



SDP OPERATIONS PLAN

Document Number..... SKA-TEL-SDP-0000081
 Document Type..... MPL
 Revision..... 01C
 Author..... J. Taylor
 Date..... 2018-04-25
 Document Classification..... Unrestricted
 Status..... Draft

Name	Designation	Affiliation	Signature
Authored by:			
John Taylor	SDP Team	University of Cambridge	<i>John Taylor</i> <small>John Taylor (Apr 24, 2018)</small>
			Date: <input style="width: 80%;" type="text"/>
Owned by:			
			Date: <input style="width: 80%;" type="text"/>
Approved by:			
Paul Alexander	SDP Project Lead	University of Cambridge	<i>Paul Alexander</i> <small>Paul Alexander (Apr 24, 2018)</small>
			Date: <input style="width: 80%;" type="text"/>
Released by:			
			Date: <input style="width: 80%;" type="text"/>

DOCUMENT HISTORY

Revision	Date Of Issue	Engineering Change Number	Comments
C	2016-03-24	-	Prepared for SDP delta-PDR review
D	2016-05-24		Prepared for SDP delta-PDR close-out
01	2016-07-20		Prepared for SDP delta-PDR final sign-off
01C	2018-04-25	ECP-SDP-180001	Prepared for SDP pre-CDR review

DOCUMENT SOFTWARE

	Package	Version	Filename
Word processor	Google Docs		SKA-TEL-SKO-0000000-01_GenDocTemplate
Block diagrams			
Google docs Add-ons	Cross Reference Table of contents List of figures		Used for figure & table numbering and references. Used for heading numbering. Used to generate list of figures and tables

ORGANISATION DETAILS

Name	SKA Organisation
Registered Address	Jodrell Bank Observatory Lower Withington Macclesfield Cheshire SK11 9DL United Kingdom Registered in England & Wales Company Number: 07881918
Fax.	+44 (0)161 306 9600
Website	www.skatelescope.org

TABLE OF CONTENTS

1 Introduction	6
1.1 Purpose of the document	6
1.2 Scope of the document	6
1.2.1 Scientific success metrics	6
1.2.2 Operational success metrics	7
2 References	8
2.1 Applicable documents	8
2.2 Reference documents	8
3 SDP Overview	9
3.1 Processor Platform	10
4 Science Operations	11
5 Engineering Operations	12
5.1 Availability	12
5.2 Standards	13
5.3 Operational performance monitoring	13
5.3.1 Health and Status Monitoring	13
5.3.2 Fault Handling and Status Reporting	14
5.3.3 Error Detection	15
5.3.4 Error Recovery	16
5.4 Maintenance Process	16
5.4.1 DevOps	16
5.4.2 Maintenance Plan	16
5.4.3 Support Tools and Instrumentation	17
5.4.4 Hardware Lifecycle Management	18
5.4.5 Acceptance Process	18
6 Operations Plan Timeline	19
7 Appendix 1 SDP Hardware	25
8 Appendix 2 SDP Software Modules	26
9 Appendix 3 L2 Operational Requirements	26

LIST OF FIGURES

Figure 1	Platform Services Component and Connector View	10
Figure 2	Component and Connector View showing External Interfaces to SDP	14

LIST OF TABLES

Table 1	Tasks Involved in Maintenance Strategy	17
Table 2	Acceptance Tests and Location	19
Table 3	Preliminary Overview of Acceptance Tests in line with Construction Plan	23
Table 4	Product Breakdown Structure and LRU assignation (Preliminary)	25
Table 5	High Level Software Modules and High Availability Strategy	26
Table 6	L2 Operational Requirements	26

LIST OF ABBREVIATIONS

SKA	Square Kilometre Array
SKAO	SKA Project Office
AAAI	Authentication, Authorisation, Access and Identification
Ai	Inherent Availability
AIV	Assembly Integration and Verification
AR	Array Release
BOM	Bill Of Material
C&C	Component and Connector
CDR	Critical Design Review
CICD	Continuous Integration Continuous Delivery/Deployment
COTS	Commercial off the Shelf
CSP	Central Signal Processing
ECC	Error Correcting Memory
FAT	Factory Acceptance Tests
FMEA	Failure Modes and Effects Analysis
FMECA	Failure Modes, Effects and Criticality Analysis
FRACAS	Failure Reporting and Corrective Action System
H/A	High Availability
ILS	Integrated Logistic Support
iOBL	Interim Operational Baseline
LOW	Low Frequency : Telescope
LRU	Line Replaceable Unit
LSA	Logistic Support Analysis
MID	Mid Frequency : Telescope
MTBcF	Mean Time Between Critical Failures
MTBF	Mean Time Between Failure
MTTR	Mean Time To Repair
NIC	Network Interface Card (typically Ethernet)
NSDN	Non-Science Data Network
OBL	Operational Baseline
OEM	Original Equipment Manufacturer
PBS	Product Breakdown Structure
PHS&T	Packaging, Handling, Storage & Transport
UAT	User Acceptance Tests
RAM	Reliability, Availability, Maintainability
SAT	Systems Acceptance Tests
SIT	Systems Integration Tests
SPC	Science Processing Centre
SRU	Shop Replaceable Unit
TM	Telescope Manager

1 Introduction

1.1 Purpose of the document

This document provides an Operations Plan for the Science Data Processor (SDP) for the SKA1 Observatory (SKAO) based on the operational context of the SKAO taking into account the criteria for success and the operating constraints of the Science Processing Centres in which the SDP will be housed. Primary drivers for the SDP Operations focus on the availability criteria and ensuring that the SDP maintains optimal efficiency and performance to meet the higher-level observing efficiency, which in turn will influence observing project completion. SDP Operations, like other elements, will be bound by constraints imposed by hosting agreements with the AUS/RSA authorities, relevant Health & Safety and Environmental considerations, and Operational budget constraints.

1.2 Scope of the document

The Operations Plan will evolve over time as the SDP Commissioning and AIV support system provides valuable input into this plan in particular with respect to the necessary tools for maintenance, configuration management, performance monitoring, continuous integration method, benchmarks and documentation.

The document refers to SDP Level 2 Operational Requirements as listed in the Operational Requirements section at the end of the document (Appendix 3). Operational Requirements are labelled [SDP_REQ_XXX], where XXX refers to three numbers relating to the specific requirement.

For consistency with the SDP Hardware PBS [AD21] specific terminology is used in this document. The term “server” is used to refer to a compute server. Where an allocation of the server to a specific component or attribute is inferred, the term “node” is used. For example where a particular server will support a component of Platform Services, this is referred to a Platform Services node.

The operational model described in this document is designed to maximise the operational success of the SKA Observatory. Success, in this context, is defined by the science and engineering metrics described below taken from [AD01] and the methods proposed for Operational Performance.

1.2.1 Scientific success metrics

Several scientific success metrics will be monitored once the observatory becomes operational:

- the over-subscription of observing time is a measure of community demand for access to the facility. This metric will be determined during each time allocation cycle;
- the number of publications, subject to defined acceptance criteria including peer review, is a measure of the observatory’s productivity. This metric will be tabulated at least annually through a combination of web searches and manual reviews of the literature;
- the number of citations to included publications is a measure of scientific impact. This metric will be tabulated as required, primarily through web services such as the NASA Astrophysical Data System;
- the number of publications or citations per unit cost is a measure of value for money.

This list is not exhaustive, and may evolve over time; it nevertheless encapsulates the primary drivers for the operational model.

1.2.2 Operational success metrics

Scientific success and operational success are intimately linked: a highly-efficient observatory, for example, will enable more science time on sky, which should in turn produce more science papers per unit time.

The following operational success metrics will be monitored once the Observatory becomes operational:

- system down time due to faults;
- system down time due to unavailability of computational resources;
- system down time due to planned maintenance;
- operational availability;
- operational availability of specific capabilities (specific bands and specific observing modes);
- observing efficiency (integration time per unit available time);
- observing project completion; and
- safety record.

This list is not exhaustive, and may evolve over time; it nevertheless encapsulates the primary drivers for the operational model.

2 References

2.1 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.

- [AD1] SKA-TEL-SKO-0000307 SKA1 Operational Concept Document
- [AD2] SKA-TEL-SDP-0000013 System-level Module Decomposition and Dependency View
- [AD3] SKA-TEL-SDP-0000047 SDP Construction Plan
- [AD4] SKA-TEL-SDP-0000013 Platform Component and Connector View
- [AD5] SKA-TEL-SKO-0000915 Engineering Report Nov 2017 to Mar 2018
- [AD6] SKA-TEL-SDP-0000050 SDP Preliminary Integrated Logistics Support Plan
- [AD7] SKA1 MID SDP - CSP ICD 300-000000-002
- [AD8] SKA1 LOW SDP - CSP ICD 100-000000-002
- [AD9] SKA1 MID TM - SDP ICD 300-000000-029
- [AD10] SKA1 LOW TM - SDP ICD 100-000000-029
- [AD11] SKA1 INTERFACE CONTROL DOCUMENT
- [AD12] SDP TO INFRA-AUS AND SKA SA SKA-TEL- SKO-0000484
- [AD13] SKA-TEL-SDP-0000013 SDP Architectural Overview
- [AD14] SKA-TEL-SDP-0000013 System-level Security View
- [AD15] SKA-TEL-SDP-0000013 Operational System Component and Connector View
- [AD16] SKA-TEL-SDP-0000013 Processing Component Module View
- [AD17] SKA-TEL-SDP-0000033 SDP L2 Requirements
- [AD18] SKA-TEL-SDP-0000054 Prototyping Report
- [AD19] SKA-TEL-SDP-0000013 Science Pipeline Management Use Case View
- [AD20] SKA-TEL-SDP-0000013 Delivery-centric Component and Connector view
- [AD21] SKA-TEL-SDP-0000064 SDP Product Breakdown Structure

2.2 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.

- [RD1] <https://www.openstack.org/science/>
- [RD2] <https://docs.openstack.org/openstack-ansible/latest/>
- [RD3] <https://github.com/SKA-ScienceDataProcessor/p3-appliances>
- [RD4] <https://wiki.openstack.org/wiki/Ironic>
- [RD5] <https://wiki.openstack.org/wiki/Monasca#Overview>
- [RD6] SDP RAM Report SKA-TEL-SDP-0000115

3 SDP Overview

In some respects the SDP is consistent with traditional High Performance Computing (HPC) systems in terms of the types of applications it needs to support, the components it needs to achieve the performance it requires (of order 100s of Pflops coupled with many PBytes of multi-tiered storage). Unlike traditional HPC facilities however, the SDP has to operate as an integral part of the SKA observatory and be scheduled and coordinated in real time with the other elements of the SKA. This close coordination of data ingest and data processing, storage and delivery gives rise to important operational requirements on the SDP. In particular the SDP must be able to quickly configure itself to accept input data from the telescopes under instruction from the Telescope Manager, it must be able to calculate in advance how long it will take to process certain observations and derive Scheduling and Processing Blocks and subsequently it must be bounded within that estimated time. Additionally, in order to achieve availability requirements of the SKA telescope, it is necessary that maintenance schedules of the SDP are aligned with maintenance schedules for the rest of the SKA telescope with steady-state maintenance being conducted while the SDP is available. Finally, it must offer a Delivery service to the overall Observatory for access to Observatory Data Products via a Preservation product. These aspects are discussed in [AD13].

These requirements together have led to an architectural representation of the SDP couched in a framework best exemplified by the System Level Module and Dependency View [AD02], such that services required by the Execution Frameworks under the auspices of Execution Control are clearly separated in terms of services to support Science Pipeline Workflows and Platform, SDP and System Services providing infrastructure, domain specific and underlying system services. This makes for an architecture that promotes adoption of standard practise in cloud techniques (automated provisioning of compute and storage infrastructure for multiple tenants, fault isolation and recovery, monitoring and logging, resource reservation, elasticity and automation of self-test) and promotes best practice in modern data centre operations and potentially leads to a scenario in which Operational Costs can be optimised (SDP_REQ_745). Such properties are clearly identified by the quality attributes of these services in [AD02] and define distinct areas for development for both domain and non-domain functionality as described in [AD03].

Based on the above and the assumption that operational aspects will exploit COTS components and widely accepted standards, technologies and techniques, the implementation and allocation of software modules is readily achievable. By adopting these practises, the necessary interfaces to Data Centre Infrastructure Management standards will exist and the SDP will satisfy [SDP_REQ_757] and the necessary interfaces with INFRA-ICD [AD12]. Maintaining this strategy further ensures that hardware refreshes do not prevent obstacles to the re-implementation of SDP on such hardware [SDP_REQ_816].

3.1 Processor Platform

A key aspect of the Operational aspects of the SDP rests with Platform Services and the Platform C&C View [AD04] (reproduced in Figure 2 for context) and corresponding Operational States of the SDP described therein.

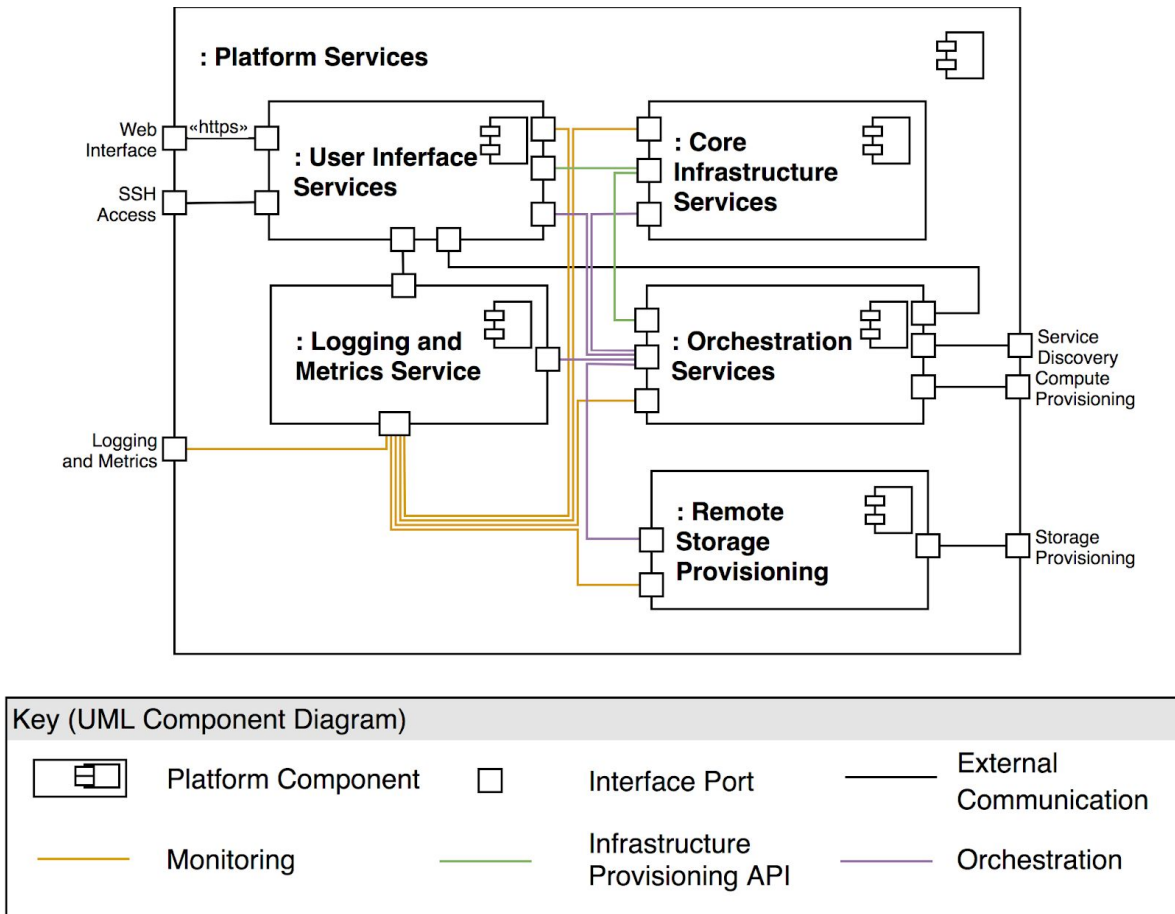


Figure 1: Platform Services C&C View from [AD2]

Platform Services provide critical services for the operation of the SDP in conjunction with Execution Control. As such these services and any persistent state require allocation to the appropriate hardware in order to maintain availability and also graceful degradation of the SDP in terms of shutdown [SDP_REQ_30]. Hardware allocated to the support of these critical services will require appropriate back-up power to avoid any power interruption [SDP_REQ_782]. On start-up of the SDP the initial state of the SDP will be in low-power mode [SDP_REQ_783] as defined by [AD4] the Platform C&C View.

Platform Operators have access to the SDP via standard mechanisms (SDP_REQ_285), such as ssh, with the addition that Health Monitoring and logging are accessible from HTTP interfaces which provide appropriate dashboards onto the SDP. [SDP_REQ_4, 723]

Core Infrastructure Services is the component that handles the provisioning, management and control of the SDP hardware (software-defined compute, storage and networking). The scale of the SDP systems is such that automation of these functions is essential and we are currently tracking

developments in cloud software stacks such as OpenStack [RD1] and infrastructure-as-code environments, such as OpenStack-Ansible [RD2]. Examples of Ansible-Playbooks are already available on the SDP Github [RD3] which will evolve into a suite of CICD tools for use by Platform Operators. In addition to the developments of Cloud ecosystems, the SKAO is in collaboration with CERN on OpenStack which could provide shared opportunities in the development of Operation aspects and tooling of the SDP and other sub-elements [AD5].

These tools will be made available for Platform Operators to ensure that optimal performance of the system can be achieved. This varies from benchmarking tools for various parts of the system (compute, data transport, buffer, etc.) to network analysers and low-level debugging tools. Mostly these are the tools that any competent system administrator and software developer will be aware of and trained to use. In addition, these also include tools to locally access and analyse the performance counters and sensor information available in modern hardware.

4 Science Operations

Science Operations encompass those activities which relate to the delivery of science. There is substantial interaction between the overall operations of the SKA and the Science Operations of the SDP as much of the former activity has either a direct or indirect impact on SDP.

The overall operation of SDP, how it is scheduled, which data products are produced etc. is discussed in the SDP Architectural Overview [AD13]. Details of how SDP deals with scheduled processing is discussed in the behaviour section of the SDP Operational System Component and Connector View [AD15].

The science operations activities specific to SDP operations are:

- Preparation, maintenance, modification and testing of new workflows / algorithms;
- Development, maintenance and testing of new processing components to be used in workflows;
- Scientific assessment of data quality, algorithmic and telescope performance via data visualisation and mining;
- Delivery of science data to SKA Regional Centres for astronomer access.

Science algorithms are defined by parameterised workflows formed from scripts / programs driving execution frameworks which themselves make use of processing components to perform all processing. Developing, programming and testing these workflows is a core aspect of the science operations of the SDP. The SDP architecture has been developed with a major architectural driver being the maintainability of these workflows by the observatory science, programming and operations team. This is achieved architecturally by the use of execution frameworks which provide the framework in which to write highly scalable data-driven workflows. Processing components can then be written, typically, to run on individual nodes without the complexity of developing software for a highly parallel, distributed, environment. Only validated workflows and components are run by the SDP. The operations activity of developing, modifying and maintaining workflows is addressed in the architecture documentation Science Pipeline Management Use Case View [AD19].

A specific example of the development of workflows is to allow for the scientific assessment of data quality together with the visualisation and analysis of derived information. This is achieved by the development and deployment of workflows which include Quality Assessment components. These quality assessment components provide the mechanism by which operators and science staff can analyse and visualise information produced during a workflow (e.g. statistics on flagging, or quality of the gain solution etc.). Modifying workflows to include Quality Assessment components is a standard aspect of science operations. Additionally science operations staff may access SDP data products via the Delivery interface and perform analysis on these products outside of the SDP within the environment provided by Observatory Support Tools. Modification of workflows to define new data products (perhaps by identifying intermediate data items as data products for the purposes of science operations) is an additional and complementary route to analysis.

Access to data products whether by the observatory staff or end-user astronomers is via the Delivery interface to SDP. This interface also provides the mechanism to send data products to SKA Regional Centres. Science and operations staff also have an interface to define the policies on data products. These aspects are discussed in more detail in the Delivery-centric Component and Connector View [AD20].

5 Engineering Operations

5.1 Availability

Engineering Operations comprises those activities necessary for the maintenance and development of the telescopes and the SKA infrastructure in the host countries. The scientific success of the SKA, as described above, requires that the telescopes be available for science observations for as much of the time as possible within the constraint of available resources as defined by [SDP_REQ_762].

Achieving this level of availability requires that all telescope systems (including both hardware and software) be designed for high reliability (high MTBF or more importantly high MTBcF) and simple maintainability (low MTTR). The scale of the SDP demands an industrial level of logistics engineering covered in [AD6]. The trade-off between MTBF, MTBcF and MTTR is a delicate balance, given the desire to minimise manual intervention, subject to the appropriate level of maintenance, and the cost implicit to gain higher reliability. For example, one aspect of this may be the over-provisioning of spares, which may well be hot-spares, compute servers such that in the event of a fault, the server is not replaced or fixed, but taken out-of-service until a more convenient time. Such practises are widely adopted in HPC centres and given the tolerance of the SDP architecture to such failures, this may well provide a more cost-efficient practise. For other services, such as storage related, appropriate use of redundancy, for example RAID or erasure coding [SDP_REQ_824] together with H/A configurations will be employed. These will be coupled with recovery of components through the configuration database [AD15]. Failures of hardware in this respect may require lower MTTR to ensure availability criteria are met. These are also identified in the RAM Analysis [RD6].

5.2 Standards

The wide adoption of suitable standards [SDP_REQ_745] is a means to mitigate diversity and the inevitable increases in costs for operation as a result. The SDP is active in the investigation of

common approaches to solutions that could be adopted both within the SDP sub-systems as well across sub-elements. Of particular concern for operational aspects are:

- automated configuration [RD2] and deployment [RD4] tools;
- containerised application environments such as Docker and Singularity;
- Platform Management tools and Cloud-based environments such as found in OpenStack and previously identified in the quality attributes associated with the SDP Module View;
- System Services based, for example, on widely available technologies such as Linux, Object and File Storage Systems, etc.;
- Customised SDP Services.

5.3 Operational performance monitoring

These are the primary metrics that will be used to measure the operational performance of the SDP. The operational requirements to track these metrics are covered elsewhere in this section. In summary:

- System downtime due to faults - tracked through the observing logs and fault database;
- Meeting Operational availability through implementation of the ILS Plan [AD6] and monitoring to prevent critical failures [SDP_REQ_825];
- Maintain an SDP Performance Model [SDP_REQ-749] to ensure that scheduled observations can be processed within known constraints;
- Ratio of processing to observing time.

5.3.1 Health and Status Monitoring

Platform Services will provide a wide range of standard monitoring points that will be provided to Execution Control to allow TM [AD9 & AD10] to have sufficient system wide visibility to make informed decisions on the operation of the SDP [SDP_REQ_814]. The SDP will handle monitoring in a hierarchical and consequently more manageable fashion by querying aggregate information from the SDP sub-elements such as the Compute and Interconnect Racks, Delivery and Preservation. This information will provide an overall health state with respect to the SDP, together with the actual status of the SDP in terms of usage and operational state. The following diagram (Figure 3) provides a view on the Security View [AD14] representation showing component interfaces between Observatory Support and the NSDN.

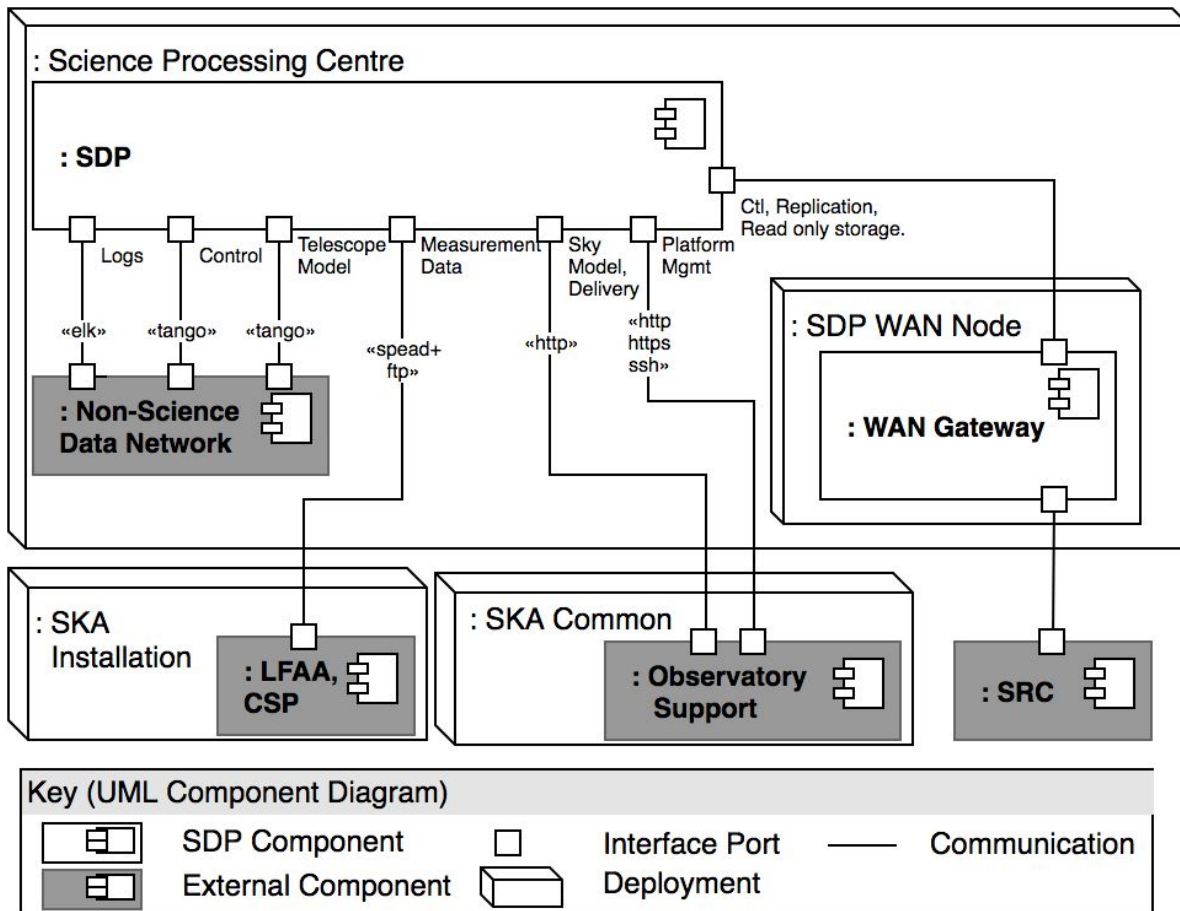


Figure 2: Component and Connector View showing External Interfaces to SDP.

5.3.2 Fault Handling and Status Reporting

The logging of faults will be detected by Platform Services and the nature of the fault automatically described in the log from the error condition and/or fault code allowing information to flow to the Help Desk for Operator intervention which may lead to a Maintenance action. The appropriate course of action will depend entirely on the severity and impact of the fault and defined in the Failure Reporting and Corrective Action System (FRACAS) which will be incorporated as part of the ILS Plan. **In exceptional circumstances faults may trigger specific events where risk of damage is, or may be about, to occur, e.g. where temperature of a hardware component has breached a soft or hard threshold, loss of power or communications. Such faults will preempt any actions by the Operator and render the SDP to a fail-safe state [SDP_REQ_52,755].**

The following provides a list of Fault handling and status reports which should be integrated with the SKA-wide fault database. At the present time a FMECA has not been performed and will be undertaken as part of the ILS Plan [SDP_REQ_823] [AD7].

- Be informed that a fault has occurred, performed automatically by Platform Services [SDP_REQ_821].
- Identify the nature and cause of the fault.
 - A fault database of currently open and closed faults, and their resolution (or otherwise), should be accessed if the fault or its solution is not familiar to the

operator. (e.g. a compute server failure may warrant no further action until an appropriate point of time - scheduled maintenance, taking into account impact on Operational Availability and RAM analysis [RD06]). A software, or configuration, fault may well be able to be corrected while in operation.

- Attempt limited troubleshooting, depending on the nature and severity of the fault (e.g reboot server, run diagnostic self-test, etc) limited to 10 mins [SDP_REQ_818, 819].
- Isolate the faulty system from the rest of the SDP if this is not done automatically.
 - E.g. Soft-errors in a memory sub-system may warrant shut-down of a server and maintenance.
 - Quality Assessment reports should inform the operator whether there is any significant degradation to the data compared to expectations. [SDP_REQ_822]
- Continue processing if possible.
 - If it is not possible to continue with the present Processing Block, based on the appropriate policies to be decided, then the next feasible Processing Block in the plan should be commenced. In most cases, this will occur automatically.
 - The SDP status will be updated automatically once a critical fault is realised so that the telescope manager can flexibly and dynamically change the short-term plan to choose the next feasible observation to execute utilising the available resources or escalate appropriately [SDP_REQ_764].
- Annotate the fault report with information that should contain:
 - A narrative describing the fault, its characteristics and the impact to observations not already described by the automated report;
 - Any corrective actions taken and the amount of observing time lost, if any.
- Notify appropriate personnel of the faults.

5.3.3 Error Detection

Platform Services will be responsible for the aspects of error detection not available through standard OS-level tools by providing methods to combine standard monitoring of the sub-systems alongside Quality Assessment mechanisms implicit to the pipelines. Firstly, errors and state information are broadly classified into alarms and events (SDP_REQ_35). Alarms are events that without further scope broadening represent a critical failure [SDP_REQ_763] and escalate to a telescope fault and require operator notification as described above (although intervention may still be automatic within the SDP). Events are points of interest, that when taken in a broader system view, perhaps via automatic aggregation, could reveal a higher level issue.

It is also important to be able to mark alarms, and other errors, with a variety of states, such as 'known failure', to manage and prioritize faults. This is especially important for alarms that get triggered at a regular cadence (e.g ECC memory errors or temperatures above a soft threshold which if left uncorrected may lead over time to a critical failure) – the system must be able to suppress further warnings when directed, but still track the alarm events themselves. Such alarms will be stored and updated within the engineering database and engineering support function requirements [AD04]. Where faults will potentially impact on the inherent Availability (viz. critical) these will be monitored in order to prevent operational impact. [SDP_REQ_825].

5.3.4 Error Recovery

The primary assumption of the SDP is that, as much as possible, sufficient loose-coupling exists at the level of the Processing together with SDP and Platform Services to provide roll-back without becoming limited by any particular serial process or global synchronisation [SDP_REQ_820] and discussed in the Rationale in the Operational System Level C&C View [AD15] and Processing Component Module View [AD16]. This requirement is also fundamental to resilience and error recovery. In effect, and where possible, errors will be detected, and that particular operation will simply be discarded until such time as the error can be corrected. Where appropriate as identified in the PBS (see Appendices I and II), redundancy, high availability strategies will be required to ensure that appropriate mitigations of such failures do not lead to critical failures and hence loss of service and to the unavailability of the SDP (SDP_REQ_723).

5.4 Maintenance Process

5.4.1 DevOps

DevOps tools are expected to fit into one or more of these categories, reflective of key aspects of the development and support and maintenance process:

- Code - development and review, source code management tools, code merging;
- Build - continuous integration (CI) tools and build status such as Jenkins methods [SDP_REQ_759];
- Test - continuous delivery/deployment tools allied to the CI tools provide feedback on risks;
- Package - package management for application pre-deployment staging, this may be afforded by use of Containers, for example, Docker and Singularity;
- Release - change management, release approvals and release automation;
- Configure - infrastructure configuration and management, Infrastructure as Code tools such as, for example, OpenStack Ansible;
- Monitor - applications performance and infrastructure monitoring such as provided by , for example, OpenStack Monasca [RD6].

The development of the DevOps environment is expected, by its very nature, to be an ongoing process, although should be in place at the beginning of SDP Operations by Q3 2020 [AD3]. Further discussion with the SKAO will be required as input into the SKAO Operations Plan.

5.4.2 Maintenance Plan

A detailed maintenance plan will be an output of Integrated Logistics Support Plan [AD6]. By adoption of suitable tools and processes the maintenance strategy will be able to diagnose faults down to LRU level so that maintenance can be scheduled appropriately [SDP_REQ_756]. For the most part, maintenance will be able to be performed while the rest of the Telescope is in operation by judicious use of redundancy in the design together with appropriate LRU specified in the Product Breakdown Structure (Appendix 1) [AD13]. This process is identified by the following tasks and discussed in detail in the document:

Maintenance Planning	<p>Define the actions and all the support aspects necessary to ensure that the SDP attains the specified system readiness objectives with minimum Life Cycle Cost;</p> <p>State specific maintenance tasks to be performed on the machine;</p> <p>Define levels of replacements, repair, task times, testability requirements, support equipment and automatic test equipment needs, training, manpower skills and facilities;</p> <p>Develop the preventive maintenance programme in accordance with observation periods and refine this from experience gained;</p> <p>Analyse the proposed work environment on the health and safety of operations team, and define relevant qualifications and/or training requirements as well as safety equipment;</p> <p>Minimize the use of hazardous materials and take into account local regulations for disposal methods for hazardous, recyclable and non-recyclable waste;</p> <p>Obsolescence Management. This will comprise of regular meetings with technology suppliers to ensure lifetime of current equipment and roadmaps for components are well understood. An equipment asset database will be maintained to track the duty-cycle of the Bill-of-Materials and suppliers should provide an obsolescence management plan (OMP) for all equipment supplied. See sections below on Hardware Lifecycle Management.</p>
Support Technical Documentation	<p>Technical manuals;</p> <p>Spare parts list;</p> <p>Preventative (scheduled and unscheduled) maintenance instructions;</p> <p>Corrective maintenance instructions including fault tracking;</p> <p>Installation, test, commissioning, acceptance and change management procedures and reports;</p> <p>Drawings and part lists;</p> <p>Specifications;</p> <p>System Software.</p>
Support Tools and Instrumentation	<p>As much as possible automated and Interactive performance measurement tools to assess individual components (e.g. functionality, power consumption, performance, reliability) and sub-systems. Where possible these should reflect the aspects of the actual processing pipelines [AD09] but also consist of custom and standard industry tests;</p> <p>Equipment for in-situ and ex-situ diagnosis and repair of SDP components (LRU and SRU);</p>
Support Personnel	<p>An assumption is made that Support Personnel are Contracted out to External Organisation(s) - although in the SDP Costing Basis of Estimate [AD10] a number of dedicated support personnel to ensure the operational success of the SDP are identified. These staff will augment the 3rd Party support organisation and provide additional capability in fully understanding the complexity of the SDP from a systematic point-of-view rather than fulfilling maintenance activities.</p>

Table 1: Tasks Underlying the Maintenance Strategy

5.4.3 Support Tools and Instrumentation

As has been noted, the SDP has some similarities with traditional HPC systems in particular around the use of High Performance Compute Server components and high-end networking in order to meet

the challenges of multi-Pflop computational capability. This capability typically pushes the solution into the realm of state-of-the-art processing technologies and can lead to the early adoption of components which have very little field data and use. Such circumstances can lead to failures which are particularly difficult to root-cause and can lead to undesirable maintenance situations. Acceptance criterion and strict acceptance tests, both in the Factory (FAT) where typically only suitable modular hardware can be tested and at the SPC (SAT) where final integration happens, are thus paramount in order to avoid these situations. Such circumstances should be taken into consideration for SDP Operational reasons and the AIV programme of work too, as discussed in the Construction Plan. In addition to the acceptance tests, close collaboration with systems vendors and 3rd party suppliers (e.g. accelerator providers) should be maintained. This may involve the use of vendor specific simulators to assess performance well-ahead of general availability of these components.

5.4.4 Hardware Lifecycle Management

The following process will be taken into account:

- A database to store all hardware attributes as part of the inventory, e.g. Manufacturer, product revision, firmware version, etc.;
- Flexible and complete Hardware introspection;
- Flexible API to add/query server attributes (e.g. OpenStack Nova);
- Burn in and acceptance process (CICD);
- Define when resources are available to application through State workflow [AD4];
- Policy needs to allow segregated access to the different teams (operators, developers), which may involve multi-tenancy;
- Clear retirement procedure or obsolescence management.

5.4.5 Acceptance Process

The acceptance testing strategy will be defined to stress the individual sub-systems that comprise the SDP PBS and its implementation, alongside the Construction Plan. Tests assume that a certain level of vendor supplied tests (burn-in) have already been performed to remove failed components early on in the system-build (preferably in the factory but also repeated to a lesser extent at the respective SPC to flush-out Packaging, Handling, Storage and Transport (PHS&T) effects) and then performance tests to ensure that the system is behaving in a manner that meets expectations and satisfies requirements. These performance tests will be governed by the performance verification suite commensurate with the Construction Plan [AD3]. These will consist of both domain specific tests and industry standard tests to exercise specific hardware (Storage, Compute Server and Networking). The activities to be conducted during Acceptance Testing of each Construction phase (as appropriate - see Table 2) will include, but not be limited to, the following:

- BOM Consistency and Configuration Management;
- Vendor-supplied system diagnostic tests to verify that all components of the system are functioning within specifications.;
- Interface testing;
- The system's capacity and capability measured using an SDP Benchmark Suite. (At the time of writing this does not exist but it will be included as part of the DevOps built-in test environment [AD18]);

- A workload composed of components of the SDP Benchmark Suite [AD18];
- Testing of the remote management and other operational and administrative attributes of the system, including placing simulated maintenance and support requests to demonstrate that service levels are in place;
- Specific timed and untimed activities to evaluate particular aspects of the system, including but not limited to: measuring system shutdown and cold-start time (including fail-safe state), removal and addition of hardware from/to an operational system, time required to perform a complete backup and subsequent full bare-metal restore of a server's software and configuration to an operational state as described in the [AD4];
- Simulated failover of and recovery of scientific workflows operating under Execution Frameworks.

The following Table provides a description of potential Acceptance Criteria and appropriate Tests. These tests will be developed as part of the CI DevOps Built-in Test framework.

Acceptance	Test	Description/Examples	Place
FAT	Specification	Document Defining the solution and specification of the system	Factory
	Inventory	MAC Address, Firmware Version, Model and Serial Number, Platform Services	Factory
	Performance	Limited set of tests to allow shipment	Factory
SAT	Verification	Connectivity (IP Address), Operating System, SDP Libraries, System Service, SDP Services, Size (e.g Bytes), Visual Inspection, Repeat Factory Acceptance to trigger payments	SPC
	Performance	Industry Standard, SDP Benchmark, CI Tests, Platform and SDP Services	SPC
	Failover and Recovery	Customised Tests for H/A Services together with low-power mode and graceful degradation	SPC
SIT	Integration	Operator Interface, Dashboard Services	SPC
	Maintenance	Fault Identification Logging and Repair, Safety	SPC
UAT	User	Documentation, Fault Reporting, Training	SKAO

Table 2: Acceptance Tests and Location

6 Operations Plan Timeline

The following table provides a preliminary assessment of the Operational Tests as applied to the Hardware and Software Product Breakdown. Tests will be refined and updated during each of the Milestones specified.

Mile stone	Key Engineering Goals	Operational Test Software (Appendix II)	Operational Test Hardware (Appendix I)
Tools	SAFe & tools training, testing of tools. DevOps and test platforms in place. SAFe management processes in place and roles filled.	Support Tools and Instrumentation. Configuration Management and DevOps see Section 5.1 covering Platform Services and System Services	P.2.1.1.2 Service Nodes to support Devops on P.2.1.1 Compute Rack
Simulators/ emulators	Provide a SDP-TM interface simulator for early testing by TM Receive visibilities	Verification and Tests available for Receive Server and High Throughput Ethernet together with Platform , System and SDP Services infrastructure	P.2.1.1.2 Receive Server and P2.1.1.3 High Throughput Ethernet Network
PV1	Demonstration of Model Databases and Buffer component scalability. This is an early demonstration of scalability of two critical long-lead time components which are key to the performance of the complete system.	Verification and Tests available for Receive Server and Capacity Storage, High Throughput Ethernet together with SDP Services infrastructure	P.2.1.1.2 Receive Server and P2.1.1.2.7.1 (Capacity Storage) P2.1.1.3 High Throughput Ethernet Network
First MVP	This would be the first meaningful solution level integration from SDP. Goals: - Working high-level architecture; - External interfaces; - Evaluation & correction of existing code base (prototypes).	Initial Systems Integration and Verification Tests	P.2.1.1.2 Service Nodes to support Devops on P.2.1.1 Compute Rack
PV2	Demonstration of Receive & Real-time processing component performance. This needs to be an early sub-system performance demonstration (against a model, given the hardware requirements) since this sub-system needs to	Verification and Tests available for Receive Server and Capacity and Performance Storage, High Throughput Ethernet and Low Latency Network together with SDP Services infrastructure.	P.2.1.1.2 Receive Server and P2.1.1.2.7.1 (Capacity Storage) P2.1.1.3 High Throughput Ethernet Network

	cross over into the commissioning strand quite early on.	Test Failover and Recovery	
PV3	Demonstration of Data Queue and Quality Assessment component scalability. These are grouped together since the underlying property is solubility of stream-like data aggregation and distribution solutions.	Verification and Performance Tests available for SDP Services specified. Test Failover and Recovery	P.2.1.1.2 Service Nodes to support SDP Services
PV4	Performance and functional test of Execution Control and TM interaction.	Verification and Integration Tests available for Execution Control interfaces with SDP and Platform Services infrastructure. Test failover and recovery	P.2.1.1.2 Service Nodes to support SDP and Platform Services
PV5	Demonstration of the execution engine performance at full system scale, with a focused test that tests scheduling, task-startup, memory management and data transfer overheads. This will demonstrate execution engine performance at full scale, against a model of how fast the execution should taken given the available hardware. As a sub-system test, it won't use real computational components, or other sub-systems such as the buffer.	Performance Tests for Receive and Processing Node. Extend SDP and Platform Services infrastructure verification testing	P.2.1.1.2 Service Nodes to support SDP, Platform and System Services with P.2.1.1.2 Receive and Processing Nodes
Functional verification at modest scale	End-to-end processing from data block to calibrated image data with simplified calibration & imaging requirements.	Extend System Integration Test	P.2.1.1 Compute Rack
PV6	Test (or investigate) the architectural decomposition of a hierarchical scalable system. Test to see if the scalability within data islands and between data islands is as required for scalability to the full	Verification and Tests of Low Latency and High Throughput Ethernet Networks tests incorporated into Systems Performance	P.2.1.1 Compute Rack and P.2.1.2 Interconnect System

	system. Use a workflow that exercises different communication requirements, e.g. Direction-Dependant calibration features.	Testing Framework. Extend Platform and System Services	
PV7	Testing scalability of interactions between SDP performance components with a workflow including at least DD CAL and MFMS features. The focus of this milestone will be the efficiency of interaction between the execution engine, model databases, data queues and the batch processing.	Hot Buffer verification and Performance Tests incorporate into Systems Performance Testing Framework. Extend Platform, SDP and System Services. Failover and Recovery	P.2.1.1 Compute Rack and P.2.1.2 Interconnect System
PV8	Demonstration of whole system performance, at the full required scale, when running the ingest, real-time calibration and fast imaging pipelines. This should be a separate (and earlier) milestone to PV10 since these features have quite different requirements to PV10 and are likely to be used at their full scale earlier in the scientific operations of the telescope. Although a system test, this milestone mostly tests the performance of the real-time processing components.	Systems Performance Testing Framework Completed	P.2.1.1 Compute Rack and P.2.1.2 Interconnect System with focus on P.2.1.1.2 Receive Server P.2.1.1.2 Storage Node P.2.1.1.2.7.1 Capacity Storage
Software suite for hardware vendors	Software suite for SKA to evaluate (benchmark) hardware. Required for procurement of AA4 hardware therefore s/w needed 12 months before AA4 SDP deployment.	Systems Acceptance Testing Framework Released	N/A
PV9	Demonstration of whole system linear (or near linear) scalability up to full required scale with a complex feature set, e.g., MFMS synthesis with DD calibration. This is the scalability (but not performance) precursor to milestone PV10.	Hot Buffer Verification and Performance Tests	P.2.1.1 Compute Rack and P.2.1.2 Interconnect System Exercising P.2.1.1.2 Processing Node
SPC MID integration	Integration of SDP system with solution level once the long distance SaDT links (from site to	High Throughput Ethernet verification tests complete. Extend	P.2.1.1 Compute Rack and P.2.1.2 Interconnect

	SPC) are available (AA2). This would be the first time SDP systems are not running at ITF or on site, but in the SPC. This is ahead of SDP hardware deployment for AA3 at the SPC. Look at issues with respect to messaging, packet loss, etc.	Platform Services	System with focus on P.2.1.1.2 Receive Server P.2.1.1.2 Storage Node P.2.1.1.2.7.1 Capacity Storage
SPC LOW integration	Integration of SDP system with solution level once the long distance SaDT links (from site to SPC) are available (AA2). This would be the first time SDP systems are not running at ITF or on site, but in the SPC. This is ahead of SDP hardware deployment for AA3 at the SPC. Look at issues with respect to messaging, packet loss, etc.	High Throughput Ethernet performance tests complete. Failover and Recovery	P.2.1.1 Compute Rack and P.2.1.2 Interconnect System with focus on P.2.1.1.2 Receive Server P.2.1.1.2 Storage Node P.2.1.1.2.7.1 Capacity Storage
SDP AA3 s/w product handover	Sufficient functionality to perform real-time calibration and basic imaging workloads in the context of a system controlled by TM with appropriate control, monitoring and reporting. Minimal set of functionality for the AA3 release of the SDP Operational System to work at AA3 scale.	Systems Integration and User Acceptance	P.2.1.1.2 Service Nodes
Centi-SDP H/W ready (AA3)	Integrate SDP hardware deployment (at the MID & LOW SPC) at solution level & integrate SDP software with centi-SDP hardware before AA3.	Systems Integration Complete	P.2.1 SDP Compute Hardware
PV10	Demonstration of performance of whole SDP system at the required scale and performance level when running a spectral line imaging pipeline. This is a natural precursor to PV11 below and while this will not test the intricacies of data movements and scheduling of the most complex pipelines it will be a good whole system throughput test. Although this tests the complete system, the feature being tested will most exercise the Buffer component and the efficiency of the computational tasks. The stress	Buffer and Processing Server Performance and Verification	P.2.1 SDP Compute Hardware

	on the execution engine will be small.		
SDP 1st operational deployment. (AA4 product handover)	Integrate SDP hardware deployment (at the MID & LOW SPC) at solution level & integrate SDP software with SDP hardware for AA4. Full software and hardware functionality and performance available at both SPCs.	System Integration Tests Complete	P.2.1 SDP Compute Hardware
PV11	Demonstration of SDP whole-system performance at the required scale and performance level and with a full feature set. E.g.: full-scale MFMS synthesis with DD calibration stages; full-scale fast-imaging and calibration test. This would be the final performance milestone: essentially a complete demonstration that SDP is achieving the performance that is required (or expected).	System Acceptance Test, Systems Integrations Test, Preliminary UAT	P.2.1 SDP Compute Hardware
SDP 2nd operational deployment. (post construction phase)	Deployment of SDP full scale hardware during operations.	User Acceptance Test complete	

Table 3: Preliminary Overview of Acceptance Tests in line with Construction Plan

7 Appendix 1 SDP Hardware

P.2.1 SDP Compute Hardware	Inventory	Spares
P.2.1.1 Compute Rack	SRU	
P.2.1.1.1 Compute Rack Management Switch	LRU	Yes
P.2.1.1.2 Server [Service, Receive, Processing, Storage] P.2.1.1.2.1 Latency optimized cores P.2.1.1.2.2 Main memory P.2.1.1.2.3 Throughput optimized cores P.2.1.1.2.4 High Throughput Ethernet NIC P.2.1.1.2.5 Low Latency network NIC P.2.1.1.2.6 Out-of-band management unit (BMC)	LRU SRU LRU SRU LRU LRU SRU	Yes
P.2.1.1.2.7 Storage P.2.1.1.2.7.1 Capacity Storage P.2.1.1.2.7.2 Performance Storage	LRU	Yes
P.2.1.1.3 Compute Rack High Throughput Ethernet Switch	LRU	Yes
P.2.1.1.4 Compute Rack Low Latency Switch	LRU	Yes
P.2.1.1.5 Cabling	LRU	Yes
P.2.1.1.6 Racks Infrastructure P.2.1.1.6.1 PDU P.2.1.1.6.2 Bulk power supply P.2.1.1.6.3 Cable Management P.2.1.1.6.4 Local cooling P.2.1.1.6.5 Rack	SRU LRU LRU LRU LRU SRU	Yes
P.2.1.2 Interconnect System P.2.1.2.1 Low Latency Network Core Switch P.2.1.2.2 Management Network Core Switch P.2.1.2.3 High Throughput Ethernet Core Switch P.2.1.2.3.1 Pluggable optics P.2.1.2.3.2 Fibre P.2.1.2.3.3 Copper cabling P.2.1.2.3.4 Patch panel	LRU	Yes Yes Yes Yes Yes Yes
P.2.1.2.4 Edge Security P.2.1.2.5 Interconnect System Rack	LRU SRU	Yes
P.2.1.3 Inter-rack Infrastructure P.2.1.3.1 Hot Aisle/Cold Aisle equipment P.2.1.3.2 High capacity power equipment P.2.1.3.3 Liquid cooling equipment (hoses, etc) P.2.1.3.4 Cross-rack cable management	SRU	

P.2.2 SDP Preservation Hardware	SRU	
P.2.2.1 Hierarchical Storage Management		
P.2.2.2 Intermediate Storage	LRU	Yes
P.2.2.3 Long Term Storage		

Table 4: Product Breakdown Structure and LRU assignation (Preliminary)

8 Appendix 2 SDP Software Modules

The Operational System Level Component and Connector Views [AD15] and Processing Component Module View [AD16] discuss software reliability in more detail. Table 5 highlights aspects of the reliability strategy for software modules.

Module	Reliability
Execution Control	H/A Configuration
Platform Services	H/A Configuration
SDP Services	H/A Configuration
System Services	Handled by Platform Services
Processing and Other Modules	Handled by Execution Control

Table 5: High Level Software Modules and High Availability Strategy

9 Appendix 3 L2 Operational Requirements

A subset of the Operational Requirements are provided here with a brief description and a pointer to the relevant section where they are discussed. Full details are tabulated in [AD17].

No.	Name	Description	Covered In Section X of document
SDP_REQ-4	SDP Resource Reporting	The SDP shall, on request from the TM, provide a summary of overall resource availability or resource availability for the requested scheduling block.	3.1
SDP_REQ-30	Graceful degradation	The failure of a single component should not cause the SDP to become unavailable.	5.3
SDP_REQ-35	Alarms	The SDP shall support the generation of notification events, in the form of alarms, to the TM based on predefined conditions.	5.3

SDP_ REQ- 52	Failsafe	The SDP shall actively ensure that internal failures do not result in a hazardous situation to the systems and personnel with which it interfaces.	5.3
SDP_ REQ- 285	Accessibility	The SDP shall enable per user access to SDP resources (hardware and software) using the Authentication and Authorisation facilities provided by the SKA (as per EN 50600-2-5. Data centre facilities and infrastructures. Part 2-5. Security systems).	3.1
SDP_ REQ- 723	State logging	The SDP shall log its operational states at all times.	3.1
SDP_ REQ- 728	Master controller failover latency	The SDP master controller shall support failover to an operational state within 1s.	5.3
SDP_ REQ- 745	Affordability	The SDP shall be affordable, i.e., the chosen architecture should ideally minimise expenditure on capital and operational costs. The cost associated with technology readiness, risk and uncertainty needs to be taken into account when minimising capital and operational expenditure. 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 & 5.2
SDP_ REQ- 755	Electronic equipment safety	All SDP equipment shall conform to South African and Australian occupational health, safety and environmental standards for electronic equipment as per local legislation in South Africa and Australia.	5.3
SDP_ REQ- 756	SDP part identification	All SDP parts, cables and connection panels shall be uniquely identified down to LRU level.	5.4
SDP_ REQ- 757	SDP equipment installation	All SDP equipment shall follow standard data centre or HPC centre installation practices.	1
SDP_ REQ- 759	SDP Software update downtime	The SDP shall not require the telescope to be offline while performing software updates. Major software updates shall be performed during engineering and maintenance down time periods of the telescope.	5.4

SDP_ REQ- 762	SDP Inherent Availability (Ai)	The SDP shall have an Inherent Availability (Ai) higher than or equal to 99.9%.	5.1
SDP_ REQ- 763	SDP Critical failure identificatio n	The SDP shall identify more than 99% of all critical failures and report them to the TM.	5.3
SDP_ REQ- 764	SDP Isolation of critical failures	The SDP shall isolate 95% of all critical failures and report it to TM.	5.3
SDP_ REQ- 782	Power interruption survivability	The SDP shall survive a main power interruption, at an arbitrary time.	3.1
SDP_ REQ- 783	Start-up power mode	On start-up, the SDP shall enter low power mode until commanded otherwise.	3.1
SDP_ REQ- 812	Portability of SDP to SKA-SRCs	SKA needs portability of SDP to SKA-SRCs. Portability to environments without TM and CSP could enable deployment/development of SDP at other locations or for other purposes. See	Not covered in this document
SDP_ REQ- 814	Level of Monitoring	Scenario(s) Level of monitoring required of SDP via TM. (QAS not refined or complete yet)	5.3
SDP_ REQ- 816	Portability when hardware is refreshed	Scenario(s) Moving to a different hardware platform when hardware is refreshed.	3
SDP_ REQ- 818	Software Reboot Time	Software failures of the SDP (TBC-084) software that requires rebooting in order to recover from the failure, shall have a MTTR (recover time) of less than or equal to 10 minutes.	5.3
SDP_ REQ- 819	Software Maximum Allowable Recovery Time	Software failures of the SDP (TBC-0085) software that require a software fix, shall fall back on to a previous working state or isolate the problem in such a way to achieve a MTTR (recovery time) of less than or equal to TBD (>10 mins <<8h) minutes.	5.3

SDP_ REQ- 820	Software failure requiring Rollback Recovery	The following design concerns were identified to enable quick rollback recovery from failures: <ul style="list-style-type: none"> - Design the system to minimise the need for schema modification to any shared databases. Addition to the schema is easier to roll-back. - Loose coupling between SDP components so that can roll back on a sub-system basis, reducing impact and time to rollback. - There shall be a process to allow rollback which includes a change to the DB schema. 	5.2
SDP_ REQ- 821	Failure detection to Achieve Ai	The SDP shall detect failures to allow recovery within the time windows specified in SDP_REQ-818 and SDP_REQ-819.	5.3
SDP_ REQ- 822	Node failures recovery	The following design requirements were identified to allow quick recovery from node failures: <ul style="list-style-type: none"> - Detection of node failure and setup of spare node shall be done within TBD seconds. The time will vary according to the amount of data loss during the recovery window and this depends on what the node is doing at time of the failure, i.e. short time for real-time processing and longer for offline processing. Re-routing of real-time (incl ingest from CSP) data streams to a spare node when an ingest node fails to ingest shall be done within TBD seconds. Changing or updating execution graph following a node failure shall be done within TBD seconds/minutes. 	5.3
SDP_ REQ- 823	Failure Prevention	The SDP shall monitor specific variables (as identified by failure analysis / FMECA) that allow detection of critical failures before they occur to allow preventative maintenance or actions (i.e. change in processing schedule).	5.3
SDP_ REQ- 824	Ingest and Buffer Failure Prevention	The SDP shall prevent degraded performance of the ingest & buffer subsystem causing critical failures. Failure prevention strategies could include: <ul style="list-style-type: none"> - Graceful (managed) degradation - Performance loss can be traded off against resilience (minimising performance loss due to storage failures by reducing storage resilience thereby accepting a certain level of data loss) - Localise performance degradation and prevent escalation to a critical failure through partitioning and isolating the parts of the sub-system. 	5.1

SDP_ REQ- 825	Monitoring to prevent critical failures	SDP Software shall monitor key performance indicators that could lead to a critical failure either quickly or after a period of time to allow preventative maintenance or actions to occur. The following critical failures (or failures that lead to critical failures) have been identified: - Buffer full; - Caused by failures or degradation; - Caused by processing taking longer than expected.	5.3
---------------------	--	---	-----

Table 6: L2 Operational Requirements.