be different for different types. Example:

/archive-root/eodata/atsr_md/v1.0/2010/06/02/atsr_md_201006.nc

Types currently foreseen for input data are (→ SST-SR-1180 Validation datasets, SST-SR-3110 Ingest observations of various types)

- atsr/atsr1, atsr/atsr2, atsr/aatsr
- avhrr/noaa_15, avhrr/noaa_16, ..., avhrr/metop1
- metop, seviri
- amsre, tmi
- seaice, aai, nwp
- in-situ (time line), atsr_md

Pre-computed extracts are generated during matchup processing for all data that otherwise require several large files to be read when generating an MMD. The generation of these pre-computed extracts is a function of the MMS. Note that this not the same as the externally pre-generated single-sensor matchups used in the prototype. It is only an optimisation within the MMS. Examples are subscenes of satellite data that are extracted during matchup computation (sub), and NWP interpolated data related to the matchups (nwp) (→ SST-SR-3140 Extract satellite subscenes, SST-SR-3150 Interpolate NWP). The extracted subscenes are stored in internal MMDs. The MMS refers to them instead of the original satellite data files, and generates MMDs for users from them. Types currently foreseen for pre-computed data are:

- atsr1/sub, atsr2/sub, aatsr/sub
- atsr1/nwp, atsr2/nwp, aatsr/nwp
- avhrr_n15/sub, avhrr_n16/sub, ..., avhrr-metop1/sub
- avhrr_n15/nwp, avhrr_n16/nwp, ..., avhrr-metop1/nwp
- metop-l2/sub, seviri-l2/sub

Geostationary data (SEVIRI) will be provided as spatio-temporal extract including a history. New types can be added on demand by providing configuration and data readers (→ SST-SR-3230 Extendable by new readers, SST-SR-3240 Configurable transformation of inputs). Among them are MMDs themselves (→ SST-SR-3130 Update flag fields). The purpose is to ingest processing results into the MMS for inter-comparison and analysis. MMDs with the processing results of subscenes are ingested, optionally for different versions of processors. (→ SST-SR-3160 Process SST, SST-SR-3190 Support concurrent versions), e.g.:

- arc-1.2-atsr1, ...

6.3.5.2 Matchup database schema

While the earth observation data remains in files matchups themselves are represented in a database. The database schema comprises four most relevant tables related to each other (Figure 6-26):

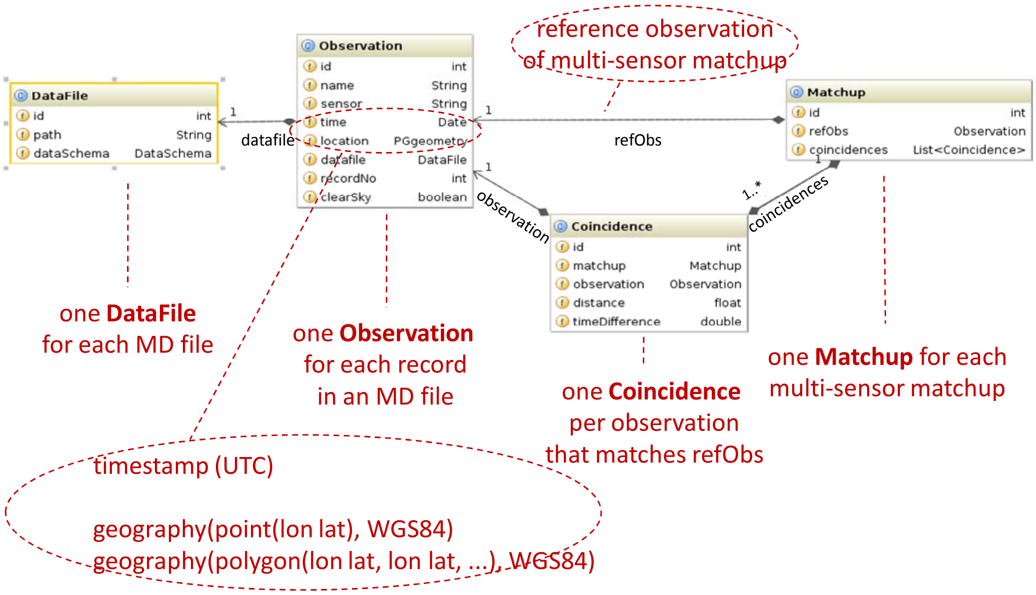


Figure 6-26: Matchup database schema with observations and matchups and their relations

- There are four database tables representing entities of the MMS. The most fundamental one is the observation. Each observation denotes one satellite image, one in-situ time line, one auxiliary data file, or one record in an MMD. Observations have sensor, time and location attributes that distinguish them.
- Matchup is the table to represent matchups of several observations. There is one reference observation of a matchup, and other observations related to the matchup by the Coincidences table. All observations related to the matchup are temporally and spatially overlapping or close to the reference observation of the matchup. The reference observation always is one of the main satellite data used (ATSR, AVHRR, METOP, SEVIRI). One of the related observations usually is an in-situ observation.
- DataFile is the table to represent files with input data or extracted MMDs in the MMS file system. Each entry refers to one file. Each file contains one or several observations. If it contains several observations (MMD, MD) a record number field in Observations selects the respective record (Figure 6-27).

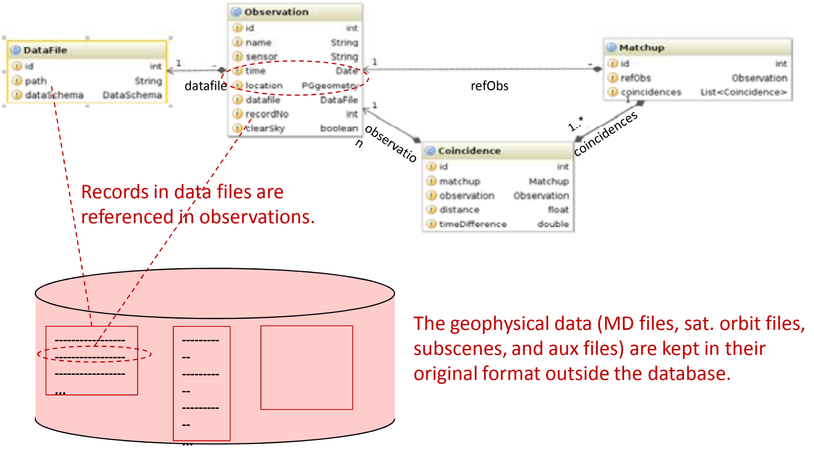


Figure 6-27: Referencing input files from observations

6.3.5.3 Matchup system dynamic model

Based on this file system and database schema processes are defined. The main processes are matchup generation, subscene processing, MMD extraction and analysis. Figure 6-28 shows these three processes in a repeat cycle (→ SST-SR-3260 Optimised for reprocessing cycles).

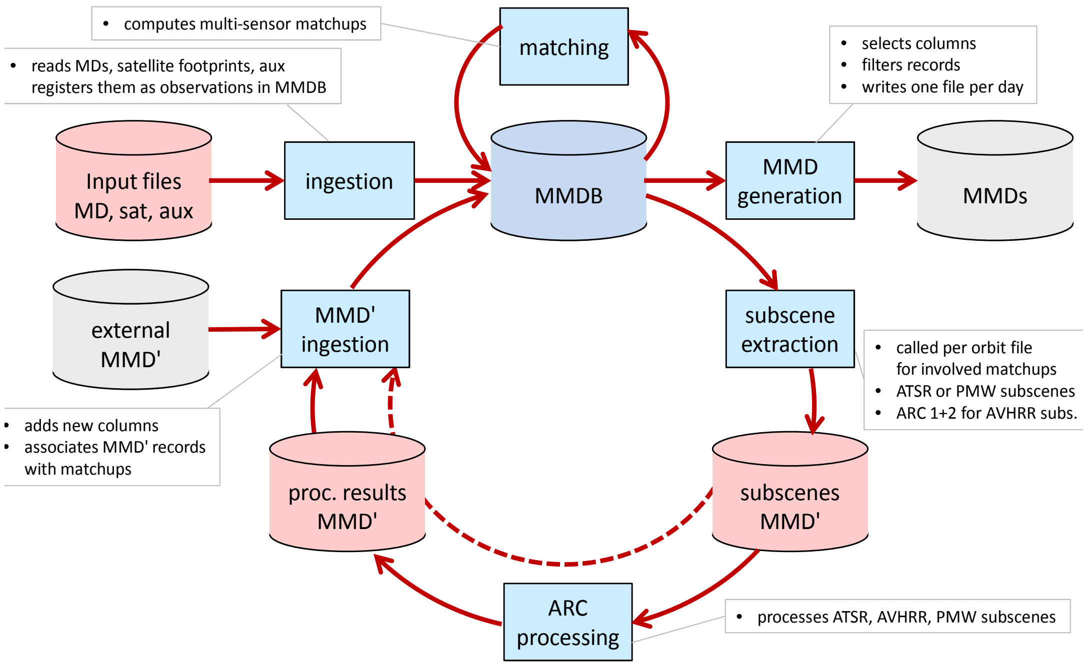


Figure 6-28: Test and validation cycles with the Multi-sensor Matchup System

The processes shown in the diagram are the following:

- Input files with satellite data, in-situ data or auxiliary data are ingested by registering them in the MMDB (→ SST-SR-2125 Validation data update, SST-SR-3110 Ingest observations of various types). Matchups are determined and added to the MMDB (→ SST-SR-3120 Compute multi-sensor matchups), SST-SR-3125 Incremental matchup computation), and subscenes are be extracted from satellite data for repeated use.

- For every new version of a processor, or to compare different processors subscenes are processed and the results are re-ingested. This cycle can also be run via MMD extraction and external processing. The result of external processing can be ingested, too, either as new "input files" or as MMD' if it is in this format. (→ SST-SR-3160 Process SST, SST-SR-3180 Ingest external MMD inputs, SST-SR-3190 Support concurrent versions. SST-SR-5140 Matchup dataset processing)
- MMDs are generated with various combinations of variables, and optionally on request by users. The matchups to include can be selected using criteria like time range and location, or sensors involved in matchups. (→ SST-SR-3170 Extract multi-sensor matchup datasets)

The processing of subscenes, and optionally the matchup generation itself can either be done in a test environment or with support of the parallelised production environment. Depending on the middleware selection this may be different:

- With Grid Engine the solution of the prototype for concurrent subscene generation and concurrent processing of subscenes can be continued. It had required staging in the prototype. The modification would be to do subscene extraction only once, allowing for reprocessing of the pre-generated matchups without staging satellite data.
- With Calvalus/Hadoop matchup generation and processing can be done in a data-local manner. It is not necessary though possible to pre-compute all matchups and to extract subscenes beforehand. The map-reduce programming scheme determines single-sensor matchups and processes subscenes of the satellite data inputs in its map step and merges multi-sensor matchups and formats MMDs in its reduce step. This is a non-trivial extension of the prototype.

The ingestion of versioned processing results and of external input increases the amount of data to be managed over time. An operator has to maintain this dataset as a librarian and has to decide which information to keep and which to replace or delete when obsolete. (→ SST-SR-3200 Operations procedures for MMS use cases, SST-SR-3210 Operational procedures for maintenance).

The interface of the MMS comprises a set of functions for the different processes explained above and the data interfaces for input data and the MMD format. This is summarised in Table 6-4.

Table 6-4: Interfaces of the MMS

Ifc ID	Interface Name	Source, Dest.	Function/Format	Interface item description
Ifc-11	**Input data interface** Satellite data Ancillary data In-situ data (timelines)	Local file system, MMS	File access Various formats, BEAM reader and column configuration required for each new format	DARD
Ifc-12	**MMD extraction data interface**	MMS, local file system or remote file system	File access, HTTP for remote access Content specification regarding columns contained, temporal and spatial selection	MMD content specification [AD 6]
Ifc-13	**MMD' ingestion data interface**	Local file system or remote file system, MMS	File access, HTTP POST for remote access Content specification regarding columns contained, metadata for origin and dataset	MMD content specification [AD 6]
Ifc-14	**MMS command interface**	MMS client, MMS service	Command line tools, HTTP Functions for staging, ingestion, matchup generation, subscene extraction, NWP interpolation, ARC processing, MMD extraction, MMD' ingestion, destaging	MMD content specification [AD 6]
Ifc-15	**MMS management interface**	MMS client, MMS service	Command line tools, HTTP Functions for dataset definitions, column definitions, data conversion configurations	MMS configuration

6.3.5.4 MMS development work

As mentioned in the beginning of this section the prototype of the MMS had been optimised for one-time MMD generation for nineteen years of input data based on pre-matched single-sensor matchup files. In order to develop this prototype to a full MMS system three functional extensions are required:

- Matchups shall be generated from in-situ and satellite data instead of pre-generated single sensor matchups. This requires the exchange of one work-flow step with adapted rules of what shall be considered a matchup. The prototype rules had taken all single sensor matchups as candidates and had sorted out duplicates that refer to the same in-situ buoy and the same satellite swath. The adapted rules have to select the buoy measurement from the in-situ history closest to the satellite data acquisition time with spatial coincidence considering cloud-free matching pixels only.
- The ingestion of newly acquired in-situ and satellite data shall be implemented as an automated process to extend the matchup database into the future with newly acquired data. As with all aggregation workflows the tricky point is to determine the time up to which to wait for additional data. If the in-situ data is available the preference rules for certain satellites makes an incremental approach for matchup computing difficult if the preferred input data (AATSR in the prototype, SLSTR in the future) is not yet available. A strategy to wait a certain time and to re-process if input data arrives later solves this difficulty.
- The MMS shall provide a query and extraction interface for external and internal users, supporting spatial (region of interest) and temporal (time period) criteria as well as sensor combinations contained, and the selection of columns to be included. The function creates logically defined named subsets of the matchups, and it generates user-defined MMDs for download.
- There has been an operational procedure how to add new columns to the MMDB in the prototype, e.g. for the ingestion of MMD' files provided by users. This addition of new columns shall be automated. Also the substitution of data by new versions shall be a function of the MMS. One element of this is a formal check of a user-provided MMD' whether it matches the MMDB content, i.e. whether its structure follows the MMD specification and all matchup identifiers listed are known in the MMDB. Another element is user management: Added information resides in database spaces of the respective user. Users can publish this data to be visible by others. Operators can move this data to become part of the common data model.
- Typical use cases and monitoring functions shall be automated workflows that are encapsulated and can be triggered in a web-based operator interface by request forms, among them the re-generation of the MMDB or parts of it, the addition of a satellite or other observation type, a processing cycle with ARC on defined subsets of matchups, the management of MMDB columns added by operators or by users, and the removal of obsolete data.
- The speed of the (ARC) processing cycle with MMD datasets depends on the amount of data to be read. The prototype has handled processing cycles by on-the-fly extraction of subscenes from satellite data and by the on-the-fly interpolation of NWP parameters before processing. The advantage is

the variable subscene size possible. This has been used for larger high-latitude extracts in the prototype. The disadvantage is that all original satellite data images of the respective time period must be loaded for extraction. In the next version of the MMS the subscenes of all found matchups shall be automatically extracted with the standard subscene size (10 km), and NWP parameters required for processing shall be interpolated. The extracted subscenes are cached in MMDs and referred to in the MMDB. Their access for processing reads only a few files instead of all the satellite inputs. This makes fast cycles for processor improvement feasible.

- Plots and simple intercomparison functions, scatter plots, maps and time line plots shall be added as ready-to-use functions.

These extensions make a system for repeated and continuous use out of a prototype that was optimised for its one-time application. For the SST CCI project the MMS remains a server-based, web based service because a stand-alone PC application is not required by the development team that works in the SST CCI environment.

6.4 System documentation

The system documentation comprises requirement documents, design and interface control documents, test documents, manuals, and maintenance information. Compared to this there are additional project documents related to the system, e.g. the SST CCI ATBD and the SST CCI PUG (→ SST-SR-1366 Product User Guide).

- The existing SRD and this SSD define requirements and design of the system. They may be updated in CCI phase 2 as the scenario for the scope of the system is finalised. An ICD for the main interfaces complements this for machine-to-machine user services, repository and deployment, request submission, processor integration, processing storage and inventory, and data exchange.
- An operations manual includes an instruction part with step-by-step descriptions of different use cases, and a reference part related to the functions and functional components and their capabilities, how to use them. There are different parts for the three subsystems user services, production environment, and development. (→ SST-SR-1367 Operator's manual)
- An installation and administration manual describes the initial setup and configuration of the SST CCI system, how to upgrade the system to a newer version, how to do maintenance and backup, how to extend the system with additional hardware and for new sensors.
- The processor integration guide describes the most important internal interface of the system and is a form of an ICD. The focus is on how to integrate the SST processors.
- The system verification documents define a set of tests and report about their results for the versions of the system that have been provided.
- The software release notes describe valid combinations of versions of components and software packages and they identify the corresponding documentation. They identify the versions currently in use.

- The issue tracking system documents system issues, among others, and their status.

This set of documents has been selected because they are considered most relevant and most used. (→ SST-SR-1525 ECSS compliance, SST-SR-1365 System level documentation)

7. DEVELOPMENT, LIFE CYCLE, COST AND PERFORMANCE

This section is a collection of further analyses regarding re-use of components, system life cycle, and cost and performance.

7.1 Re-use and development

Table 7-1 lists software packages that can be adapted, configured and integrated for use in the SST CCI system.

Table 7-1: Re-used software and its integration effort

Software	Role	Adaptation	Integration and Configuration
ARC-CCI	Cloud detection for IR sensors. SST retrieval processor.	Algorithm improvement. Possibly extend use of Bayesian cloud detection to more sensors. SST CCI data format	Provision of auxiliary data for the different platforms and sensors wrapper scripts for NWP interpolation and for processor call
CDO (climate data operators from MPI)	used in ARC-CCI for NWP interpolation	none	wrapper scripts (see above)
OSI-SAF and CMS processor	Used for METOP and SEVIRI processing	Development from prototype. (METOP processing may be moved to an adapted ARC-CCI, depending on Phase 2 scenario.)	parameterisation, wrapper scripts, auxiliary data handling
L3 processor	aggregate and reproject single sensor outputs	development from prototype tools	Parameterisation
L4 analysis system	Merge multi-sensor inputs and analyse them	development from prototype	parameterisation, wrapper scripts, auxiliary data handling
User tools	Aggregation, regridding, data analysis	development from prototype tools	none
Apache Hadoop	processing system middleware (alternative 1)	none	Hadoop setup

Software	Role	Adaptation	Integration and Configuration
Calvalus	processing system (alternative 1)	none	workflow definitions, parameterisation
Grid Engine	processing system middleware (alternative 2)	none	Grid engine setup, network file system setup start scripts and run scripts for processing
Calvalus processing monitor	processing system (alternative 2)	none	workflow definitions, parameterisation
BEAM	used in Calvalus, in tools, and in L3 processor	SST CCI data readers extension for uncertainty handling	none
Liferay	portal framework	none	Web design, content, system configuration, forum setup, integration of other services (MERCI, THREDDS, Jira)
MERCI	catalogue	LDAP user management	CCI product discovery metadata definition CCI product quality check criteria CCI product quicklook and thumbnail generation
THREDDS	online data server	none	setup of THREDDS OPeNDAP, WMS
vsftpd	Online data server	none	directory structure and access control configuration
Jira (optional)	Issue tracking	none	Setup of Jira, SST project configuration
KnowledgeTree (optional)	Document management	none	Setup of KT, tree structure configuration
OpenLDAP	User management backend	none	LDAP configuration

7.2 System life cycle drivers and considerations

The SST CCI system life cycle will not be completely static and pre-planned because of the desired inclusion of new requirements over time. Driving forces for an evolutionary development are

- availability of the SST CCI prototype as a starting point for further development
- incremental functional extension of the system
- improvement of algorithms, addition of new workflows and processing chains
- Sentinel 3 and other missions, increased data volume, options for synergistic use
- continuous improvements in hardware and data centre environments
- concurrent development for other ECVs

The evolutionary development will provide several versions of the system over time. To respond to the first two points, the system is initially based on the prototype. The initial system comprises only a subset of functions and components, but it is already hosted on the target platform. It may start with a core processing chain for ATSR and AVHRR using prototype processors and a core user service with the Web portal and an FTP data service for outputs. Incrementally, additional components and functions are added and interfaces to data providers and users are extended.

The third point of algorithm development requires the addition of validation capabilities, e.g. the MMS and tools, user feedback e.g. via the forum, and configuration control for the software. The next stage of the system shall support these functions.

The increased data volume and the new products are a qualitative change, too. The existing methods need to be adapted to make use of new channels. The amount of data is a challenge to currently available hardware for storage, memory and throughput. Optimisation will be required to avoid new bottlenecks. Example: To keep all input data online the system may grow on a monthly basis. This would increase the necessary resources to keep reprocessing time constant. Before the new data are available the system includes the necessary extensions and optimisations.

For the next three years phase of SST CCI it is reasonable to decide for a certain hardware and data centre environment and to keep this constant. But for the longer perspective also renewal of hardware and optional change of software layers can be expected. The system is prepared for this by the modularity of its functional components.

Last, the concurrent development for other ECVs is kept independent initially to avoid delays. But if supported, ECVs may exchange results and use them at least for consistency checks once they are available. SST CCI intends to use sea level, sea ice, ocean colour, and aerosol products for comparison.

7.3 Sizing and performance analysis

This section defines the budgets for data storage and processing capabilities. The budget for data storage mainly depends on the amount of input data to be managed (→ SST-SR-1450 Input storage size). This comprises both historical data and future data acquired continuously.

Table 7-2 summarises the input (satellite and anciliary) data that will be needed during the next phase (next 3 years) of the SST CCI project. The table lists the period of the data needed and their estimated total volume (compressed) for this period and for future years. Data volume estimates refer to the storage needed on tape, since it may not be practicable or affordable to hold all datasets online in the long run, in particular for Sentinel-3. Additional columns specify the fraction of the data required to be available online and the respective period for which online availability is needed. Note that most of the required input data are hold at the Centre for Environmental Data Archival (CEDA) on tape.

Table 7-2: Input data volumes for SST CCI Phase 2 (estimates)

ID	Type	Period	Data for period (TB)	Future data (TB / year)	Avail-able at CEDA	Online fraction (%)	Online start (KO+m)	Online end (KO+m)
A	ATSR	1991 – 2012	26.0	-	Yes	100	0	30
B1	AVHRR-GAC	1978 – 2013	13.0		No	100	0	30
B2	AVHRR-GAC	2014 – 2016	1.5	0.50	No	100	25	36
C1	AVHRR-A	2005 – 2014	24.0	-	Yes	10	12	15
C2	AVHRR-A	2005 – 2015	26.4	-	Yes	15	15	30
C3	AVHRR-A	2016	2.4	2.40	Yes	15	25	36
E	PMW	1997 – 2013	1.0	0.05	No	100	7	30
F1	SLSTR	2016 Q1 & Q2	56.0	-	Yes	100	28	32
F2	SLSTR	2016 Q3 & Q4	56.0	112.00	Yes	15	33	36
Z	ERA-Interim	1991 – 2013	9.0	0.25	No	100	0	36

Similarly Table 7-3 summarises the volumes of output data that will be produced in Phase 2 of the SST CCI project and when these data will be required to be available online. Three reprocessings will be carried out in this phase, the ordinal of which is indicated by the first digit of the dataset identifier in the first column.

Note that the volume of L2P and L3U outputs depends on the amount of inputs, while the volume of L4 outputs is constant per year (→ SST-SR-1460 Output storage size). Also note that volume of the L4 outputs for the 3rd reprocessing includes *all* outputs of the OSTIA system, whereas the estimates for the 1st and 2nd reprocessing include L4 SST CCI products only.

The hardware storage budget has to foresee some spare volume for unpacking compressed files, test results and other intermediate data. In addition, the budget may include a certain factor for storing different internal versions of output products to be kept concurrently.

For the volume estimates we adopt 15% spare and 100% of the total output volume for storing different versions.

Table 7-3: Output data volumes for SST CCI Phase 2 (estimates)

ID	Type	Period	Data for period (TB)	Future data (TB / year)	Online fraction (%)	Online start (KO+m)	Online end (KO+m)

ID	Type	Period	Data for period (TB)	Future data (TB / year)	Online fraction (%)	Online start (KO+m)	Online end (KO+m)
1A	SST CCI L3U (ATSR)	1991 – 2013	0.19	-	100	6	30
1B	SST CCI L2P (AVHRR-GAC)	1991 – 2013	2.10	0.15	100	6	30
1O	SST CCI L4 (OSTIA)	1991 – 2013	0.15	-	100	6	30
2A	SST CCI L3U (ATSR)	1991 – 1996	0.03	-	100	18	30
2B	SST CCI L2P (AVHRR-GAC)	1978 – 1996	2.50	-	100	18	30
2C	SST CCI L3U (AVHRR-A)	2005 – 2014	0.50	-	100	18	30
2O	SST CCI L4 (OSTIA)	1991 – 2014	0.21	-	100	18	30
3A	SST CCI L3U (ATSR)	1991 – 2013	0.19	-	100	28	36
3B	SST CCI L2P (AVHRR-GAC)	1978 – 2016	4.50	0.15	100	28	36
3C	SST CCI L3U (AVHRR-A)	2005 – 2016	0.60	0.05	100	28	36
3F	SST CCI L3U (SLSTR)	2016	0.05	0.05	100	28	36
3G	SST CCI L2P (ATSR)	1991 – 2012	20.00	-	100	28	36
3O	SST CCI L4 (OSTIA, all outputs)	1991 – 2015	7.00	0.6	100	28	36

Figure 7-1 below illustrates the online data volumes estimated in Table 7-2 and Table 7-3 for Phase 2 of the SST CCI project. The online storage will be about 55 TB initially, and will reach up to about 210 TB when processing the first six months of SLSTR data. The data volumes needed offline (i.e. on tape) are illustrated in Figure 7-2.

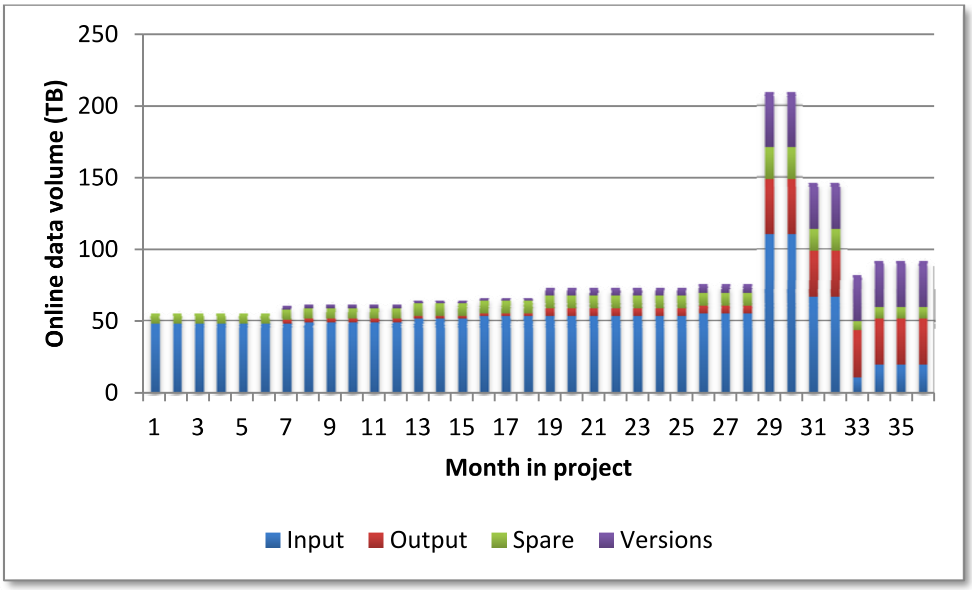


Figure 7-1: Online data volumes needed for SST CCI Phase 2 (estimates)

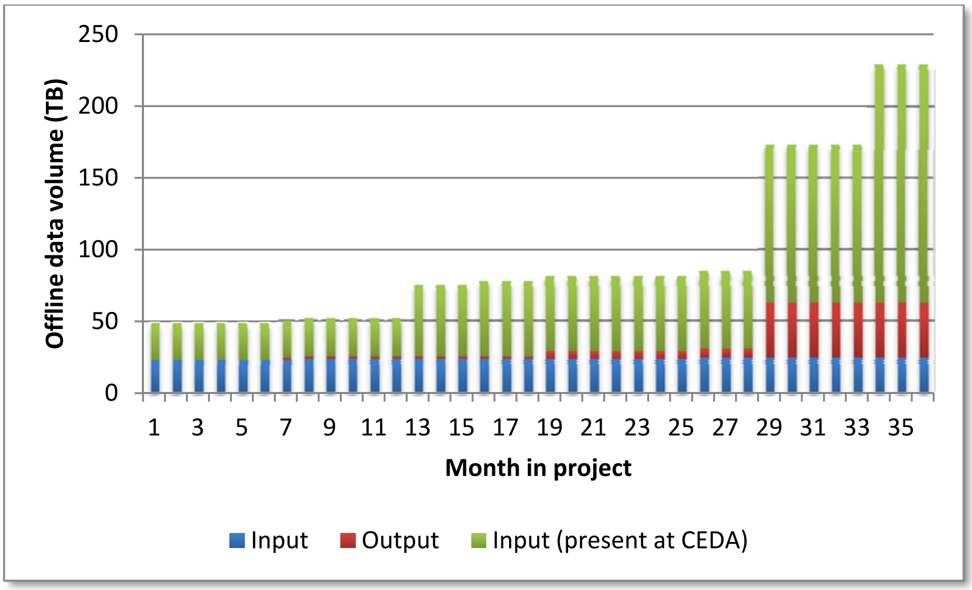


Figure 7-2: Offline data volumes needed for SST CCI Phase 2 (estimates)

The processing capabilities are mainly defined by the speed of processing (single product or day on single machine core) and the allowed overall time for reprocessing (assuming the time needed for staging data from offline storage to online storage is negligible compared to the time needed for processing the staged data).

Type	Amount of inputs for one reprocessing	CPU Core hours for one reprocessing
ARC CCI SST	50 TB of inputs for 1st reprocessing (1st year)	50000 h (based on estimation of 41280 h for 40 TB inputs)
ARC CCI SST	75 TB of inputs for 2nd reprocessing (2nd year)	75000 h (based on estimation of 41280 h for 40 TB inputs)
ARC CCI SST	200 TB of inputs for 3rd reprocessing (3rd year)	200000 h (based on estimation of 41280 h for 40 TB inputs)
OSTIA L4 SST	Up to 26 years of L2P and L3U inputs from ARC CCI (all years)	TBD (based on estimation of 65000 h for 20 years at resolution of 0.05°)
Maximum		200000 + TBD h

The allowed reprocessing time of 1 month (→ SST-SR-1490 Three days per year (48h/a 1981-2012)) leads to a concurrency of up to 280 + TBD for the 3rd reprocessing.

The estimated concurrency is also sufficient to re-generate the MMD in the required 1 day per data year (→ SST-SR-3225 MMDB rebuild).

7.4 Cost analysis

The costs for the system are composed of costs for using the CEMS/CEDA infrastructure, development and integration, and operations including scientific data verification.

Since the MMS mainly uses the same input data, it can use the same infrastructure and can use the spare processing power for MMD rebuild if required. A dedicated (virtual) machine used as database server with sufficient main memory (48 GB) is foreseen. (→ SST-SR-3220 20 million matchups, SST-SR-3225 MMDB rebuild)

The data access is facilitated by CEDA. (→ SST-SR-6370 User service availability).

Note that the cost estimates for using the CEMS/CEDA infrastructure in the manner estimiated in the previous section is a matter of discussion at the time of writing this document.

Regardless of the costs for the CEMS/CEDA infrastructure, the effort for operations is about one person year per year (~**150,000 €**). Development and integration is about the same amount for the first year (~**150,000 €**) and a smaller amount in subsequent years (~**75,000 €**).

Table 7-4: Cost estimates per year (CEMS/CEDA costs yet unknown)

	First year	Second year	Third year
CEMS/CEDA	TBD	TBD	TBD
Development	150,000 €	75,000 €	75,000 €
Operations	150,000 €	150,000 €	150,000 €
Sum			

Résumé

The numbers provided are an estimate based on 2012 prices and performance and the experiences with the current prototype systems. The purpose of the estimates is to provide an orientation in the space of options. It does not necessarily define the final price of the solution.

Looking at the balance between resources for scientific work and resource for reprocessing within SST CCI there may be limitations by an overall budget for SST CCI Phase 2. The justification for reprocessing still is that it supports scientific progress. Leaving no budget for the scientific part would make reprocessing useless.

Therefore, depending on the height of the budget, sub-optimal solutions for reprocessing may have to be selected:

- Less frequent reprocessing could reduce operations cost and costs for use of the CEMS/CEDA infrastucture.
- Less demanding reprocessing times for the new missions could reduce CEMS/CEDA costs.

- To continue using the prototype instead of a new development could reduce the development costs (but this may increase operations costs in the end)

Without limitation by an overall budget the recommendation is to implement the full system. It would allow scientific development without restrictions by a limited processing system.

8. REQUIREMENTS TRACEABILITY

This section traces input requirements (SST-SR-xxxx) of the SRD [AD 7] to sections of this document (§) or to particular SST system components that implement them.

Requirement	Title	Reference
SST-SR-6110	L2P outputs	6.2.5
SST-SR-6120	L3C outputs	6.2.5
SST-SR-6130	L3U outputs	6.2.5
SST-SR-6140	L4 outputs	6.2.5
SST-SR-6150	L4 diurnal outputs	6.2.5
SST-SR-6160	Output format and naming	6.1.2, 6.2.5
SST-SR-6170	Product features	6.1.5, 6.1.6
SST-SR-1110	(A)ATSR inputs	6.2.5
SST-SR-1120	AVHRR GAC inputs	6.2.5
SST-SR-1130	AVHRR METOP inputs	6.2.5
SST-SR-1135	PMW inputs	6.2.5
SST-SR-1140	Sentinel inputs	6.2.5, 6.2.3
SST-SR-1150	VIIRS inputs	6.2.5, 6.2.3
SST-SR-1160	Sentinel data reduction	6.2.3
SST-SR-1165	VIIRS data reduction	6.2.3
SST-SR-1170	Auxiliary inputs	6.2.5
SST-SR-1180	Validation datasets	6.2.5, 6.3.5
SST-SR-1190	Input data interface	6.2.3, 6.2.1
SST-SR-1200	Output preservation	3.3
SST-SR-1210	Structured storage	6.1.2, 6.2.2
SST-SR-1220	Input versions	6.2.3, 4.3
SST-SR-1230	Reprocessing input versions	6.2.3, 4.3
SST-SR-1240	Output versions	3.1, 6.1.3, 6.2.2
SST-SR-1241	Long-term stewardship	3.3
SST-SR-1250	SST retrieval	6.2.4, 6.2.5
SST-SR-1260	Error characterisation	6.2.4, 6.2.5
SST-SR-1270	Projection and compositing	6.2.4, 6.2.5
SST-SR-1280	Sensor merging	6.2.4, 6.2.5
SST-SR-1290	Output products	6.2.4, 6.2.5
SST-SR-1300	Quality check	6.2.6
SST-SR-1310	Matchup analysis	6.3.4, 6.3.5
SST-SR-1320	Trend analysis	6.1.6
SST-SR-1330	Processor configuration control	6.3.2

Requirement	Title	Reference
SST-SR-1340	Concurrent processor versions	6.3.2
SST-SR-1350	Processor framework	6.2.1, 6.3.1
SST-SR-1360	Executable integration	6.2.1, 6.3.1
SST-SR-1370	Processing chain	6.2.5
SST-SR-1380	Bulk reprocessing	6.2.5
SST-SR-2110	Input ingestion	6.2.3
SST-SR-2120	Auxiliary update	6.2.3
SST-SR-2125	Validation data update	6.2.3, 6.3.5
SST-SR-2130	Data-driven processing	6.2.5
SST-SR-2140	Wait for ancillary	6.2.5
SST-SR-2150	CDR extension	6.2.5
SST-SR-1390	Operator	4.3, 6.1.7, 6.2.5
SST-SR-1400	Automation	6.2.5
SST-SR-1410	System status	6.2.1
SST-SR-1420	Operating tool	6.2.1
SST-SR-1430	Error handling	6.2.5
SST-SR-1440	Cancel and resume	6.2.1, 6.2.5
SST-SR-2160	Online processing status	6.2.1, 6.2.5
SST-SR-2170	Online processing error handling	6.2.3, 6.2.5
SST-SR-1450	Input storage size	6.2.1, 7.3
SST-SR-1460	Output storage size	6.2.1, 7.3
SST-SR-1470	Online storage	6.2.1, 6.2.3
SST-SR-2180	Short-delay processing	6.2.3, 6.2.5
SST-SR-1480	Parallel processing	6.2.1
SST-SR-1490	Three days per year	7.3
SST-SR-1500	Backup archive	3.3, 5.3.1
SST-SR-1501	Bulk archive retrieval	3.3
SST-SR-1510	Archive reliability	5.6
SST-SR-1520	Autonomous system	5.3
SST-SR-1525	ECSS compliance	6.4
SST-SR-1365	System level documentation	6.4
SST-SR-1366	Product User Guide	6.1.6, 6.4
SST-SR-1295	Processor documentation	3.3, 6.1.1
SST-SR-3110	Ingest observations of various types	6.3.5
SST-SR-3120	Compute multi-sensor matchups	6.3.5
SST-SR-3125	Incremental matchup computation	6.3.5
SST-SR-3130	Update flag fields	6.3.4, 6.3.5
SST-SR-3140	Extract satellite subscenes	6.3.4, 6.3.5

Requirement	Title	Reference
SST-SR-3150	Interpolate NWP	6.3.4, 6.3.5
SST-SR-3160	Process SST	6.3.4, 6.3.5
SST-SR-3170	Extract multi-sensor matchup datasets	6.3.5
SST-SR-3180	Ingest external MMD inputs	6.1.3, 6.3.5
SST-SR-3190	Support concurrent versions	6.3.2, 6.3.5
SST-SR-3200	Operations procedures for MMS use cases	6.3.5
SST-SR-3210	Operational procedures for maintenance	6.3.5
SST-SR-3220	20 million matchups	7.4
SST-SR-3225	MMDB rebuild	7.3, 7.4
SST-SR-3230	Extendable by new readers	6.3.5
SST-SR-3240	Configurable transformation of inputs	6.3.5
SST-SR-3250	Structured storage of original inputs	6.3.5
SST-SR-3260	Optimised for reprocessing cycles	6.3.5
SST-SR-6180	Output provision	6.1.1, 6.1.3, 6.2.6
SST-SR-6190	FTP access	6.1.1, 6.1.3
SST-SR-6200	Web access	6.1.3
SST-SR-6210	OPeNDAP access	6.1.3
SST-SR-6220	Bulk access	6.1.3, 6.1.3.4
SST-SR-4110	Subsetting and regridding	6.1.6
SST-SR-6230	Online subsetting and regridding	6.1.1
SST-SR-4120	Visualisation and data analysis	6.1.6
SST-SR-4130	Data access software	6.1.6
SST-SR-6240	Web site	6.1.16.1.1, 6.1.5
SST-SR-6250	Catalogue	6.1.4
SST-SR-6260	News feed	6.1.1, 6.1.5
SST-SR-6270	Forum or help desk	6.1.1, 6.1.5
SST-SR-6280	Processor repository	6.1.6, 6.3.2
SST-SR-6290	External algorithm development	6.1.3, 6.1.6
SST-SR-6300	Matchup datasets	6.1.3
SST-SR-6310	Matchup inputs	6.1.3
SST-SR-6330	Web site	6.1.5
SST-SR-6340	Science issues	0, 6.1.5
SST-SR-5110	Community process	0, 6.1.5
SST-SR-6360	Forum maintainer	6.1.5
SST-SR-6370	User services availability	5.6
SST-SR-4140	Open source tools	6.1.6
SST-SR-5120	Development environment	6.3.3
SST-SR-5130	Development versions	6.3.2

Requirement	Title	Reference
SST-SR-5140	Matchup dataset processing	6.3.5, 6.1.3
SST-SR-5150	Full timeline access	6.3.3
SST-SR-5160	Other sensor's inputs	6.2.3
SST-SR-5170	New input type	6.2.3, 6.2.5
SST-SR-5180	Other ECVs as ancillary	6.2.3
SST-SR-5190	New ancillary type	6.2.3
SST-SR-5200	New processing chain	6.2.5
SST-SR-5210	Transfer to operations	6.3.2, 6.3.3
SST-SR-5215	L4 analysis system update	5.1.1
SST-SR-5220	Development team	0
SST-SR-5230	Agile requirements selection	3.1, 0
SST-SR-5240	Development and evaluation	0
SST-SR-5250	Version decisions	3.1, 6.3.2
SST-SR-5255	Overlapping versions	3.1
SST-SR-5260	Version release process	3.1, 6.3.2
SST-SR-5265	Version management	3.1, 6.2.5
SST-SR-5270	Short development cycle	3.1, 6.3.3
SST-SR-5275	Internal version storage	6.2.2, 7.3
SST-SR-5280	Storage extension	5.5
SST-SR-5290	Storage extension	5.5
SST-SR-5300	Performance scalability	5.5
SST-SR-5310	Measures for stability	6.2.2, 6.3.2, 6.3.3
SST-SR-1367	Operator's Manual	6.4

9. SUMMARY AND CONCLUSION

This SSD defines the design for the SST CCI system for the next CCI phase. It combines components and functions for user services, production, and algorithm development. The overall purpose of the system is to serve users with climate quality products. User services itself is kept relatively simple and conventional, except for the functions for user feedback. The focus of the system design is on high performance reprocessing and on algorithm improvement because climate quality still requires a lot of research and algorithm development, and the capabilities for fast reprocessing support this by better assessments of changes.

One key element for SST CCI in this improvement cycle is the Multi-sensor Matchup System MMS. For SST there is a large amount of in-situ data mainly from drifting buoys. The MMS is designed to match this with satellite data from missions relevant for SST retrieval and with auxiliary data. The main idea is that validation that uses the in-situ data to compare it with processed satellite data can be done on processed subscenes matching the in-situ data instead of the complete satellite images. This is much faster and allows cycles of algorithm modification and validation.

The conclusions of the trade-off analysis has defined which environment is preferred by SST CCI. The large amount of the expected Sentinel 3 SLSTR data is the main driver to aim for a shared processing environment that hosts this data already for other purposes. A common centralised, but open environment is preferred over a distributed SST CCI system because it avoids transfer of intermediate products. And if climate users or projects using large amounts of SST CCI outputs can do their analysis in the same shared environment where SST CCI is hosted then transfer of output products can be avoided in addition. The question whether cloud or cluster technology is used to achieve concurrency stays behind the data access questions, but unless there are other constraints a cluster environment with support for massive parallel processing with Hadoop as middleware is preferred by SST CCI because of its scalability and the support for immediate data-local concurrent processing without needs for staging. SST CCI is open to share parts of user services data stewardship with other ECVs as long as this can be done with low harmonisation effort.