SDP Architecture

Document Number……………………………………………...……...SKA-TEL-SDP-0000013
Document Type...………………………………..……………………………………..….DRE
Revision…………………………………………………………..……...…………………02C
Authors……………………………..……………………………...…..P. Alexander, V. Allan,
R. Bolton, P. C. Broekema, G. van Diepen, S. Gounden, Á. Mika, R. Nijboer, B. Nikolic,
S. Ratcliffe, A. Scaife, R. Simmonds, J. Taylor, A. Wicenec
Release Date ...……………………………………………………………...…….2016-03-24
Document Classification………………….………………………………...……. Unrestricted
Status……………………………...…...……………………………...………………. Draft

Author:	Designation	Affiliation
Paul Alexander	SDP Project Lead	University of Cambridge
Signature & Date:		
Agnes Mika	SDP COMP, PIP, ARCH project manager	ASTRON
Signature & Date:		

Version	**Date of Issue**	**Prepared by**	**Comments**
01B	2016-02-29	Paul Alexander	Issued for the System review
02C	2016-03-24	Paul Alexander	Issued for the SDP Delta PDR

ORGANISATION DETAILS

Name	Science Process Data Consortium

1. Table of Contents

2. List of Figures

3. List of Tables

4. List of Abbreviations

Abbreviation	Expansion
AAAI	Authentication Authorisation Allocation and Identity (Management)
CA	Certificate Authority
CSP	Central Signal Processor
FFT	Fast Fourier Transform
FOV	Field of View
GSM	Global Sky Model
GUI	Graphical User Interface
HDF5	Hierarchical Data Format (Version 5)
HPC	High Performance Computing
HTML	Hyper-Text Markup Language
HTTP	Hyper-Text Transfer Protocol
IdP	Identity Provider
IVOA	International Virtual Observatory Alliance
L1, L2 etc.	Level 1, Level 2, etc.
LMC	Local Monitoring and Control
LSM	Local Sky Model
LTS	Local Telescope State

MAGMA	Matrix Algebra on GPU and Multicore Architectures
MDL	Manage Data Lifecycle
PSF	Point Spread Function
PSP	Preserve Science Products
PSS	Pulsar Search
PST	Pulsar Timing
QA	Quality Assessment
RC	Regional Centre
RESTful	Representational State Transfer
RFI	Radio Frequency Interference
SaDT	Signal and Data Transport
SDP	Science Data Processor
SIA	Simple Image Access
SPEAD	Streaming Protocol for Exchanging Astronomical Data
TBC	To Be Confirmed
TBD	To Be Determined
TM	Telescope Manager
UPC	Polytechnic University of Catalonia
URI	Unique Resource Indicator

5. References

5.1. Applicable Documents

The following documents are applicable to the extent stated herein. In the event of conflict between the contents of the applicable documents and this document, **the applicable documents** shall take precedence.

Reference Number	Reference
AD01	SKA-TEL-SDP-0000033 Requirements Analysis and Allocations
AD02	ECP-150007
AD03	SKA-TEL-SDP-0000049 SDP Compliance Matrix
AD04	100-000000-002. SKA1-Low Interface Control document SDP to CSP, Rev 1
AD05	300-000000-002. SKA1-Mid Interface Control document SDP to CSP, Rev 1
AD06	100-000000-029 SKA1 Interface Control Document SDP to TM Low, Rev 1
AD07	300-000000-029 SKA1 Interface Control Document SDP to TM Mid, Rev 1

5.2. Reference Documents

The following documents are referenced in this document. In the event of conflict between the contents of the referenced documents and this document, **this document** shall take precedence.

Reference Number	Reference
RD01	SKA-TEL-SDP-0000014 SDP Assumptions and Non-Conformance
RD02	SKA-TEL-SDP-0000015 SDP Execution Framework Design
RD03	SKA-TEL-SDP-0000018 SDP Data Processor Platform Design
RD05	SKA-TEL-SDP-0000023 SDP Preservation Design

RD06	SKA-TEL-SDP-0000025 SDP Delivery Design
RD07	SKA-TEL-SDP-0000026 SDP Local Monitoring and Control Design
RD08	SKA-TEL-SDP-0000027 SDP Pipelines Design
RD09	SKA-TEL-SDP-0000079 SDP Observatory Tools Design
RD10	SKA-TEL-SDP-0000077 SDP Level 4 Interfaces
RD11	SKA-TEL-SDP-0000064 SDP Product Tree
RD12	SKA-TEL-SDP-0000071 SDP Product Tree Descriptive Spreadsheet

6. Document scope

This document describes the Science Data Processor (SDP) high-level architecture, placing it in the context of the SKA telescopes. The main SDP functions and products are decomposed at Level 2 (L2). The allocation between the SDP functions and products is presented. The SDP Level 2 requirements which this architecture is designed to meet are listed in [AD01].

The architecture is presented in a series of dynamic and deployment views. The L2 architectural elements are defined and described briefly. A detailed description and further decompositions are presented in a series of design documents [RD02 to RD12].

7. System Overview

7.1. The SDP Challenge

The Science Data Processor (SDP) challenge has aspects which, when considered together, make it unique among comparable systems in astronomy. These include:

1. The SDP is an intrinsic element of the SKA telescopes and not a separately scheduled, remote processing facility. Hence:
    - The SDP will need to be scheduled as an integral part of the observatory, i.e., the data ingest, the raw data storage and processing (into science data products) will need to be carefully coordinated. In contrast, in typical observatories the data ingest and data processing are largely decoupled by an archive that permanently stores all of the raw data.
    - It is also very different from standard HPC facilities, which do not usually need to manage near real-time systems with long distance data delivery and very high data delivery rates.
2. The SDP processes the incoming data via a set of pipelines. The computational requirements to process this incoming data into scientifically useful data products are significantly greater (by approximately two orders of magnitude) than the largest systems currently used in astronomy and must be able to operate largely autonomously.
    - For this reason the capital costs and operational costs associated with the compute hardware become very important considerations. To achieve the required processing in a time frame comparable to that on which the data are collected, a high degree of parallelism will need to be applied to the processing of each observation. This is in contrast to the typical situation at radio-interferometric facilities where many observations are indeed processed in parallel but with a limited degree of parallelism in the processing of each observation.
3. The incoming data rate is so high that raw data are unlikely to be kept permanently. Also, the temporary storage of raw data will need to be minimised (to perhaps as short as 6 hours). This has the implication that data processing and Quality

Assurance (QA) will need to be automated with little or no possibility for intervention by operators or scientists.
4. The SDP will need to perform some of the data processing within strict deadlines (e.g., around 15 s for real-time calibration).
5. The SKA telescopes are novel and very large facilities. Past experience with (at their time) similarly ground-breaking facilities has shown that once the SDP is online, considerable scientific benefits can be achieved through modifying, improving and adding to the algorithms exploited in the SDP. This means that the SDP must have sufficient flexibility to allow such long-term improvement. Furthermore, if the antenna arrays are upgraded, the SDP will need a corresponding upgrade in processing ability.
6. The requirements for the SDP element are evolving and will continue to do so into the operational phase of the telescope. During the designed 50 years lifetime of the Observatory the key science objectives will almost certainly change significantly and thus the requirements for the SDP element will evolve as well.
7. The lifetime of the telescope, the lifetime of the compute hardware and the need to minimise power consumption are such that the hardware element of the SDP will need to be refreshed, or completely replaced, on a relatively frequent timescale. The software may is very likely to need corresponding updates.

7.2. High-level Design Principles

The SDP architecture was designed with the following four high-level design considerations in mind:

1. **Maintainability and extensibility**: it must be possible for the SKA Observatory to keep the SDP software running efficiently as the data processing algorithms evolve and the underlying hardware is refreshed.
2. **Affordability**: the SDP must be affordable, i.e., the chosen architecture should ideally minimise expenditure on capital and operational costs. This may be in conflict with the other design principles and in that case the selected architecture should not cost significantly more than the cost of other possible architectures.
3. **Support for *current* best-practice algorithms**: the SDP must support all of the current best-practice algorithms used in radio interferometry and in particular those used by the pathfinders and precursor instruments.
4. **Scalability**: the SDP should be scalable to handle a range of computational and data throughput requirements. This is in contrast to a potential architecture which aims to achieve a solution for a particular design parameter point. In particular, the architecture must scale down efficiently since the current system sizing defines a maximal capability for the SDP which is likely to only be achieved after some years of operation.

The operational cost of electrical power for the SDP means that the hardware refresh cycle is going to be relatively rapid so as to make best use of the increasing power efficiency of new computing hardware. It is important that investment continues in the SDP software to match these hardware refreshes to ensure that the system remains efficient and maintainable.

Experience with the SKA pathfinders (and other radio telescopes) has shown that once commissioning and scientific observations with a new telescope begin, new requirements for the SDP software will emerge. Examples are that very frequently new algorithms and techniques must be developed to correct unexpected effects found in the telescope system, the telescope algorithms or in the atmosphere. This operational necessity therefore requires that the design should allow for the functionality of the SDP to be updated with reasonably limited effort.

Additionally, the design and construction phases of SDP are sufficiently long that the system maintainability will become important long before the SDP reaches full operations. For example, the time period between the time of writing and the beginning of construction in 2018 is *shorter* than the anticipated time between the beginning of construction and full science operation of the SKA in 2023.

It is very likely that some novel algorithms will be required to get the best possible science from the SKA telescopes. It is not, however, the purpose of the SDP design work to be researching these new algorithms – the current SDP design phase has to focus on delivering a workable and complete design and that precludes reliance on not-yet-discovered algorithms. At the same time, the SDP processing will be very challenging. Therefore the SDP will support an appropriate selection of current best-practice algorithms for interferometric data reduction tailored for the SKA functions and requirements.

A critical corollary is that iterative algorithms, which are the basis of current state-of-the-art imaging pipelines must therefore be supported and this is a major design driver for the top-level SDP architecture.

7.3. Scope of the SDP

The SKA Observatory is distributed across three physical sites (see Figure 1). The headquarters are located in the UK and the two observatory sites in South Africa (SKA1 Mid) and Australia (SKA1 Low). The SDP will be deployed to each of the observatory sites: there will be separate platform deployments at the SDP data centres in Cape Town and Perth; however, the SDP software is a common deployment to both of these platforms.

The SKA has adopted a Tiered model for data delivery with SKA Regional Centres (SRCs) playing a formal role of accepting/requesting SKA-SDP data products and making these available to astronomers together with processing resources and support. The SKA Regional Centres will enforce the SKA data access policies.

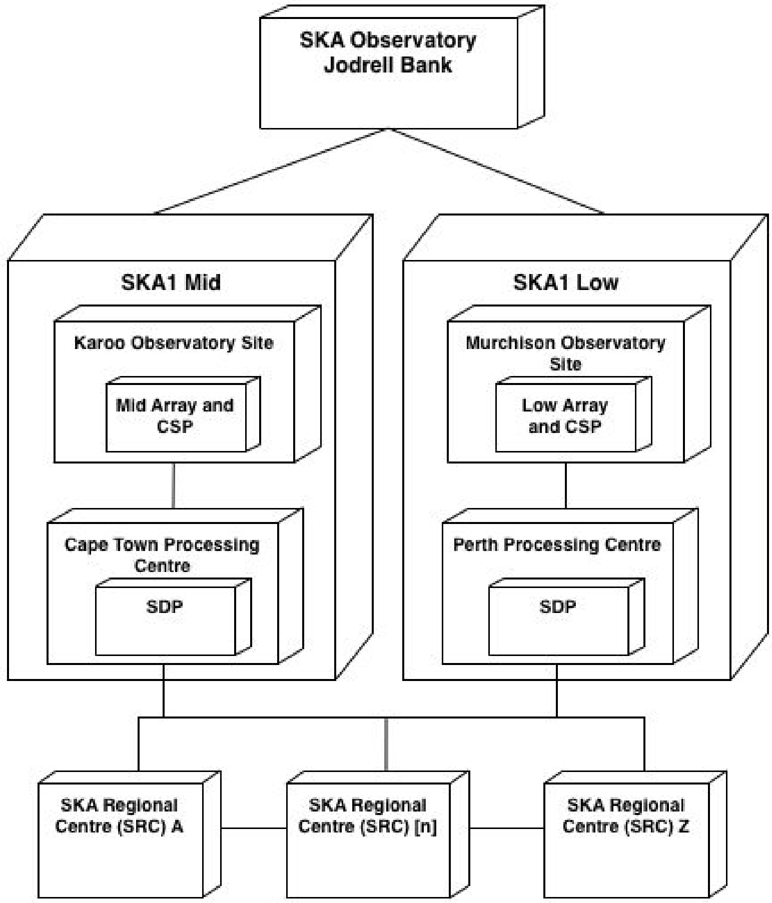


Figure 1: *Deployment of the SDP within the SKA Observatory. The SKA Observatory has its headquarters in the UK at Jodrell Bank and two observatory sites in South Africa and Western Australia. There will be two physical platform deployments to processing centres in Cape Town (South Africa) and Perth (Australia). The interconnects between the physical deployments are shown by solid lines. The SDP software will be common across the observatory and will be deployed to each of the SKA processing centres associated with SKA1 Mid and SKA1 Low. The SKA Regional Centres (SRCs) form a federated group of data and science support centres: SKA data products may be moved between them.*

The SDP is responsible for the processing of various observed data into science-ready data products, the long term preservation of these data products, and the delivery of these products to the SKA Observatory, across both SDP sites. The science data products may be queried by duly authorised users at each SDP site with further processing and analysis being performed at SKA Regional Centres (SRCs).

The SDP is also responsible for:

- Computing feedback information to the SKA system for calibration solutions;
- Alert generation;
- Providing additional metadata to describe the provenance of data;
- Quality Assurance information to assess allow evaluation of the efficacy and scientific quality of the processing.

The SDP receives observed (raw) data from the Central Signal Processor (CSP) element while associated metadata (describing the telescope configuration, observation being performed, etc.) are received from the Telescope Manager (TM) element. The SDP is controlled, scheduled and monitored by the TM. The data transfer between the CSP and SDP as well as the TM and SDP is performed by the Signal and Data Transport (SaDT) element. The scope of the SDP is shown in context in Figure 2.

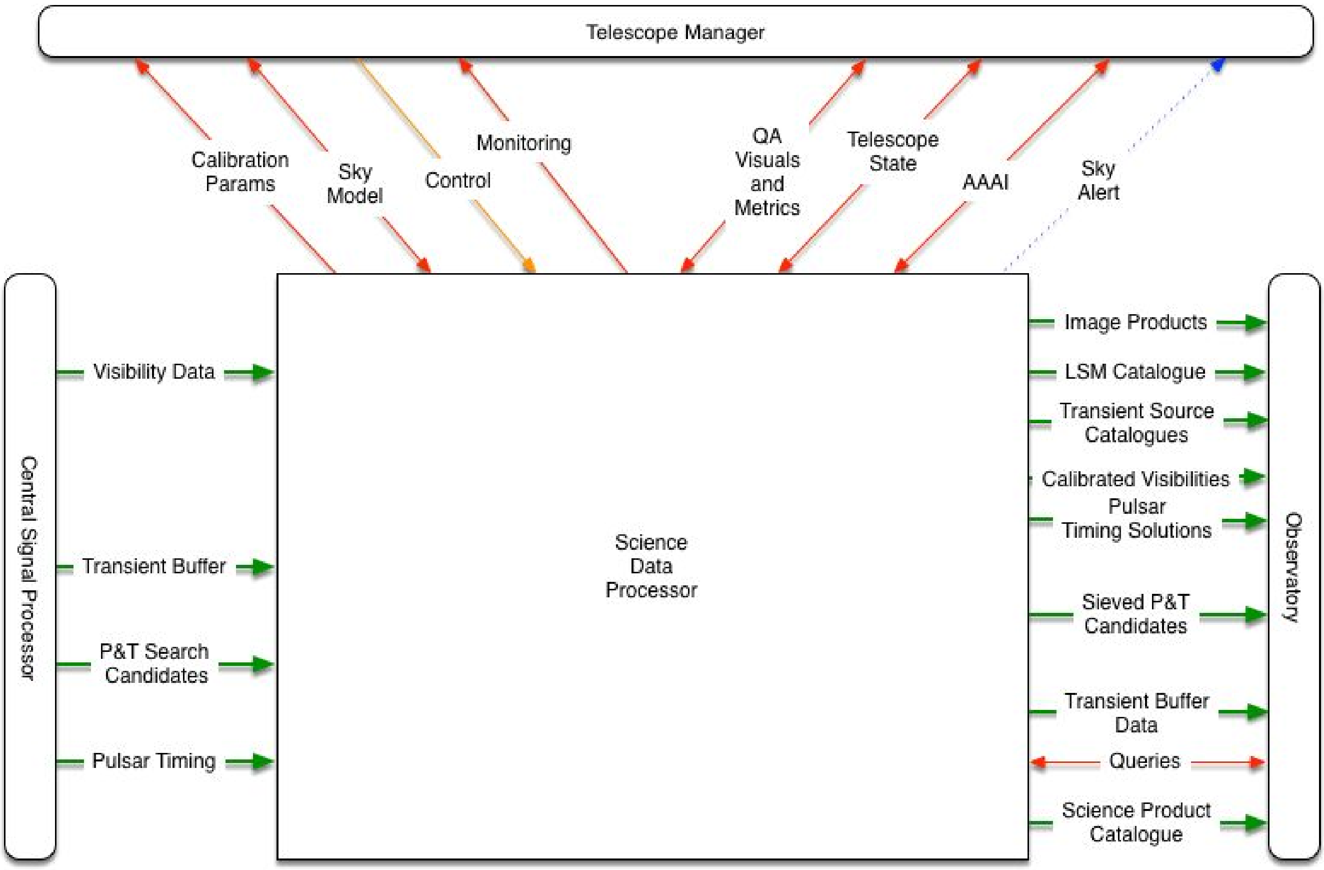


Figure 2: *Context Diagram depicting the scope of the SDP, illustrating the raw astronomic data being input from the CSP and combined with data and metadata input from the TM. These data are processed and output as Science Products which can be queried. The Central Signal Processor, the Telescope Manager and the Observatory are external to the SDP. Red arrows indicate the flow of metadata and control information; green arrows indicate the flow of data and data products; blue arrows represent sky alerts. Input data are received by the SDP from the Central Signal Processor and data products are delivered to the Observatory (an abstract external entity decomposed further below).*

The SDP receives two types of observational data from CSP. These are Visibility Data received as a continuous flow to be processed and imaged, and non-imaging data (Transient buffer, Pulsar and Transient Search Candidates and Pulsar Timing Data) received as discrete chunks. The SDP provides the ability to process both these data independently (as may be the case during commissioning) or commensally (as expected during operation).

Additionally, the SDP supports commensal operation and the processing of multiple observational programmes each of which utilise a subset of the telescope — the so-called sub-arraying or multi-beaming for aperture arrays. The precise definition of sub-arrays is under discussion within the Telescope Resolution Teams.

Along with observational data from the CSP, Control, Monitoring, Feedback and Event information is provided from and to the Telescope Manager as shown in Figure 2. These data and metadata (operating under different cadence depending on the processing being performed) provide information on:

- real-time **Calibration** solutions for updating the SKA system
- continuous **Sky Model** feedback: information on Local Sky Model (LSM) updates to the Global Sky Model (GSM)
- **Control** information to steer processing pipelines
- **Monitoring** information describing the health-state of the SDP and providing resource information for managing the execution of the scheduling blocks
- **Quality Assurance** data in the form of visual information for Operator intervention and metrics for automated Quality Assurance (QA).
- metadata describing the **Telescope State** including configuration information, numerical models deployed, empirical parameters used and conventions followed.
- **Alerts** which can be generated and promulgated from the SDP, via TM, to the SKA to permit follow-on processing.

The SDP allows appropriately privileged external users and users within the Observatory to query the metadata associated with the data products. The required Authentication, Authorisation, Allocation and Identity (AAAI) management information about users is requested from TM. The result of such a query may be the bulk transfer of data to a Regional Centre (RC) mediated by SaDT over International WANs and National Research and Education Networks (NRENs).

The SKA Regional Centres will be required to support a query client and to support a function to receive data from the SDP. The SDP produces a number of standard data products which may be maintained at both SDP sites. The standard data products are described in Table 1.

7.4. Data Products

The following table provides a list and description of the data products which the SDP will deliver.

Table 1: SDP Data Products

Data Product	Description (For each product a QA and Processing Log will also be maintained)
Image Products 1: Image Cubes	1. Imaging data for Continuum, as cleaned restored Taylor term images (n.b. no image products for Slow Transients detection have been specified – maps are made, searched and discarded) 2. Residual image (i.e. residuals after applying CLEAN) in continuum 3. Clean component image (or a table, which could be smaller). 4. Spectral line cube after continuum subtracted 5. Residual spectral line image (i.e. residuals after clean applied) 6. Representative Point Spread Function for observations (cutout, small in size compared to the field of view (FOV))
Image Products 2: UV-grids	1. Calibrated visibilities, gridded at the spatial and frequency resolution required by the experiment. One grid per facet (so this grid is the FFT of the dirty map of each facet). c.f. ECP150007 [AD02] 2. Accumulated Weights for each uv cell in each grid (without additional weighting applied).
Calibrated Visibilities	Calibrated visibility data (for example for EoR experiments) and direction-dependent calibration information, with time and frequency averaging performed as requested to reduce the data volume.
LSM Catalogue	Catalogue of a subset of the Global Sky Model (GSM) containing the sources relevant for the scheduling block being processed. These are the sources in the FOV, as well as, potentially, strong sources outside of the current FOV. Initially, the LSM is filled from the GSM; during the data processing the sources found in the images are added to the LSM.

Transient Source Catalogue	Time-ordered catalogue of candidate transient objects pertaining to each detection alert from the real-time, so-called, Fast Imaging.
Pulsar Timing Solutions	For each of the observed pulsars the output data from the pulsar timing section will include the original input data as well as averaged versions of these data products (either averaged in polarisation, frequency or time) in PSRFITs format. The arrival time of the pulse. The residuals from the current best-fit model for the pulsar. An updated model of the arrival times.
Transient Buffer Data	Voltage data passed through from the CSP when the transient buffer is triggered.
Sieved Pulsar and Transient Candidates	A data cube which will be folded and dedispersed at the best Dispersion Measure (DM), period and period derivative determined from the search. A single ranked list of non-imaging transient candidates from each scheduling block. For those transients deemed of sufficient interest, the associated “filterbank” data will also be archived. A set of diagnostics/heuristics that will include metadata associated with the scheduling block and observation. If a sufficiently interesting pulsar is discovered this will generate an alert as well as being recorded in a Log. (While we have a requirement to report single pulse events, it is not clear whether we have to provide alerts for anything other than single pulses. This is being referred to the Telescope Teams.)
Science Alerts Catalogue	Catalogue of Science Alerts produced and communicated by the SDP. The alerts themselves are IVOA alerts; this catalogue provides a searchable and retrievable record of past alerts.
Science Product Catalogue	A database relating to all Science Products processed by the SDP. It includes associated scientific metadata that can be queried and searched and includes all information so that the result of a query can lead to the delivery of data.

7.5. Overview of Architectural Drivers and Constraints

Based on the considerations on architectural drivers and constraints described in previous sections, we summarise the principles to which the SDP architecture must adhere:

- Meet the formal requirements on the SDP.
- Allow for the separation of the data flows according to their nature: data for real-time processing (fast imaging, real-time calibration), low-volume data (transient buffer, pulsar data) and high-volume data.
- Allow for an evolving hardware platform.
- Allow downward scalability: a subset of the SDP might need to be built for commissioning or cost reasons.
- Enable the incremental delivery of the system in a modular fashion, with well-defined interfaces for testing and integration.
- Enable the SKA to build an affordable SDP.
- Allow for the use of state-of-the-art algorithms.
- Support only standard data products (see their definition under "Scope of the SDP"). Producing other data products is the responsibility of the Regional Centres.
- Support visualisation (real-time view) of QA metrics for observatory use.
- Support metadata queries on the Science Data Products (but no direct queries on the science data).
- Manage highly distributed data across a large cluster.
- Accommodate component failures and thus maintain availability.
- Exploit the parallelism inherent in the data.
- Deal with sub-arraying and commensal observing by having multiple instances of the processing steps.
- Provide a subset of IVOA-type services exposed to end-user astronomers: a fuller set of IVOA services and further science processing beyond standard products should be done by the Regional Centres.
- Simplify the design by assuming that observations are independent and that the SDP produces Science Data Products on an observation-by-observation basis. Thus we assume that the SDP does not routinely do averaging outside of a single observation: such averaging is assumed to occur in Regional Centres.

- Restoring data from external sites is not part of the current design, but data may be imported from the alternate SDP site.

- Data products are formally delivered to the SKA Observatory: the observatory sets policies on data access, delivery to Regional Centres etc.

- The SDP will not archive raw visibilities in full operation, only calibrated visibilities together with calibration solutions. These will be archived and delivered as a specifically requested data product. Furthermore, since the data volumes in this case are very large, it is assumed that this will only happen for science programmes for which this product is essential (such as the Epoch of Reionisation).

- Design a system which is only required to run at the SKA Science Data Processing Centres. We are not planning or designing the SDP software to run at the Regional Centres. If we can provide software to Regional Science and Data Centres (RCs) for minimal extra cost, we do not rule out doing that; however, it does not guide our design of the SDP software.

Starting from these principles, the Level one and two (L1 and L2) functional architectures were developed and are presented in the following sections.

8. SDP High-Level Architecture

We begin by presenting the SDP Architecture in terms of a functional decomposition of the system at Level 1 (L1) and Level 2 (L2). In addition to the functional decomposition, we introduce a view of the system which illustrates the flow of data and information between the functions.

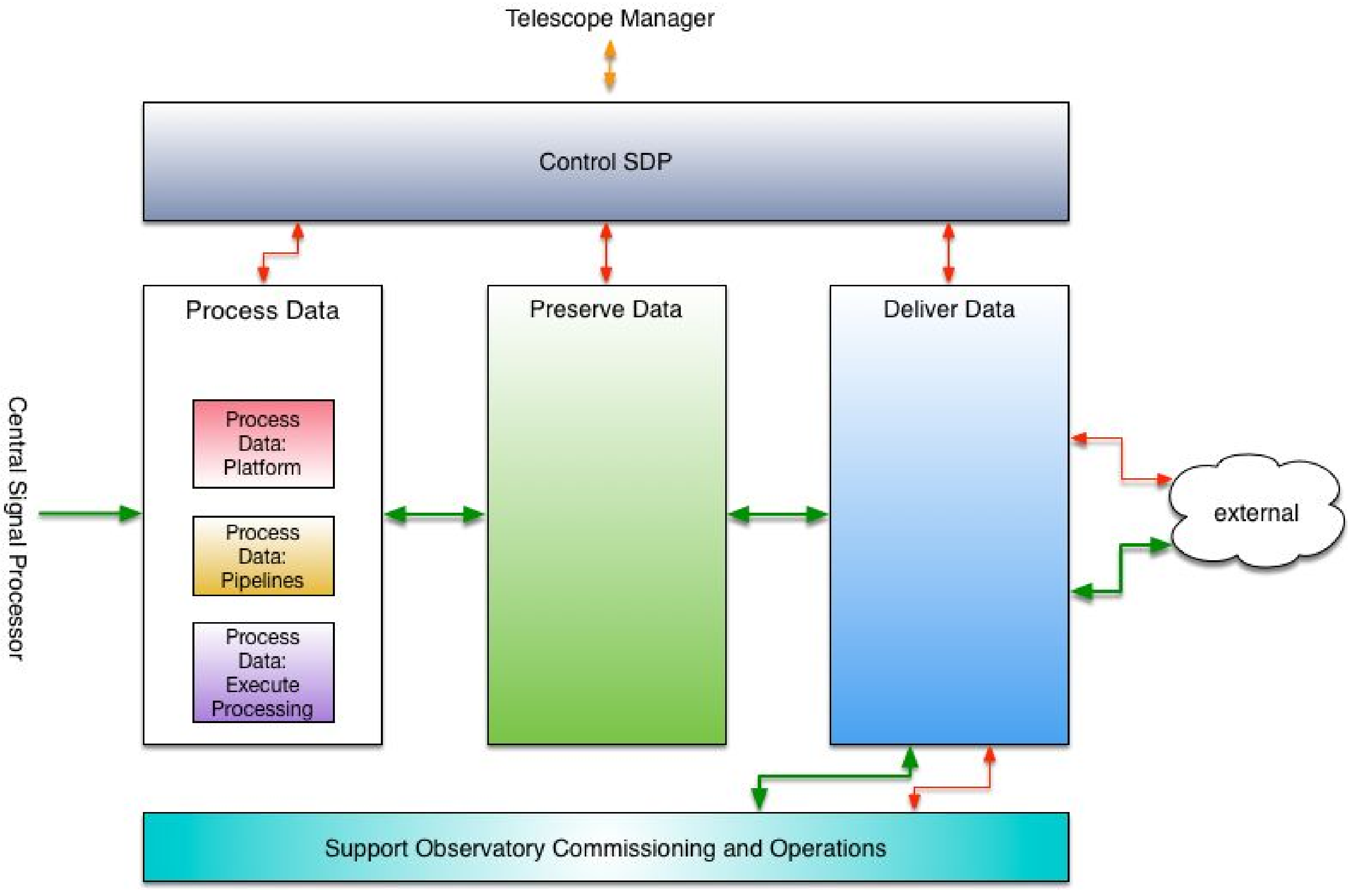


Figure 3: *Functional and data flow at L1, showing the SDP functions at L1 as in Figure 3, but illustrating additionally the flow of data and information between them. Arrows indicate the interfaces over which data, metadata or control information passes and the directions of the flow: green arrows indicate data flow and red arrows control and/or metadata. The block colours are for cross referencing to the functional decomposition at level 2 (L2), where we further subdivide Process Data into: Process Data: Platform, Process Data: Pipelines and Process Data: Execute Processing. Seven design documents corresponding to each of these blocks provide architectural descriptions down to at least level 3 (L3).*

8.1. L1 Functional Architecture

The functional decomposition of the SDP element is shown in Figure 3. The SDP architecture, broken down to L1, contains a Control function which communicates with Process Data, preserve Data and Deliver Data and provides the external interface with the Telescope Manager (TM). The data received from the Central Signal Processor are processed within the Process Data function. The resulting Science Data Products are then handled by the Preserve Data function, which provides all persistence functionality and finally, by the Deliver Data function. Deliver Data provides the external interface with the Observatory and the Regional Centres and interfaces to the Support Observatory Commissioning and Operations function.

8.2. L2 Functional Architecture

The SDP architecture is decomposed at L2 as shown below in Figure 4. Here, we also show the flow of data and information through the system during execution. The key concepts of the architecture are now summarised.

- SDP is scheduled and controlled by TM. TM interfaces to the **Control SDP** function which is decomposed into a number of functions and services:
  - **Master Control** is the control interface and has a control interface to all SDP functions
  - Processing is organised into a set of capabilities. A Capability is the minimum-sized task which can be scheduled by TM. In general, more than one capability will be executed by the SDP at any time. Capabilities are discussed in more detail below.
  - TM maintains a full description of the telescope in a logical structure called the *Telescope State*. For a given capability, a *Local Telescope State* (LTS) is produced and made available to all functions that need to either read or update the state (**Manage Local Telescope State**):
    - **Manage Local Telescope State** is implemented as a service.
    - Changes in the LTS are transmitted back to the TM-owned Telescope State and are persisted as a set of time-stamped data.
    - Information on the Telescope State which is required at high cadence is made available via the Produce Fast Telescope State.
    - The LTS is also used to construct SDP data products and the index of data products.
  - Astronomical information is contained in the Global Sky Model by TM and abstracted into a Local Sky Model (LSM) in a similar way to the Telescope State. **Manage Local Sky Model** is implemented as an updatable service. The LSM may also be queried in the production of data products.
- Data received by SDP from CSP is in a variety of formats depending on the experiment being performed. All such data are transmitted over a switched network infrastructure (**Switching**):
  - **Receive** functions associated with each type of data take the data, together with metadata, and package the received data in a SDP internal format which we refer to as data drops (see below).
- All processing of data is scheduled as a capability.
  - SDP capabilities operate both on streaming data and data which have been buffered (**Buffer Data**) over the course of a complete synthesis observation.
  - Capabilities can be scheduled which process previously buffered data.
  - Capabilities can be scheduled which process data in real time with requirements on latency and cadence: for example, to produce calibration

    solutions in real time which are transmitted via TM to CSP to be applied prior to beamforming.
  - Processing pipelines produce data products together with Quality Assessment metrics – the latter are aggregated (**Aggregate QA metrics**) and a visualisation provided for Observatory use (**Visualise QA Metrics**): the aggregated metrics can also be persisted as an SDP data product.
- Processing within SDP may generate astronomical Alerts. These are managed within SDP (**Manage Science Events**) and packaged into IVOA Alerts which are transmitted by Telescope Manager to the outside world. A catalogue of all events generated is constructed and maintained as an SDP data product.
- SDP is conceived as a highly parallel physical architecture. Processing within a capability in this highly distributed environment is managed by the **Execute Processing** function. Data management is similarly a critical function: the data are structured as a large number of distributed data objects (drops), and these data objects managed by the **Manage Data Lifecycle** function.
- Data products are logically assembled from the distributed data objects via the L1 **Persist Data** function which decomposes into:
  - **Stage data Products** forming the interface between the processing system and the concept of data products
  - **Index Science Products** which collects science metadata and provenance data for the data products and produces an index which enables scientific queries against the products. It also establishes the relation to the set of drops comprising the data product and their physical location.
  - Indexed data science data products are persisted (**Persist Science Products**) and can be backed up (**Backup Science Products**). All data objects through the system including persisted objects are subject to data lifecycle management (**Manage Data Lifecycle**).
- Access to data products by any external actor (Observatory staff, Astronomer, Regional Centre etc.) is managed in a coherent way by the L1 function Deliver Data which decomposes into:
  - A **Query and Request Data** function providing the external interface and the means to query the SDP science data product index – this itself being : the latter is itself a data product which may be sent to regional centres for efficient query management.
  - Data access policies and other security policies are implemented via an **AAAI** function with an interface to TM.
  - Queries result in the assembling of (parts of) SDP data products into data to deliver to the authorised users (**Prepare and Deliver Data**)
- Observatory staff are treated in the same functional way as external users but with different policies enabled as encapsulated in the L1 function **Support Observatory Commissioning and Operations**.

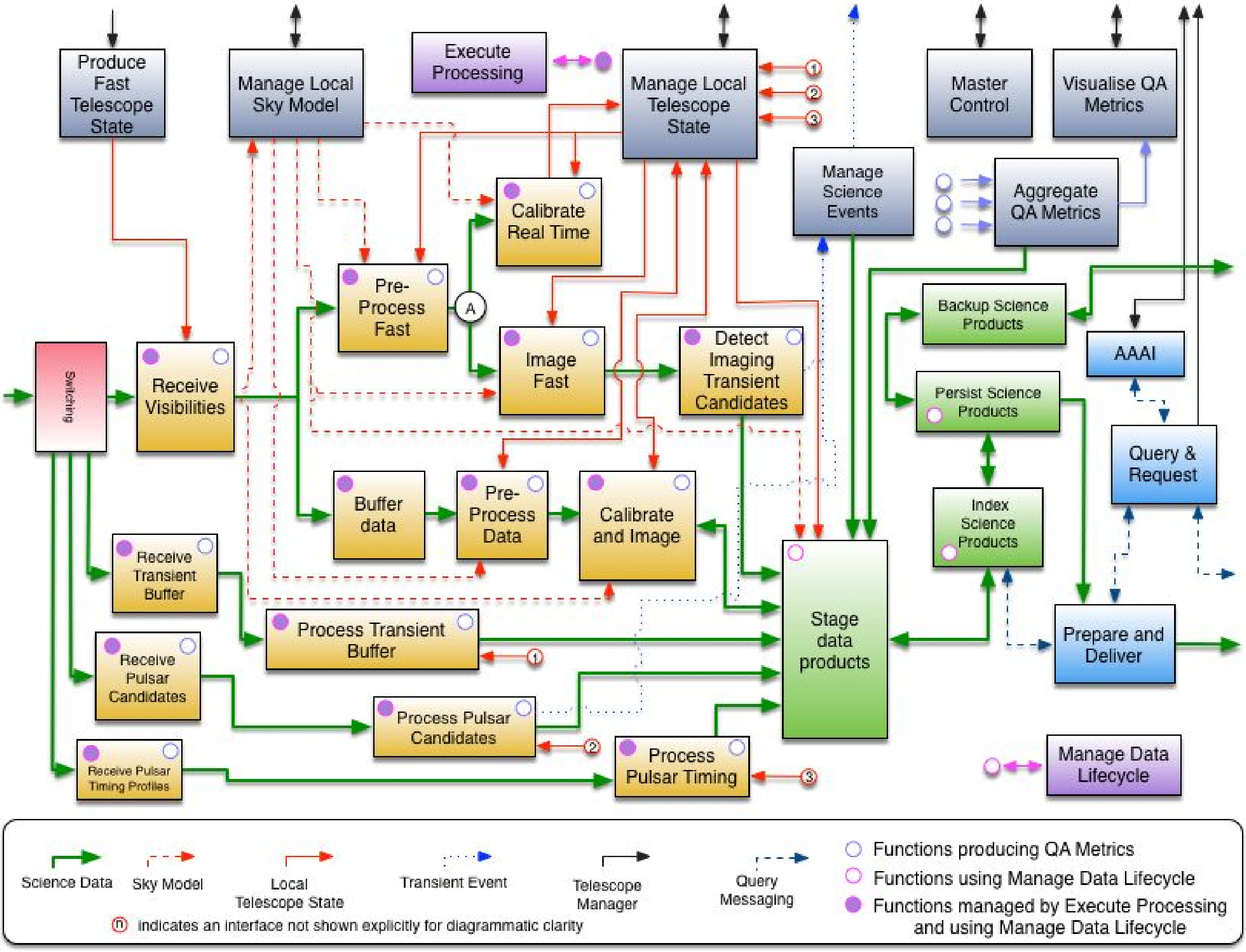


Figure 4: *Functional and data flow at L2, showing the SDP functions at L2 and the flow of data and information between them. Arrows indicate the interfaces over which data, metadata or control information passes and the directions of the flow: green arrows indicate data flow, red arrows control and/or metadata, blue event information, black interfaces to Telescope Manager (TM) and dark blue Deliver Data messaging. Solid and dashed lines have no additional meaning and are used only to help in the visual presentation. The control and metadata interfaces for Manage Data Lifecycle and Aggregate QA Metrics are shown using colour-coded circles within function blocks. The block colours provide a cross reference to the functional decomposition at L1 including the subdivision of Process Data into Process Data: Platform, Pipelines and Execute Processing. The Execute Processing Function has an interface to all of the Process Data: Pipelines functions. The numbers within red circles are used to show the linkage between some functions and the Manage Local Telescope State Function.*

8.3. Capabilities

The Functional Architecture provides a complete picture of the functions that the SDP can perform. However, not all of this functionality will be associated with, or required for, a given scheduled observation or scheduled processing task. We therefore define the concept of a *capability*. In general, a capability is a grouping of SDP components and products to provide a particular pipeline analysis function. In the context of the SDP, a capability maps directly to a particular analysis or engineering task. More specifically for SDP:

- A capability is the minimum-sized task which can be scheduled by TM – TM does not control what happens within a capability;
- Capabilities do not share information directly, but information (e.g. Telescope State information) may be made available between capabilities via TM;
- A capability is defined by a set of configuration parameters, functions and products which are required to produce a defined product;
- When a capability is executed, further parameters may be defined that refine the behaviour of the capability and the SDP resources are allocated for execution;
- The SDP will support multiple simultaneous executing capabilities – this will be a standard situation with for example one executing capability being responsible for receiving data for the current observation while another executing capability is processing data stored within the Buffer.

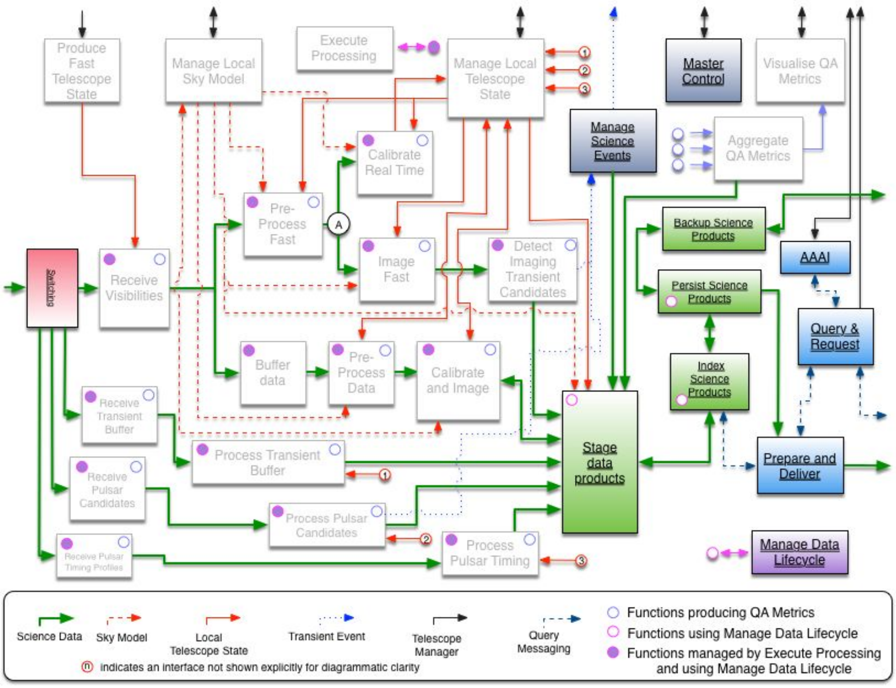


Figure 5: *Functional and data flow at L2 showing those functions which exist independently of capabilities – singleton functions. The diagram uses the same notations as Figure 4: functions associated with capabilities are shown in grey, singleton functions existing independently of capabilities are shown in the same style as Figure 4 and with their function names highlighted with a double underline.*

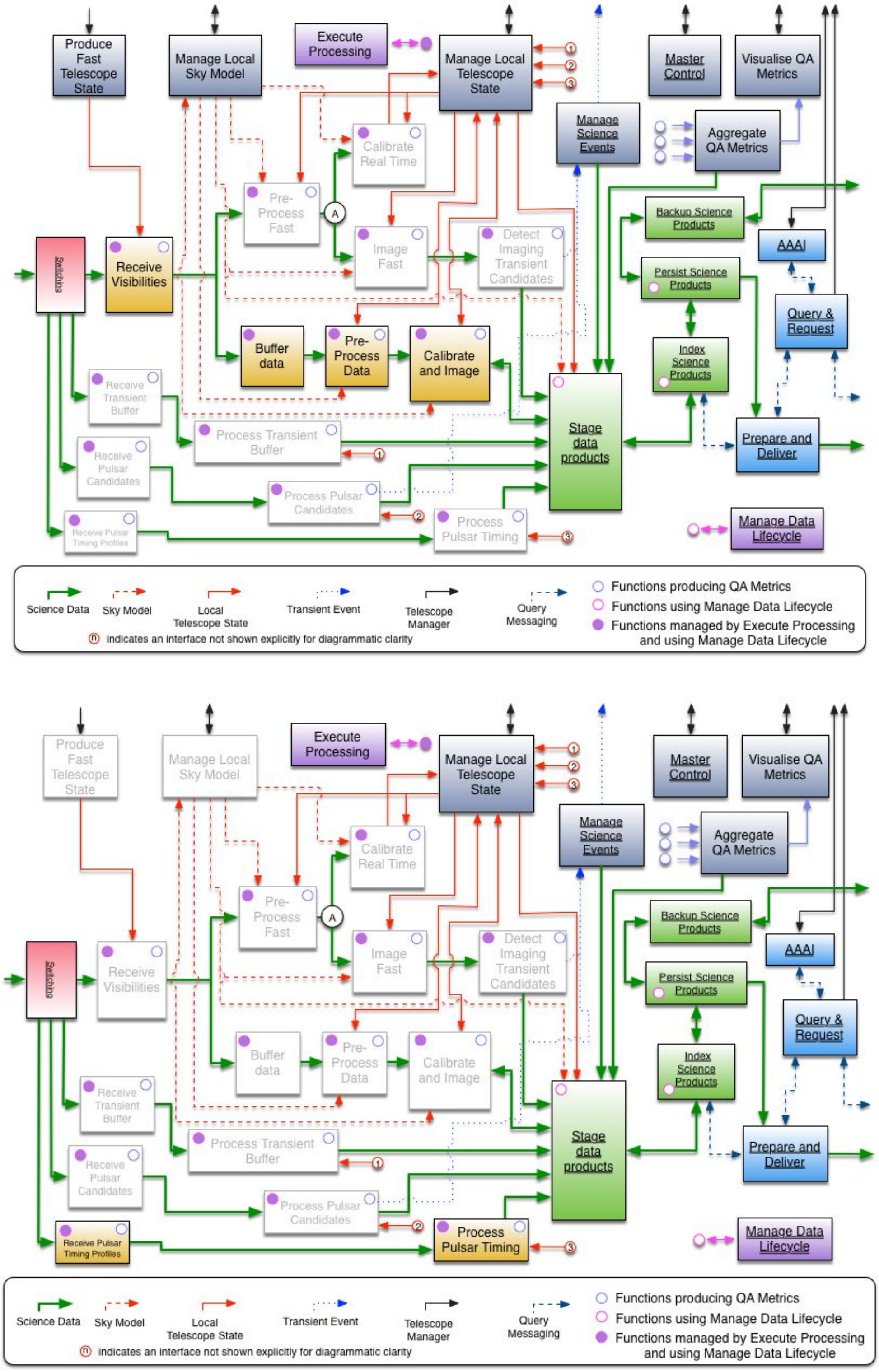


Figure 6: *Functional and data flow at L2 illustrating specific capabilities. The upper panel schematic shows those functions associated with an imaging capability plus singleton functions (highlighted with a double underline of the function name) while the lower panel shows those functions associated with processing pulsar candidates. The diagram uses the same notations as Figure 4: singleton functions are those functions shown highlighted in Figure 5.*

Not all SDP functions are associated with individual capabilities. We define those functions that exist independently of capabilities as “singleton functions”. Figure 5 shows (as coloured, non greyed functions) those singleton functions which persist and which are present for all capabilities. Figure 6 shows two examples of capabilities defined in terms of the SDP functional decomposition at L2.

A capability may further be regarded as a package of functions which are required for a given operation: not all of this functionality may be required for every capability. As discussed below and in more detail in the Execution Framework document [RD02], in the SDP architecture, pipelines are executed within a data-driven framework. The data dependencies and their execution within this framework is graph driven and so-called ‘logical graph-templates’ provide a basic description of this functionality. A capability may also be constructed internally by SDP from one or more logical graph templates, each describing an aspect of the capability, which are then formed into a single graph execution framework. Figure 7 illustrates the structure of a capability.

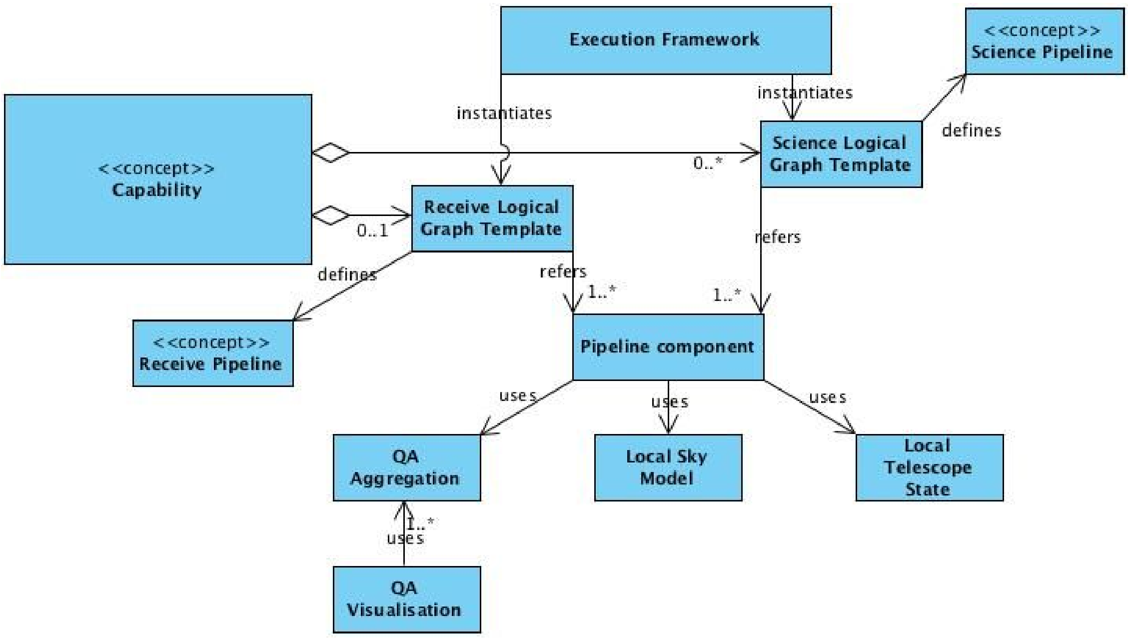


Figure 7: *A capability viewed as a package in terms of products: the capability includes a data-driven pipeline execution framework, but also provides access to services in particular the capability also includes access to an instance of the Local Sky Model and Local Telescope State. The baseline SDP architecture uses a graph-based data-driven architecture for executing processing and therefore a capability must also include graph artefacts defining the processing in detail including specific parameters required for a specific scheduled instance.*

Capabilities are the minimal-sized tasks which can be scheduled by TM, as stated above. The relationship between capabilities, which are exposed to TM by SDP, and the scheduling block used by TM needs further definition and clarity. As currently envisaged, the SDP capabilities are not formally exposed to observers, but rather TM translates project requirements into a series of scheduled SDP capabilities. Further system-level consideration needs to be undertaken to decide if this overall architectural approach is optimal.

8.4. Execution Framework

The execution framework implements the Execute Process function and is a critical element of the system from the point of view of delivering scalability. As discussed in the system overview, the SDP architecture is designed to exploit the inherent data parallelism of the problem. The baseline SDP architecture for the execution framework therefore employs a data-driven approach.

The adoption of this approach is motivated by:

1. the need to exploit the intrinsic data parallelism in the SDP challenge without strong coupling to the hardware architecture and sizing;
2. the requirement to achieve suitable efficiency and in particular to load-balance the system;
3. the desire to separate architecturally and in an explicit way the domain-specific functionality from the aspects of the system designed to achieve scalability and flexibility in a large-compute environment.

In the SDP context, the data-driven architecture has the following key concepts:

- The processing is divided into components which explicitly specify all of their required inputs and outputs and their execution is driven by the availability of data.
- The architecture aims to minimise data movement through the system by exploiting the explicit specification of data dependencies – required data and data movements are fully specified by the definition of a given pipeline.
- The SDP problem naturally allows the control aspects of the data-driven architecture to be structured in a hierarchical way which greatly simplifies the route to achieving scalability – data access by frequency channel provides a primary index on which coarse partitioning of the work is possible.
- The architecture allows several avenues for implementing fault tolerance explicitly, for example:
    - restarting processing based on data dependencies;
    - data policies with regard to the loss of input or intermediate data;
    - reallocation of work across the hardware.

This description is in essence a task-based approach to achieving scalability. Critical to the effectiveness of this task-based approach is the judicious use of global and localised synchronisation points. For example, any global synchronisation point, i.e. forcing a rendez-vous between all participating computational entities, would require extremely accurate load balancing, otherwise computation would proceed at the speed of the slowest task. For best effect, of course, all tasks should be independent. For the SDP pipelines some global synchronisation is required. In order to enable effective load balancing between synchronisation points, the number of tasks to be executed must be large enough to allow for it.

To realise this architecture, a language is required to represent the logical structure: the language we adopt is that of the "directed graph" which has become a much-used language for expressing such problems.

8.5. Control Concept

The SDP control concept is a combination of centrally orchestrated control by the Master Controller and devolved control by the Execution Framework. For many of the components the control from the Master Controller will be very lightweight and will simply ensure component availability, base configuration and health monitoring.

The boundaries of this control concept are established by the definition of an SDP capability. All singleton functions and associated processes not within capabilities are managed directly by the Master Controller and the Master Controller has the responsibility for triggering the setup and execution of a capability. Within an executing capability, the execution framework manages processes which are instantiated by the requirements of the graph-driven execution framework. Within an executing capability, processes have access to services provided by components under the direct control of the Master Controller (e.g. the LTS, LSM and metric aggregation).

In practical terms, a control plane will exist that handles all Control SDP communication including control, metadata exchange, health monitoring, logging and alarm events. This control plane is not shown explicitly in the functional diagrams, apart from links indicating which functions interact directly with the Local Sky Model and Local Telescope Model.

The SDP states should cover the following operating conditions:

- Complete shutdown of the entire SDP system. In normal operation transition out of this state will require an external intervention to apply power to the essential components required to start the Master Controller;
- A state in which the Master Controller is fully powered and operational and sufficient platform elements are started to allow the master controller to start further processes and establish full SDP functionality;
- A state in which all SDP capabilities are available for configuration and scheduling by TM. In general it is not required that every SDP component is available for the overall system to be in the available state, as the system will be built with some excess capacity to handle failures;
- The SDP may be functional when not all capabilities may be scheduled, this may for example be due to unavailability of the required resources for the capability. In this state TM can schedule some capabilities;
- A state in which no capabilities are available to be scheduled but the components representing singleton functions are executing;
- An Error state: a special state used purely to handle the case in which the master controller fails to start or suffers catastrophic failure;
- Fall-over of processing nodes or other platform elements which affect the operation of capabilities are managed as states of that capability: if it is possible to reschedule

the physical graph deployment and restructure the data deployment (or continue with lost data) then the capability may recover and continue.

8.6. Science Product Lifecycle and Delivery

The science product lifecycle encompasses Science Products from their creation within the processing system managed by the Staging function, through curation in Preservation and Persistence, through to delivery to Regional Centres.

The delivery services running at an SDP site enable searching for Science Products and for transferring these products to designated geographically distributed computing centres, referred to as Regional Centres. Product transfers from an SDP site are controlled from that site so that network links can be used most effectively to meet the observatory's objectives.

Services provided enable the searching for Science Products created by the observatory at both of the SDP sites. A location service enables replicas of Science Products to be found at SRCs which could remove the need to transfer the same products out of the observatory multiple times. IVOA services will be supported at the SDP sites, though only those relating to searching for Science Products will be exposed to external users. The intention is that the full set of IVOA services be supported at Regional Centres, but these are not within the scope of the SDP.

The SDP will deliver its Science Products in standard formats to external users and Regional Centres. Data are preserved within the SDP long-term preservation system in an internal format (data drop). A single Science data product may, and in general will, be preserved as multiple data drops with each data drop being a subset of the data product. Conversion to standard formats is done by the Prepare Science Products function before delivery.

8.7. Data Preservation

The data products generated by the various pipelines are in a common internal format – "data drops" (see [RD02]). Each data drop has a unique global id. To achieve data locality within the data-driven architecture it may be necessary to have multiple copies of the same data drop, where each copy has its own id. At the end of the data processing, the data drops are prepared for long-term preservation by the Staging function. Within the Staging function, data from QA are aggregated and merged into the Science Products so that future queries may be made against the quantitative data quality metrics. Other metadata are similarly aggregated (e.g. metadata that applies to the observation, metadata that applies to the scheduling block and information from the Local Telescope Model).

An index and location for all extant data drops is maintained by the Data Lifecycle Management function which links data drops to observational metadata and hence to Science Products.

The Staging function also needs data associated with services which is not in the internal Drop (e.g. LTS and LSM). Staging will query these services to extract information which is also used to construct the index of science data products and to produce metadata data objects which may be persisted as data products.

Data are migrated into Preservation and are managed via a set of policies. They allow for removing data at a future stage from Preservation, for example deleting preserved data based on QA metrics in order to manage the overall preserved data volume. These policies also manage the movement of data between storage tiers. Data can also be migrated from Preservation via the Staging function into the processing system.

8.8. Architecture of data discovery and retrieval

All new connections to the delivery platform cause authentication credentials to be passed to the AAAI module. This uses authoritative sources (e.g., trusted IdPs or CAs) to verify the identity of the requester. For a verified user the AAAI module returns to a session started to service this request a list of groups this user belongs to. This list is compared to ACLs associated with all services and data to determine if a request to access them should be satisfied.

A view of information describing all Science Products is available in the delivery platform. This information can be obtained by queries to the Preservation function or by using a replica of the catalogue containing this information. The queries could be performed through a web UI or using IVOA-compliant calls. Requests can be made for lists of Science Products that relate to particular metadata. This metadata might refer to a particular observation, survey project or some other characteristic of the data. As users may not be authorised to see all Science Products, information is only returned about the set of Science Products which the user is authorised to see.

A global location service is used to locate instances of a particular Science Product in the case where copies exist other than the one at the SDP site where the product was created. This contains a URI for each instance of the Product that is discoverable. In the case that a Product needs to be transferred from an SDP site, a request for the Science Product is made to the Preservation function using the local URI. This causes the drop (or drops) to be pushed to Prepare and Deliver, which converts the drop(s) into the requested Science Product and format and informs the transfer service that this is ready. The transfer service then transfers this Product to the desired destination once network capacity is available and the transfer can be made efficiently. Once the transfer is complete, the local copy of the Science Product created in a format for delivery can be marked for garbage collection. The receiving site should register the presence of the product so that this information is available to the global location service.

In the implementation there will need to be a way of implementing transfer policies and controlling the work done by the Prepare and Deliver function. Transfer policies may for example favour requests from particular projects or provide fairness between projects. A priority-based scheduling tool can implement this policy by ordering and controlling the flow of requests for data sent to the Preservation function.

8.9. Observatory Support Tools

These represent a set of tools that are needed by astronomers with direct access to the SDP sites to perform data analysis in support of the observatory. These tools will be essential to support the commissioning and operation of the telescopes.

The observatory support tools will include a more complete set of IVOA services than is available from the Deliver Data system and visualisation tools to enable visual analysis of large data cubes created by the SDP. These functions will be called via a Web UI to provide a standardised interface. A data download function will be included so that Science Products could be downloaded to an appropriate device if needed (e.g. to the desktop of an observatory staff member during commissioning). In general, Science Products will be too large for this to be done, so a remote visualisation service will be provided to support the visual analysis of Science Products directly from the Staging function. The additional IVOA functions supported here provide access to data extraction using the Simple Image Access (SIA) protocol and they provide the ability to link related data using the DataLink protocol.

Other observatory support tools are defined in [RD09].

8.10. Local models

The pipelines, which are processing the data observed in a scheduling block, make use of some local data models, i.e., data not shared with other capabilities. There are two such models:

1. The Local Sky Model (LSM) contains the description of all celestial sources used during the data processing. It contains information such as name, sky direction, flux for all Stokes, spectral index, etc. Point sources, as well as extended sources can be described. Possibly sources can be grouped to describe an extended source as a collection of other sources. Shapelets should be supported if needed (TBD). The LSM can hold images, possibly with Fourier term images for spectral variation. Initially, the LSM will be filled with the relevant sources from the Global Sky Model (GSM): the sources in the FOV and bright sources outside the FOV affecting the visibilities. During data processing, newly found sources will be added to the LSM and information on existing sources may be updated. At the end of the data processing the final LSM will be merged into the GSM, possibly depending on the image quality.
2. The Local Telescope State (LTS) contains all other parameters needed for the data processing. Initially it contains a subset of the overall Telescope Model data which consists more or less of the time-invariant metadata contained in the subtables (such as ANTENNA) of the MeasurementSet. During data processing the LTS is extended with the calibration parameter solutions (usually per time/frequency window,

antenna/polarisation and possibly direction). Note that demixing (i.e., removal of strong sources outside the FOV) also results in calibration parameters. Flagging statistics could also be part of the LTS. A direction/time/frequency-dependent ionospheric model can also be part of the LTS. Applications and monitoring software can query the LTM for various information.
The LTS does not contain all of the calibration gain solutions. In particular, the direction-dependent gain solutions and gain solutions with a high cadence are managed as an explicit data stream and are not managed via the LTS.

The number of calibration parameters can be very high (e.g. per second, channel, antenna, direction on the sky, polarisation). The architecture supports the implementation of large-volume calibration data as an explicit intermediate data product managed in the same way as other high-volume data such as visibility data. Similar considerations apply to, for example, Clean Components generated during the imaging iterative cycles.

Both the LSM and LTS are populated by, and may update the models maintained by, the Telescope Manager for the Global Sky Model and Telescope State, respectively.

The SDP architecture implements the LSM and LTS as services which may be queried and updated by relevant components which require access. The implementation of these functions as services is the SDP baseline; however, there is an issue of whether such an implementation can scale to the SDP size. An alternate approach which will be investigated as a 'formal issue' will be to consider an architecture change in which these data (LTS and LSM) should be treated as explicit data within the data-driven approach.

8.11. Quality Assessment

QA metrics are generated by many of the SDP functions. The QA system acts as a service function to aggregate and summarise these metrics for presentation via LMC to TM and to generate data which is further aggregated into data products via Staging.

There is potential to generate a large volume of QA data: the implementation of the QA service function should support hierarchical aggregation and filtering. Such a hierarchy must be structured so as to scale to the SKA1 system size without degradation of performance. QA metrics will be aggregated. Metrics must be generated and aggregated with a cadence throughout the processing so that appropriate intervention is possible.

QA metrics will include information about data loss/removal (e.g. RFI flagging, computing hardware failure resulting in loss of data, curtailment of straggling processes) and data quality (e.g. poor signal-to-noise ratio). The aggregated QA metrics will then be visualised by the QA visualisation function. This allows operators to see quality metrics in near real-time and abort the observation if the situation warrants it (or reschedule parts of a specific observing programme with particular signal-to-noise requirements, for example).

The Master Controller will also receive data for monitoring the health of the system; this may be managed separately from the Science QA reporting.

At the end of an observation, quality information is written out as a data product, so astronomers can see the quality information for the data set they are interested in.

8.12. Receivers

The receiver functions are responsible for receiving all data from the CSP element and also, in the case of visibility data, the merging of "fast" telescope metadata. The four receiving functions are distinct as they all have different requirements and the ICDs with CSP specify qualitatively different protocols for each of them [AD04, AD05]. The receiving functions all receive data from the Switch function – this reflects the possibility that all of the four data streams can be re-routed to different hardware and that fully dedicated hardware is not required for any of the functions.

8.13. Fast Imaging and Real-time Calibration

8.13.1. High-level assumptions and requirements

The assumption here is that both Fast Imaging as well as the Real-time Calibration will have to be executed all the time when the telescope is operational and observing. This leads to two main requirements for these two pipelines:

1. Both functions have to run continuously whenever data is collected.
2. Data coming from the CSP will need to be multiplexed into these functions as well as going to the main Imaging function.

The SDP baseline architecture is to manage fast imaging and real-time calibration using the data-driven approach including full Data Lifecycle Management. At present, system requirements on latency, cadence and models against which calibration should be performed are not well advanced. This represents therefore a significant risk and issue for the SDP which will need to be examined, including the possible need to manage the execution of these elements outside of the rest of the execution framework.

The output of the Real-time Calibration are telescope calibration parameters which are written into the Local Telescope State and hence managed by LMC and transmitted to TM. The products from Fast imaging are transmitted to Staging and are then managed by the Data Lifecycle Management for preservation and dissemination. Alerts from Fast Imaging are transmitted to TM via LMC.

Both the Real-time Calibration and the Fast Imager will make use of the Local Sky Model Management function of the LMC, which also extracts and buffers the Local Sky Model from the Global Sky Model maintained by TM. Both of these functions use as input information from the Local Telescope Model.

8.13.2. Real-time Calibration

Real-time Calibration is a function which calculates gain solutions and other varying calibration parameters that are recorded in the Local Telescope State and transmitted by Control SDP to TM for use by other SKA elements. This function is critical for the central beam former mode of CSP and fast imaging in order to determine sufficiently accurate gain solutions and delays.

Without this function the CSP will not be able to work properly and thus this function is critical and will require some careful availability and reliability planning. Likely calibration parameters to be fed back include complex gains, change of complex gains and dispersive and non-dispersive delays (the former are caused by the troposphere, the latter by the ionosphere). The algorithms required to derive these parameters are quite well understood, but some analysis is required for the scale of the SKA system, including the long baselines and the ionospheric corrections, in particular for SKA1-LOW.

The calculations have to be made to a latency limit as the corrections will be applied in real time within the CSP and the effects being calibrated are expected to vary relatively quickly. The calibration is made against sources in the LSM derived from the Global Sky Model.

It is assumed that the SKA will calibrate the telescope(s), not the data. This means that calibration solutions may be re-used across observations, if possible. It also means that standard calibration scans are owned by the observatory, accounted for as observatory time and can be used and distributed to multiple observing program PIs/teams.

8.13.3. Fast Imaging and Transient Detection

Fast Imaging is required for supported science programmes which need either or both of:

- Fast time-resolved imaging, e.g. to detect transients on timescales comparable to the resolution of the imaging time scale and where:
    - Timescales as short as a correlator dump time could be required;
    - A sub-selection or an average of the data for the highest time-resolution could be required.
- Imaging with a short latency requirement in order to trigger an alert.

This mode may be commensal with other observing modes. One such use case is the detection of so-called slow transients.

At observatory level, it is required that the light curves of such transients be measured. This is supported at low-signal-to-noise directly by fast imaging, but will also be an analysis mode on the full visibility data, however with a long latency to the production of the product.

The Fast Imaging function is designed to form continuum images for every correlator integration time after the relevant portion of the GSM has been subtracted and then run a detection of transient sources. There is no deconvolution of the imaged data. The data products shall include a catalogue of found sources, a sensitivity image and a representative PSF image. The Fast Imaging function will also be able to generate alerts.

Additional assumptions about the required outputs of the Fast Imaging function are:

- Slow transient events to be issued to TM
- The detect function may also be supplied with additional catalogue information about known transients or targets within the field against which the analysis can be performed.
- QA metrics
- Images in this pipeline will not require a full clean, just an initial removal of the relevant portion of the GSM
- Images produced by this pipeline will only image an area out to the first null or less
- Images will not be permanently stored, only the catalogues.

Detailed requirements for the Fast Imaging are to be defined: in particular the approach and recording of variability and different cadence measurements of variability.

9. Discussion of Allocations

The L2 requirement allocation to functions and products is discussed in the detailed design documents [RD02, RD03, RD05, RD06, RD07, RD08, RD09]. We show our compliance to the L1 requirements in our requirements analysis [AD01] and monitor this in our compliance matrix [AD03]. Non-conformance to these requirements is discussed in RD01.

The functional architecture described above is translated into a structured tree of products. The product tree is structured so as to provide an as clean as possible allocation of functions to the products. The top-level product tree follows the functional decomposition at L1 with the top-level products being:

- C.1 Data Processor
- C.2 Delivery System
- C.3 SDP Local Monitor and Control
- C.4 Preservation system
- C.5 Observatory support tools

Each of these products is then decomposed into a software and platform product as shown in Figure 8.

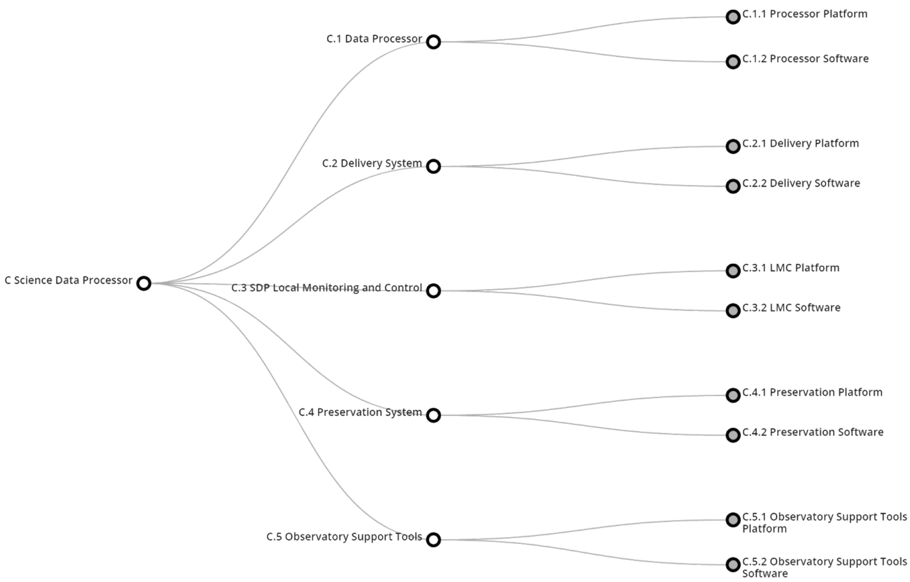
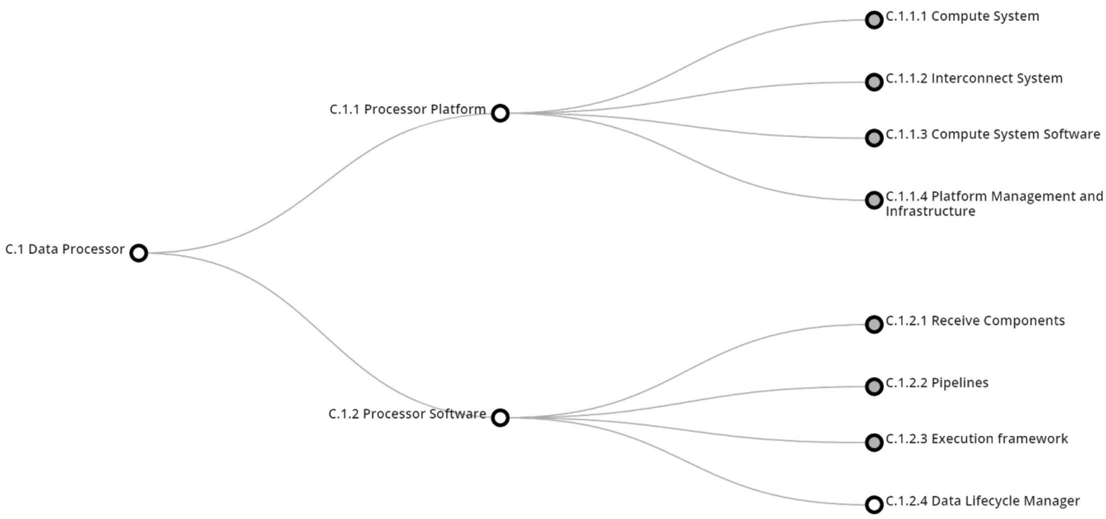


Figure 8: *The SDP element Product Tree decomposed into the main products which follow the same structure as the L1 functions and thence to a platform and software product for each of the five SDP main products.*

The SDP element Product Tree is further decomposed as shown in Figure 9 on the following pages.



.

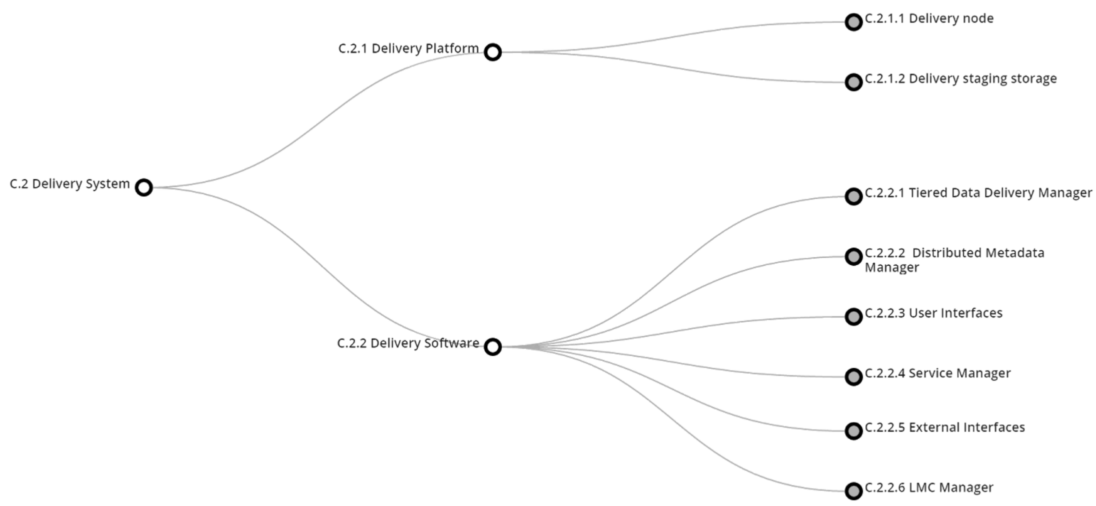


Figure 9a: *The SDP element Product Tree is further decomposed. Shown here is the decomposition for the Data Processor and Delivery System.*

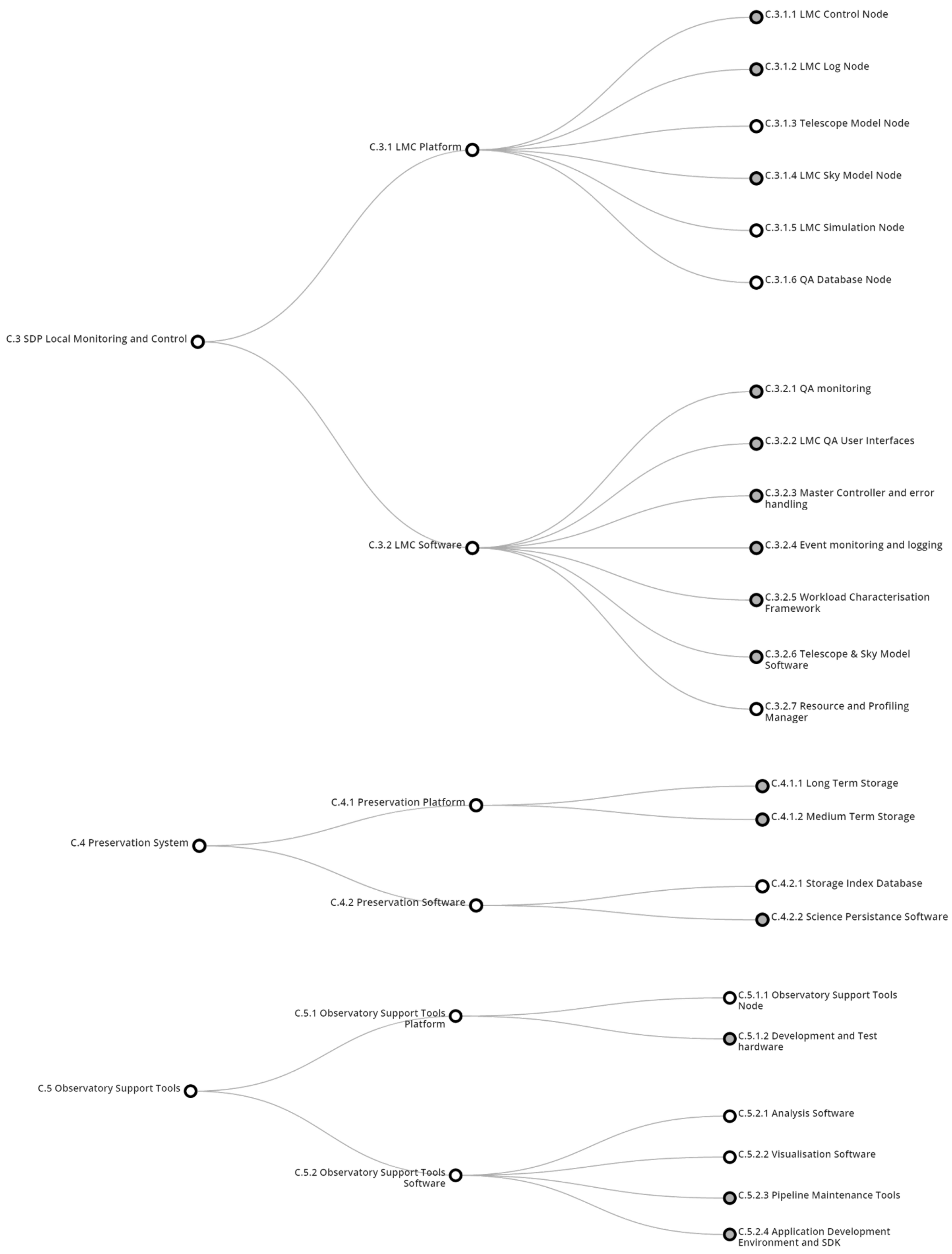


Figure 9b: *The SDP element Product Tree is further decomposed. Shown here is the decomposition for the Local Monitor and Control, Preservation System and Observatory Support Tools.*

The functional allocation to these products is shown in Figure 10.

Functions \ Products	C Science Data Processor	**C.1 Data Processor**	C.1.1 Processor Platform	C.1.2 Processor Software	**C.2 Delivery System**	C.2.1 Delivery Platform	C.2.2 Delivery Software	**C.3 SDP Local Monitoring and Control**	C.3.1 LMC Platform	C.3.2 LMC Software	**C.4 Preservation System**	C.4.1 Preservation Platform	C.4.2 Preservation Software	**C.5 Observatory Support Tools**	C.5.1 Observatory Support Tools Platform	C.5.2 Observatory Support Tools Software
F SDP Functions	X															
F.1 Process Data		X														
F.1.1 Switching			X													
F.1.2 Receive Visibilities			X	X												
F.1.3 Pre-process Fast			X	X												
F.1.4 Calibrate Real-time			X	X												
F.1.5 Image Fast			X	X												
F.1.6 Detect Imaging Transient Candidates			X	X												
F.1.7 Buffer Data			X	X												
F.1.8 Pre-process data			X	X												
F.1.9 Calibrate and Image			X	X												
F.1.10 Receive Pulsar Timing Profiles			X	X												
F.1.11 Receive Pulsar Candidates			X	X												
F.1.12 Receive Transient Buffer			X	X												
F.1.13 Process Pulsar Timing			X	X												
F.1.14 Process Pulsar Candidates			X	X												
F.1.15 Process Transient Buffer			X	X												
F.1.16 Manage Data Lifecycle			X	X												
F.1.17 Execute Processing				X												
F.2 Preserve Data											X					
F.2.1 Persist Science Products												X				
F.2.2 Index Science Products													X			
F.2.3 Stage data products												X	X			
F.2.4 Backup Science Products												X				
F.3 Deliver Data					X											
F.3.1 AAAI							X									
F.3.2 Prepare Science Products						X	X									
F.3.3 Query and Request Data						X	X									
F.3.4 Deliver Data to User						X	X									
F.4 Control SDP								X								
F.4.1 Visualise Quality Assessment Metrics									X	X						
F.4.2 Manage Local Telescope State									X	X						
F.4.3 Aggregate QA Metrics									X	X						
F.4.4 Manage Local Sky Model									X	X						
F.4.5 Produce Fast Telescope State									X	X						
F.4.6 Master Control									X	X						
F.4.7 Manage Science Events									X	X						
F.5 Support Observatory Commissioning & Operations														X		
F.5.1 Allow Observatory Access to Query																X
F.5.2 Allow Observatory Access to Request and Deliver																X
F.5.3 Visualise Data															X	X
F.5.4 Analyse Data															X	X
F.5.5 Create and Maintain Pipelines															X	X

Figure 10: *Allocation of functions to products.*

10. Interfaces

Interfaces are defined in terms of SKA system decomposition levels. Interfaces between the SDP element and external entities are defined to be at interface-level 3. Internal SDP interfaces between our L1 functions and associated physical interfaces are therefore at interface-level 4.

10.1. Interface-level 3

Interface-level 3 interfaces are defined in the associated ICDs. For completeness we summarise these interfaces and provide references to the interface IDs. The detailed ICDs are baselined.

10.1.1. C.1 Data Processor-CSP [AD04, AD05]

Interfacing products	ID	Name	Description	Item(s) transferred by the interface
CSP to C.1 Data Processor	I.S1L.SDP_CSP.001 I.S1M.SDP_CSP.001	SKA1-Low SDP_CSP Visibility Data Interface SKA1-Mid SDP_CSP Visibility Data Interface		Visibilities (CSP) Visibility Metadata (CSP)
	I.S1L.SDP_CSP.002 I.S1M.SDP_CSP.002	SKA1-Low SDP_CSP Pulsar Search Data Interface SKA1-Mid SDP_CSP Pulsar Search Data Interface		Pulsar Candidates (CSP) Single Pulse Candidates (CSP)
	I.S1L.SDP_CSP.003 I.S1M.SDP_CSP.003	SKA1-Low SDP_CSP Pulsar Timing Data Interface SKA1-Mid SDP_CSP Pulsar Timing Data Interface		Pulsar integrated pulse profiles (CSP)
				Transient Buffer Data (CSP)

10.1.2. C.3 Local Monitor and Control - TM [AD06, AD07]

Interfacing products	ID	Name	Description	Item(s) transferred by the interface
C.3 Local Monitor and Control to TM	I.S1L.SDP_TM.003 .S1M.SDP_TM.003	SKA1-Low SDP_TM Science Event Interface SKA1-Mid SDP_TM Science Event Interface		Transient Detection Alerts (TM)
	I.S1L.SDP_TM.005 I.S1M.SDP_TM.005	SKA1-Low SDP_TM Sky Model Interface SKA1- Mid SDP_TM Sky Model Interface		Updates to Global Sky Model (TM)
	I.S1L.SDP_TM.002 I.S1M.SDP_TM.002	SKA1-Low SDP_TM Telescope State Information Interface SKA1-Mid SDP_TM Telescope State Information Interface		Telescope State Updates (TM)
	I.S1L.SDP_TM.004 I.S1M.SDP_TM.004	SKA1-Low SDP_TM QA Interface SKA1- Mid SDP_TM QA Interface		QA Visualisation (TM)
	I.S1L.SDP_TM.001 I.S1M.SDP_TM.001	SKA1-Low SDP_TM Control and Monitoring Interface SKA1-Mid SDP_TM Control and Monitoring Interface		Capability Availability (TM) SDP Configuration (TM)
TM to C.3 Local Monitor and Control	I.S1L.SDP_TM.001 I.S1M.SDP_TM.001	SKA1-Low SDP_TM Control and Monitoring Interface SKA1-Mid SDP_TM Control and Monitoring Interface		SDP Control (TM) SDP Configuration (TM) Capability Request (TM)
	I.S1L.SDP_TM.004 I.S1M.SDP_TM.004	SKA1-Low SDP_TM QA Interface SKA1- Mid SDP_TM QA Interface		QA User Control (TM)
	I.S1L.SDP_TM.005 I.S1M.SDP_TM.005	SKA1- Low SDP_TM Sky Model Interface SKA1- Mid SDP_TM Sky Model Interface		Local Sky Model (TM)
	I.S1L.SDP_TM.002 I.S1M.SDP_TM.002	SKA1-Low SDP_TM Telescope State Information Interface SKA1-Mid SDP_TM Telescope State Information Interface		Telescope State - Unstructured (TM)
	I.S1L.SDP_TM.003 I.S1M.SDP_TM.003	“SKA1-Low SDP_TM Science Event Interface” “SKA1-Mid SDP_TM Science Event Interface”		Transient Detection Thresholds (TM)

10.1.4. C.2 Delivery System - External (currently not an official interface with an ICD)

Interfacing products	ID	Name	Description	Item(s) transferred by the interface
C.2 Delivery System to External		SDP to Regional Centre Interface	This is the bulk data transport interface. It is assumed that this will be connected to a network that has been optimised for high bandwidth communication with a small number of Tier 1 sites. Also requests for the locations of specific Science Products will be sent on this interface.	Science Product Catalogue (regional centre) Science Data Products (regional centre) Requests for Science Product locations (i.e., Tier 1 sites holding specific Science Products.
		AAAI interface	This is used to connect to trusted identity providers to request user authentication. It will also be used to connect to an authorisation service to determine user group membership and access privileges.	IdP Authentication request (IDP) and group membership request
External to C.2 Delivery System		Web portal interface	Web GUI interfaces for performing online search and research product requests.	Science Product Query (end-user)
		IVOA RESTful interface	RESTful HTTP based API for communicating directly with IVOA services.	RESTful query requests and HTML based query responses
		SDP to Regional Centre Interface	Responses to Science Product location requests.	Science Product locations (regional centre)
		AAAI interface	Information returned from trusted identity providers and from an authorisation service.	IdP authentication response (IDP) and group membership response

10.2. Interface-level 4

These functional interfaces were identified in the L1 functional decomposition. A description of the physical L4 interfaces is given in [RD10] and a summary in Figure 11.

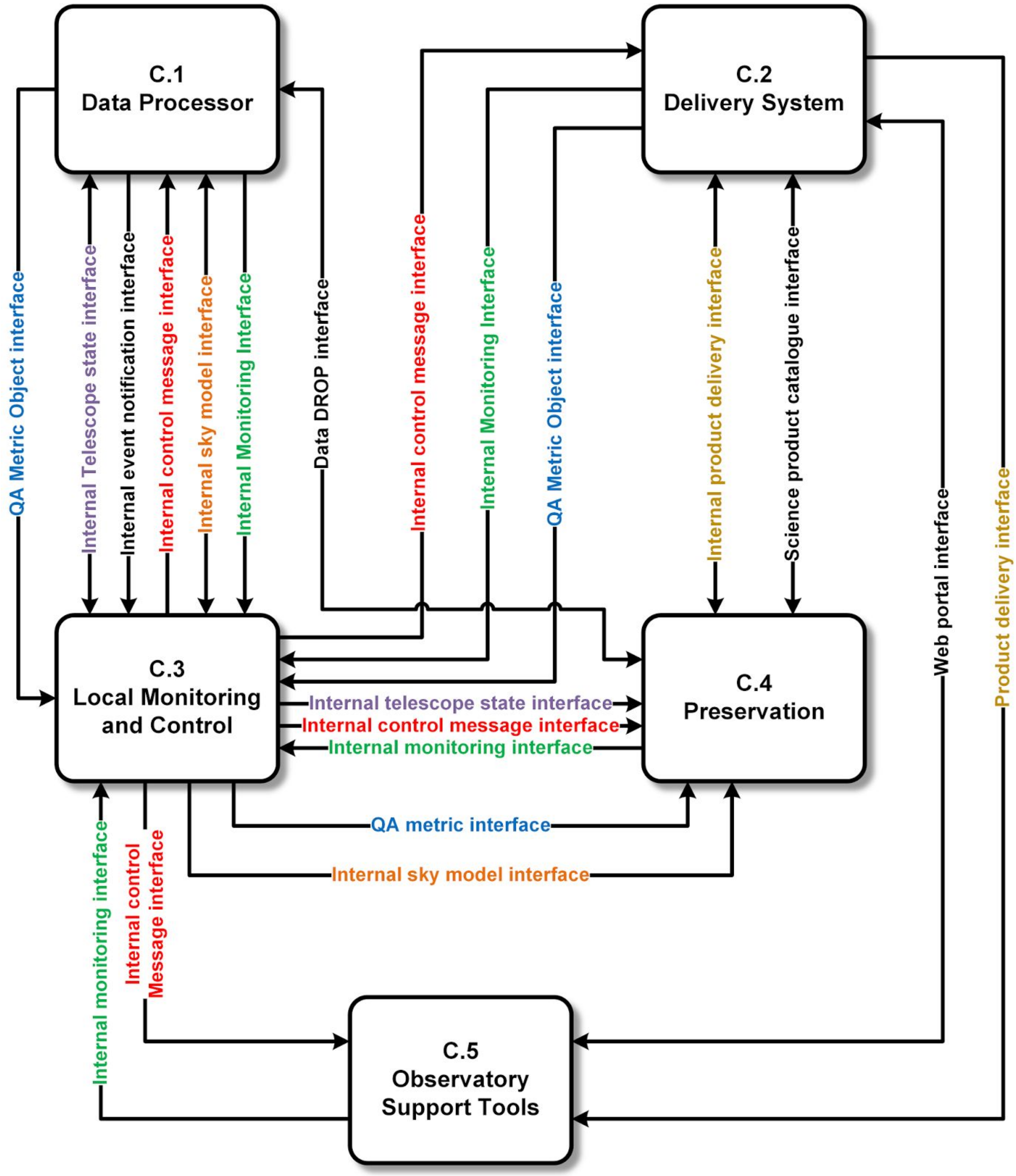


Figure 11: *Physical interface at interface-level 4 are between SDP products at the SDP L1 level. The interfaces are identified in this figure and the nature of the items transferred by the interface is shown in summary form. The associated tables give more information on these physical interfaces.*

11. Platform Architecture Software Deployment and Software Dependency

The product structure described above associates a platform and software product with each of the top-level SDP functions and associated products. This decomposition is intended to provide, as far as possible, a straightforward deployment of software to platform within each high-level product. There is, however, some cross-deployment which is an inevitable consequence of requiring cohesion across the system. The deployment is shown in Figure 12.

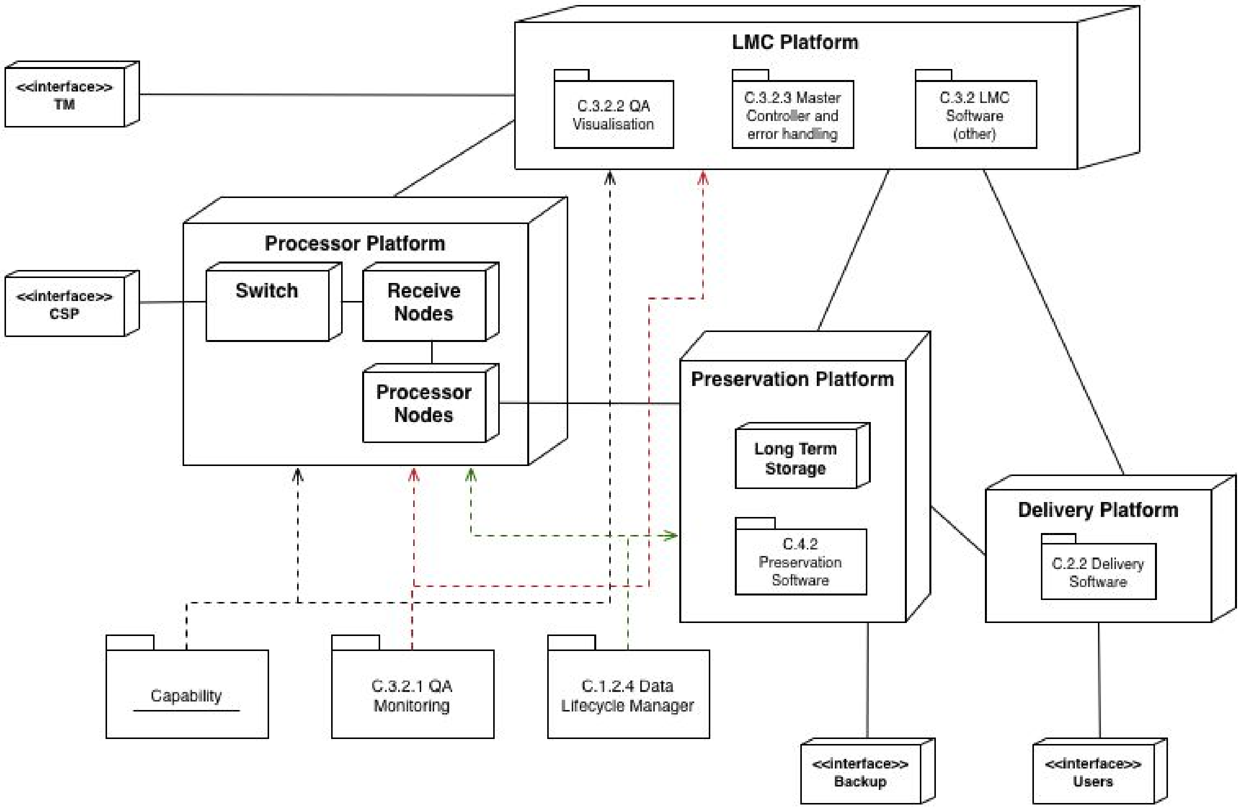


Figure 12: *Software deployment for the SDP. The SDP consists of four platform products. Processor Platform is here shown further decomposed into switch, receive and processor nodes (these may share a common implementation). Deliver and Preservation software are deployed completely to their respective platform products. All of the LMC software product is deployed to the LMC platform, however some elements are additionally deployed to the processor platform as services such as the LTS and LSM that are required during the execution of a capability (and indeed are defined as part of a capability). Therefore, those functions associated with a capability are shown deployed across both Processor and LMC platforms as are the QA monitoring components. Similarly, the Data Lifecycle Manager must be deployed across three platforms to provide for an integrated approach to data life cycle management.*

The functional decomposition and associated structure of the product tree are designed to reduce interdependencies. The most complicated dependencies exist within the software components of a capability. As an illustration of the way the current product tree provides a structured, layered software architecture, we show an illustrative dependency diagram in Figure 13.

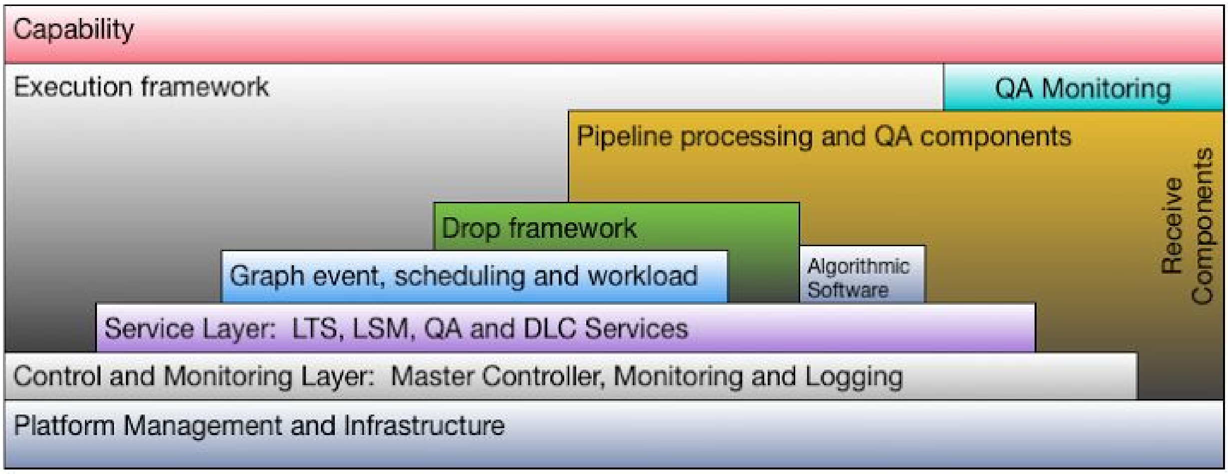


Figure 13: *Software stack for the processing of pipelines via capabilities. The diagram shows the layers of the software and interdependencies in an illustrative stack-diagram. The Service Layer refers to services supporting the Local Telescope State (LTS), Local Sky Model (LSM), Data Lifecycle Management (DLC) and Quality Assurance (QA).*

12. Appendix A

12.1. Description of the L2 Functional Elements

12.1.1. Switching

Input:

- *correlated visibilities from CSP using UDP/IP over 100 Gigabit Ethernet*
- *pulsar integrated pulse profiles from CSP using TCP/IP over 10 Gigabit Ethernet*
- *Pulsar candidates from CSP using TCP/IP over 10 Gigabit Ethernet*
- *Single pulse candidates from CSP using TCP/IP over 10 Gigabit Ethernet*
- *Transient buffer data from CSP using UDP/IP over 100 Gigabit Ethernet*

Output:

- *correlated visibilities*
- *pulsar integrated pulse profiles*
- *pulsar candidates, single pulse candidates*
- *transient buffer data*

The switch is the physical interface between CSP and SDP. It receives data from the CSP, possibly some hundreds of kilometres away, and distributes it to the hardware resources within SDP. The CSP sends data to IP addresses within the Process Data Platform, the switching function is responsible for directing the packets to the appropriate hardware resources.

The Switch is part of the L1 Process Data function.

12.1.2. Receive Visibilities

Input:

- *visibility data from the Switch (as Jumbo packets)*
- *fast telescope state*

Output:

- *visibility data + metadata*

The Receive Visibilities function receives visibility data as jumbo SPEAD packets, as well as the fast telescope state. It then reconstructs SPEAD heaps with appropriate metadata. These are then packaged in the internal SDP format and made available for further processing to the Fast Pre-processing function and to Buffering.

The Receive Visibilities function is part of the L1 Process Data function.

12.1.3. Pre-Process Fast

Input:

- *visibility data and metadata*
- *Local Sky Model (LSM) data*
- *Local Telescope State (LTS)*

Output:

- *visibility data and metadata*
- *QA metrics*

Fast pre-processing involves steps that must be done on all data before they can be used in the fast imaging and real-time calibration pipelines. The fast pre-processing has the ability to perform RFI flagging and to process the LSM to remove strong sources outside the field of view. This is a real-time function with strict latency requirements. Quality metrics are reported back to the QA metric aggregator (e.g. flagged/excised channels).
Functions performed within the Pre-Process Fast function include:

- Fast data flagging
- Applying the current gain solution
- Removing out-of-field strong sources using the Local Sky Model after possibly solving for local calibration solutions towards them.

Note that in-field sources are not removed as these are needed for the real-time calibration. The Local Telescop State is required for metadata and access to the current gain solutions.

The Fast Pre-processing function is part of the L1 Process Data function.

12.1.4. Calibrate Real Time

Input:

- *visibility data and metadata*
- *Local Sky Model (LSM) data*
- *Local Telescope State*

Output:

- *gain solutions*
- *Local Telescope State*
- *QA metrics*

The purpose of the real-time calibration is to solve for real-time calibration parameters. These calibration parameters are used to update the Local Telescope State hence they are transmitted to TM by the Control SDP. These calibration parameters will be employed elsewhere in the telescope, importantly in the central beam former. The function also produces quality metrics which are reported to the QA Metric Aggregator.
The Calibrate Real Time function is part of the L1 Process Data function.

12.1.5. Image Fast

Input:

- *visibility data and metadata,*
- *Local Sky Model (LSM) data*
- *Local Telescope State (including calibration solutions)*

Output:

- *image data and metadata*
- *QA metrics*

The purpose of the Image Fast function is to produce snapshot images with low latency and high time resolution. Functionality within Image Fast includes:

- Apply real-time calibration solutions
- Within-field LSM subtraction possibly after solving for the local calibration solution
- Visibility gridding and Fourier Transforms
- Quality metrics are reported back to the QA metric aggregator.

These images are used for source detection in order to identify radio sources that exhibit variability with respect to the LSM.

The Image Fast function is part of the L1 Process Data function.

12.1.6. Detect Imaging Transient Candidates

Input:

- *Image data and metadata*

Output:

- *Transient Candidates*
- *QA metrics*
- *Alerts*

The purpose of the candidate detection is to identify radio sources in image data and measure their position, flux density and other characteristics. These data, which are intrinsically time resolved, are passed to staging to form a Science Data Product – the transient source catalogue. This product may be queried to, for example, construct a light curve for a transient detection. Such a query is managed via standard methods implemented within the Deliver Data L1 function.

The Detect Imaging Transient Candidates function may trigger an Alert Event directly. This, together with astronomical data, is then passed to Manage Science Events to produce a Virtual Observatory Alert Event which is sent to TM to be communicated externally.

Quality metrics are reported back to the Aggregate QA Metrics function.
The Detect Candidates function is part of the L1 Process Data function.

12.1.7. Pre-Process Data

Input:

- *visibility data and metadata*
- *Local Sky Model (LSM) data*

- *Local Telescope State (LTS)*

Output:
- *visibility data and metadata*
- *QA metrics*
- *LTS calibration solutions (for strong source subtraction)*

Pre-processing involves steps that must be done on all data before they can be used for calibration and imaging. Functions include RFI flagging (possibly using a large time window) and removal of strong sources outside the Field of View (FoV), after having solved for the gains in their directions. Note that pre-processing does not have to wait until the entire observation has been done (unless the entire time window is needed for proper RFI flagging). Quality metrics are reported back to the QA metric aggregator (e.g. flagged/excised channels). The LSM is used for the source models of the strong sources outside the FoV and possibly bright sources in the target field (to get good gain solutions in the directions of the sources to be removed). The LTS is used for the observation's meta data and to get the beam model (probably needed for SKA1-LOW). Furthermore, the gain solutions of strong source removal can be sent to the LTS for later inspection.

The Pre-process Data function is part of the L1 Process Data function.

12.1.8. Buffer Data

Input:
- *visibility data and metadata*

Output:
- *visibility data and metadata*

The purpose of the Buffer Data function is to provide short-term storage to accumulate visibility data which are input for the Imaging Pipeline. The function will accumulate the data of a complete scheduling block which will subsequently be analysed by the Imaging Pipeline. The function must support double buffering.

The Buffer Data function is part of the L1 Process Data function.

12.1.9. Calibrate and Image

Input:
- *visibility data and metadata*
- *image data and metadata*
- *Local Sky Model (LSM) data*
- *Local Telescope State (LTS)*

Output:
- *image data and metadata*
- *Local Sky Model (LSM) data*
- *Local Telescope State*
- *QA metrics*

The purpose of the Calibrate and Image function is to take visibility data and produce calibrated image data. Multiple pipelines are supported and new or modified pipelines can easily be defined by using a combination of components. Pipeline behaviour is controlled by parameters defined in the processing definition which is part of the observational parameters provided by to the SDP by TM.

Defined pipelines include continuum, spectral line and specific pipelines for observations of the Epoch of Reionisation. Pipelines may support iteration, in particular the Imaging Pipeline incorporates an iterative loop with the calibration pipeline to update the LSM and LTS. Functions within the Imaging Pipeline include visibility gridding, FFT and deconvolution. Quality metrics are reported back to the QA metric aggregator.

The Calibrate and Image function is part of the L1 Process Data function.

12.1.10. Receive Transient Buffer

Input:

- *voltage data stream with at least 100 μs time resolution*

Output:

- *packaged voltage data stream with at least 100 μs time resolution*

Buffer sizes are 63.8 GB for SKA1-Low (for 150 MHz bandwidth, 425 seconds) and 6.24 GB for SKA1-Mid (300 MHz bandwidth, 20.8 seconds), but by looking at the latency of the entire system, the buffers may need to be 10 to 15 seconds longer than this.

The Receive Transient Buffer function receives data from the switch and passes it through to the processing function in the SDP internal format.

The Receive Transient Buffer function is part of the L1 Process Data function.

12.1.11. Process Transient Buffer

Input:

- *transient buffer data*
- *Local Telescope Model (LTM)*

Output:

- *processed transient buffer data*
- *QA Metrics*

When the Transient Pipeline detects a fast transient, it sends an alert to TM, which in turn alerts the beamformer to save the transient buffer. The buffer will arrive at the SDP only to be transmitted to Preserve Data and hence to be made available as a Science Data Product (TBC). The system should also be able to respond to external triggers, but the SDP process is exactly the same as for an internal trigger.

The output data are transmitted to Staging in the internal SDP data format. Processing of the data is possible, but the SDP baseline is that this is a pass-through function.

The Transient Buffer Processing function part of the L1 Process Data function.

12.1.12. Receive Pulsar Candidates

Input:

- *pulsar candidate data cubes in PSRFITS format, of size (number of frequency channels) x (number of pulse phase bins) x (number of sub-integrations)*
- *pulsar candidate metadata in HDF5 format; pulsar candidate parameter lists in plain text (1 per beam).*
- *fast transient candidate data cubes in PSRFITS format, of size (number of frequency channels) x (number of samples) x (number of polarisations)*
- *fast transient candidate metadata in HDF5 format; fast transient candidate parameter list in plain text.*

Output:

- *pulsar timing data for processing*

This receive function receives Pulsar and Single Pulse Candidate data from the switch and passes it through to the processing function in the internal SDP format.

The Receive Pulsar Candidates function is part of the L1 Process Data function.

12.1.13. Process Pulsar Candidates

Input:

- *pulsar timing and candidate data*
- *Local Telescope State*

Output:

- *processed pulsar timing and candidate data*
- *QA metrics*
- *alerts*

This function takes all the candidates identified by the processing done within CSP and filters them to sort out the most likely real sources from the noise. The analysis in CSP compares candidate sources from different beams first, meaning that the candidate lists from all the beams need to be compared. The remaining tasks, including heuristic value calculation and machine learning, are carried out on each pulsar candidate. After processing has finished, the metadata and parameter lists for all candidates received from the CSP, as well as all data cubes for all candidates that pass the coincidence test, are sent to Staging. Only those candidates that are identified as likely pulsars by the machine learning generate an alert, which may or may not be acted upon depending on the urgency of follow-up.

The output data are transmitted to Staging in the internal data format.

The Process Pulsar and Timing Candidate function is part of the L1 Process Data function.

12.1.14. Receive Pulsar Timing Profiles

Input:

- *data cubes in PSRFITS format from up to 16 pulsars, each of size (number of frequency channels) x (number of pulse phase bins) x (number of polarizations).*

Output:

- *timing data for processing*

Relevant metadata will be included in the PSRFITS files. The LTM will provide parameters for RFI mitigation, calibration and data cube averaging. The LSM will provide the pulse template, the currently best timing model and the arrival times from earlier observations to the pipeline (TBC).

The Receive Pulsar Timing Profiles function receives data from the Switch and passes it through to the associated processing function.

This function is part of the L1 Process Data function.

12.1.15. Process Pulsar Timing

Input:

- *timing data*
- *Local Telescope State*

Output:

- *processed timing data,*
- *QA metrics*

The Process Pulsar Timing function covers the second part of the complete SKA pulsar timing analysis: the first part occurs within CSP and provides "folded" pulsar data files to the SDP, one file per sub-integration and pulsar. The SDP timing function will add the sub-integrations of each pulsar together, calibrate the data, generate a number of data products that are different representations of the folded data, generate times-of-arrival and finally update the current timing model using the new data. The determination of the arrival time of the pulse is essential as part of the quality assurance checks but also potentially for identifying interesting scientific events. While this pipeline will run in near real time, it is likely that further processing will be required and therefore the delivered science data products must accommodate this possibility. The output consist of cleaned and calibrated PSRFITS-format data cubes in various averaged forms, the times of arrival (plain text), the timing residuals (plain text) and the updated timing model (plain text).

The output data are transmitted to Staging in the internal data format.

This function is part of the L1 Process Data function.

12.1.16. Stage Data Products

Input:

- *data products in internal format from all pipelines*
- *QA aggregated data*
- *Local Telescope State*
- *Local Sky Model*

Output:

- *Science Products in internal format*

The Stage Data Products function packages LTS, QA and LSM data in the SDP internal format and sends them to Index Science Products and Persist Science Products for association with Science Products. It also reorganises Science Data Products into forms suitable for long-term preservation. This function also acts as interface between the processing functions and Preserve Data.

This function is part of the L1 Preserve Data function.

12.1.17. Index Science Products

Input:

- *from Stage Data: Science Products*
- *from Stage Data: Science Product metadata*
- *from Query, and Request Data: Science Catalogue Queries*
- *from Query and Request Data: Requests for Science Products to be moved to Prepare Products including Delivery Data Format Information*

Output:

- *to Persist Science Products: Request for Science Products to be moved to Prepare Products.*
- *to Persist Science Products: Uniform Resource Identifiers (URIs) for data in our internal data format (drops) relating to Science Products local to this SDP site.*
- *to Query, and Request Data: Science Catalogue Query results*
- *to Query, and Request Data:Science Catalogue contents for replication*
- *data usage statistics to LMC*

The Index Science Products function has two aspects:

1. It maintains the local catalogue of Science Products; i.e., the ones created at the SDP site, it therefore provides a service function interface.
2. It provides functionality to manage the preservation of data into and out of Persistence and organise and index Science Products
    a. Associates data with metadata in the Science Products
    b. Creates entries in the Science Product Catalogue for new Science Products

Note that this function does not move data around, it simply provides information to other functions about where the data is via URIs.

The Science Product catalogue includes all of the astronomy metadata that is needed to identify Science Products. The function accepts metadata queries to the local catalogue and can provide a dump of the local catalogue so this can be replicated to other locations if required. It also accepts queries for specific Science Products to be copied to the Prepare Science Products function.

The function also reports data usage statistics to LMC.

This function is part of the L1 Preserve Data function.

12.1.18. Persist Science Products

Input:

- *Science Products*

Output:

- *Science Products*

The Persist Science Products function receives Science Products from the Preservation and Index Science Products function. It maintains Science Products long term, including the generation of multiple copies and/or parity or erasure code protection, checksumming and other means to ensure the resilience of the Science Products.

This function is part of the L1 Preserve Data function.

12.1.19. Backup Science Products

Input:

- *Science Products*

Output:

- *Science Products*

This function provides the backup of persisted Science Data Products potentially off site from the main SDP: a possible implementation of offsite backup could be the utilisation of the alternate SDP instance, specific off-site facilities or Regional Centres.

This function is part of the L1 Preserve Data function.

12.1.20. Prepare and Deliver

Input:

- *Delivery Request (format, destination, SKA user credentials, RC user credential)*
- *drops*
- *Science Catalogue*
- *Science Product URI List*

Output:

- *Science Product Location*
- *Science Products*

- *Catalogue Replication*
- *Delivery Request Status Update*

The Prepare and Deliver function enables Delivery Requests to be queued and prioritised. It manages the delivery of Science Products to the requested destination which includes stripping away the drop format signature and converting to the format requested by the user. It also manages the replication of the Science Catalogue to the requested destination. The Delivery System functions need local storage for working space and for buffering the Science Products being transferred. A Delivery Request is the part of a User Request to be executed at a specific SDP.

This function also manages the delivery of data to specified locations as a result of a Query managed by the Query and Request function.

This function is part of the L1 Deliver Data function.

12.1.21. Query and Request

Input:

- *Group Membership List*
- *updated Delivery Request*
- *Delivery Request Status*
- *drop URI List*
- *User Queries*
- *User Selections*
- *SKA User credentials*
- *RC User Credentials*

Output:

- *Science Catalogue Query Result*
- *monitor User Request Status*
- *User Request Completion notification*
- *Science Product Usage Statistics*
- *Delivery Requests (to Prepare and Deliver)*

This function provides an external interface to users and observatory staff for the SDP through web services or a graphical user interface. This allows both external users and Observatory Staff to query and select Science Products that have been created by the SDP for transfer. It supports both public and proprietary access to Science Catalogue entries and Science Products. It provides a set of the IVOA-compliant services that relate to searching and access. The function supports ad-hoc User Requests through the GUI or through the web service and a subscription service for Science Products that are described by pre-selected query constraints on the Science Catalogue.

This function is part of the L1 Deliver Data function.

12.1.22. AAAI

Input from SDP:

- *user credentials (username and password or certificate).*
- *input from TM:*
  - *list of trusted IdPs*
  - *list of trusted CAs*
  - *group membership information*

Output:

- *Username mapping and group membership list, or null*

The AAAI (Authentication, Authorisation, Allocation and Identity management) function is used to interface to appropriate authentication and authorisation mechanisms. For authentication it is assumed that both federated identity providers (IdPs) and X509 Certificate Authorities (CAs) will be supported. Lists of trusted IdPs and CAs will be obtained from TM and used to authenticate user credentials. TM will also supply group membership information which will be returned by the AAAI function and used to determine access privileges.

This function is part of the L1 Deliver Data function.

12.1.23. Aggregate QA Metrics

Input:

- *SDP function specific QA information*

Output:

- *aggregated SDP QA information*
- *Science Product metric to Stage Data*

The Quality Assurance (QA) Aggregator function receives QA information from all SDP functions that produce QA information. These are identified with a circle in the top left corner of each such function in the Level 2 functional flow diagram (Figure 3).
This function interfaces to a dataflow system that does not perform all of the operations in a prescribed order. Therefore the QA aggregator will need to have a buffering sub-function which allows us to re-assemble data processed at different times to display QA which logically belongs together, i.e., data observed at the same time. It should also have the ability to make QA data with incomplete inputs.

This function is part of the L1 Control SDP function.

12.1.24. Visualise QA Metrics

Input from QA Aggregator:

- *aggregated SDP QA information*
- *input from TM:*
  - *QA User Control*

Output:

- *graphical views of QA information*

- *aggregated QA Metrics as a Science Data Product*

The Quality Assurance Visualisation function creates visual representations of the SDP QA data.

This function is part of the L1 Control SDP function.

12.1.25. Produce Fast Telescope State

Input:

- *Local Telescope State*

Output:

- *Local Telescope State Information for the Receive Visibilities function*

The Produce Fast Telescope State function is responsible for ensuring the timely delivery of critical telescope state information from the Telescope Manager to the SDP Receive function. Such critical state information is needed at the point at which SPEAD data is received from the CSP and decoded. If such data is not present at the time of receiving a particular correlator dump, the dump must either be buffered or discarded.
The ICD between SDP and TM defines those pieces of telescope state information that are deemed to be critical, but in general will include information needed to unambiguously assign the incoming data to a particular observation and to correctly interpret the axes of the data.

This function is part of the L1 Control SDP function.

12.1.26. Manage Local Sky Model

Input:

- *Global Sky Model*
- *source finder output*

Output:

- *list of sources*
- *model of a source*
- *(TBD) images of extended sources*
- *updates to the Global Sky Model*

This service manages the Local Sky Model described in section 'Local Models'. It has functionality such as initial population, query and insertion of new sources.

This function is part of the L1 Control SDP function.

12.1.27. Manage Local Telescope State

Input:

- *Telescope State*
- *calibration parameters*

Output:

- *observation metadata*
- *calibration parameters*

This service manages the Local Telescope State described in section 'Local Models'. It offers functionality to get the time-invariant metadata of the observation and to store and retrieve calibration parameters which can depend on antenna, time, frequency, polarisation, and/or direction. It also has functionality to calculate the beam for a given antenna, time, frequency and direction and (TBD) functionality to calculate ionospheric TEC from direction-dependent phase solutions and back from interpolated TEC to phase. The calibration parameter part of the service might be highly distributed to accommodate the possibly high data rates. Additionally, high-volume data such as the full gain tables can be managed explicitly as data within the data-driven architecture.

This function is part of the L1 Control SDP function.

12.1.28. Master Control

The Master Control function encompasses a wide variety of functionality that is needed to ensure that the appropriate emergent behaviour of the SDP system is achieved in response to command requests originating from the Telescope Manager. In general, this will involve the receipt of a command to activate a particular capability (see the `Control Concept' section for definition of a capability). The setup and initial control of the capability is the responsibility of the Master Controller. The Master Control function will also include functionality to estimate resource usage of a particular capability to allow both internal planning and external scheduling by the Telescope Manager.

In the SDP data-driven system, fine-grained control of components does not require direct overview from the Master Controller. It simply plays a supervisory role, ensuring that all needed components have been started with an appropriate configuration and that such components persist in a functional state for the duration of the requested capability.

The Master Control function is also responsible for maintaining the overall SDP state machine, and in particular it handles the critical power-on to idle and power-off transitions.

In the context of the L2 diagram (Figure 3), the control plane is not shown and is deemed to be a cross-cutting underpinning layer that will explicitly connect to each of the L2 components. In some cases direct control connections may extend lower in the diagram, but in general these internal interfaces should be positioned at a high enough level so as to logically group similar functionality. For example, it is preferable that a single control connection to the Master Drop Manager is maintained, with connections to the subordinates managed by the Master Drop Manager itself.

This function is part of the L1 Control SDP function.

12.1.29. Manage Science Events

Input:

- *Science Events from pipeline components*

Output:

- *IVOA-compliant science event alert*
- *update of the Science Event Catalogue*

This function accepts alerts generated through the processing and prepares these in a standard format for an IVOA alert and passes this to TM for further transmission. An entry into a persistent catalogue of these events is also made.

This function is part of the L1 Control SDP function.

12.1.30. Manage Data Lifecycle

The Manage Data Lifecycle (MDL) is a service function which provides a range of functionalities to monitor and control the life cycle of the internal data (drops) in the SDP system when some degree of persistence is required. The MDL manages these data from their initial instantiation to deletion.

The MDL interfaces as a service to the Execute Processing implementation of a data-driven approach of the architecture discussed in the next section, but the MDL exists independently of a given capability.

The range of functions that the MDL implements on the internal SDP data (held in data drops) includes:

- Garbage collection
- Management of expiry policies
- Registry services for data drops including data location
- Configurable heuristics for persisted data drops which implement explicit migration of the drops through a storage hierarchy

Drops can exist simultaneously in various identical copies on different physical storage media. In order to be scalable, the MDL will need to be implemented in a distributed way. Such functions are typically implemented using a Hierarchical Storage Manager product of some kind; the detailed requirement analysis has to reveal which, if any, of the existing products would fulfil these requirements. The MDL will also need to keep some kind of history either in the form of logs or as a complete history of all drops within a certain time window in order to allow for debugging and tracing.

This function is part of the L1 Process Data function.

12.1.31. Execute Processing

The Execute Processing function manages the execution of the functions associated with the processing pipeline and it is closely associated with the concept of capabilities. It is fully described in [RD02].

This function is part of the L1 Process Data function.

12.1.32. Allow Observatory Access to Query

This function provides local access to querying SDP Science Data Products.

This function is part of the L1 Support Observatory Commissioning and Operations function.

12.1.33. Allow Observatory Access to Request and Deliver

This function provides local access to querying SDP Science Data Products.

This function is part of the L1 Support Observatory Commissioning and Operations function.

12.1.34. Visualise Data

This function provides visualisation tools for visualising data by observatory staff: requirements are to be defined.

This function is part of the L1 Support Observatory Commissioning and Operations function.

12.1.35. Analyse Data

This function provides analysis tools for visualising data by observatory staff: requirements are to be defined.

This function is part of the L1 Support Observatory Commissioning and Operations function.

12.1.36. Create and Maintain Pipelines

This function provides pipeline maintenance and creation tools for visualising data by observatory staff: requirements are to be defined.

This function is part of the L1 Support Observatory Commissioning and Operations function.