Customer	:	**ESRIN**	**Document Ref**	:	**SST_CCI-SSD-BC-201**
Contract No	:	**4000109848/13/I-NB**	**Issue Date**	:	**2.0**
WP No	:	**34**	**Issue**	:	**10 February 2016**

Project : **SST-CCI-Phase-II**

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

Abstract : This document contains the SSD for the ESA SST CCI project.

Author :

T.Block
Brockmann Consult GmbH

Checked :

C. Merchant, Science Leader
University of Reading

H. Kelliher, Project Manager
Space ConneXions Ltd.

Accepted :

Digitally signed by Craig Donlon
DN: cn=Craig Donlon, o=European Space Agency, ou=ESTEC, email=craig.donlon@esa.int, c=NL
Date: 2016.04.01 13:14:27 +01'00'

C. Donlon
ESA Technical Officer

Distribution : SST_cci team members

ESA (Craig Donlon)

EUROPEAN SPACE AGENCY
CONTRACT REPORT

The work described in this report was done under ESA contract. Responsibility for the contents resides in the author or organisation that prepared it.

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
1.0	31.10.2014	Initial version
2.0	10.02.2016	Integrated updates implemented in SST-CCI project Phase-II

RECORD OF CHANGES IN THIS ISSUE

Issue	Page/Sec.	Reason	Change
2.0	Various	Correction	Corrected formatting and typos
2.0	Various	Correction	Corrected usage of BEAM and SNAP
2.0	1.2	Update	Updated reference document versions
2.0	2.2	Update	Updated input dataset
2.0	2.4	Update	Updated processor description for L2/L3 generation
2.0	2.6	Update	Updated paragraph re Open Data Portal Project and data storage
2.0	2.7.6	Update	Added current status of L4 processor migration
2.0	2.7.7	Update	Updated pre-processing status
2.0	3.1	Update	Updated external interfaces summary
2.0	5.3	Update	Updated paragraph wrt. Open Data Portal Project
2.0	6.2.4	Update	Updated list of processors
2.0	6.2.5	Update	Updated processor descriptions for L2/L3
2.0	6.3.4, 6.3.5	Update	Updated to latest developments in MMS
2.0	7.3	Update	Updated data interfaces and volumes

EXECUTIVE SUMMARY

This Sea Surface Temperature (SST) System Specification Document (SSD) specifies the design of an operational system for SST for the ESA Climate Change Initiative (CCI). The design is based on general operational use cases and decisions on design alternatives made at the beginning of CCI Phase-II. 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

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 Purpose and scope

This System Specification Document (SSD) of the European Space Agency (ESA) Climate Change Initiative (CCI) for Sea Surface Temperature (SST) specifies the design for an operational system for SST for CCI Phase-II and beyond. SST is one of currently 13 Essential Climate Variables (ECV) considered in the CCI. The SST system covers production, user services, and long-term stewardship for Climate Data Records (CDR) generated by the SST CCI project.

This document comprises the third evolution of the initial SST-CCI SSD for Phase-I, which was an immediate response to the SST System Requirements Document (SRD) [AD 7]. It is a formal deliverable of the SST CCI project requested in the Statement of Work (SoW) [AD 1]. System evolutions from Phase-I to Phase-II mainly comprised decisions on infrastructure in order to address requirements on sustainability in the SoW and updates in light of the upcoming ESA ITTs for a common CCI Data Access Portal and a common CCI User Toolbox. Further evolutions within Phase-II of the project focus on performance gains, algorithmic improvements and steps to facilitate re-processing.

In Phase-I the SSD described the (maximum) desirable future system under the assumption that Phase-II focuses on developing an operational system and services. But since Phase-II is focused on new ambitious scientific goals, the emphasis in this SSD is on scientific improvements and achieving a sustainable system, which facilitates proactive scientific developments and is operated by the SST team in a cost-effective manner.

This document version is based on achievements documented in the earlier versions with a focus on the evolutionary steps undertaken since the last delivered version.

1.2 Applicable and referenced documents

Applicable documents are listed in the table below.

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	1.0	2013-07-02
[AD 2]	ESA CCI Phase-II Sea Surface Temperature (SST) Proposal		2013-09-02
[AD 3]	Sea Surface Temperature CCI User Requirements Document, SST_CCI-URD-UKMO-001 (URD)	3	2015-12-14
[AD 4]	Sea Surface Temperature Data Access Requirements Document, SST_CCI-DARD-UOL-201 (DARD)	2	2014-09-23
[AD 5]	Sea Surface Temperature Product Specification Document, SST_CCI-PSD-UKMO-201 (PSD)	2	2014-04-11
[AD 6]	Sea Surface Temperature MMD Content Specification, SST_CCI-REP-UOL-001	1	2012-05-04

ID	Title	Issue	Date
[AD 7]	Sea Surface Temperature System Requirements Document, SST_CCI-SRD-BC-001 (SRD)	1.2	2012-04-30
[AD 8]	Sea Surface Temperature Detailed Processing Model, SST_CCI-DPM-BC-001 (DPM)	1.0	2012-10-04
[AD 9]	Sea Surface Temperature System Prototype Description, SST_CCI-SPD-BC-001 (SPD)	1.1	2013-07-28
[AD 10]	Sea Surface Temperature Product Validation Plan, SST_CCI-PVP-UoL-001 (PVP)	2	2014-02-04
[AD 11]	Sea Surface Temperature Algorithm Selection Report, SST_CCI-ASR-UOE-001 (ASR)	1.0	2012-06-30
[AD 12]	Sea Surface Temperature Algorithm Theoretical Basis Document, SST_CCI-ATBDv0-UOE-001 (ATBD)	1.0	2013-05-17
[AD 13]	CCI System Requirements, CCI-PRGM-EOPS-TN-12-0031	1.0	2013-07-02
[AD 14]	Data Standards Requirements for CCI Data Producers, CCI-PRGM-EOPS-TN-13-0009	1.1	2013-05-24
[AD 15]	SST CCI Reference Document List, SST_CCI-REP-UOE-001	1	2011-09-27
[AD 16]	SST CCI Acronyms List, SST_CCI-REP-UOE-002	1	2011-09-27

Referenced documents are listed in the table below. Additional referenced documents are listed in [AD 15].

ID	Title	Issue	Date
[TN 1]	SST CCI Phase-II – Technical Note: MMS Implementation Plan	1	2014-07-01
[TN 2]	SST CCI Phase-II – Technical Note: Review of SEWG SRD	1	2014-04-31
[DAP]	CCI Data Access Portal, CCI-PRGM-EOPS-SW-14-0030	1	2014-08-07

1.3 Acronyms

Acronyms used in this document are listed below. Additional acronyms are listed in [AD 16].

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

Acronym	Definition
CCI	Climate Change Initiative
CDR	Climate Data Record
CEMS	Climate and Environmental Monitoring from Space (Facility)
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
PMW	Passive Microwave
SDI	Saharan Dust Index
SCL	Space ConneXions Limited
SEVIRI	Spinning Enhanced Visible and Infrared Imager
SGE	Sun Grid Engine
SST	Sea Surface Temperature
UoE	University of Edinburgh
UoR	University of Reading

1.4 Notation

As this document is a response to requirements documents [AD 3, 4, and 7], requirements are traced back to subsections of the documents in Section 8. The backward references from sections to requirements are interspersed within the text. Backward referencing is denoted by an arrow followed by the requirement's identifier and the requirement's title, usually in parenthesis, e.g. (→ SST-SR-1240 Output versions), meaning that the text describes the design for the referenced requirement.

1.5 Document outline

After this formal introduction, the document is structured as follows:

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 summarises the decisions on infrastructure alternatives that were made at the beginning of Phase-II due to new requirements

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-usage of components, system life cycle, and cost and performance

Section 8 traces system requirements back to sections of this document

2. THE SST CCI PROJECT

2.1 Overview

The ESA CCI SST project 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) for Phase-I 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-I, a prototype production system was built, which generated world-class SST products [AD 5] (fulfilling CR-2) and was used for a range of activities (see below) for climate quality algorithm development and testing (satisfying CR-1).

For addressing CR-4, the SST CCI team 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 future system (hereafter referred to as "the operational system") in the System Requirements Document [AD 7]. The technical specification of this operational system was described in Phase-I and is further developed in this document.

In Phase-I the SST CCI prototype processor was 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). Due to new CCI cardinal requirements for Phase-II, in particular the integration of major new data sources like the Sentinels, Earth Explorers, and non-ESA Missions (CR-5), the decision was made to migrate the SST ECDF prototype and MetOffice OSTIA systems to the Climate and Environmental Monitoring from Space (CEMS) facility hosted at the UK Centre for Environmental Data Archival (CEDA). CEMS is a purpose-built facility offering space-based Climate Change and Earth Observation (EO) data and services. Its goal is to nurture growth in EO and climate-based services by providing, within a single facility, high performance computing, extensive data collections and various user services and software applications. In particular the provision and stewardship of Sentinel-3 SLSTR and Eumetsat AVHRR datasets is of essential interest in the SST CCI project.

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). Since the end of Phase-I, SST products have been available for the long-term (1991–2010) and demonstration (June and August 2007 and January to March 2012) periods. By the end of Phase II, version 2 products will be available for L2P, L3U/C and L4, covering 1981 to 2016 (this being the present target, not yet scientifically demonstrated). The products have been fully specified in the PSD [AD 5].

Several elements of the prototype production system have been re-used in the migration to the operational software system. Nevertheless, some software elements, 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 a future CCI-based system (operating beyond Phase-II under programmes not yet known) as specified by the team (Figure 2-1, trapezoidal shapes indicate manual activities).

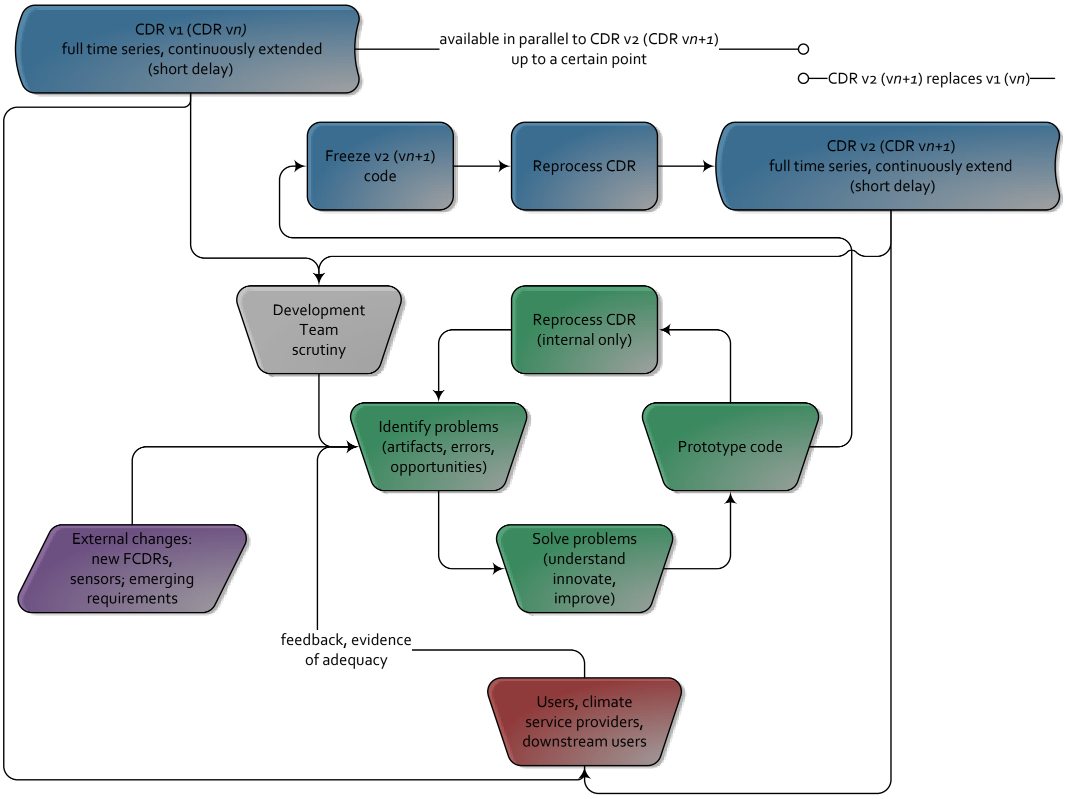


Figure 2-1: Algorithm improvement triggered by user feedback

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

1. There is a stable climate data record 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 the 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 by 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) were implemented in Phase-I of SST CCI; implementing an automated operation within the operational system is foreseen for Phase-II. 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 operational 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 were prototyped in Phase-I; plain letters denote elements that were prototyped for the demonstration but not for the long-term production in Phase-I; italic letters indicate functions that were not prototyped in Phase-I, but will be addressed for implementing the operational system in Phase-II and beyond.

The remainder of this project summary briefly describes and discusses the system elements and functions in turn. Where these elements have supported major activities within Phases-I and II, these activities and the role they have fulfilled are also described, with references to where in this document these elements are specified.

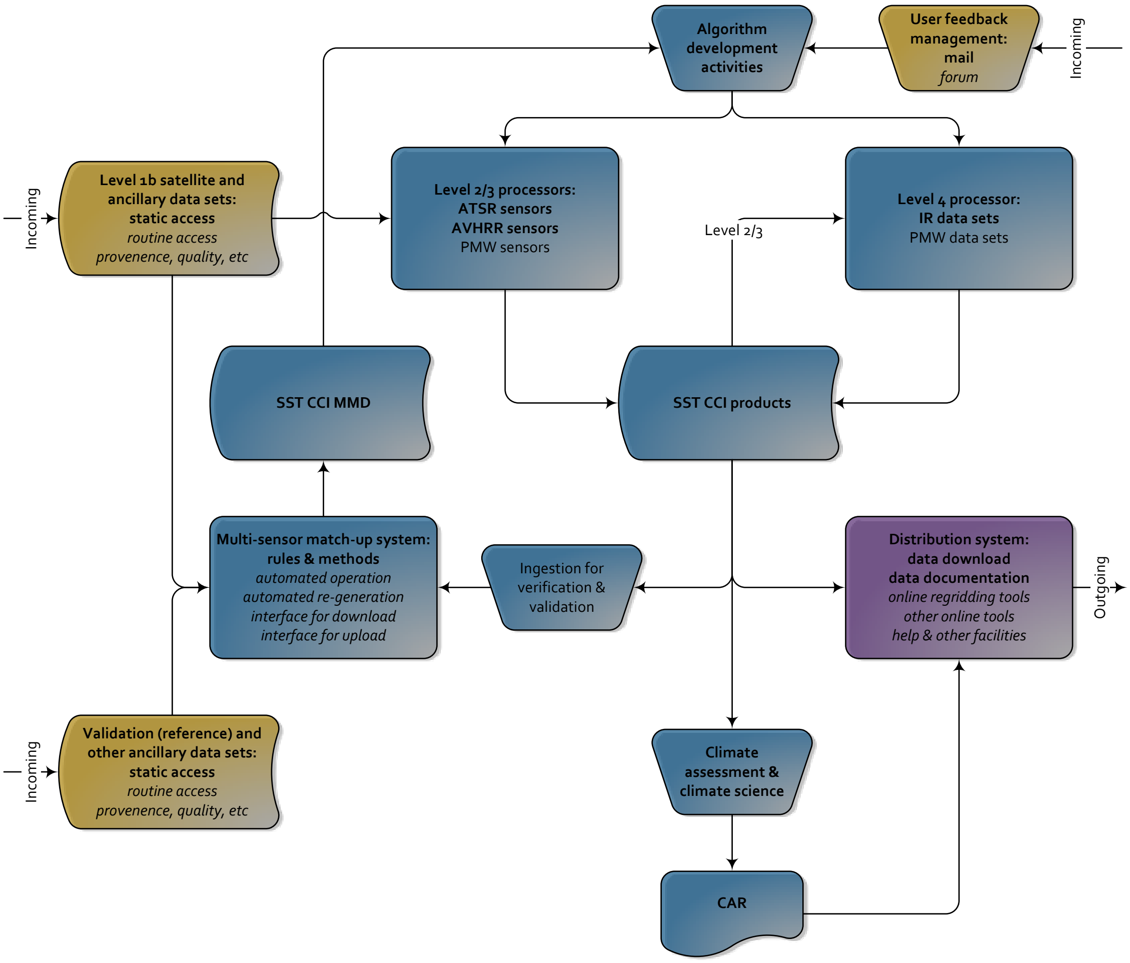

Algorithm development activities
User feedback management: mail *forum*
Incoming
Level 1b satellite and ancillary data sets: static access *routine access provenence, quality, etc*
Incoming
Level 2/3 processors: ATSR sensors AVHRR sensors PMW sensors
Level 2/3
Level 4 processor: IR data sets PMW data sets
SST CCI MMD
SST CCI products
Multi-sensor match-up system: rules & methods *automated operation automated re-generation interface for download interface for upload*
Ingestion for verification & validation
Distribution system: data download data documentation *online regridding tools other online tools help & other facilities*
Outgoing
Validation (reference) and other ancillary data sets: static access *routine access provenence, quality, etc*
Incoming
Climate assessment & climate science
CAR

Figure 2-2: High-level decomposition of the operational 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, EUMETSAT AVHRR FRAC Level 1 and MetOp internal Level 1b)
- 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.

For the prototype, the data are accessible statically. For the operational 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 operational system is specified in Section 6.2.

2.3 Multi sensor match-up system (MMS)

The MMS is a component that has been started as a novel development for the prototype. It is capable of computing match-ups between satellite data from different sensors, and of generating multi-sensor match-up datasets (MMD) of match-ups 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 match-ups have been based on pre-matched single sensor match-ups. The data types and sensors that have been or will be included are:

- Single-sensor match-ups from ATSR1, ATSR2, AATSR, MetOp AVHRR, SEVIRI, and NOAA AVHRR
- Level 1 satellite images from ATSR1, ATSR2, AATSR
- Level 1 satellite images from AVHRR from EUMETSAT MetOp-A and -B
- Level 1 satellite images from AVHRR from NOAA-06 to NOAA-19
- Level 1 satellite images from AMSR-2
- 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 SST CCI processing in MMD format

In an early period of Phase-I 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 match-ups. These MMDs were the basis for doing further work. The initial complete run of the MMS revealed several types of errors in input data and processing, which required a semi-automated approach for analysing the cause of errors and for handling them. The MMS was verified by comparing its results to the outputs of an independent implementation of the corresponding match-up and extraction algorithms. In addition, all types of outputs were manually inspected and compared with corresponding input data values on a sample basis.

The MMS databases contain all information necessary for doing queries and extracts on spatial and temporal criteria, and criteria of satellite combinations contained in match-ups. The MMS provides the infrastructure to compute match-ups on the basis of sub-random sampling or *in situ* locations and satellite data without pre-computed single sensor match-ups, and contains functions to extend the match-ups into the future with newly ingested inputs.

In Phase-I the MMS 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.

Developments of the MMS in Phase-II concentrated mainly on improving the performance and implementing an automated update procedure. The target for the automation is to have the MMS generate match-ups in parallel with the extension of the ECV time series data using current satellite data acquisitions. Generating match-ups in parallel supports the routine quality checks on the SST dataset. The operational MMS developments are specified in Section 6.3.

2.4 Level 2/3 processor

The Level 2/3 processor applies the SST retrieval and cloud detection algorithms to generate L2P and L3U SST products from Level 1 satellite brightness temperature and NWP inputs. It is based on software developed at UoE and now maintained at UoR. The software was originally developed within the GBCS [RD 181] project for applying Bayesian cloud detection techniques to SEVIRI imagery. The ARC [RD 296] project adapted it to generate a SST CDR from ATSR data, added coefficient-based and optimal estimation (first version, denoted OE1) retrieval of SST, and adaptations for bulk processing of complete satellite datasets. 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.

Work within CCI Phase-I involved several modifications to the ARC/GBCS software including modification of auxiliary files (e.g., updated coefficient files) and development of new software modules. In particular, new modules have been developed for optimal estimation retrieval (second version, denoted OE2) for ATSRs and AVHRR-GAC, and for conversion of outputs to L2P (AVHRR-GAC) and L3U (ATSR) compliant with the Product Specification Document [AD 5]. The prototype Level 2/3 processor is described in further detail in the System Prototype Description [AD 9].

Additionally, for the Phase-I demonstration period several external SST products were converted to CCI L2P / L3U format. These include: MetOp AVHRR SSTs from OSI-SAF, SEVIRI SSTs from OSI-SAF, and passive microwave SSTs from Remote Sensing Systems (RSS).

A summary of the structure of the infrared (IR) processor, showing the origin of different elements, is illustrated in Figure 2-3, blue boxes indicating new developments.

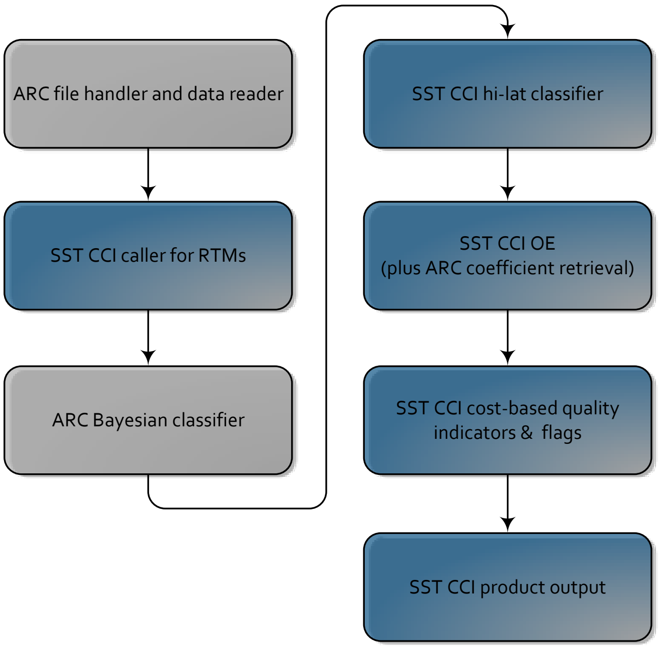

ARC file handler and data reader
SST CCI caller for RTMs
ARC Bayesian classifier
SST CCI hi-lat classifier
SST CCI OE
(plus ARC coefficient retrieval)
SST CCI cost-based quality indicators & flags
SST CCI product output

Figure 2-3: Schematic of IR processor with new SST CCI and ARC heritage components

The current L2/L3 processor can operate on AVHRR pre-processed NOAA GAC L1b* (NOAA-6 to NOAA-19 + Metop-A), EUMETSAT FRAC L1b (Metop-A and Metop-B) AVHRR data as well as Envisat-formatted ATSR L1b data. It interpolates auxiliary NWP (ERA-interim data) to satellite swath sensor resolution.

An SST CCI hi-latitudes classifier for AVHRR GAC (NOAA-12 onwards) data is applied. Cloud detection is performed by the latest Bayesian Cloud Detection algorithm for AVHRR FRAC and ATSR data. For AVHRR GAC data the Bayesian Cloud Detection is not fully supported, instead the CLAVR/x algorithm is applied.

The SST retrieval algorithm applied is depending on the input sensor namely ARC retrieval is used for ATSR and a smoothed optimal estimation is used for AVHRR GAC and FRAC data. The L2/3 processor generates both L2P and gridded L3U outputs.

Since the generation of the EXP1.2 dataset, the following improvements have been implemented:

- Support for early AVHRR sensors (NOAA-6 to NOAA-11)
- Support Eumetsat AVHRR FRAC data
- Support L2P and L3U output for all sensors
- Common quality level flagging is applied across all sensors
- Include quality level 2 through 4 data in outputs
- Add atmospheric correction smoothing

The operational 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].

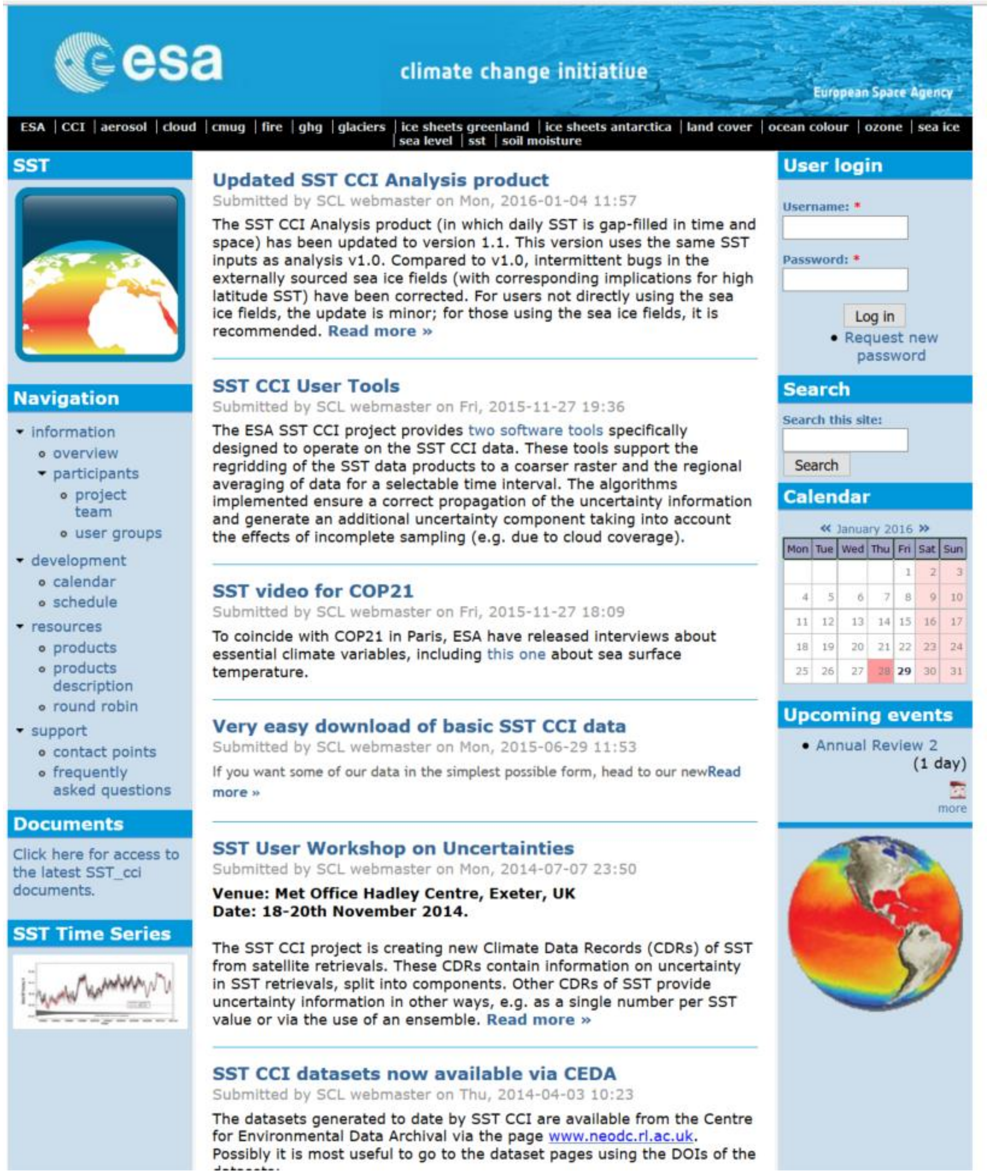
The operational Level 4 processing and production is specified in Section 6.2.

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 CEDA, using the SST-CCI project workspace. 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-II of the project, as outlined in the systems requirement document [AD 9].

The CDRP is linked from the SST CCI website, but physically served from a remote FTP service hosted by CEDA. A simple registration on the CEDA 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 are available for download from the SST CCI project website, including all ancillary data required. Integration of these functions into a web service, taking account and working with the CCI Portal project, is required for the operational system to be developed in Phase-II of the project [AD 9].



Updated SST CCI Analysis product
Submitted by SCL webmaster on Mon, 2016-01-04 11:57

The SST CCI Analysis product (in which daily SST is gap-filled in time and space) has been updated to version 1.1. This version uses the same SST inputs as analysis v1.0. Compared to v1.0, intermittent bugs in the externally sourced sea ice fields (with corresponding implications for high latitude SST) have been corrected. For users not directly using the sea ice fields, the update is minor; for those using the sea ice fields, it is recommended. **Read more »**

SST CCI User Tools
Submitted by SCL webmaster on Fri, 2015-11-27 19:36

The ESA SST CCI project provides two software tools specifically designed to operate on the SST CCI data. These tools support the regridding of the SST data products to a coarser raster and the regional averaging of data for a selectable time interval. The algorithms implemented ensure a correct propagation of the uncertainty information and generate an additional uncertainty component taking into account the effects of incomplete sampling (e.g. due to cloud coverage).

SST video for COP21
Submitted by SCL webmaster on Fri, 2015-11-27 18:09

To coincide with COP21 in Paris, ESA have released interviews about essential climate variables, including this one about sea surface temperature.

Very easy download of basic SST CCI data
Submitted by SCL webmaster on Mon, 2015-06-29 11:53

If you want some of our data in the simplest possible form, head to our new**Read more »**

SST User Workshop on Uncertainties
Submitted by SCL webmaster on Mon, 2014-07-07 23:50

Venue: Met Office Hadley Centre, Exeter, UK
Date: 18-20th November 2014.

The SST CCI project is creating new Climate Data Records (CDRs) of SST from satellite retrievals. These CDRs contain information on uncertainty in SST retrievals, split into components. Other CDRs of SST provide uncertainty information in other ways, e.g. as a single number per SST value or via the use of an ensemble. **Read more »**

SST CCI datasets now available via CEDA
Submitted by SCL webmaster on Thu, 2014-04-03 10:23

The datasets generated to date by SST CCI are available from the Centre for Environmental Data Archival via the page www.neodc.rl.ac.uk. Possibly it is most useful to go to the dataset pages using the DOIs of the datasets:

Click here for access to the latest SST_cci documents.

Mon	Tue	Wed	Thu	Fri	Sat	Sun
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Figure 2-4: Entry page to SST CCI website

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.

With the CCI Open Data Portal project being kicked-off in 2015, the distribution modes and the web-presentation of the SST CCI project will be re-considered. The CCI Open Data Portal is proposed to be the single access point for all CCI projects. The SST CCI team is in constant contact with members of the data portal team. It is expected that all CDR data sets will be delivered to the public using the Portal facilities, once the infrastructure for the portal has been implemented and tested. SST CCI tools for re-gridding and sub-setting will be made available to the Open Data Portal. It is planned to implement the tools as a web-based service, allowing users to pre-process the data before download.

2.7 Activities in SST CCI

2.7.1 Algorithm development for SST retrieval (Phase-I)

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

Earlier optimal estimation (OE) retrieval had by design been tied to the calibration of *in situ* observations, mostly from drifting buoy observations because of their availability throughout the 1991 to 2010 period.

Optimal estimation methods require use of a fast radiative transfer model (RTM), and give low-bias (GCOS compliant) SSTs only, 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 OE retrieval is illustrated in

Figure 2-5 (blue boxes indicate new SST CCI developments).

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

2.7.2 Other algorithm developments (Phase-I)

Besides the algorithm developments for SST retrieval summarised in Section 2.7.1, work was carried out 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 facilitated the algorithm development and validation tasks.

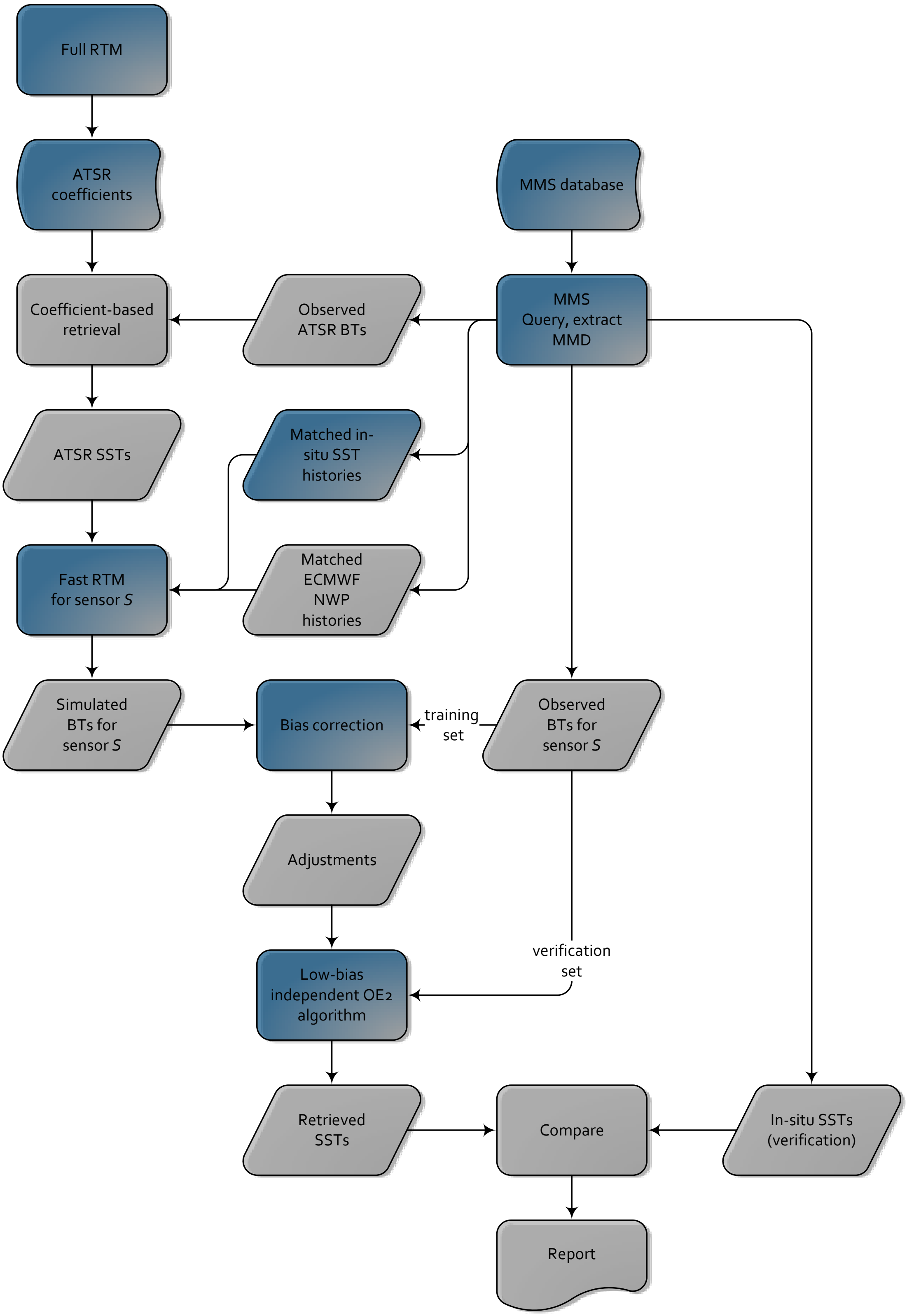

Full RTM
ATSR coefficients
MMS database
Coefficient-based retrieval
Observed ATSR BTs
MMS Query, extract MMD
ATSR SSTs
Matched in-situ SST histories
Fast RTM for sensor *S*
Matched ECMWF NWP histories
Simulated BTs for sensor *S*
Bias correction
training set
Observed BTs for sensor *S*
Adjustments
verification set
Low-bias independent OE2 algorithm
Retrieved SSTs
Compare
In-situ SSTs (verification)
Report

Figure 2-5: Development of *in situ* independent OE2 retrieval for AVHRR and SEVIRI

2.7.3 Round-robin algorithm comparison and selection (Phase-I)

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 (the “Round Robin”) followed by selection of algorithms according to criteria defined in the Product Validation Plan [AD 10].

The exercise was open over a four-month period ending 31 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) match-up 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 was 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 an 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.

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 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 match-up 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 [AD 9].

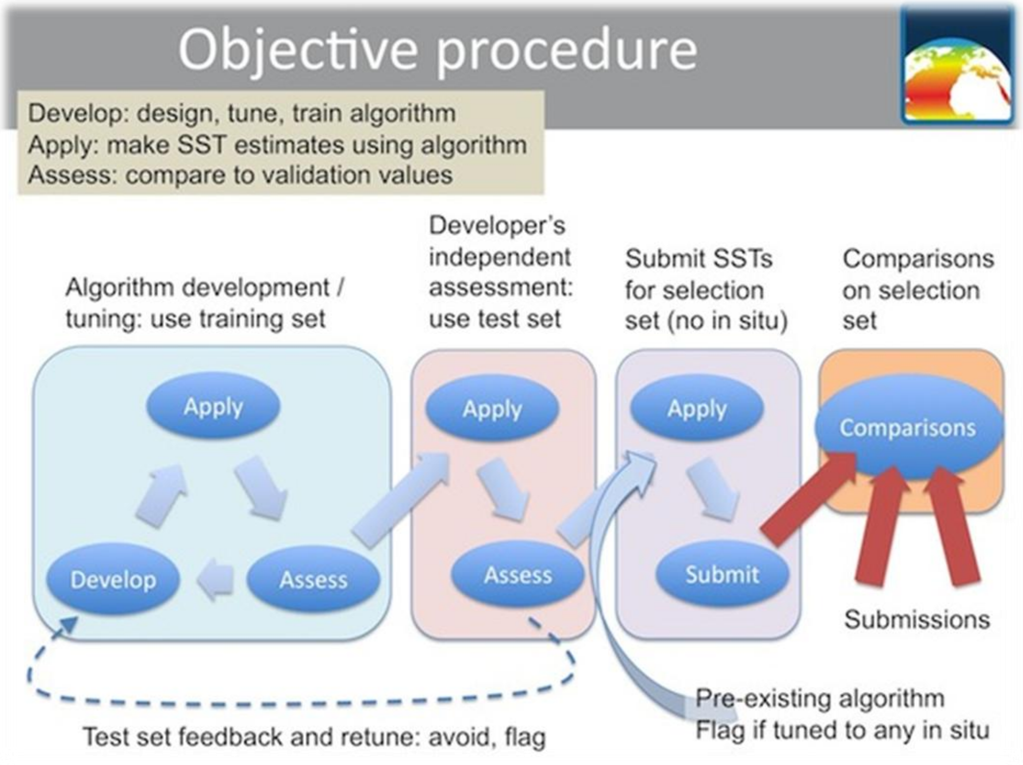

Objective procedure
Develop: design, tune, train algorithm
Apply: make SST estimates using algorithm
Assess: compare to validation values
Algorithm development / tuning: use training set
Developer's independent assessment: use test set
Submit SSTs for selection set (no in situ)
Comparisons on selection set
Apply
Develop
Assess
Apply
Assess
Apply
Submit
Comparisons
Submissions
Test set feedback and retune: avoid, flag
Pre-existing algorithm
Flag if tuned to any in situ

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

2.7.4 System verification (Phase-I)

The verification activities in SST CCI Phase-I are fully described in the System Verification Report (SVR). The activities covered 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 2.7.3) had been an early activity within the project.

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

2.7.5 Product validation and climate assessment (Phase-I)

SST CCI Level 2, Level 3 and Level 4 products were independently validated using high-quality SST measurements made *in situ* from a number of sources. In addition, the SST CCI Level 4 products were 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 also includes other kinds of assessment, as detailed in the Product Validation Plan [AD 10].

2.7.6 Migration from ECDF to CEMS (Phase-II)

As a response to the new cardinal requirements for Phase-II, the Project Team decided to migrate the components of the distributed prototype system at ECDF and Met Office to CEMS. Migrating the ECDF prototype started in the last month of Phase-I and was completed in the second month of Phase-II. All development and production tasks for the early months in Phase-II were conducted at CEMS.

Migrating the OSTIA L4 production system from MetOffice to CEMS is currently being implemented.

2.7.7 Pre-processing of AVHRR-GAC (Phase-II)

Rather than using the original AVHRR-GAC Level-1b data from NOAA, the Project Team decided to pre-process to Level-1c before usage in the MMS and SST production systems. Pre-processing of AVHRR-GAC Level-1 data tackles several problems:

- The different AVHRR data formats used by NOAA are harmonised to use the same data types for variables and other information.
- The Level-1b data are supplemented with a cloud mask resulting from a common cloud screening algorithm at Level-1c. In early Phase-II the CLAVR-X mask has been used, but improved algorithms can be used later.
- Using the pre-processed AVHRR data as input to the MMS and SST production system simplifies creating updated MMD and future re-processing datasets; the pre-processor will apply improved AVHRR calibration wherever this is available.
- Original AVHRR data are often affected by missing lines. In the pre-processing, missing lines are detected and filled. Pre-processed data do not exhibit missing lines; formerly missing pixels are inserted and flagged, but geo-location information is properly calculated.
- Orbit overlaps as information is received from different stations can be detected.

The Level-1c data format is netCDF with metadata complying with Climate Forecast (CF) conventions. In this manner the AVHRR data are harmonised and are accessible to users that are not familiar with the original formats.

2.7.8 Summary of project activities

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

Table 2-1: Summary of project activities

Activity	Prototype system (Phase-I)	Operational system (Phase-II and beyond)
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), triggered by ECV updates with new satellite acquisitions.
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. For Metop FRAC, involves implementation of Bayesian cloud detection. For SLSTR, preparatory activities will be undertaken.
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, leading to automated generation of standard validation products for a given reprocessing version.
Climate assessment	To be carried out using products and MMS outputs	Requires automatic generation of standard assessment metrics on generation of new products

3. TECHNICAL OVERVIEW OF THE SST CCI SYSTEM

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

3.1 Scientific and technical context

The SST CCI system has two aims. The most obvious is to produce “climate data records” for sea surface temperature. SST is an “essential climate variable”, which means that to understand and track climate variability and change, high quality records of how SST changes over time are needed. With sufficient remote sensing know-how, this can be achieved using infrared imagery from Earth observing satellites.

The project has a limited lifetime, but the need for climate data records will not disappear. The second aim therefore is, in the process of delivering data, to build a software system that can be sustained to provide data in future. Irrespective of future funding, in Phase-II the project will develop its software to a stage of being “pre-operational”.

The system will not be able to be simply run unattended, even in a sustained (or "operational") mode, without maintenance of the science, and cyclic improvement to used techniques. Satellites come and go, and getting new observations to the standards defined for inclusion in the SST CDR requires ongoing Earth observation development work. Moreover, over time, it will be possible to make CDRs made better, and therefore the whole dataset should be reprocessed consistently in the improved manner.

The SST CCI system includes both software and human experts, in a cycle of sustained production of data (as the observations come in) and periodic reprocessing. The cycle is illustrated in

Figure 3-1 and further 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).

Different groups of users provide feedback. Users and the Development Team both contribute to problem identification and options for improvements. Problems are selected for being solved. New versions of inputs, upcoming data from new sensors, and emerging new requirements initiate the short cycle (→ SST-SR-5230 Agile requirements selection).

In the short cycle, the Development Team solves newly identified problems by analysis, prototyping, internal reprocessing and validation. This may already require a reprocessing of the full CDR (→ SST-SR-5270 Short development cycle). When the Development Team decides to release a new CDR version, the source code is frozen and the full CDR is reprocessed, if necessary. The new version is made available for validation by external validators and expert users as a release candidate (→ SST-SR-5250 Version decisions, SST-SR-5260 Version release process). For a certain period of time the release candidate exists concurrently with the previous version to allow feedback, optional improvements, and switching over (→ SST-SR-5255 Overlapping versions, SST-SR-5265 Version management).

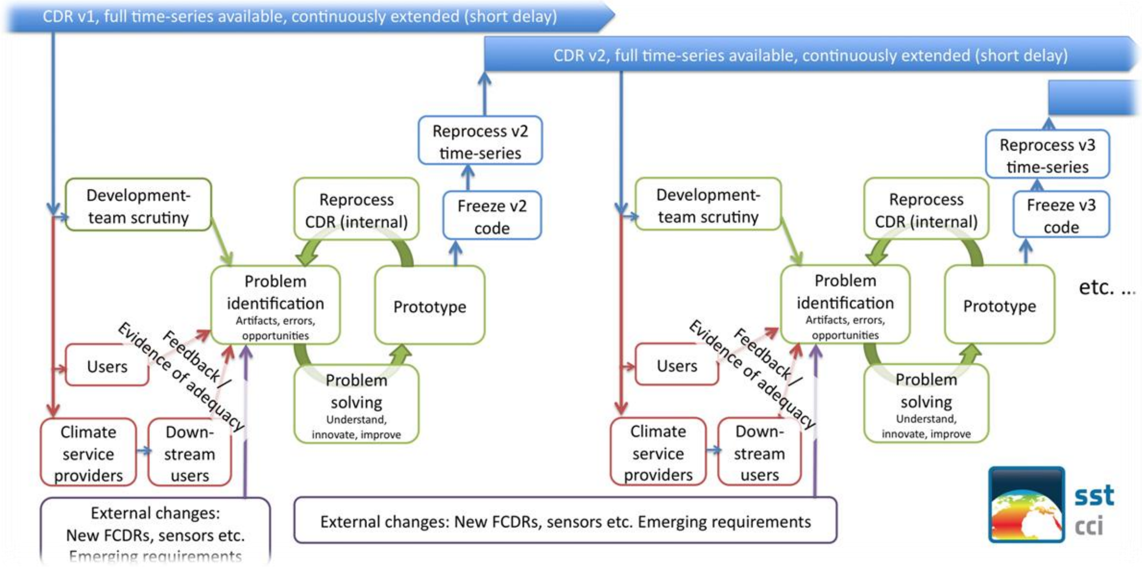

CDR v1, full time-series available, continuously extended (short delay)
CDR v2, full time-series available, continuously extended (short delay)
Development-team scrutiny
Reprocess CDR (internal)
Freeze v2 code
Reprocess v2 time-series
Problem identification
Artifacts, errors, opportunities
Prototype
Problem solving
Understand, innovate, improve
Users
Evidence of adequacy
Feedback /
Climate service providers
Down-stream users
External changes: New FCDRs, sensors etc. Emerging requirements
External changes: New FCDRs, sensors etc. Emerging requirements
Development-team scrutiny
Reprocess CDR (internal)
Freeze v3 code
Reprocess v3 time-series
Problem identification
Artifacts, errors, opportunities
Prototype
etc. ...
Problem solving
Understand, innovate, improve
Users
Evidence of adequacy
Feedback /
Climate service providers
Down-stream users
sst cci

Figure 3-1: Improvement cycles driven by user feedback and external changes

Technically, the SST CCI system provides certain interfaces to the outer world. Figure 3-2 illustrates the SST CCI system in its full context, as explained in the following paragraphs.

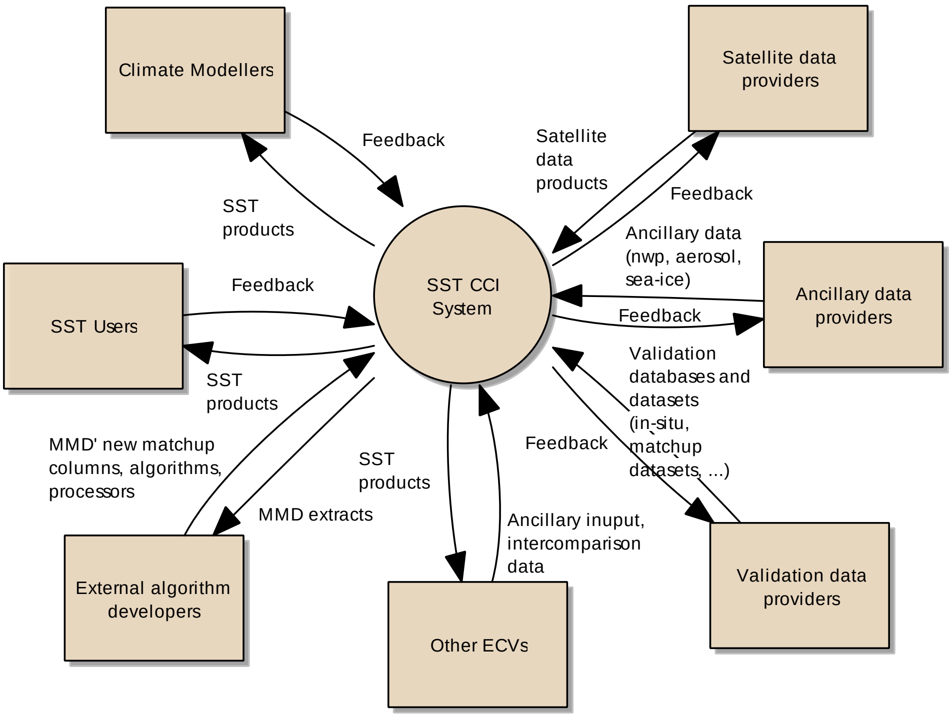

Climate Modellers
Satellite data providers
Feedback
Satellite data products
SST products
Feedback
Ancillary data (nwp, aerosol, sea-ice)
SST Users
Feedback
SST CCI System
Ancillary data providers
Feedback
SST products
Validation databases and datasets (in-situ, matchup datasets, ...)
MMD' new matchup columns, algorithms, processors
SST products
Feedback
MMD extracts
Ancillary inuput, intercomparison data
External algorithm developers
Other ECVs
Validation data providers

Figure 3-2: Context of the SST CCI system with data providers and users

Climate researchers and users from the SST community receive data products along with necessary documentation and other information. The SST CCI Development Team (see below) proactively 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.

The SST CCI Development Team consists of experts in SST that can take the scientific leadership of the project, but also includes software developers and operators. The Development Team supports, directs, and prioritizes operations within an agreed scope, and ensures developing SST CDRs to satisfy user requirements. The Development Team has a mandate defined by terms of reference to support evidence-driven improvements in the development cycle. Note that the Development Team is an inherent part of the SST CCI system and therefore does not appear in Figure 3-2 (→ 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).

Validation is a two-step activity: Firstly, internal validation is performed as a quality assurance step, also detecting possible anomalies, by the Development Team before a new CDR is provided to users. Secondly, independent validation by external expert users provides additional feedback.

External algorithm developers may also carry out algorithm development and validation. The SST CCI system foresees this activity as an interaction where the MMD managed by a MMS are exchanged between the Development Team and external users.

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 CCI output data on the other end. Some interfaces have several endpoints and are split into sub-interfaces. Interfaces for historical satellite data are listed because of future reprocessing. In order to avoid duplication with the DARD [AD 4] the list of sources for the validation data are not repeated. DARD Sections 6 and 7 provide a comprehensive list including access information. Note that the interfaces to data providers may change during the project due to new data sources becoming available or different needs by 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**	**various**	**DARD 4**
Ifc-1.1	ATSR input data interface	DECC/NERC/ ESA	Direct access from CEMS	DARD 4.1
Ifc-1.2a	AVHRR GAC L1 input data interface	NOAA CLASS, University of Maryland	FTP	DARD 4.2
Ifc-1.2b	AVHRR GAC L1 input data interface	NOAA CLASS	FTP	
Ifc-1.3a	MetOp A and B AVHRR L1B input data interface	Eumetsat	Direct access from CEMS	
Ifc-1.3b	MetOp A and B IASI input data interface	Eumetsat	To be defined	

Ifc-1.4	Sentinel 3 SLSTR	ESA	Direct access from CEMS	
Ifc-2	**Ancillary input data interface**	**various**	**various**	**DARD 5**
Ifc-2.1	ERA-Interim	ECMWF	Direct access from CEMS	DARD 5.1
Ifc-2.6	OSI-401 SSM/I Sea ice concentration maps OSI-409 Sea ice concentration reprocessing	OSI SAF	FTP from OSI SAF High Latitude processing centre	DARD 5.3, 5.4
Ifc-2.7	TOMS OMI GOME-1 GOME-2 Absorbing aerosol index	NASA GSFC, TEMIS	Download from NASA GSFC and TEMIS websites	DARD 5.5
Ifc-2.8	SAGE II Aerosol	NASA	NASA	
Ifc-3	**Validation input data interface**	**various**	**FTP pull**	**see below**
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**	
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**	**various**	**see below**
Ifc-6.1	SST CCI web interface	SST CCI	HTTP	Section 6.1.5 below
Ifc-6.2	SST CCI feedback interface	SST CCI	Email, web forms (issue tracking)	Section 6.1.5 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 achieve its aims, the SST CCI system provides three high level functions:

- Producing “world-class” SST CDRs
- Disseminating CDRs, documentation, and other information
- Improving SST retrieval algorithms as a response to feedback from users and other external events

For production, the focus is on offline processing (and re-processing) of complete missions, repeated as necessary due to requirements or requests for improvements from users or other external reasons. Essential functions for production are:

- Storing data to hold and make available inputs, intermediates, outputs and auxiliary data
- Processing, i.e. transforming input data into outputs
- Controlling processing workflows and massive parallel bulk production
- Quality checking, automated and visual
- Creating match-ups
- Creating inter-comparison and validation reports
- Generating metadata to distinguish dataset versions, writing other dataset documentation
- Ingesting new satellite and corresponding auxiliary data for immediate processing

Managing bulk production is essential. The manner in which production is managed directly influences the agility of improvement cycles and the quality of outputs.

For dissemination, the focus is on services for climate users. Essential functions for dissemination are:

- Providing project information, data discovery, catalogue service
- Providing (bulk) data access, online and optionally on media
- Providing data customisation tools available as service and optionally as installable software
- Providing validation support, providing access to reference data and reports
- Providing access to documents on products and algorithms, example products
- Handling feedback, tracking issues, communicating with users (forum and e-mail)
- Long-term preservation of CDRs and all information necessary to understand them

Serving climate users rapidly with the data and information they need is a key aspect. Serving and interacting with the SST community is essential to get feedback on data and methods support for the SST CCI project.

For algorithm improvement, the focus is on establishing an environment to exchange and compare processor versions and configurations with little effort. Essential functions are:

- Testing new “snapshot” processor versions. The SST CCI system provides a processor interface where new processor versions are “plugged in”
- Creating or extending multi-sensor match-up datasets
- Validating retrieval results
- Accessing full-mission data, capability of reprocessing large sets of input data when required
- Managing versions of processors, configurations, and data, with documentation of what has been tested, updated, and released

The overall agility of the SST CCI workflow depends on a responsive, easy to use and efficient infrastructure available for day-to-day use by the SST CCI Development Team.

3.3 High-level decomposition and main internal interfaces

The functions declared in the previous subsection are implemented by functional components. These functional components can be grouped into subsystems. The following paragraphs provide an overview of the high-level components of the SST CCI system, which provide a starting point for drafting the main operational scenarios (see Section 4). A detailed specification with all components, functions and interfaces is given in Section 6.

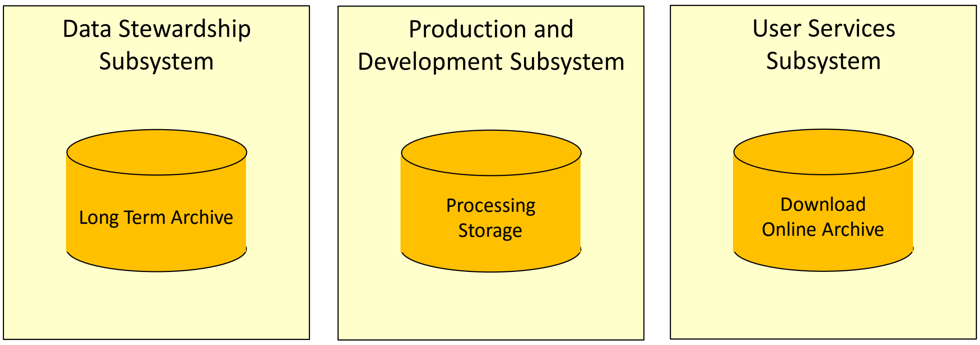


Figure 3-3: Subsystems for production, user services and long term archiving

Figure 3-3 identifies the three high-level subsystems: data stewardship, production and development, and user services. Note that these subsystems encapsulate Earth Observation data in different manners:

- The long-term data archive within the data stewardship subsystem facilitates reliable long-term storage of data and all data representation information required for using it. Typical uses constitute request of data in large chunks, for example a complete product set of a certain version for a time period of several years.
- Processing storage of the production and development subsystem is accessed by data processors and facilitates highest data throughput. Processing storage is protected against direct and concurrent access by external users.
- The online archive within the user services subsystem provides direct and concurrent download access for external users.

Different use cases imply different features and specialisations (or optimisations). Specialisation and separation of concerns does not prohibit integrating two subsystems into one or hosting all subsystems by one institution. The (data) interfaces between subsystems facilitate separation.

Figure 3-4 illustrates the individual components of the three subsystems. The data stewardship subsystem stores the input data required by SST CCI and its outputs. It provides long-term data preservation and bulk data provision on request. Depending on whether some functions are provided externally, separate components for the preservation of input and output data 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).

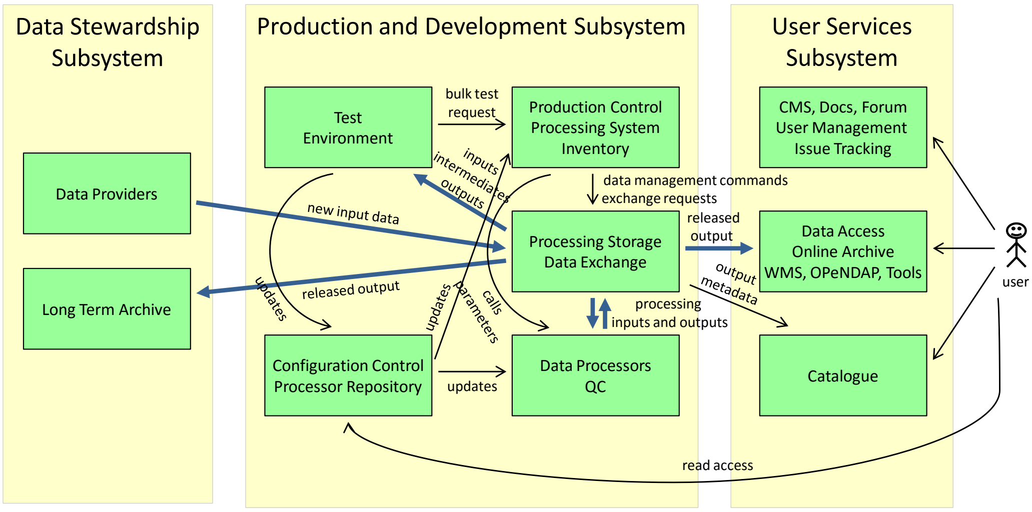

Data Stewardship Subsystem
Production and Development Subsystem
User Services Subsystem
Data Providers
Long Term Archive
Test Environment
bulk test request
Production Control Processing System Inventory
CMS, Docs, Forum User Management Issue Tracking
inputs intermediates outputs
data management commands exchange requests
new input data
released output
Processing Storage Data Exchange
Data Access Online Archive WMS, OPeNDAP, Tools
user
released output
output metadata
updates
updates
parameters
calls
processing inputs and outputs
Configuration Control Processor Repository
updates
Data Processors QC
Catalogue
read access

Figure 3-4: Components of the SST CCI subsystems

The production and development subsystem has components for production control, processing storage and data processors, which provide the essential infrastructure for processing. A test environment with read access to all data and the option to use the production infrastructure for bulk tests facilitates rapid development. For supporting bulk production, an intermediate software layer (middleware) implements the main functions of the processing system and the processing storage. The middleware makes it easier for software engineers to develop applications; examples are generic batch queuing and scheduling services and cluster management software. Depending on the middleware, the two functions of processing control and data storage may be tightly coupled.

The user services subsystem consists of at least three components: a web representation, data access by users, and a catalogue. The web representation includes a user forum and an issue tracking software. Data access offers different protocols and supports online re-gridding 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 requested functionality will be covered by an operational version of the Open Data Portal.

The main internal interfaces of the SST CCI system are those for data exchange between production and user services and between production and a long-term stewardship system. Table 3-2 lists the internal interfaces with references to the description of the exchanged data items. Interfaces between a subsystem's components are described in 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	Direct access to data stewardship subsystem (BADC/CEDA)	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 or SFTP pull by SST CCI user services	PSD [AD 5]

4. SST CCI OPERATIONAL SCENARIOS

The main SST CCI operational scenarios are user data access and information, processing and validation of the full or partial CDR, and algorithm improvement. The specification of the elements facilitating these scenarios is elaborated in detail in Section 6.

4.1 User roles

Two major groups of users who interact with the SST CCI system are considered. These are the (internal) Development Team and the (external) users, including climate users and the SST community. Both are actors in the operational scenarios. Users can take different roles, depending on how they use the system.

- Climate users
  - Are interested in consistent stable datasets, with occasional version upgrades
  - Need certain data formats compatible with their models
  - Provide feedback resulting from the use of SST CCI data in climate modelling
- SST community users
  - Are interested in the best SST products
  - Are themselves skilled in retrieving SST and provide feedback from their own external validation activities
  - Provide proposals for alternative methods and improvements
  - Are invited to perform (external) algorithm development and comparison with multi-sensor data sets
- Development Team with scientists, operators, and system engineers
  - Pushes forward SST CCI, decides on requirements to analyse and implement, decides which algorithms are to be tested and selected
  - Maintains the system, manages the data archives, initiates and monitors production
  - Interacts with users to collect feedback and new requirements
  - Cyclically improves the system and releases new CDR versions
- ESA
  - Supervises the project or makes oversight arrangements
  - Decides on the overall direction of the project

Data providers are not considered as users of the system. Nevertheless, the Development Team interacts with data providers and gives feedback on input data quality in order to improve the inputs.

4.2 User information and data access

Figure 4-1 illustrates the typical scenario for information and data access by a climate or SST community user. Here, a new user visits the SST CCI portal, informs himself about the project, and eventually requests SST CDR data for use in his project.

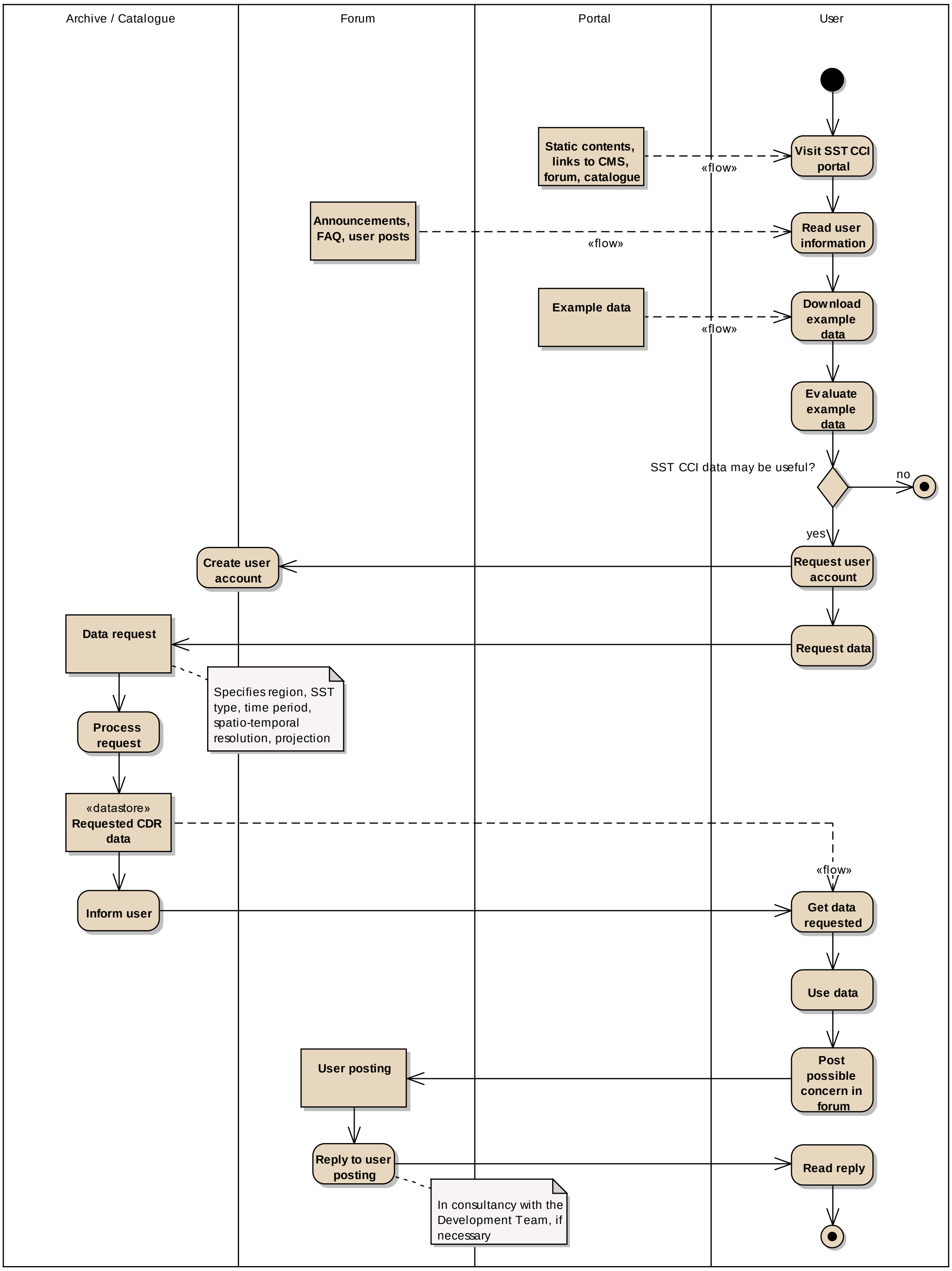


Figure 4-1: User information, data access, and user feedback handling activities

Static web pages provide links to dynamic contents in a content management system (CMS), a 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 their specific application or not.

If the SST CDR is considered useful, the user requests an SST CCI account, which is valid for both the data archive and the forum. After the account has been created and 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, the time period of interest, the required spatial and temporal resolutions, and the map projection.

A data request is processed, and when completed, the user is informed. When 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 posted report is replied to, in consultancy with the Development Team, if necessary. The user reads the reply and may be satisfied. If concerns or issues remain, the user may start a discussion with the forum. Eventually the user feedback might point out an issue of the CDR processing algorithm, which then is 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 necessarily 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, then available in the CMS.

4.3 Processing and validation

The typical scenario for the activities carried out during the processing and the internal (or 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 (see Section 4.4). Here reprocessing and processing are used as synonyms.

The integration of reprocessing into the algorithm improvement cycle is considered in Figure 4-2, where the starting point represents the activity of algorithm improvement within the cycle, which precedes the reprocessing activity itself.

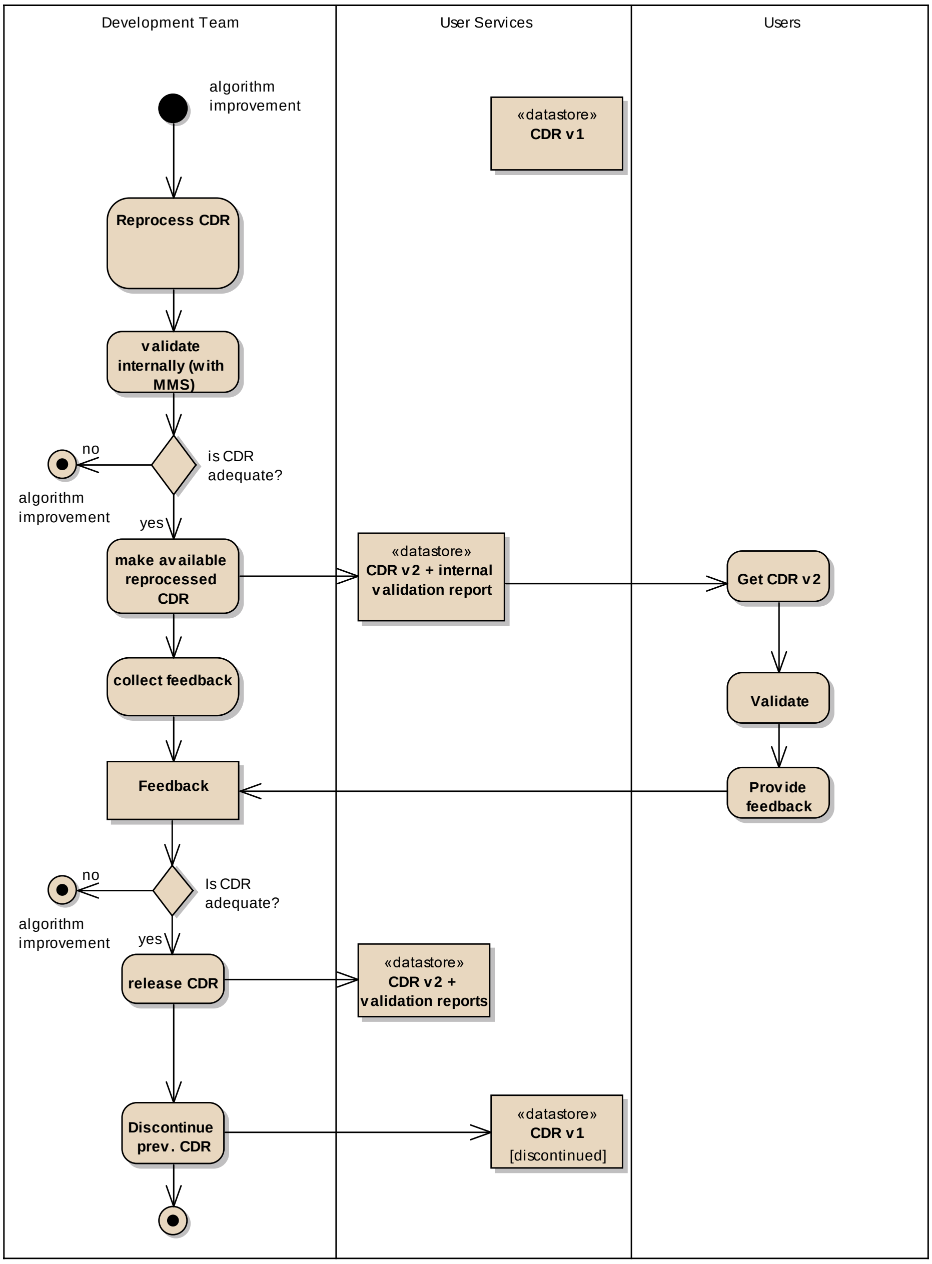


Figure 4-2: Reprocessing and validation activities

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 by means of the MMS. The internally validated 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.

A new CDR does not necessarily replace its predecessor. External expert users may validate the new CDR and provide feedback to the Development Team. Feedback is to be reported in the form of validation reports and may include match-up data sets in MMD format [AD 6]. After scrutinising and evaluating the feedback from experts, the Development Team decides whether the reprocessed CDR is adequate with respect to the requirements that have initiated the improvement cycle. If it is not adequate, another improvement cycle may be triggered. Modalities are decided by the Development Team. If the new CDR is adequate, it is released along 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. After this period the previous CDR is discontinued.

Reprocessing is an internal activity of the Development Team. It is illustrated in detail in Figure 4-3 below. Either Development Team scientists or operators conduct individual actions. The initial action is to define the bulk-processing task. Team scientists specify 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)

Configuring and starting the bulk-processing task is in the responsibility of the team operators, which also ensure staging of the data needed into fast storage and schedule the production. In addition, team operators monitor production, and, if necessary, the operators handle exceptions by trying to repeat the affected part of the processing or by investigating and delegating to the team scientists. (→ SST-SR-1390 Operator).

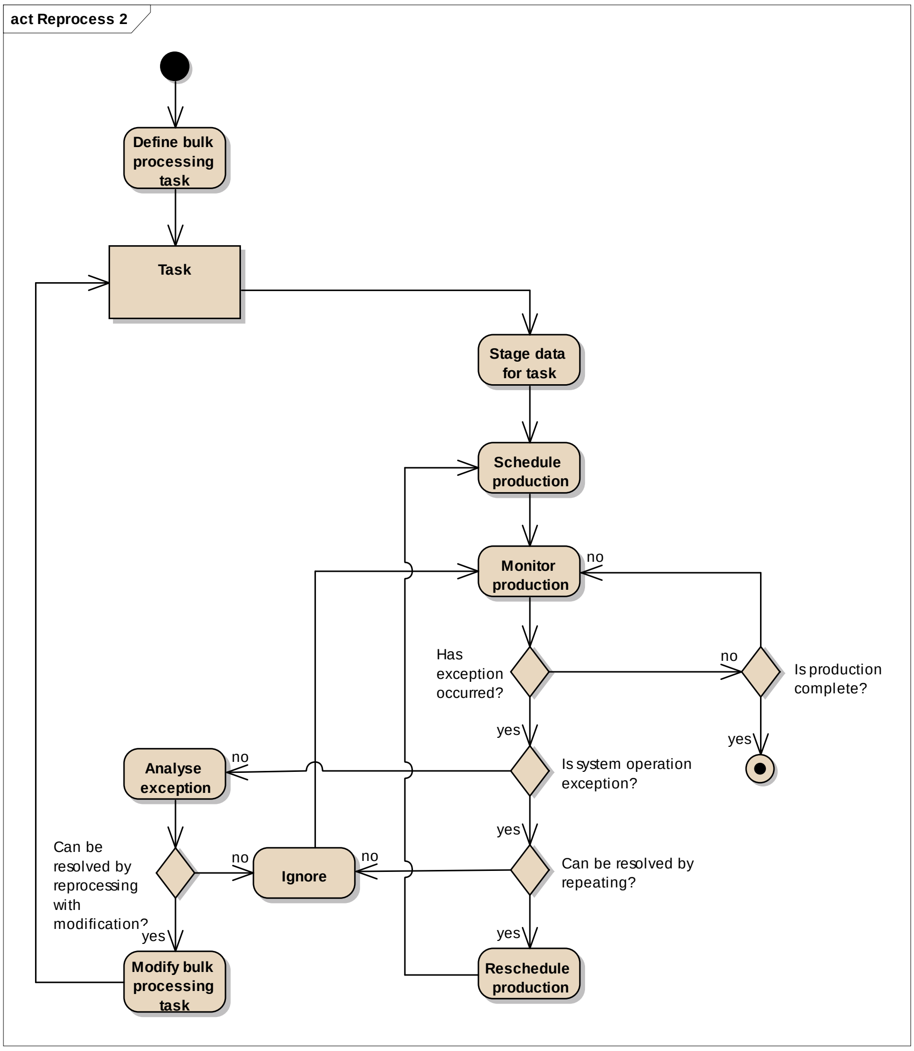


Figure 4-3: Bulk-processing activities

4.4 Algorithm improvement

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.

External experts using the SST CDR report problems to the SST CCI team by means of the user forum, possibly attached with additional *in situ* data or MMD files. During post-release activities, the Development Team scrutinises reports, investigates issues, and decides whether these shall be resolved. If an issue is to 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 tracking system 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 an issue can be resolved, any *in situ* data and MMD files attached to the problem report are ingested into the MMS. If necessary, new match-up data records are calculated and sub-scene MMD files are extracted. In a new branch of the processor software source code (which includes all production and scientific code), team scientists and developers implement the necessary changes that will resolve the problem. Then sub-scene MMD files are processed with the modified processor. Team operators ingest result MMD files into the MMS and then extract an MMD file containing all data required for validation.

The Development Team conducts the validation of the modified code’s processing results; if the results have not improved, the issue is reanalysed, starting another iteration of activities. If results have improved, the full CDR is reprocessed into a new release candidate and validated thoroughly to ensure that the new results do not deteriorate in other respects.

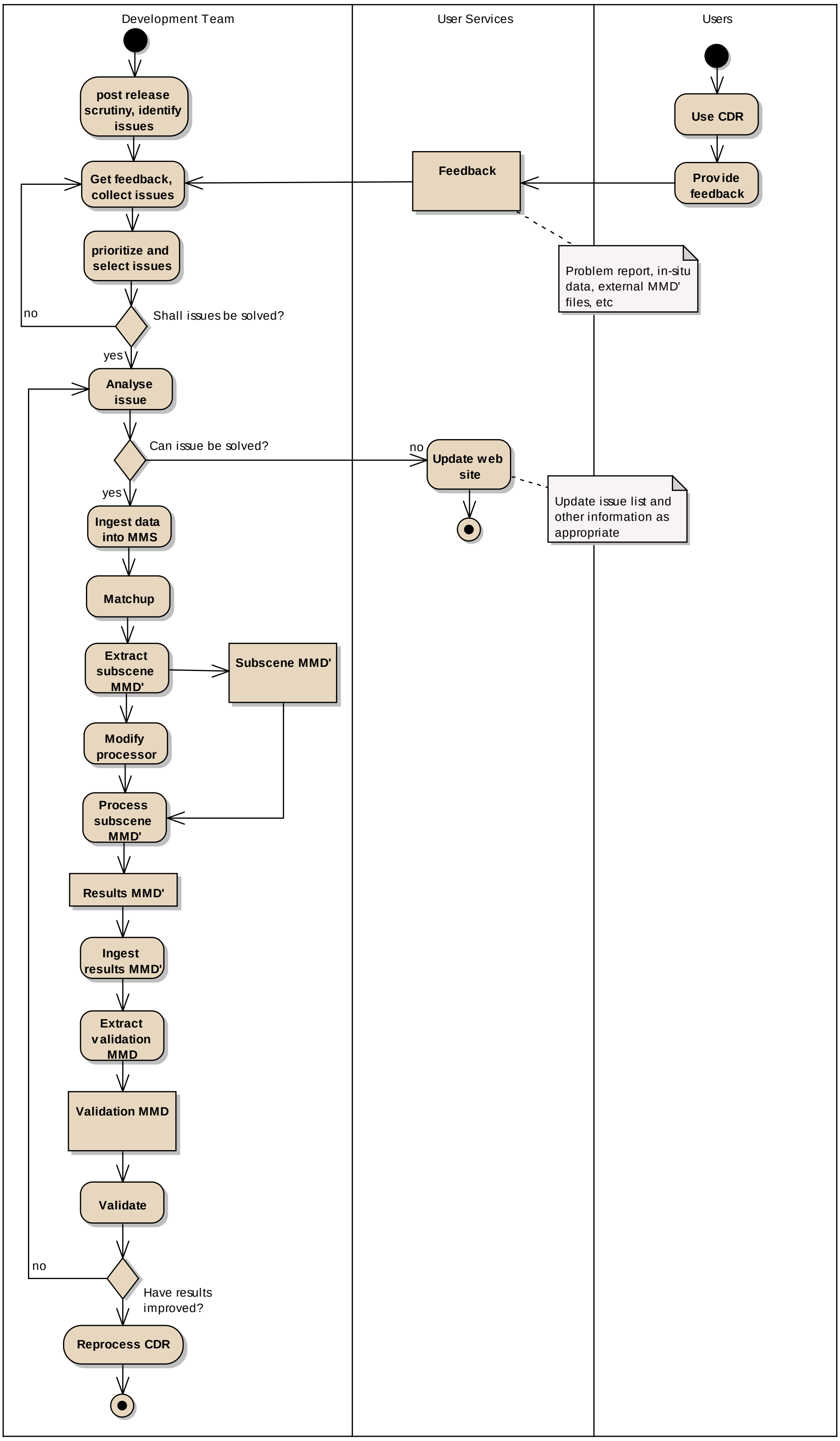


Figure 4-4: Algorithm improvement activities

5. TECHNICAL INFRASTRUCTURE DECISIONS

Section 5 of the Phase-I SSD exhibited a trade-off analysis concerning several fundamental infrastructure alternatives for the future SST CCI system. In particular:

1. To what extent shall the operational system use and be built on the prototype system? Shall the operational system be distributed like the prototype or be consolidated into a new central SST infrastructure?
2. Shall the operational system be implemented using the infrastructure of an already existing Earth Observation data processing centre?
3. What functions or subsystems of the operational system are the best candidates for sharing with other CCI essential climate variables?
4. Shall the operational system run in a high-performance cluster environment like the prototype or in a completely virtualised cloud?
5. What kind of processing middleware shall be used by the operational system? A standard grid engine or a cluster resource management layer like Apache Hadoop, which facilitates data-local processing?

Note that the infrastructure alternatives are not completely independent of each other. For example, using the infrastructure of an existing data processing centre excludes the deployment of the system into a cloud, and also the question of the processing middleware may already be decided.

The following sections describe the infrastructure decisions that were made between the end of Phase-I and the beginning of Phase-II, some of which have already been realised.

5.1 Consolidating the distributed prototype

As explained in Section 2.1, the Phase-I prototype system was distributed among ECDF (Level 2/3 development and production), the UK Met Office (OSTIA Level-4 development and production) and the CMS (developments and production for integrating MetOp AVHRR and SEVIRI). For Phase-II the project team decided to migrate the ECDF and Met Office components to the CEDA/CEMS infrastructure. Both components will remain separated logically, but will share the CEMS infrastructure (see Section 5.2 below). The data streams between CMS and the Level 2/3 development and production in Phase-II are different than in Phase-I and new interfaces between these components will be defined in Phase-II.

Whereas migrating the Level 2/3 components formerly running at ECDF has required minimal effort, developing the stand-alone version of the OSTIA system, which then can be migrated to different environments easily, is in progress.

By deciding to consolidate two of the three components of the distributed prototype system into the same infrastructure environment, the project team has reduced the external data traffic and has prepared the ground for speeding up reprocessing and algorithm improvement workflows.

5.2 Migrating to the CEDA/CEMS infrastructure

CEDA provides the JASMIN and CEMS data processing environments. The JASMIN super-data-cluster is deployed on behalf of NCAS at the STFC Rutherford Appleton Laboratory (RAL) and supports the data analysis requirements of the UK and European climate and earth system modelling community. It consists of multi-Petabyte (Pb) fast storage co-located with data analysis computing facilities, with satellite installations at Bristol, Leeds and Reading Universities. It shares infrastructure with CEMS, an equivalent facility in the Earth Observation domain.

JASMIN is deployed in the e-Science department at RAL to deliver three main functions: the infrastructure for the data storage and services of CEDA, including the NCAS British Atmospheric Data Centre; an environment for data intensive scientific computation for the climate and earth system science communities; and flexible access to high-volume and complex data for the climate and earth observation communities.

Together with CEMS, a total of 4.6 Pb of fast storage is deployed at RAL, connected via its own low-latency network to the JASMIN and CEMS data computer facilities. Satellite systems at Bristol, Leeds and Reading Universities consist of significant disk (500, 100 and 150 Tb, respectively) coupled with additional computer resources. It is envisaged that virtual machines for analysis can be constructed for particular purposes at satellite sites, then migrated to the central system for processing against the major data store.

The SST CCI project uses a dedicated group workspace within CEMS. Group workspaces are portions of storage allocated for particular projects to manage themselves, enabling collaborating scientists to share network accessible disks. Users can pull data from external sites to a common cache, process and analyse their data, and where allowed, exploit data available from other group workspaces and from the CEDA archive. Besides SST CCI, the Ocean Colour CCI project has acquired a dedicated group workspace for disseminating their CDR, which facilitates both ECVs sharing a joint mini portal to their respective CDRs (see Section 6.1.3).

Group Workspaces are usually exploited in conjunction with project specific computing resources, configured and deployed as virtual machines in the JASMIN infrastructure. Such machines can generally mount their own GWS and the CEDA archive. The SST CCI project uses a single virtual machine for running the PostgreSQL database of the MMS.

It is important to understand that these workspaces are not the same as the CEDA archive. Data in a group workspace can be earmarked for ingestion into the CEDA archive, but this is a process that should be discussed directly with CEDA, it is not automated in any way.

Data within group workspaces are under the responsibility of the designated group workspace manager and are not backed up by CEDA. The manager of the SST CCI workspace is a member of the Development Team.

An “Elastic Tape” system is available, which enables group workspace managers to manage secondary copies of their data and to move less-used data out to near-line tape to enable optimal use of high performance disk space.

By deciding to migrate the SST CCI system to the CEDA/CEMS infrastructure the project has gained the immediate benefit that Eumetsat AVHHR (FRAC) are available on fast network accessible disks. ESA Sentinel-3 SLSTR data will be available when the mission has passed commissioning. By migrating to CEMS, the SST CCI system has become ready for Phase-II and the future beyond.

5.3 Sharing a common CCI portal across ECVs

As explained in Section 5.3 of the Phase-I SSD, the user services subsystem is a good candidate for sharing among several or all CCI ECVs. In fact, ESA has recently commissioned a CCI Data Access Portal project [DAP].

The CCI Portal project covers a major upgrade of the existing CCI Portal to facilitate free, open and easy access to ECV data products generated through the CCI for the international climate user community. Besides improvement and operation of the CCI programme web site and improvement of the common elements of the design of the CCI project web sites and assistance to the project teams to implement these improvements and maintain their CCI web sites, the CCI Portal project activities include, in particular, development, implementation and operation of a CCI Central Data Archive and Metadata Catalogue, and development, implementation and operation of multiple interfaces to access ECV data and metadata from the CCI Central Data Archive and Metadata Catalogue. In addition, the CCI portal project aims to provide the primary communication and promotion web site for the ESA CCI programme including a moderated facility for handling user-queries, feeding appropriate questions back to the individual CCI projects (Science Leaders or their delegates), and recording all feedback.

Our understanding of the CCI Portal Statement of Work [DAP] is that the existing SST CCI data access mechanisms are already sufficient or do require little work to provide interfaces to the future CCI Portal. The SST-CCI team is in constant contact with the CCI Portal team to facilitate the migration of services and tools to the new web-environment.

5.4 Using the CEMS data processing environment

The decision to migrate the SST CCI system to the CEDA/CEMS infrastructure (see Section 5.2) implies using the high performance data processing environment implemented at CEMS. Although the CEMS environment offers cloud computing and virtualisation of computing resources (the SST CCI project uses a dedicated virtual machine for the MMS database) the computational work for SST CCI is conducted in the cluster environment, which offers the simplest access and highest performance.

5.5 Using the LOTUS batch processing system

Using the CEMS cluster environment implies using the LOTUS batch processing system. LOTUS is a heterogeneous cluster, with mostly Intel high performance processors of different speeds and memory combinations and one large memory AMD SMP host. Batch jobs are submitted from a front-end node using the LSF scheduler, which is a kind of grid engine software. Scheduling is done so that every user gets a fair share. Figure 5-1 illustrates an example: three users (left column) have jobs in the queue (middle column) that are waiting to run on the cluster nodes (right column). As the blue user's job finishes (row T2), all three users could potentially use the two job slots that become available. However, the orange and purple users already have jobs running, whereas the blue user does not, and therefore the blue user's jobs that are run (row T3). Giving the scheduler as much information on a job as possible, such as job length and memory required can increase the chance of getting jobs run.

Exploiting data locality is not important in the CEMS environment, because all data are available from all nodes with highest performance equivalent to or even better than local disk drives.

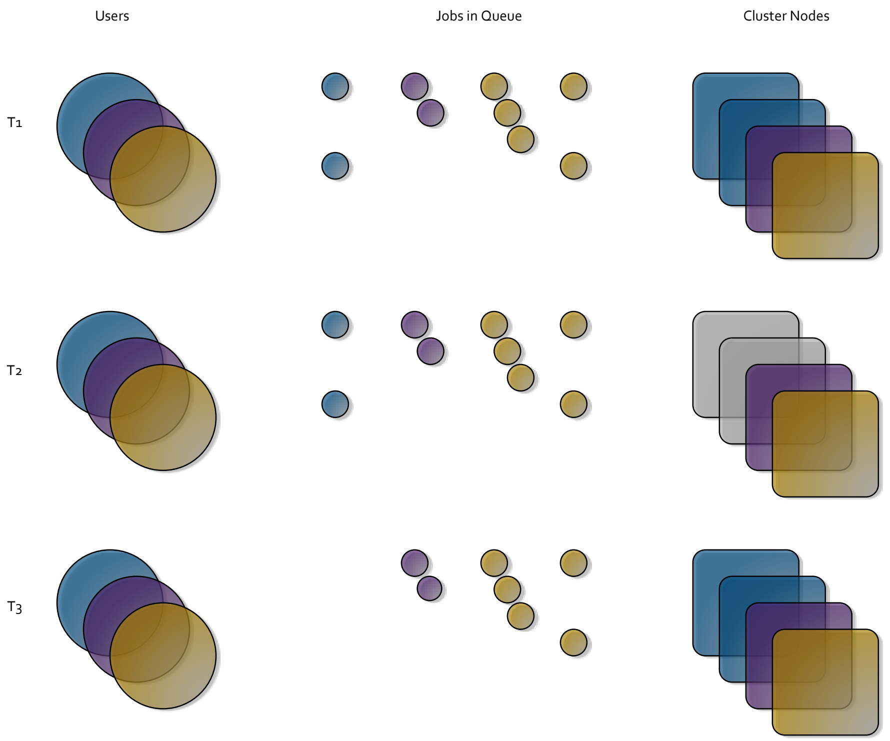


Figure 5-1: LOTUS scheduling example

6. FUNCTIONAL DESIGN

Dedicated functional components implement the functions for user services, production management and the continuous improvement cycle. 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 User services

User services constitute the functions and interfaces that external users expect and use to interact with the SST CCI system. The SST CCI project uses these services to present itself and to interact with the user community. Public resources provide information on the project, the data produced and the algorithms used for production. Datasets shall be made available using several access mechanisms so that users can choose their preferred way of obtaining data.

A web resource shall allow registered users to easily download (and upload) data, tools, and documents. In addition, registered as well as anonymous users shall have access to information exchange through a managed forum and news feeds. Users shall be able to access data, information, and catalogue services through a central CCI web portal, which also bundles links to SST CCI web resources implemented independently. These independent SST CCI user services mainly cover two functional aspects:

- Access to the data generated by the processing system, online resources for regional averaging and re-gridding as well as CMIP5 re-formatting, access to match-up datasets
- A mini portal to visualise SST CCI data products via diagnostic plots (e.g. canonical time series plots, Hovmöller plots, spatial maps, animations of time series) in a format suitable for use in promotional activities

6.1.1 Components and interfaces

The high level functional aspects are implemented using several dedicated software packages that implement specific web functionalities (

Figure 6-1). Blue symbols indicate components operated by the SST CCI Team, whereas white boxes are outsourced or shared with other CCI ECVs. Most likely this functionality is completely covered by the Open Data Portal project; consultations between the team members will ensure that the requested functionality is being made available to the public users.

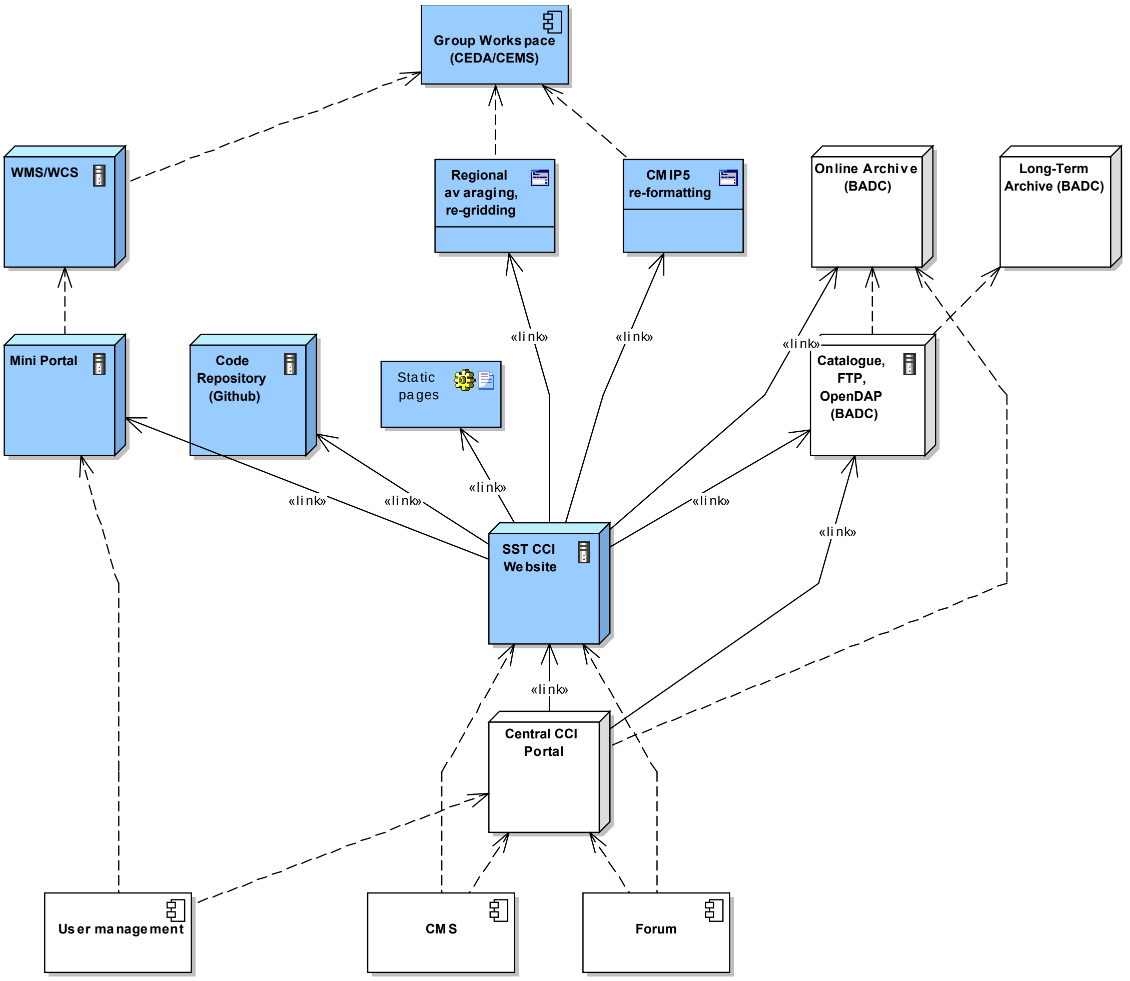


Figure 6-1: Services for SST CCI users

An SST CCI website bundles all the resources available to the users (→ SST-SR-6240 Website). The website is preferably implemented using a content management system (CMS). All of the distributed functional components are connected using links from the central portal. The background components have standard interfaces that may be integrated into other portals, like the upcoming central CCI Portal.

Particular user-accessible services are WMS and functions for customised data by sub-setting, regional averaging, and re-gridding (Section 6.1.3.5), access to the code repository to publish a subset of the project documents and algorithm implementations (section 6.1.6). The online and long-term archives are implemented by integrating the SST CCI data into the data storage and delivery system implemented by the Open Data, which offers catalogue, HTTP archive browsing, FTP, OpenDAP and other protocols for data downloading. (→ SST-SR-1295 Processor documentation, SST-SR-6180 Output provision, SST-SR-6190 FTP access, SST-SR-6230 Online sub-setting and re-gridding).

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	Website interface	Website External user's browser	Web pages, forms	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, web server External user's FTP client, browser	Data products	FTP, FTP, HTTP
Ifc-9.4a	OpenDAP interface	OpenDAP server External user's OpenDAP client, browser	Requests, data subsets	HTTP, OpenDAP
Ifc-9.4b	WMS interface	WMS server External user's WMS client, browser	Requests, images	HTTP, OGC WMS
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	CMS User's web browser	Documents, versioning information	HTTPS
Ifc-9.7	User management interface		Authentication and authorisation information, user profile information	

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 user's requests. The production subsystem adds new data and new CDR versions for publication.

A simple directory tree structures the online archive 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:

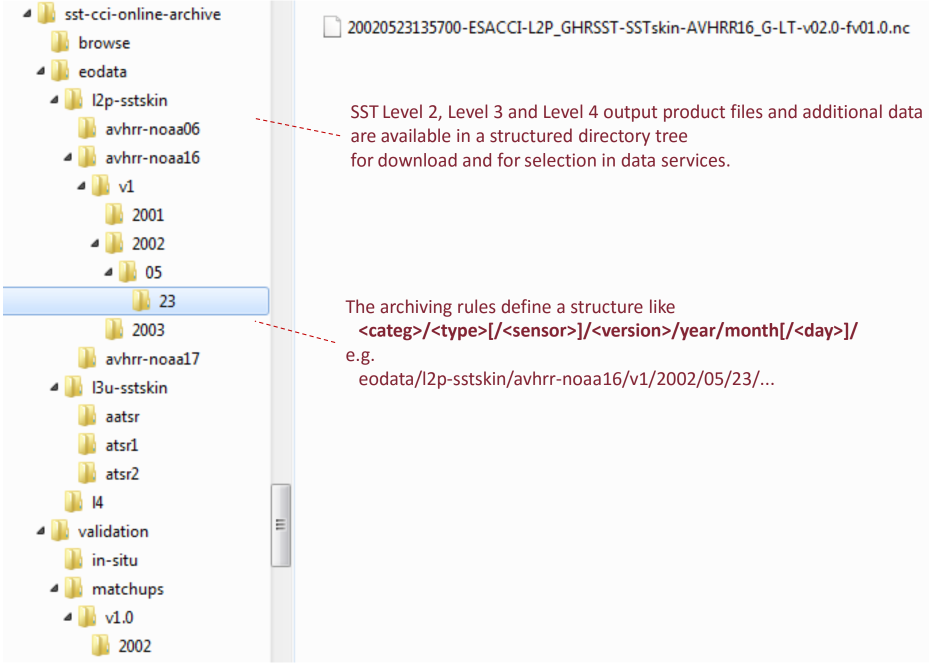
<category>/<type>[/<sensor>]/<version>/year/month[/<day>]/...

that generates paths like:

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

Figure 6-2 depicts an example archive structure following the rule above. The archiving rules and hence the structure of the directory tree distinguishes type information, version, and time information.

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



SST Level 2, Level 3 and Level 4 output product files and additional data are available in a structured directory tree for download and for selection in data services.

The archiving rules define a structure like
<categ>/<type>[/<sensor>]/<version>/year/month[/<day>]/
e.g.
eodata/l2p-sstskin/avhrr-noaa16/v1/2002/05/23/...

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

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 (WMS) serves images of the data, an OpenDAP service serves subsets, and a catalogue service (see Section 6.1.4) allows search and retrieval through 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 downloads 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.

Data access to SST CCI products has been implemented by providing the CDR to CEDA, which offers FTP, HTTP, OpenDAP and catalogue services.

6.1.3.1 FTP and HTTP

The complete data archive of output products shall be 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 similarly as in Figure 6-2.

Data access to SST CCI products by means of FTP (and HTTP) has been implemented by providing the CDR to CEDA.

6.1.3.2 OpenDAP and WMS

Access to SST CCI data shall also include protocols that concentrate on the exploration of datasets, in contrast to the file-based mechanisms described in the previous subsection. Protocols implemented for this purpose are OpenDAP and OGC WMS.

The open source Thematic Real-time Environmental Distributed Data Services (THREDDS) supports both access protocols and directly works with the netCDF format of the SST CCI products. The THREDDS server can be configured to use the structure of the output data archive suggested in Section 0 as catalogue base structure.

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.

Users can navigate through the archive in a similar way as he browses through an FTP archive. THREDDS offers a simple HTTP based user interface that implements basic directory browsing. This way, specific files in the archive can be located in a very intuitive and natural way.

Data access to SST CCI products by means of OpenDAP has been implemented by providing the CDR to the DEMS file system. WMS (and WCS) access mechanisms are a planned protocol that will be implemented by the Open Data Portal.

6.1.3.3 Match-up data

As a service for interacting with the community with respect to algorithm development and validation, the user services publish match-up datasets in the MMD format [AD 6] (→ SST-SR-6290 External algorithm development, SST-SR-6300 Match-up datasets, SST-SR-5140 Match-up dataset processing).

Besides the access via FTP, the MMS will provide a web interface to generate custom-made multi-sensor subsets of the MMD, 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 match-ups. An FTP-server contains a specific folder that is open for uploaded match-up data. The uploaded MMD may include additional satellite data or processed output from some algorithm for inter-comparison. There will be online checking and reporting of MMD file format incompliances and improper referencing of match-ups. In addition to this, a mandatory web or email form will be 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 Match-up inputs).

6.1.3.4 Bulk access

By disseminating the CDR through CEDA, the particular problem of delivering a complete reprocessed SST CDR to the climate modelling community has been tackled (→ SST-SR-6220). The CDR for the existing sensors comprises approximately 60 TB, and reprocessing activities may imply delivering an update of the complete CDR. The amount of data will increase with new sensors, in particular SLSTR, and CEDA provides the necessary bandwidth for download.

A supplementary mechanism to copy the CDR to suitable physical media that are shipped to the users using standard parcel services is currently being assessed. This feature will be implemented if user consultations prove an urgent need for physical media shipping. If implemented, each user will receive the set of media, copy the data to local storage and, when finished, ship the media to the next user or back to the SST CCI team. 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.

6.1.3.5 Online tools and mini portal

As depicted in

Figure 6-1, tools for regional averaging and re-gridding the CDR to lower spatial and temporal resolutions and for providing the CDR in CMIP5 compliant data format are made available online.

In order to visualise SST CCI data products via diagnostic plots (e.g. canonical time series plots, Hovmöller plots, spatial maps, animations of time series) in a format suitable for use in promotional activities, a mini portal is implemented that is based on an instance of the Operational Ecology (OPEC) portal (see

Figure 6-3 for a screenshot). SST data layers for display and analysis are published to the mini portal by means of WMS and WCS services using a THREDDS data server. The server is configured so that the many files in the CDR (in particular Level-4 analysis) are represented as a single (time series) dataset.

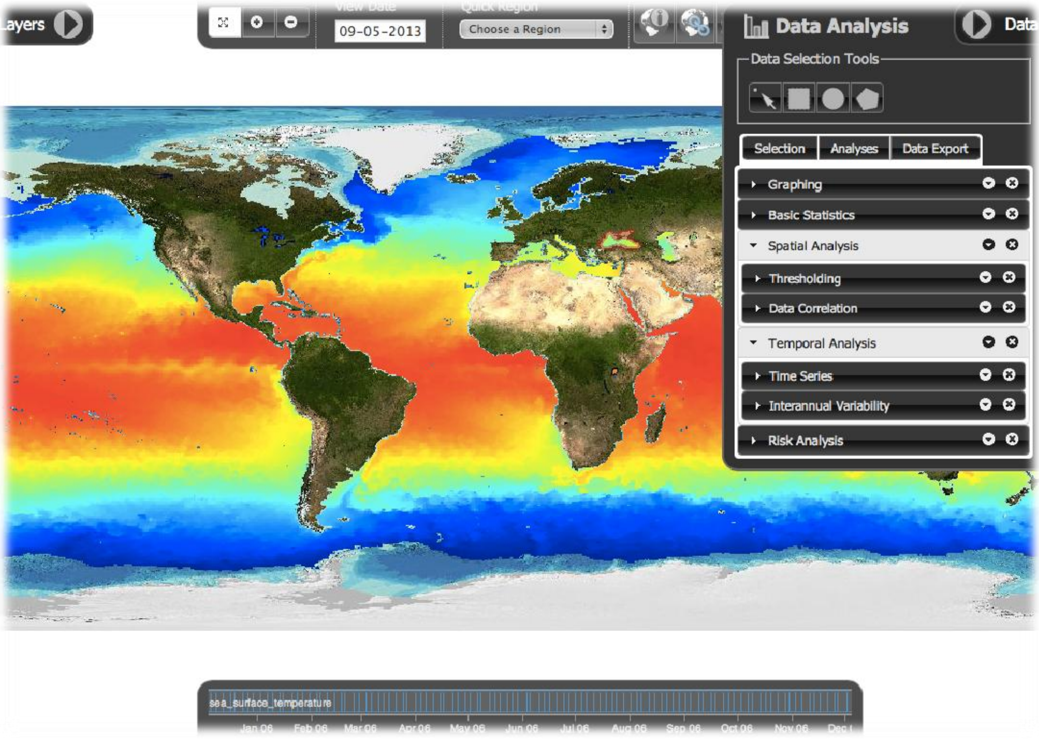


Figure 6-3: Screenshot of the OPEC portal (test version)

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). CEDA provides a basic catalogue service for discovering available datasets that match user-specified search terms. The latest version of the CEDA Metadata Catalogue has been built using CEDA's MOLES 3.4 metadata model, which is built on the series of ISO19100 standards.

Note that it is required that the SST CCI data holdings are discoverable by other metadata catalogues, 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. This functionality is planned to be implemented by the Open Data Portal project.

6.1.5 Web presentation and community interaction

The SST CCI website is the central starting point to explore achievements, data and other resources available to the users (→ SST-SR-6240 Website). It provides 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 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 used for the SST CCI web site (and also for the upcoming CCI Portal project) is the open-source solution Drupal, which provides, among other things:

- Separation of content and layout, corporate web site layout using CSS (core)
- Support for authoring by separated creation and publishing, dedicated approval step (core)
- Management of links independent from web pages, links to services and data access (core)
- User management (module)
- Document management (module)
- Forum (module)
- Issue tracking (module)
- News feed (module)

The basic functionality of any CMS is to separate the design of the website from the content. This allows keeping a consistent look-and-feel that is automatically applied to all new pages added to the website. In Drupal, 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.

Creating content is supported from any remote computer. The CMS separates the editing and formatting process from the publishing. This simplifies editing as new content can be created in its final external form (style, layout, colour-scheme), reviewed by a second administrative user and finally published. (→ SST-SR-6330 Website)

Forum

The upcoming CCI Portal project is going to provide the infrastructure for a central moderated CCI forum for all ECVs. It will provide the following functionality:

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 in discussions about specific issues or questions, which facilitates following the evolution of an idea, or the solution of a problem. To operate a forum that is valuable to the users, it is required that the forum is managed by an administrator who is capable of answering basic questions and getting in contact with specialists to delegate more complex problems. Such an administrator is the key to an informative and valuable forum. (→ SST-SR-6270 Forum or help desk, SST-SR-6360 Forum maintainer)

In addition to the forum, the website 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.

News feed

Drupal includes an RSS implementation. Any announcement published in Drupal 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. (→ 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 website will contain an issue-tracking module (→ 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.

Community interaction

Despite interacting with the user community by means of the services described above (forum, news feed, issue tracking) 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, in particular explaining uncertainties and how to use them

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

6.1.6 Access to tools, documentation and algorithm implementations

The SST CCI project provides tools, documentation, and software source code to interested users. Tools can be downloaded and used locally to work with the SST CCI data. Documents intended for users are the Product User Guide (PUG), validation and intercomparison reports, climate assessment reports, and the algorithm theoretical basis documents (→ SST-SR-6170 Product features, SST-SR-1366 Product User Guide). These are accessible via the project website. The product user guide also includes sample code in different programming languages illustrating how to read SST CCI data (→ SST-SR-4130 Data access software).

Access to software source code is provided to meet requirements for transparency. Commented public code can be reviewed and is available for possible improvements. Access may be provided by different means:

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

The following tools are provided (→ SST-SR-4110 Sub-setting and re-gridding, SST-SR-4120 Visualisation and data analysis, SST-SR-1320 Trend analysis):

- A re-gridding (and sub-setting) tool to extract data on grids with lower spatial resolution (to meet a wider range of user requirements). This tool is also available as an online service
- A regional averaging tool. This tool is also available as an online service
- Visualisation tools (Sentinel Toolbox, CCI Toolbox)

These tools are offered as desktop applications suitable to execute on all major operating systems. Platform-specific installer software installs the tools at the user's premises.

The source code of the SST CCI processing system, MMS and tools are all under configuration control. Section 6.3.2 Processor version concept describes the version control approach. The source code is publicly readable (→ SST-SR-4140 Open source tools). This facilitates confidence in the final data products because any user, in principle, can reproduce each step of the processing. In addition to the source code, the software repository also contains instructions for building the software from its sources (→ SST-SR-6290 External algorithm development, SST-SR-6280 Processor repository).

The software repository contains the actual processing code and the complete former code history. The Development Team decides whether the current development branch is made accessible. All versions can be accessed using a simple mechanism to choose a tag and download the source code of the selected version.

Write access to the processor repository is restricted to the Development Team including or extended by algorithm developers, if external. Operators add working configurations to version control. Because 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).

Git is the version control system used. Public code repositories like GitHub (used by SST CCI) can host open and closed projects.

6.1.7 User management

User management is an essential function of the upcoming CCI Portal. Users will be able to access general CCI as well as specific SST CCI services with the same account.

6.2 Production and reprocessing for consistent versions

This section defines the structure and functions that implement the operational production and reprocessing in the operational SST CCI system. Focus is on consistency, completeness, traceability, and efficiency of processes.

6.2.1 Components and interfaces

Further breaking down the production and development subsystem leads to the decomposition shown in

Figure 6-4.

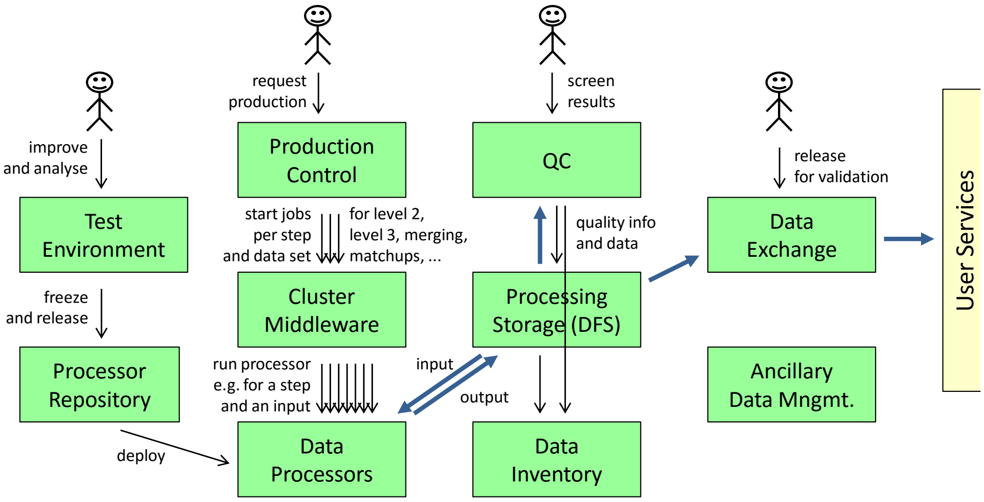


Figure 6-4: Components of the SST CCI production and development subsystem

Table 6-2 below describes the components of the production and development subsystem in terms of name, purpose and function, local data stored and managed, and 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 SST CCI products	Auxiliary data	Extensions of existing processors
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	Network file system and centralised fast storage (→ SST-SR-1470 Online storage)
Cluster middleware	Handles processing jobs, uses configuration and plug-ins to generate tasks and to call processors (→ SST-SR-1480 Parallel processing)	Job queue Status of each job	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	Production monitor, MetOffice Rose
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	Direct access to archive
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 Triggering 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 quick-looks, provides tools for systematic visual screening and for product inspection	QA working area in processing storage Quality attributes of product entries in inventory	Data product readers, tests for missing data and for geo-location issues, ESA Sentinel Toolbox framework

Component	Purpose and Function	Data	Implementation
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 Triggering rules	Ingestion modules for different protocols
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	Virtual machines
Processor repository	Stores source code of processor implementations, versioning with branches, authorship, simplified user interface to store and to retrieve versions, supports automated deployment of processor bundles	Source code, repository meta information	GitHub with public repository, git software package, tools for simplified access

The components of the production subsystem interact with each other, with other subsystems and with system users and operators. Each component provides and uses interfaces. Figure 6-5 and Table 6-3 show and describe the main interfaces in terms of interface exchange items, contents and format, and data transfer protocol used.

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

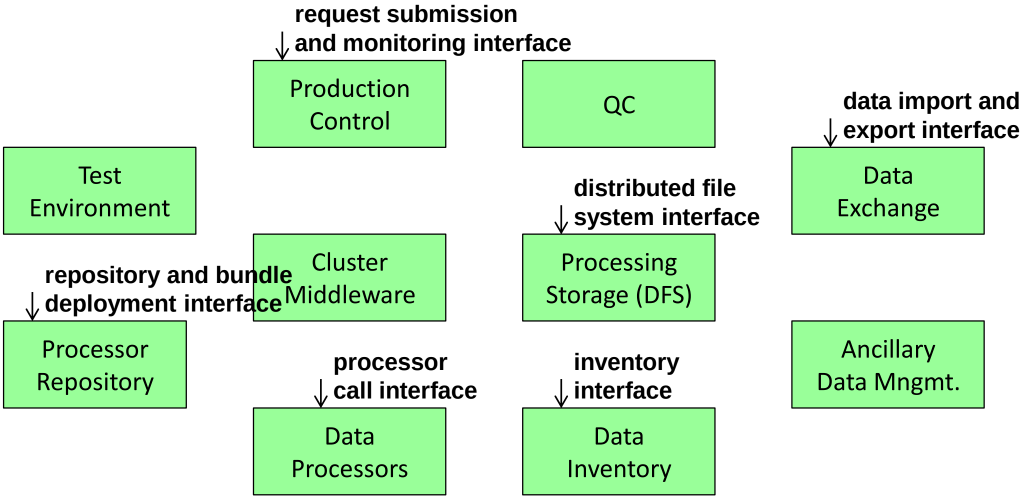


Figure 6-5: Main interfaces of the production and development subsystem

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

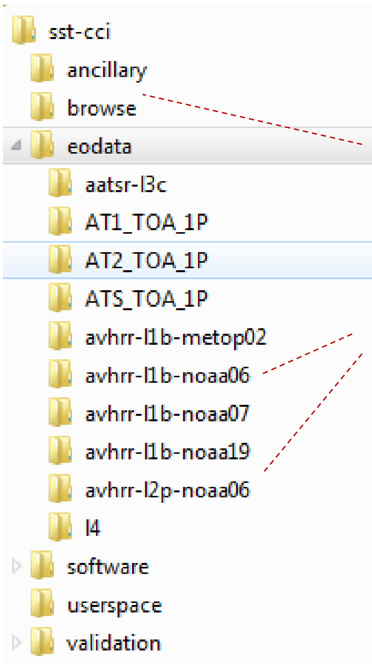
Ifc ID	Interface Name	Endpoints (provider, user)	Interface items content and format	Data exchange protocol
Ifc-10.1	Processor call interface(s) (→ SST-SR-1350 Processor framework, SST-SR-1360 Executable integration)	Data processors User: Production control via cluster middleware	Calls with parameters (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	Executable code, Unix process, inter-process communication, wrapper scripts, working directory, file interface for inputs, outputs, parameters, reports Plug-in software libraries, Unix threads, ESA SNAP 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 Unix file descriptors and file handles	Network file system, Unix commands (mkdir, cd, ls, put, get, rm, cp), Unix file open method
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 for bundle deployment
Ifc-10.4	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)

6.2.2 Structured processing storage

The definition of the directory structure for the processing storage file system is the basis for all processing functions. The processing file system hosts earth observation inputs, outputs, ancillary, and reference data, as well as processor software bundles (see Section 6.3.2 Processor version concept). The hierarchy, versioning, and naming schema – so-called archiving rules - identify each item in this structure. There is exactly one location for each item or file. Knowing the archiving rules, a processing function can identify its inputs and 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).

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

Figure 6-6 illustrates the top-level directories of the SST processing storage with Earth Observation data, browse images, ancillary data, reference data, processor software bundles, and a user space.



Directories for satellite data inputs and outputs, ancillary and browse data, processor software packages

Processing is a controlled transformation of files into other files in the SST CCI directory tree

Figure 6-6: Directory 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 instance 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 Grid Engine middleware environments with a hierarchical file system, the structure is instantiated in background storage, where data actually used is staged to fast network storage that is accessible by all processing nodes.

6.2.3 Input ingestion

Input ingestion 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 to systematically retrieve and ingest newly acquired data along with 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 6-7 shows the function and the elements involved. The ingestion function accesses a local or remote pickup point and scans it for new products. A local pickup point will typically be used to scan media obtained from data providers. Scanning descends a directory tree and uses file (or file name) filters to detect data products. In order to detect new products for data-driven processing, the ingestion function repeatedly scans the pickup points maintaining a memory of known files. Errors are handled by retry and reporting (→ SST-SR-2170 Online processing error handling). Path rules and version information are used to store the products in the processing storage directory structure (→ SST-SR-1220 Input versions, SST-SR-1470 Online storage). Ingestion initiates registration of products in the inventory (→ SST-SR-2160 Online processing status). The ingestion function triggers data-driven production from newly ingested products. Quality checks for new products are initiated. If needed, input data is reformatted, e.g. to transform and subset Sentinel or VIIRS inputs into the data format and spectral bands used by 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).

The ingestion function is implemented by the Data Exchange component.

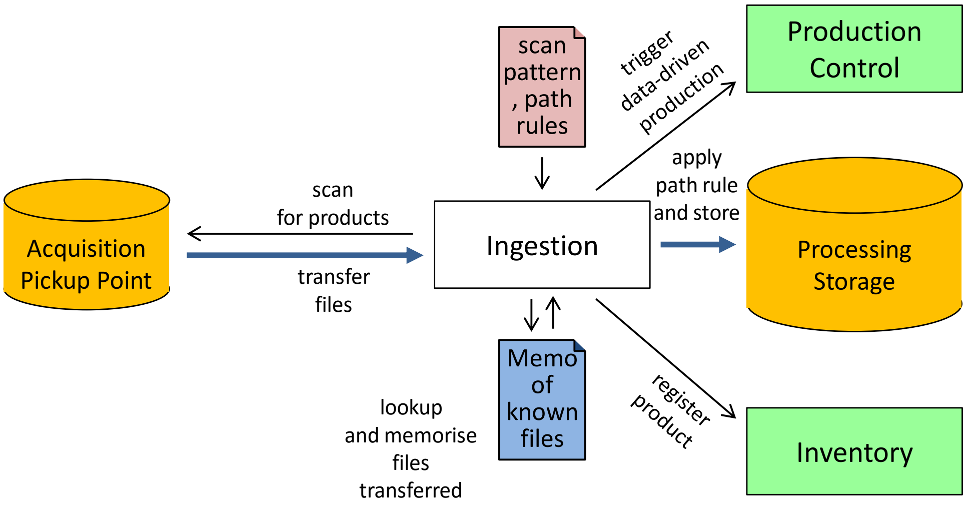


Figure 6-7: Ingestion function to pull input data from a pickup point of a data provider

6.2.4 Data processors

The data processors of the CCI SST system are versioned modules plugged into processing chains. The processors include:

- ARC/CCI SST retrieval processor (L2 processor)
- DV model processor (also does depth adjustments)
- 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 a generic cluster middleware (grid engine) and to augment it with:

- 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 is implemented by services that the operator interacts with (→ SST-SR-1390 Operator, SST-SR-1400 Automation) and that facilitates adapting workflows (→ SST-SR-5170 New input type, SST-SR-5200 New processing chain). Figure 6-8 below shows the hierarchy of services of production control including the cluster middleware and data processors. The granularity of tasks is an example of how a bulk request can be broken down into jobs and processing tasks that together generate the requested outputs.

Layer	Elements and functions	Granularity of activity
Production management layer	Request Queue Workflow engine Resource management	"produce daily L3C from L1 for all months of 2011"
Processing management layer	Jobs and tasks Cluster middleware	"produce L2 from all AATSR L1 of July 2011"
Processing infrastructure layer	Task execution on processing node	"call level 2 processor for one AATSR L1 file"

Figure 6-8: Hierarchy of services and granularity of control at different levels

Figure 6-9 illustrates the SST CCI processing workflow with the logical dependency graph of steps and the corresponding data flow (→ SST-SR-1370 Processing chain).

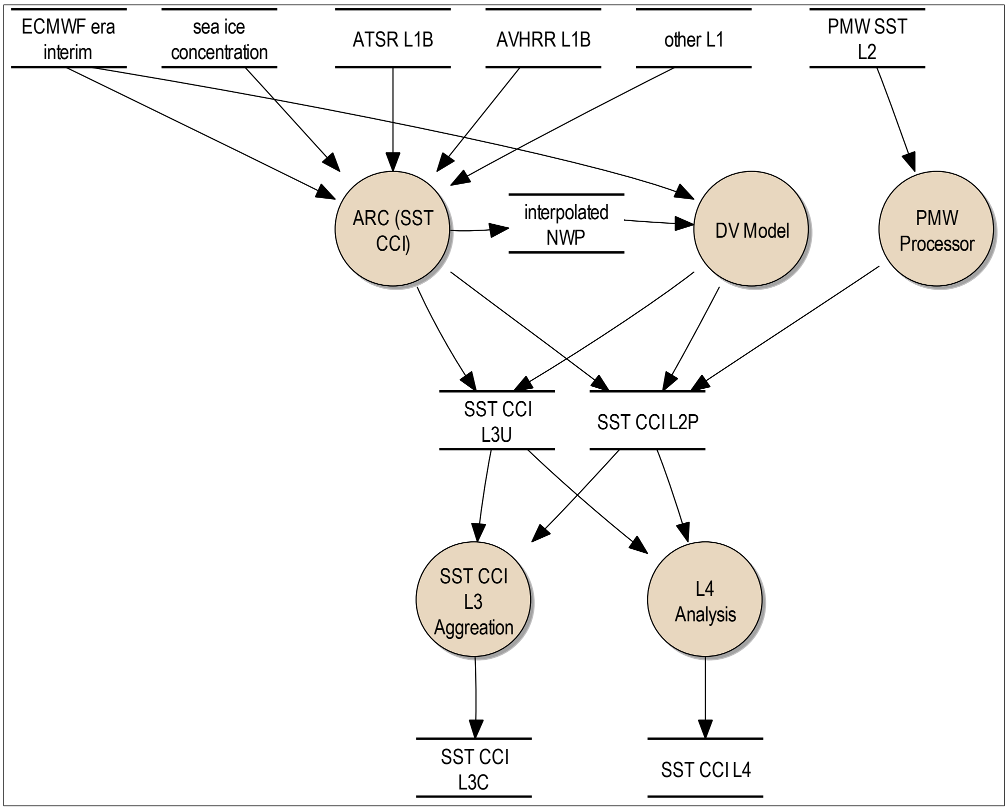


Figure 6-9: SST processing workflow from Level-1 and Level 2 satellite data to Level-4 analysis

The ARC/SST CCI processor retrieves SST from ATSR and AVHRR (METOP and NOAA AVHRR GAC) Level-1 products and generates L2P and L3U products according to the PSD [AD 5]. It is foreseen to extend the SST processor to new Level-1 inputs from SLSTR (→ 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-1250 SST retrieval, SST-SR-1260 Error characterisation, SST-SR-1290 Output products).

Currently the ARC/SST CCI processor adds the diurnal adjustment information from the DV Model to the L2P / L3U products (→ SST-SR-6150 L4 Diurnal outputs). It performs the following steps during the L2P/L3U products generation:

- access the CEDA archive of ERA-interim data and interpolate NWP to satellite raster
- (optionally) call the DV Model
- run Bayesian cloud detection and SST retrieval
- generate L2P and L3U outputs

The PMW processor used in Phase-I of the project converts microwave products into the SST format, optionally adding bias adjustments and uncertainty and quality information (→ SST-SR-1135 PMW inputs). The PMW processor is not used anymore in Phase-II.

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).

Defining steps and constraints

Production control manages a workflow by defining steps for sets of products, and by defining constraints in the form of pre-conditions and post-conditions. These conditions ensure that dependencies between steps are taken into account and that all steps are carried out.

Note that for aggregation steps there is an input set for each output product, whereas for the Level-2 processing steps there is a one-to-one relation between inputs and outputs.

Step	**SST retrieval 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
Pre-conditions	L1B sensor products of time period are available/staged
Post-conditions	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
Pre-conditions	L2P products for sensor and time period are available L3U products for sensor and time period are available
Post-conditions	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
Pre-conditions	PMW L2 products of sensor and time period are available/staged
Post-conditions	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
Pre-conditions	L2P products (with DV) for ATSR and time period are available L3U products (with DV) for ATSR and time period are available
Post-conditions	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
Pre-conditions	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
Post-conditions	L4 products for time period are available
Auxiliary	-

Bulk production and reprocessing

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

1. L1B products of AVHRR, ATSR and the PMW sensors are made available (staged, if necessary) in order to fulfil initial preconditions.
2. The steps for all sensors (i.e. the processor calls) 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.

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

Production on request

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

1. L1B products of AVHRR, ATSR or the PMW sensors are made available (staged, if necessary) in order to fulfil initial preconditions.
2. The steps for all months and sensors are generated.
3. 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:

1. The steps for this product are generated.
2. The pre-condition for this product is fulfilled by copying the product into the processing directory structure.
3. The steps are executed when pre-conditions are obeyed.
4. If the production depends on certain auxiliary data that may is not available immediately, then the required auxiliary data is formalised as pre-condition, and auxiliary ingestion triggers fulfilling of corresponding pre-conditions for data-driven processing steps (→ SST-SR-2140 Wait for ancillary, SST-SR-2180 Short-delay processing).
5. Aggregating steps (i.e. Level-3 and Level-4 steps) are generated separately on a delayed execution basis, e.g. controlled by a timer.

Production resource management and queuing

Configuration, a production request, and emerging new products can initiate production. For the production control function, the concept of a single production request is generalised in a manner that there can be a set of requests for bulk production, and that also data-driven production can generate a request for each new product to control its workflow through the system. The production request specifies what shall be processed with respect to the 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 depend on implementation: for instance, an internal format defined by a database schema, and an external format defined by an XML schema. As a minimum, a request exhibits the following fields:

Field	**Description** (or set by the ingestion system in case of data-driven production)
Identifier	A unique ID and a name provided by the user or operator (or set 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, has failed, or has succeeded.

Any infrastructure for processing requests has certain limits. The infrastructure resources considered here are:

- Processing time, number of 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 middleware monitors and controls the rest. For example, the number of jobs per host as well as the demand of main memory and CPU time (per job) can be criteria to restrict scheduling and hence concurrency.

The production control function restricts execution of steps according to the availability of monitored resources. Resources at least are the number of steps of a certain type to be executed concurrently. For example, the restriction may be to run at most for concurrent SST retrieval processing steps. Note that each step may process a complete month of inputs, which, in this example, results in the concurrent processing of four months of input data.

In addition, the production control function manages storage space. 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 type of auxiliary data required, there is a selection rule to determine the auxiliary data products required. Often, the rule is based on acquisition time. Rules required for SST are:

- Sea ice data are available as daily product files. The SST retrieval step selects sea ice data using the acquisition start time of the product to be processed. The sea ice product is selected by **t**emporal coverage. It temporarily covers the acquisition start time of the product.
- NWP data are available at a resolution of 6 hours. The SST retrieval 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 products closest before and closest after the acquisition start time are 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 SST retrieval step is a design decision. It means that the interpolated NWP is not stored and not re-used. 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. 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 has been 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 resuming, the system keeps track of all steps for a bulk request that completed successfully. The resume function skips completed steps, restarts interrupted steps, and starts all other steps (satisfying preconditions). The granularity of steps (and optionally a means within a step to detect partial results) determines the overhead caused by a failure.
- Failure to process a certain input product with a certain processor results in a qualified exception report put out by the processor. Production control records the failure and continues production of only of those steps that do not depend on the failed step. The pre-conditions of subsequent steps ensure taking into account dependencies. Simple repetition without modifications usually is not helpful in case of a processing failure. An operator may provide a corrected input or updated parameters before repeating. The operator may also decide to skip the product that has caused the failure so that subsequent aggregation steps do no longer depend on this input. This is handled by the production control function by updating the set of products in the pre-condition of the aggregation step.

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

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 then provides feedback to the system after having inspected the result. One form of integrating manual steps is to separate the workflow into the parts before and after a manual step. When the first part is complete, 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 possibility for integrating manual steps is to provide the data to be handled manually to a certain storage area (for operators) and to submit feedback via a tool. The implementation of this feedback may use constraints and set preconditions to initiate continuation of the processing. Sorting products into “good” and “bad” ones can be implemented in the same manner as exceptions are handled. Failed quality checks lead to a failed step (or a step failed for certain inputs). Production for the "good" products can be continued.

If the workflow depends on the result of a (manual) step, then the step generation for continuing 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 depends on the quality of the input data. Because the SST workflow contains an aggregation of inputs, just a single corrupted input that remains undetected can compromise a daily composite. The Quality Check (QC) 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 6-10 illustrates quality checking. Firstly, all input products are screened automatically for consistency (file format, file size, geo-location, and data content). Optionally, for all input products quick-looks are generated for visual screening. Corrupted products are marked in the inventory and removed from processing storage. Secondly, 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 the production workflow. Thirdly, the L3 and L4 output products are quality checked and validated before release.

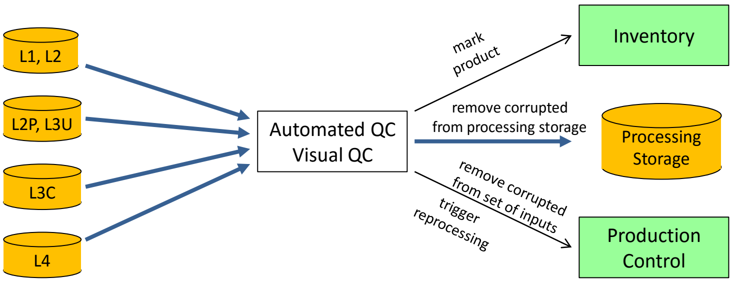


Figure 6-10: Quality check of inputs, intermediates and outputs

For data-driven short-delay processing, quality checks can be conducted after the generation of products. In case QC detects issues, products are retracted, reprocessed, and replaced if necessary. The collaboration diagram in Figure 6-11 shows how QC is integrated into its context by data and control flows.

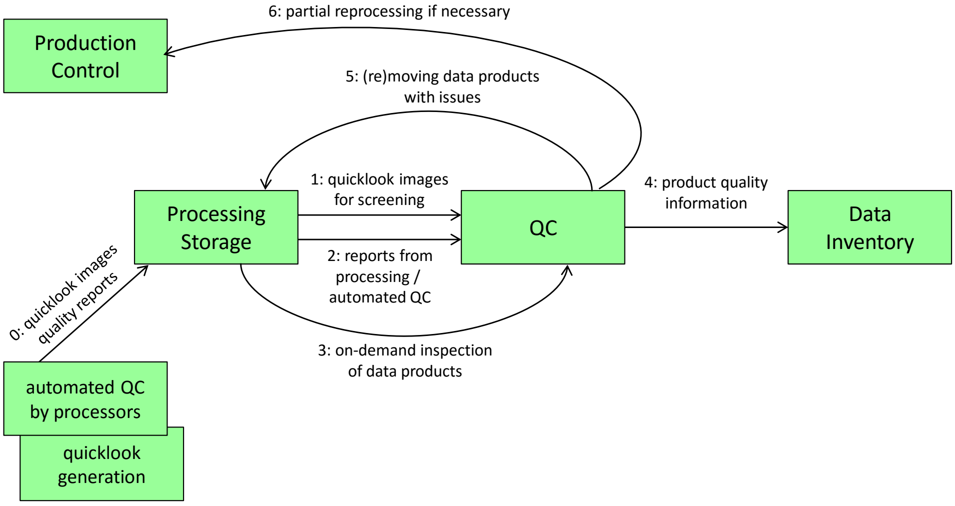

6: partial reprocessing if necessary
Production Control
5: (re)moving data products with issues
1: quicklook images for screening
4: product quality information
Processing Storage
QC
Data Inventory
2: reports from processing / automated QC
0: quicklook images quality reports
3: on-demand inspection of data products
automated QC by processors
quicklook generation

Figure 6-11: Quality Check and its interfaces

QC retrieves new quick-look images and quality reports from automated QC processors via the processing storage. Operators inspect data products from processing storage, if necessary. Product quality information is updated in the inventory, including data quality metrics that indicate performance. In case of quality issues, partial reprocessing may be triggered or affected products may be removed from the collection.

6.3 Continuous algorithm improvement

This section defines the structures and functions that extend the production and reprocessing environment to facilitate continuous improvement. Focus is on flexibility, rapid prototyping and testing, including an interface to full mission data and reprocessing capabilities also for testing and validation. The concepts described are processors, versioning, a test environment, and the multi-sensor match-up 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 transferring data and seven processors for transforming data, among the latter the ARC/CCI SST retrieval processor, the CMS OSI-SAF processor, and the OSTIA L4 analysis processor. All exhibit individual interfaces. As one of the goals of the project is to simplify the integration and operational use of scientific code, a rigorous standardisation of processor interfaces like in an ESA IPF is not feasible. Instead, adapters wrap the existing scientific processors. 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. exit on error with exit code, multi-process capable, no memory leaks)

Depending on the processing middleware used, the processor wrapper is an adapter between the middleware job interface and the respective processor (→ SST-SR-1350 Processor framework, SST-SR-1360 Executable integration). In a grid engine environment, a “start script” creates a job by submitting a “run script” to the job queue. The “run script” is the script that calls the processor; the “start script” submits the “runs-script” to the grid engine job queue. Processors have to be executable files installed in a network file system shared by all computing nodes.

Unix executable files use a file interface to access inputs and write outputs. Parameters are transmitted as command line arguments, as environment variables, or within a parameter file. Executable files do provide feedback with the exit code and with messages on standard output and error streams as well as in log files.

Figure 6-12 and Figure 6-13 show example processor wrapping scripts for a grid engine environment. The scripts constitute the environment for a processor, which is called with the necessary parameters. Note that neither the “start script” nor the “run script” do depend on the middleware explicitly. The “start script” calls functions for resuming, submitting, and waiting for completion of a processor job, the implementation of which is provided in a separate, environment-specific shell script. An example applicable to the CEMS/LOTUS batch processing system is given in Figure 6-14. For transferring the system to another type of middleware, only the environment-specific shell script has to be adapted.

```
#!/bin/bash

# usage: gbcs-start.sh <year> <month> <sensor> <usecase>
# example: gbcs-start.sh 2003 01 atsr.3 mms2


# source environment-specific functions and variables
. ${mms.home}/bin/mms-env.sh


year=$1
month=$2
sensor=$3
usecase=$4


task="gbcs"
jobname="${task}-${year}-${month}-${sensor}"
command="${task}-run.sh ${year} ${month} ${sensor} ${usecase}"


echo "`date -u +%Y%m%d-%H%M%S` submitting job '${jobname}' for usecase ${usecase}"


resume_task_jobs ${jobname}

if [ -z ${jobs} ]; then

   submit_job ${jobname} ${command}

fi

wait_for_task_jobs_completion ${jobname}
```

Figure 6-12: gbcs-start.sh – example of a "start script" (SST retrieval processor)"

```
#!/bin/bash

# usage: gbcs-run.sh <year> <month> <sensor> <usecase>
# example: gbcs-run.sh 2003 01 atsr.3 mms2


year=$1
month=$2
sensor=$3
usecase=$4


mkdir -p ${mms.archive.root}/${usecase}/arc/${sensor}/${year}


echo "`date -u +%Y%m%d-%H%M%S` gbcs ${year}/${month} sensor ${sensor}..."


${mms.home}/bin/gbcs-tool.sh -c ${mms.home}/config/${usecase}-config.properties \
-Dmms.gbcs.sensor=${sensor} \
-Dmms.gbcs.mmd.source=${mms.archive.root}/${usecase}/sub/${sensor}/${year}/${sensor}-sub-${year}-
${month}.nc \
-Dmms.gbcs.nwp.source=${mms.archive.root}/${usecase}/nwp/${sensor}/${year}/${sensor}-nwp-${year}-
${month}.nc \
-Dmms.gbcs.mmd.target=${mms.archive.root}/${usecase}/arc/${sensor}/${year}/${sensor}-arc-${year}-${month}.nc
```

Figure 6-13: gbcs-run.sh – example of a “run script” (SST retrieval processor)

```
#!/bin/bash
# useage ${mms.home}/bin/mms-env.sh  (in xxx-start.sh and xxx-run.sh)


if [ -z "${MMS_INST}" ]; then
   MMS_INST=`pwd`
fi
MMS_TASKS=${MMS_INST}/tasks
MMS_LOG=${MMS_INST}/log


read_task_jobs() {
   jobname=$1
   jobs=
   if [ -e ${MMS_TASKS}/${jobname}.tasks ]
   then
      for logandid in `cat ${MMS_TASKS}/${jobname}.tasks`
      do
         job=`basename ${logandid}`
         log=`dirname ${logandid}`
         if grep -qF 'Successfully completed.' ${log}
         then
            if [ "${jobs}" != "" ]
            then
               jobs="${jobs}|${job}"
            else
               jobs="${job}"
            fi
         fi
      done
   fi
}


wait_for_task_jobs_completion() {
   jobname=$1
   while true
   do
      sleep 10
      if bjobs -P esacci_sst | egrep -q "^$jobs\\>"
      then
         continue
      fi
      if [ -s ${MMS_TASKS}/${jobname}.tasks ]
      then
         for logandid in `cat ${MMS_TASKS}/${jobname}.tasks`
         do
            job=`basename ${logandid}`
            log=`dirname ${logandid}`
            if [ -s ${log} ]
            then
               if ! grep -qF 'Successfully completed.' ${log}
               then
                  echo "tail -n10 ${log}"
                  tail -n10 ${log}
                  echo "`date -u +%Y%m%d-%H%M%S`: tasks for ${jobname} failed (reason: see ${log})"
                  exit 1
               else
                  echo "`date -u +%Y%m%d-%H%M%S`: tasks for ${jobname} done"
                  exit 0
               fi
            else
                  echo "`date -u +%Y%m%d-%H%M%S`: logfile ${log} for job ${job} not found"
            fi
         done
      fi
   done
}
```

```
submit_job() {
    jobname=$1
    command=$2
    bsubmit="bsub -R rusage[mem=20480] -q lotus -n 1 -W 8:00 -P esacci_sst -cwd ${MMS_INST} -oo
${MMS_LOG}/${jobname}.out -eo ${MMS_LOG}/${jobname}.err -J ${jobname} ${mms.home}/bin/${command}
${@:3}"
    rm -f ${MMS_LOG}/${jobname}.out
    rm -f ${MMS_LOG}/${jobname}.err
    if hostname | grep -qF 'lotus.jc.rl.ac.uk'
    then
        echo "${bsubmit}"
        line=`${bsubmit}`
    else
        echo "ssh -A lotus.jc.rl.ac.uk ${bsubmit}"
        line=`ssh -A lotus.jc.rl.ac.uk ${bsubmit}`
    fi
    echo ${line}
    if echo ${line} | grep -qF 'is submitted'
    then
        jobs=`echo ${line} | awk '{ print substr($2,2,length($2)-2) }'`
        echo "${MMS_LOG}/${jobname}.out/${jobs}" > ${MMS_TASKS}/${jobname}.tasks
    else
        echo "`date -u +%Y%m%d-%H%M%S`: tasks for ${jobname} failed (reason: was not submitted)"
        exit 1
    fi
}
```

Figure 6-14: mms-env.sh – environment-specific functions for CEMS/LOTUS

6.3.2 Processor version concept

Processors (more precisely processor bundles) including software as well as its 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. A processor bundle has a certain runtime structure (the development structure may be different) that is packed into an installation package for deployment. The packing procedure labels the file name of the installation package with the proper version (→ SST-SR-5210 Transfer to operations).

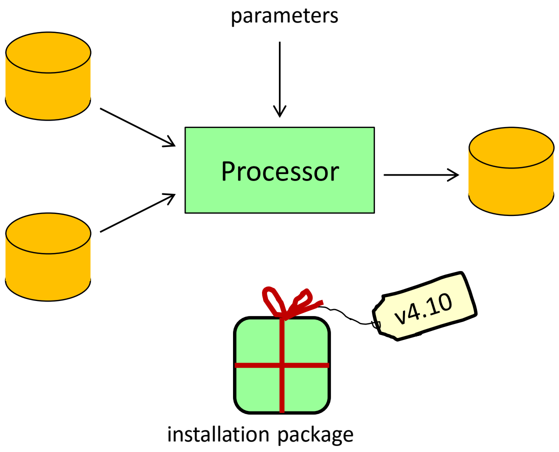
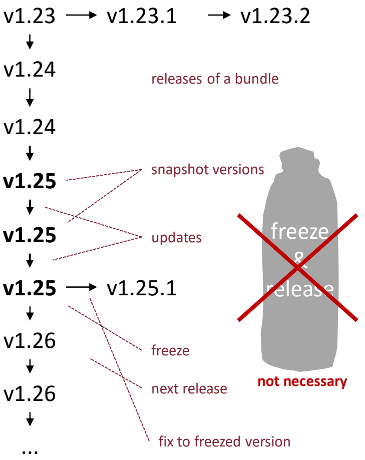


Figure 6-15: Processors and versioned processor installation packages for modularity

Processing jobs specify the bundle to be used and the processor to be run by name and version (→ SST-SR-1340 Concurrent processor versions). A combination of bundles constitutes an assembly. The SST CCI assemblies comprise a certain version of the ARC/CCI 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 can be reproduced from the same input, if needed.

In order to simplify development, the schema for development versions is less strict and follows the “snapshot approach” often used with the git version control system. The same version number can be used repeatedly to tag development versions as long as it is not frozen. Figure 6-16 illustrates the versioning schema and its principles (→ SST-SR-5130 Development versions).



- A snapshot version may be updated in development and test cycles.
- Every version initially is a snapshot and looses this attribute by a freeze.
- A freezed version has been tested before the freeze and is not renamed by the freeze (-> no additional test)
- A freezed version is never updated.
- After a freeze the next version is started with the next update (release)
- Fixes to freezed versions, if necessary, are done in a branch.

Figure 6-16: Processor and configuration versioning with freeze and release

The git version control system and a git server (e.g. GitHub) are used for the processor bundles of the SST CCI system. If there are parts that are not open source, the corresponding bundles can be kept in a non-public repository of the git server. All other bundles are in a public (for reading) repository in order to allow interested users to review the software and the configurations used, and to allow external 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; includes adding and deleting files from the directory tree, committing and pushing to the repository, tagging with the current or a new branch version number, if specified
- Convenient checkout: fetches the directory tree of the development structure of the processor bundle from the repository; fetches the main trunk or a specified branch or tagged version
- Freeze and release: commits and freezes the current version; 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 includes a corresponding marker file. Only frozen versions will be used for production and for bulk tests.

Concurrent development by experienced developers can still use the native commands of the version control system for merging and branching if required. The convenience functions can be used without detailed knowledge of a version control system, if conflicts caused by concurrent changes of the same module do not occur.

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 when it is no longer required. No long-term state is maintained in such a machine. State is instead kept in the software repository.

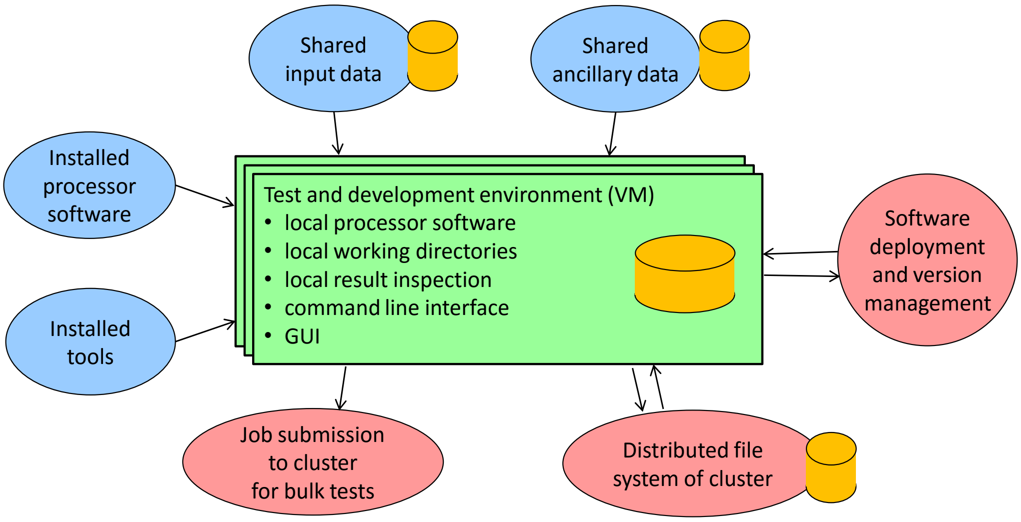

Shared input data
Shared ancillary data
Installed processor software
Installed tools
Test and development environment (VM)
• local processor software
• local working directories
• local result inspection
• command line interface
• GUI
Software deployment and version management
Job submission to cluster for bulk tests
Distributed file system of cluster

Figure 6-17: Local test environment with full data access

Figure 6-17 depicts the virtual machines and their environment. A virtual machine includes 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 into the production environment (→ SST-SR-5210 Transfer to operations, SST-SR-5270 Short development cycle). Certain tools are installed, and there is access to test data and to auxiliary data. Access to the full processing storage of the production environment is used for the analysis of problems. For bulk tests the interface for job submission to the processing environment can be used from the test and development environment (→ SST-SR-5150 Full timeline access). By limiting resources for test production, operational reprocessing is not compromised (→ SST-SR-5310 Measures for stability).

6.3.4 Multi-sensor match-up datasets

Match-ups are pairs or multiples of observations that coincide in space and time. Often, but not necessarily, one of the observations comes from *in situ* data. Match-ups allow comparing outputs of processing of satellite data with reference data (from *in situ* measurements or processing of other data). In SST CCI there are floating buoys, moored buoys and ships that collect large amounts of *in situ* data. In a multi-sensor match-up dataset, data from several satellite and *in situ* measurements are related.

Figure 6-18 constitutes a plot of multi-sensor match-ups for a single day (2010-06-02). The plot shows match-ups 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* match-ups have not been plotted for convenience, because there are too many. The *in situ* data are from drifting buoys. If there are two dots at the same location they differ in time substantially (one satellite orbit or more).

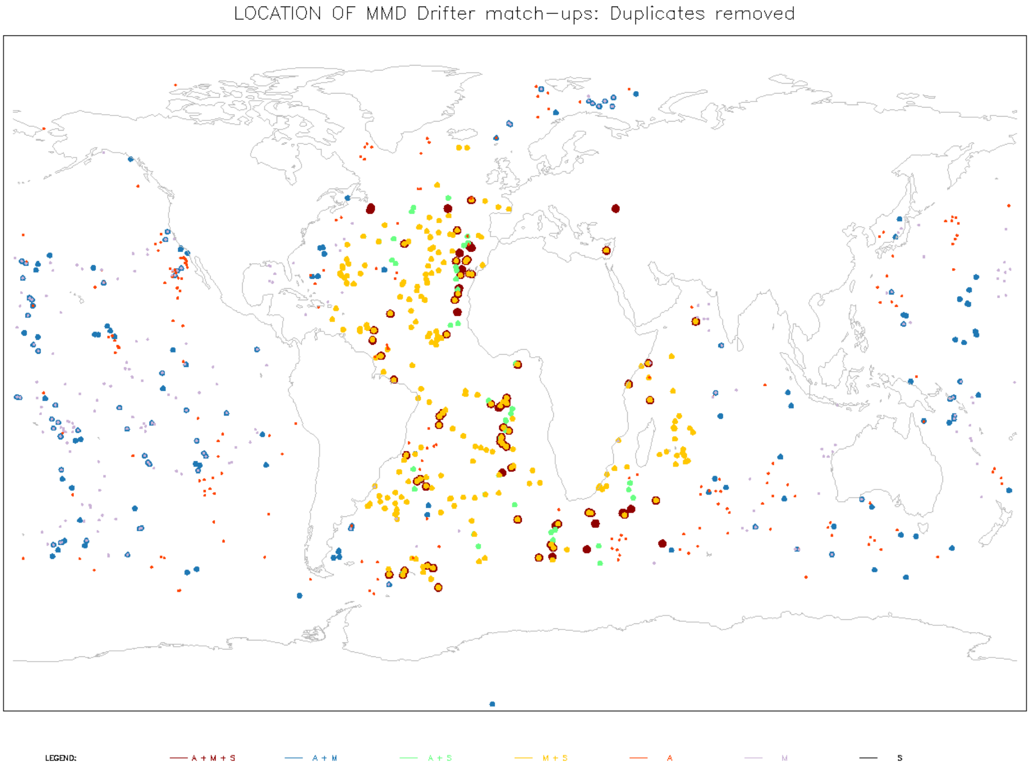
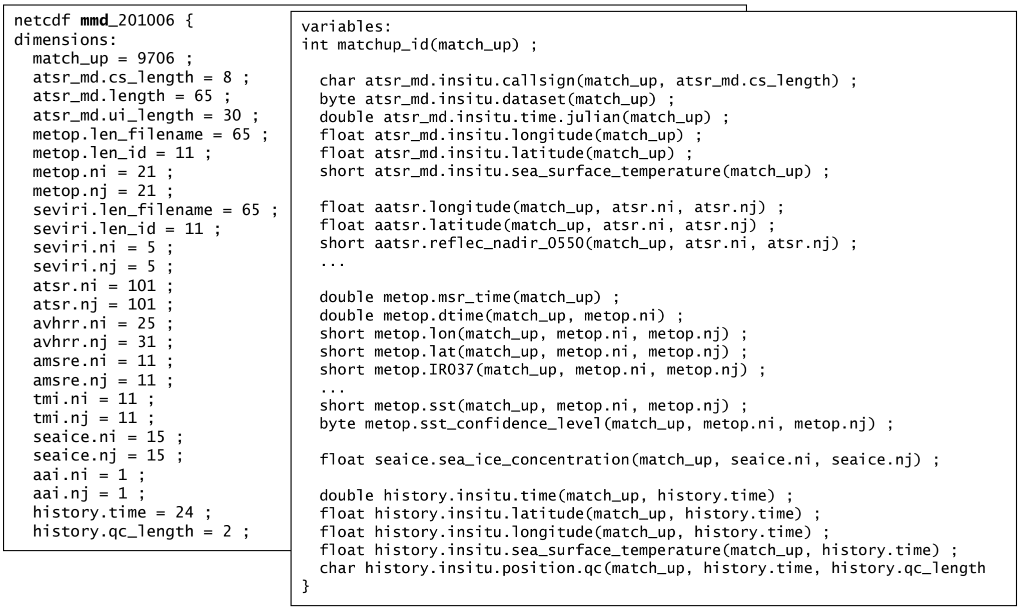


Figure 6-18: Locations of multi-sensor match-ups for a single day

Sets of match-ups are provided in so-called MMDs ([AD 6]) (→ SST-SR-1310 Match-up analysis). The MMDs are organised in records, one for each match-up. A record comprises fields for each sensor that contributes to the match-up (ATSR, AVHRR, etc.), *in situ* data, and auxiliary data (sea ice, aerosol, NWP). Additional fields provide time and location information per sensor, small sub-scene extracts from Level-1 data, and, optionally, processed sub-scenes. Fill values are used where a sensor is not part of a match-up (→ SST-SR-3140 Extract satellite sub-scenes, SST-SR-3150 interpolate NWP, SST-SR-3160 Process SST).

MMDs are produced in NetCDF format, with record fields being NetCDF variables. Figure 6-19 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 6-19: NetCDF header of MMD file with dimensions and prefixed variables

MMDs are flexible with respect to 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 a subset of fields.

The generation of the final MMD files is controlled by a set of configuration files that define the MMD content. The match-up generation engine assembles the final MMDs according to the configured specification. The engine allows:

- Definition of the set of variables to be contained
- Definition of subset sizes for data read from intermediate files on a per-variable-basis
- Variable name mappings
- Conversion of geophysical variables e.g.
  - Kelvin to Celsius
  - Counts to Brightness Temperature
  - etc.

In order to relate different MMDs to each other, each match-up and record is uniquely identified by a match-up ID. The value of the match-up ID is the same in all MMDs.

6.3.5 Multi-sensor match-up system

The SST MMS generates and processes MMDs (→ SST-SR-1310 Match-up analysis). A prototype of the MMS was developed in the SST CCI Phase-I, for the purpose of a onetime creation of MMDs for nineteen years of satellite data, starting with pre-matched single-sensor match-up files (MD). Three extensions to this prototype were developed to create match-ups of *in-situ* and satellite data from scratch rather than MD files (→ SST-SR-3120 Compute multi-sensor match-ups), to optimise sub-scene extraction, and to simplify extension with new columns (→ SST-SR-3130 Update flag fields, SST-SR-3260 Optimised for reprocessing cycles).

The subsequent sections define the Earth Observation and auxiliary data organisation, the 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 Earth Observation data are referenced and not copied. The MMS therefore requires a directory structure satisfying a certain archiving rule, which distinguishes types or sensors (→ SST-SR-3250 Structured storage of original inputs):

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

The directory structure may be rooted at any path, for example:

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

MMS input data types are listed in Table 6-4 below (→ SST-SR-1180 Validation datasets, SST-SR-3110 Ingest observations of various types).

Table 6-4: MMS data types

Type	**Type name(s)**
Satellite data	
ESA ATSR Level 1-b	orb_atsr.1, orb_atsr.2, orb_atsr.3
NOAA TIROS/AVHRR Level-1b GAC	orb_avhrr.n07, orb_avhrr.n08, ..., orb_avhrr.n19
MetOp AVHRR Level-1b GAC	orb_avhrr.m01, orb_avhrr.m02
MetOp AVHRR Level-1b FRAC	orb_avhrr_f.m01, orb_avhrr_f.m02
AMSR2 Level-1R	orb_amsr2
AMSRE Level-2	orb_amsre
TMI Level-2	orb_tmi
Auxiliary data	
OSI-SAF Sea Ice	seaice
Aerosol Index	aai
In situ data (histories)	history
Single-sensor match-up datasets (Phase-I only)	
ESA ATSR MD	atsr_md
MetOp AVHRR MD	metop_md
MetOp SEVIRI MD	seviri_md
NOAA AVHRR MD	avhrr_md

Pre-computed extracts are generated during match-up processing for data that otherwise would require reading from many large files for generating an MMD. Generating these extracts is a function of the MMS. Note that these extracts are different from the externally pre-collected single-sensor match-up datasets used in the Phase-I prototype.

Pre-generating extracts is an optimisation within the MMS workflow that facilitates a more rapid processing during improvement cycles. Examples of pre-computed extracts are sub-scenes from individual sensors and interpolated NWP data related to the sub-scenes and interpolated data related to the match-ups (→ SST-SR-3140 Extract satellite sub-scenes, SST-SR-3150 Interpolate NWP). The extracted sub-scenes and interpolated NWP data are stored in intermediate files that are stored in the MMD working directory, which the MMS can use instead of the original satellite data files for generating final MMD files. Different types of pre-computed data are listed in Table 6-5.

New data types can be added on demand by providing additional data readers and configurations (→ SST-SR-3230 Extendable by new readers, SST-SR-3240 Configurable transformation of inputs). Among them are particular types of MMD (→ SST-SR-3130 Update flag fields) in order to ingest processing results into the MMS for inter-comparison and analysis. MMDs with SST retrieval results for sub-scenes can be labelled and ingested, optionally, for different versions of processors (→ SST-SR-3160 Process SST, SST-SR-3190 Support concurrent versions). For example: arc-v1.1_atsr.3, arc-v1.2_atsr.3 (instead of arc_atsr.3).

Table 6-5: MMS-internal data types

Type	**Type name(s)**
Single-sensor sub-scene extracts	
ESA ATSR Level 1-b	sub_atsr.1, sub_atsr.2, sub_atsr.3
NOAA TIROS/AVHRR Level-1b GAC	sub_avhrr.n05, sub_avhrr.n06, ..., sub_avhrr.n19
MetOp AVHRR Level-1b GAC	sub_avhrr.m01, sub_avhrr.m02
MetOp AVHRR Level-1b FRAC	sub_avhrr_f.m01, sub_avhrr_f.m02
AMSR-2 Level-1b	sub_amsr2
NWP data for single-sensor sub-scene extracts	
ECMWF	nwp_atsr.1, nwp_atsr.2, nwp_atsr.3
ECMWF	nwp_avhrr.n05, nwp_avhrr.n06, ..., nwp_avhrr.n19
ECMWF	nwp_avhrr.m01, nwp_avhrr.m02
ECMWF	nwp_avhrr_f.m01, nwp_avhrr_f.m02
ECMWF	nwp_amsr2
NWP forecast and analysis data for a primary sensor in multi-sensor match-ups	
ECMWF	match-up
Single-sensor SST for sub-scenes	
SST CCI	arc_atsr.1, arc_atsr.2, arc_atsr.3
SST CCI	arc_avhrr.n05, arc_avhrr.n06, ..., arc_avhrr.n19
SST CCI	arc_avhrr.m01, arc_avhrr.m02
SST CCI	arc_avhrr_f.m01, arc_avhrr_f.m02
SST CCI	arc_amsr2

6.3.5.2 Match-up database schema

While Earth Observation data are stored in an archive (see Section 6.3.5.1 above), match-ups are stored in a database the schema of which is defined by the four related tables depicted in

Figure 6-20 below.

There are four database tables representing entities of the MMS. The most essential table is the *Observation*. Each observation is associated with a single Earth Observation data file, a single *in situ* data history file, a single auxiliary data file, or an MMD (or MD) file. Observations have sensor, time and location attributes. If a data file contains more than a single observation (MMD, MD) the record number attribute defines the originating record within the file (see

Figure 6-21 below).

The *Matchup* table represents matches of several observations. Each match-up has a single reference observation and is associated with other observations related by the *Coincidence* table. All observations related in a match-up are spatially and temporally overlapping or close to the reference observation. The reference observation always originates from any of the primary satellite sensors, for example ATSR or AVHRR. One of the related observations typically originates from an *in situ* observation or a secondary satellite sensor.

The *DataFile* table represents Earth Observation and auxiliary data files (also internal MMD files) within the MMS disk archive. Each entry refers to a single file. Each file contains one or more observations.

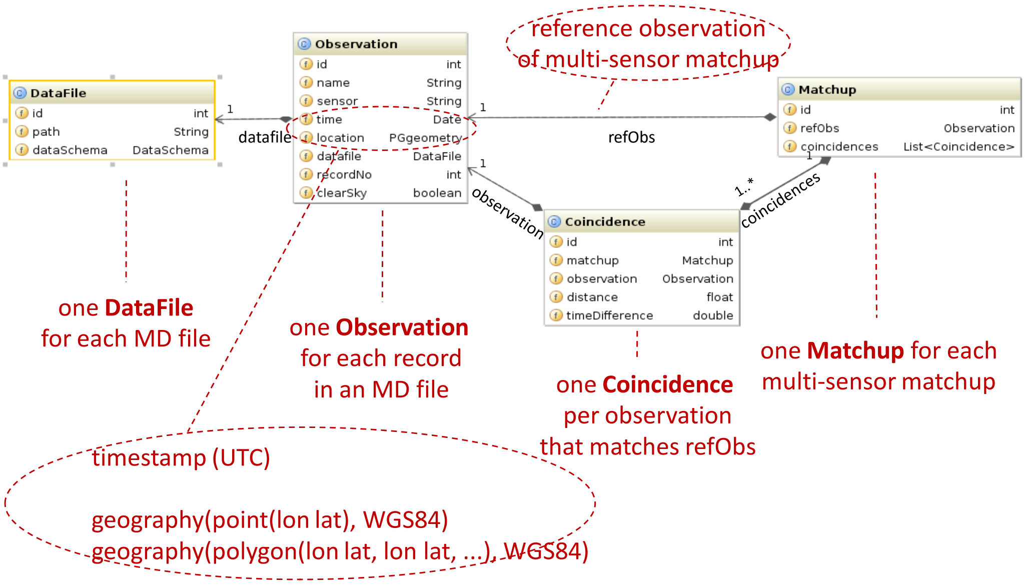
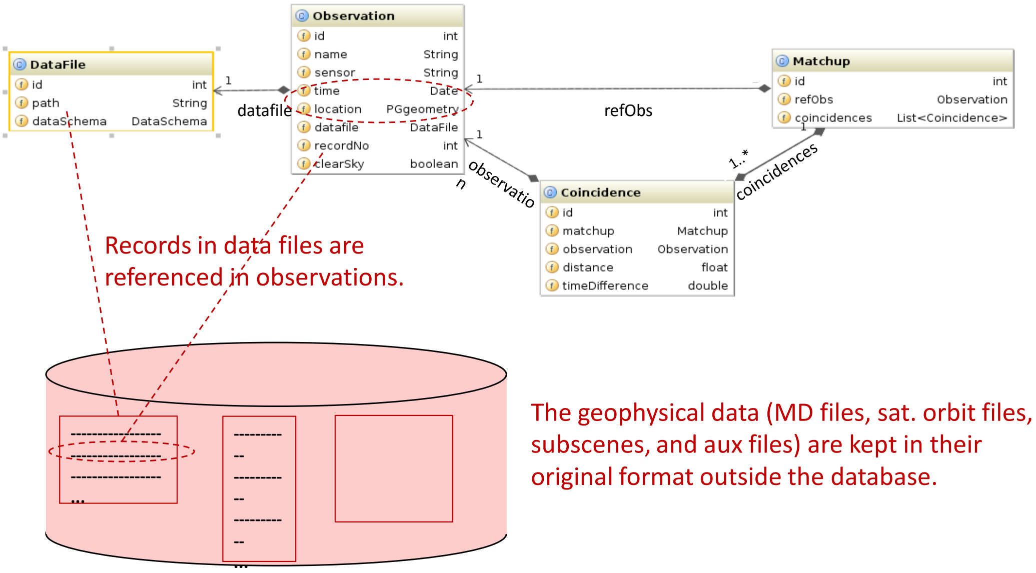


Figure 6-20: Schema of the match-up database



Records in data files are referenced in observations.

The geophysical data (MD files, sat. orbit files, subscenes, and aux files) are kept in their original format outside the database.

Figure 6-21: Referencing input files from observations

6.3.5.3 Match-up system dynamic model

Several processes are defined on the basis of the MMS file system and database schema. The main processes are match-up generation, sub-scene processing, and MMD extraction and analysis. Figure 6-22 below illustrates these processes in the context of improvement (or repeat) cycles (→ SST-SR-3260 Optimised for reprocessing cycles). The processes depicted in Figure 6-22 are the following:

1. Files with Earth Observation, *in situ*, and auxiliary data are ingested into the MMS database (MMDB) (→ SST-SR-2125 Validation data update, SST-SR-3110 Ingest observations of various types). Match-ups are created and added to the MMDB (→ SST-SR-3120 Compute multi-sensor match-ups, SST-SR-3125 Incremental match-up computation), and sub-scenes are extracted from satellite data for faster repeated use.
2. For new versions of a processor sub-scenes are processed and processing results are re-ingested into the MMDB in order to compare different processor versions. This process can also be realised by MMD extraction and external processing. The result of external processing can be ingested too, either as new “input files” or as MMD' file, if it is in MMD format (→ SST-SR-3160 Process SST, SST-SR-3180 Ingest external MMD inputs, SST-SR-3190 Support concurrent versions. SST-SR-5140 Match-up dataset processing).
3. MMD files can be generated with various combinations of variables. The match-ups to be included can be selected typically using criteria like time range and location, or sensors involved in match-ups (-→ SST-SR-3170 Extract multi-sensor match-up datasets)

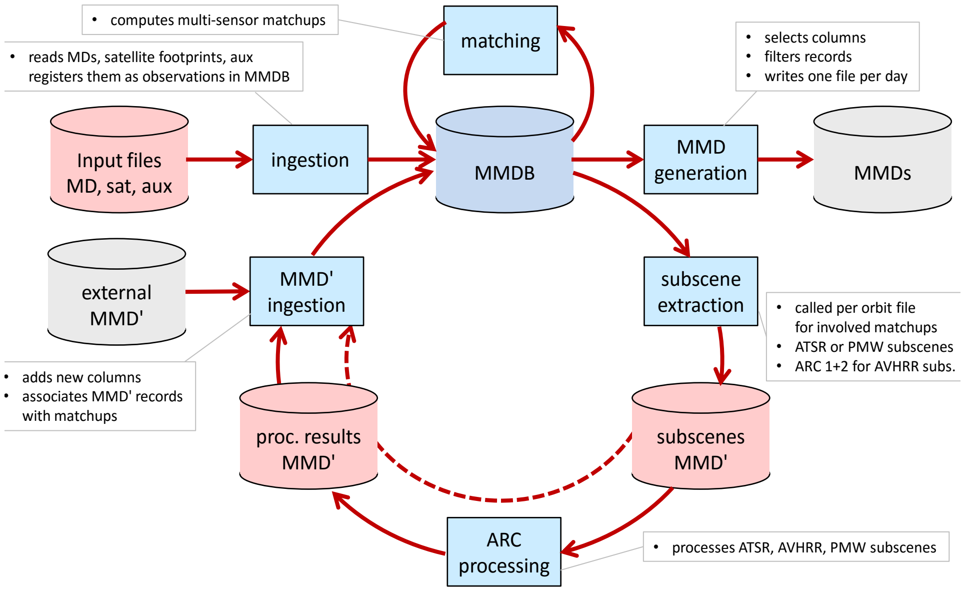


Figure 6-22: Test and validation cycles with the Multi-sensor Match-up System

The processing of sub-scenes (optionally the match-up creation itself) can be carried out either within a testing environment or with support of the parallelised production environment.

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 like a librarian. He 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 MMD files. The interfaces are summarised in

Table 6-6: Interfaces of the MMS.

Table 6-6: 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 (history)	Local file system, MMS	File access Various formats, SNAP 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, match-up generation, sub-scene extraction, NWP interpolation, ARC processing, MMD extraction, MMD' ingestion, de-staging	MMS Implementation Plan [TN 1]
Ifc-15	MMS management interface	MMS client, MMS service	Command line tools, HTTP Functions for dataset definitions, column definitions, data conversion configurations	MMS Implementation Plan [TN 1]

6.3.5.4 MMS Phase II developments

The prototype MMS in Phase-I was developed for one-time MMD creation for nineteen years of input data based on pre-matched single-sensor match-up files. For Phase-II, several functional extensions have been implemented, reflecting the changed requirements on the match-up generation process.

The match-up creation process has been changed to always start using the original input data files (satellite observation or *in situ*), the use of pre-generated single sensor match-ups has been dropped. All input data for the specified match-up, defined by:

- Primary sensor,
- (optional) secondary sensor,
- (optional) *in situ* data,
- (optional) random point generation and
- Time-range,

is registered into the MMD specific database, extracting the required metadata from the original input file, namely sensing times and observation geometry. During the registration procedure, all variables of the input data and the associated attributes are stored to the database. Support for different versions of input data is implemented using configuration files that allow the definition if input datasets to be used for the MMD processing.

Based on this information, the database is used to pre-calculate possible match-up candidates using time and geometric intersection criteria. The resulting list of match-up candidate locations per time interval is processed in subsequent steps to finally assemble the MMD.

The MMS has been updated to generate intermediate files for different purposes to avoid reading the original satellite sensor data more than once, a feature that improves the reading performance of the MMS. This storage of subset data for the possible match-up datasets is implemented for satellite input data subsetting, NWP processing, GBCS processing and more. This feature allows to use intermediate data from any MMD processing step to be re-used as input for subsequent processing steps, e.g. for algorithm fine-tuning.

An optional processing step has been implemented that allows the generation of match-up plots for the processing period. All match-ups detected in the time range are plotted onto a simple plate-carree projection, allowing to visually verify the local distribution of the match-up points. This can simplify the detection of unexpected behaviour or to identify data gaps.

Currently envisaged developments cover mainly the automated generation of MMDs triggered by the acquisition on new satellite sensor input data. For validation purposes, it is planned to implement a scripted solution that triggers the generation of new MMDs whenever a suitable amount of new acquisitions has entered the data storage and is pre-processed to a state that can be used for MMD generation.

For the SST CCI project the MMS will remain a server-based service because the Development Team does not require a stand-alone application. *Further progress on the implementation of the MMS during Phase-II is documented in the MMS Implementation Plan [TN 1].*

6.4 System documentation

The system documentation comprises requirement documents, design and interface control documents, test documents, manuals, technical notes, 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-II 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 set-up 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 have been (or 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 (in Phase-I)	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 (in Phase-I)	Used for METOP and SEVIRI processing (in Phase-I only)		
L3 processor	Aggregate and re-project single sensor outputs	Development from prototype tools	Parameterisation
L4 analysis system	Merge multi-sensor inputs and analyse them	Implementation of Met Office system at Jasmin/CEMSa	Parameterisation, wrapper scripts, auxiliary data handling
User tools	Aggregation, re-gridding, data analysis	Development from prototype tools	None
Grid Engine	Processing system middleware	None	Grid engine set-up, network file system set-up Start scripts and run scripts for processing
Processing monitor (p-monitor, MetOffice Rose)	Processing system	None	Workflow definitions, parameterisation
BEAM and Sentinel-3 Toolbox	Used in MMS, tools, and in L3 processor	SST CCI data readers Extension for uncertainty handling	None
Drupal OPEC Portal	Portal framework	None	Web design, content, system configuration, forum set-up, integration of other services (MERCI, THREDDS, Jira)
THREDDS	Online data server	None	Set-up of THREDDS, OpenDAP, WMS, WCS
Jira (optional)	Issue tracking	None	Set-up of Jira, SST project 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 instruments and spectral 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. For 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 three years in Phase-II of SST CCI, the system has been settled. But in the longer term, renewal of hardware and optional updates or changes of software components can be expected. The system is prepared for these changes by the modularity of its functional components.

The concurrent development for other ECVs is kept independent initially to avoid delays. But 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, 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 ancillary) data that is needed during Phase-II 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 held at the Centre for Environmental Data Archival (CEDA) on tape or network storage (online).

Similarly,

Table 7-3 further summarises the volumes of output data produced in Phase-II of the SST CCI project and when these data will be required to be available online. Three re-processings 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 7-2: Input data volumes for SST CCI Phase-II (estimates)

ID	**Type**	**Period**	**Data for period (TB)**	**Future data (TB / year)**	**Available 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
F1	SLSTR	2017 Q1 & Q2	56.0	112.00	Yes	100	28	32
Z	ERA-Interim	1991 – 2013	9.0	0.25	Yes	100	0	36

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

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	0.77	-	100	28	36
3B	SST CCI L2P (AVHRR-GAC)	1978 – 2016	4.56	0.12	100	28	36
3C	SST CCI L3U (AVHRR-A)	2005 – 2016	1.10	0.10	100	28	36
3F	SST CCI L3U (SLSTR)	2016	0.05	0.05	100	28	36
3G	SST CCI L2P (ATSR)	1991 – 2012	3.36	-	100	28	36
3O	SST CCI L4 (OSTIA, all outputs)	1991 – 2015	7.00	0.6	100	28	36

Figure 7-1 illustrates the online data volumes estimated in Table 7-2 and

Table 7-3 for Phase-II of the SST CCI project. The online storage will be about 55 TB initially, and will reach up to about 180 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.

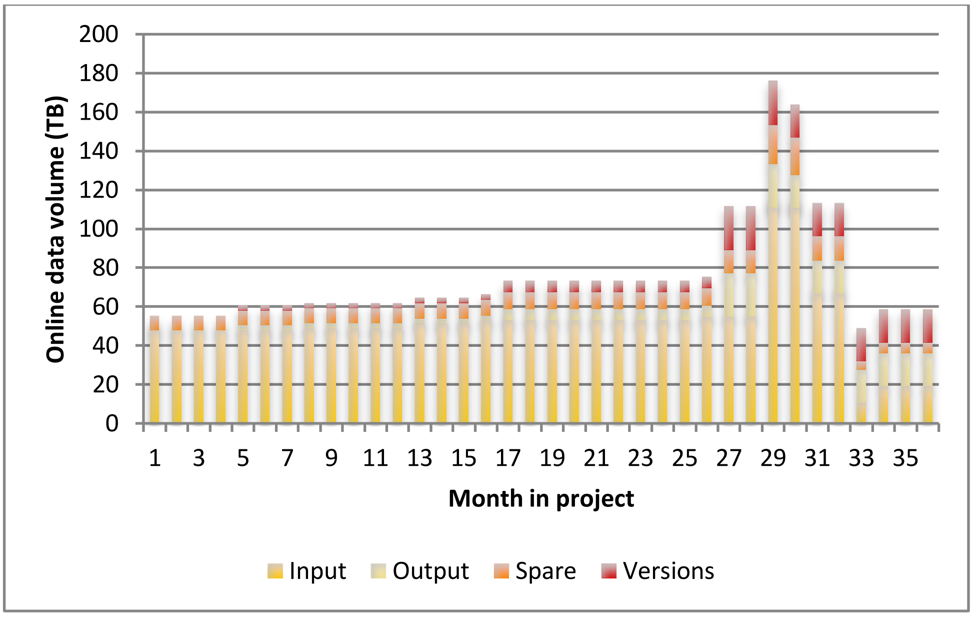


Figure 7-1: Online data volumes needed for SST CCI Phase-II (estimates)

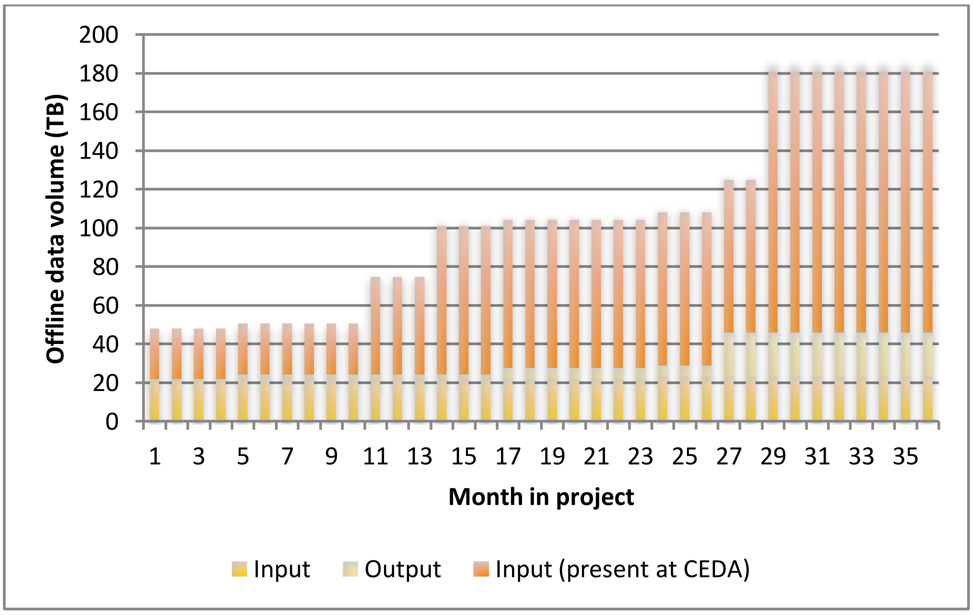


Figure 7-2: Offline data volumes needed for SST CCI Phase-II (estimates)

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).

Table 7-4: CPU core hours needed for SST-CCI Phase-II

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 CEDA/CEMS 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 match-ups, SST-SR-3225 MMDB rebuild). Data access is facilitated by CEDA (→ SST-SR-6370 User service availability).

Irrespective 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 €**).

The costs listed in **Error! Reference source not found.** below are estimates based on 2014 prices and performance and experiences with the 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. Costs for using the CEDA/CEMS infrastructure are subject to negotiation for years beyond Phase-II.

Table 7-5: Cost estimates per year

	First year	Second year	Third year
CEDA/CEMS	Brought in by the project	Brought in by the project	Brought in by the project
Development	150,000 €	75,000 €	75,000 €
Operations	150,000 €	150,000 €	150,000 €
Sum			

8. REQUIREMENTS TRACEABILITY

This section traces input requirements (SST-SR-xxxx) of the SRD [AD 7] and the Statement of Work [AD 1] for Phase-II to sections within this document (§).

Note that in addition to the table below, a statement on the projects' compliancy with the common CCI system and data standard requirements expressed in applicable documents [AD 13] and [AD 14] is provided in [TN 2].

Requirement	Title	Reference
System requirements from the SRD [AD 7]		
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	0, 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	0, 0
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, 0
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

Requirement	Title	Reference
SST-SR-1310	Match-up analysis	6.3.4, 6.3.5
SST-SR-1320	Trend analysis	6.1.6
SST-SR-1330	Processor configuration control	6.3.2
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, 5.2
SST-SR-1501	Bulk archive retrieval	3.3
SST-SR-1510	Archive reliability	5.2
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 match-ups	6.3.5

Requirement	Title	Reference
SST-SR-3125	Incremental match-up computation	6.3.5
SST-SR-3130	Update flag fields	6.3.4, 6.3.5
SST-SR-3140	Extract satellite sub-scenes	6.3.4, 6.3.5
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 match-up 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 match-ups	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	Sub-setting and re-gridding	6.1.6
SST-SR-6230	Online sub-setting and re-gridding	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	Match-up datasets	6.1.3
SST-SR-6310	Match-up inputs	6.1.3
SST-SR-6330	Web site	6.1.5
SST-SR-6340	Science issues	4, 6.1.5
SST-SR-5110	Community process	4, 6.1.5
SST-SR-6360	Forum maintainer	6.1.5
SST-SR-6370	User services availability	5.3

Requirement	Title	Reference
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
SST-SR-5140	Match-up 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	5.2
SST-SR-5220	Development Team	4
SST-SR-5230	Agile requirements selection	3.1, 4
SST-SR-5240	Development and evaluation	4
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	0, 7.3
SST-SR-5280	Storage extension	5.2
SST-SR-5290	Storage extension	5.2
SST-SR-5300	Performance scalability	5.2
SST-SR-5310	Measures for stability	0, 6.3.2, 6.3.3
SST-SR-1367	Operator's Manual	6.4
Technical requirements from the Statement of Work [AD 1]		
SST-TR-12	Online aggregation tools	6.1.3.5
SST-TR-45	Cost-effective platform for SST CCI	5
SST-TR-48	Sustainable MMS	6.3.5
SST-TR-50	Web-based visualisation tools	6.1.3.5
SST-TR-64	obs4MIPS translation tool	6.1.3.5