SKA1 SDP RAM Analysis

Document Number..... SKA-TEL-SDP-0000115
Document Type..... REP
Revision..... 01
Author..... L. Christelis, F. Graser
Date..... 2018-10-31
Document Classification..... Unrestricted
Status..... Released

Name	Designation	Affiliation	Signature	
Authored by:				
Lorita Christelis	SDP Systems Engineering Team Member	Space Advisory Company	L. Christelis (Oct 29, 2018)	
			Date:	
Owned by:				
			Date:	
Approved by:				
			Date:	
Released by:				
Paul Alexander	SDP Project Lead	University of Cambridge	Paul Alexander (Oct 30, 2018)	
			Date:	

DOCUMENT HISTORY

Revision	Date Of Issue	Engineering Change Number	Comments
C	2018-04-18		Prepared for M20 Pre-CDR review
01	2018-10-31	ECP-SDP-180002	Prepared for M21, SDP CDR review SDP PRECDRs addressed: Major: SDPPRECDR-189 CRT, Table 2 updated, MTTRs in Table 7 and 9 changed from 8 to 2. Clear def of MTTR in Ai added in section 5.4.. SDPPRECDR-193 hot spares: new section added to clarify the decision for hot spares (5.5.4. Hot spares for the Compute Rack) SDPPRECDR-195 Buffer concept: Text added in Table 5. SDPPRECDR-197 follow-up actions Section 6 updated. SDPPRECDR-198 persistent failure: text updated, but with updates in 352, this is not relevant anymore. SDPPRECDR-306 Planning for maintaining/updating RAMS report SDPPRECDR-311 SDP MDT compliance: Table 2 updated with changed (SDP_REQ-759) and new (SDP_REQ-874) requirements. Minor: SDPPRECDR-190 MTTR and MTBF figures - see RD5. SDPPRECDR-191 compute islands: all reference to Compute Island removed and replaced by Compute Rack. SDPPRECDR-196 rankings in table 10: Table 10 updated SDPPRECDR-309 MTTR for SDP critical failures SDPPRECDR-338 Consequence of SEP_REQ-030 violation unclear: SDP_REQ-30 removed. No L1. SPF exists and are managed. References to this requirement in element discussion in section 5.3 also removed. SDPPRECDR-350 execution control single point failures for SDP SDPPRECDR-351 design economization assumption for service nodes updated in section 5.5.3.1. SDPPRECDR-352 Criticality for SDP Availability: section 6 removed, section 5.3.1 added. RBD changed (Fig 3), section 6 expanded. Typo: SDPPRECDR-312 typo: fixed

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	SDP Consortium
Lead Organisation	The Chancellor, Masters and Scholars of the University of Cambridge The Old Schools Trinity Lane Cambridge CB1 ITN United Kingdom
Website	www.ska-sdp.org

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Ai	Inherent Availability
CDR	Critical Design Review
CRT	Critical Repair Time
CSP	Central Signal Processor
FLOPS	Floating Point Operations per Second
GPU	Graphics Processing Unit
HTEN	High Throughput Ethernet Network
ILS	Integrated Logistic Support
LLN	Low Latency Network
LRU	Line Replaceable Unit
MDT	Maintenance Down Time
MTBF	Mean Time Between Failures
MTTR	Mean Time to Repair
OT	Observing Time
PB	Petabyte (buffer capacity)
PSS	Pulsar Search
PST	Pulsar Timing
QA	Quality Assessment
RAM	Reliability, Availability & Maintainability
RBD	Reliability Block Diagram
SDP	Science Data Processor
SEI	Software Engineering Institute
ST	Standby Time

1. Introduction

1.1. Purpose of the Document

The Reliability, Availability & Maintainability (RAM) Requirements of the SDP are architectural drivers for the software and hardware design and selection. It also has a significant influence on capital and operational costs.

To support this design process and ensure compliance to Telescope Level Availability Requirements (section 3.1. L1 RAM Requirements), work has been done to model and analyse the SDP availability problem. Availability depends on the Reliability and Maintainability of the System, so reference in this document to Availability includes both these concepts.

This document is intended for the following audience:

- SKAO and SDP RAM teams will use the report to check compliance of the design against L1 RAM requirements and Telescope level RAM budget allocations. As implementation detail becomes known or design changes are made, the underlying model will be used to assess the impact on the RAM.
- SDP System Engineering and Architecture teams may identify from this report further design drivers or lower level requirements. This report also serves as a tool to make trade-offs concerning the reliability and maintainability characteristics of the design. This report aids in ensuring that the focus, in terms of availability, is placed on the correct products.
- ILS and Operations teams, will use the initial results in their maintenance planning. They will give feedback on the feasibility of this report's maintainability estimations, and elaborate on the implementation thereof via SLAs and spares (which includes support delay estimations not covered in this document).

This work is modelled in a spreadsheet (section 4.3.1. RAM Model). This document is a snapshot of the current state of the SDP design for SDP's CDR milestone, understood through that model.

The Operations Plan [RD1] and SDP Architectural Overview and relevant view packets [AD4] provide detail on how these RAM allocations are met and what mitigations or recovery strategies are in place.

1.2. Scope

This document is within the scope of the SDP Integrated Logistic Support (ILS) Plan [RD3] and informs the SDP Operations Plan [RD1]. This document focuses on the inherent reliability of the design. The ILS looks at the bigger logistical picture to ensure the attributes of Availability, Maintainability and Supportability are supported.

This document addresses the RAM Analysis performed on SDP hardware and software.

1.3. RAM Definitions

The following definitions are important for the context of this document:

Term	Definition
Ai (Inherent Availability)	The probability that a system is operationally capable at any point in time when used in an ideal support environment, i.e. one in which repair commences instantaneously upon failure. Allocated to SDP in [AD1]. **This is the primary focus of the availability analysis.**
Ao (Operational Availability)	The probability that a system is operationally capable at any point in time when used in a realistic support environment, i.e., one in which repair cannot commence until some time after the failure has occurred. It is thus a measure of not only reliability and maintainability, but also of the response time of the support system. Allocated only to the Telescope, allocated to SDP in terms of MDT.
Critical Functions	A function that, if defective or unavailable, will result in the telescope not being available (i.e. Telescope not available, failed observation or a revised observation schedule).
Critical Failures	A failure which may cause injury, damage, or the telescope not being available. A critical failure in this context also includes failures which may result in loss of redundancy or degradation, and if not detected or repaired could result in the telescope not being available.
Critical Repair Time	Time to repair critical failures. It excludes preventive maintenance and corrective maintenance on non-critical items. It also excludes support delays.
Direct Maintenance Hours	Time for all, scheduled and unscheduled, on-equipment maintenance. Exclude administrative and Supply Chain hours. DMH provides a measure of the maintenance personnel hours required on-site. It is limited to on-equipment maintenance.
Fault Isolation	The ability to find the root cause of a fault, by isolating the LRUs whose operational mode is not nominal.
Fault Detection	The ability to detect malfunctions in real time, as soon and as surely as possible.
Component	In the SEI context components are referred to as runtime components, and therefore have a closer relationship to typical

	System Engineering “functions” than to Systems Engineering “components”. Components are implemented by products. Products can be hardware products or software modules.
Reliability	The probability that an item can perform its intended function for a specified interval under stated conditions.
Maintainability[1]	The measure of the ability of an item to be retained in or restored to a specified condition, when maintenance is performed by personnel having specified skill levels using prescribed procedures and resources at each level of maintenance and repair.
MTBF	Mean Time Between Failures is a probabilistic failure prediction of the up time between failures. MTBF is only valid for the "useful life period", which is characterized by a relatively constant failure rate (the middle part of the "bathtub curve", between burn-in and wear-out failure rates). MTBF should not be confused with the expected life of a component.

Other Important RAM terms are Direct Maintenance Hours (DMH) and Full-time Employee (FTE). These terms are discussed in the ILS Document [RD3].

[1] This document only includes MTTR/CRT, the rest of this concept is discussed as part of the ILS Document [RD3]

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.

Reference Number	Reference
[AD1]	SKA RAM Allocation SKA-TEL-SKO-0000102 Rev 03
[AD2]	SDP L2 Requirements, SKA-TEL-SDP-0000033, Rev 03
[AD3]	SKA Phase 1 System Requirements Specification SKA-TEL-SKO-0000008, Rev 11
[AD4]	SDP Architecture Documentation,SKA-TEL-SDP-0000013, Rev 06
[AD5]	SKA1 INTERFACE CONTROL DOCUMENT, SDP TO TM MID, 300-000000-029, Rev 03A SKA1 INTERFACE CONTROL DOCUMENT, SDP TO TM LOW, 100-000000-029, Rev 03A

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.

Reference Number	Reference
[RD1]	Operations Plan, SKA-TEL-SDP-0000081 Rev 2
[RD2]	SKA-TEL-SDP-0000043, SDP Cost Model, Rev 04
[RD3]	ILS Document SKA-TEL-SDP-0000050 Rev 3
[RD4]	SKA-TEL-SDP-0000162 SDP Memo 43 Pulsar Timing Failure Analysis Rev 01
[RD5]	TESLA K40 GPU Accelerator, BD-06902-001_V05 November 2013, Board Specification https://www.nvidia.co.uk/content/PDF/kepler/Tesla-K40-PCIe-Passive-Board-Spec-BD-06902-001_v05.pdf
[RD6]	Practical Reliability (4th Edition), Patrick D.T. O' Connor, published by Wiley, ISBN 0470844620 (HB) 0470844639 (PB)
[RD7]	ReliaWiki

3. RAM analysis methodology

The methodology, as described in Figure 1, was followed for the SDP RAM analysis.

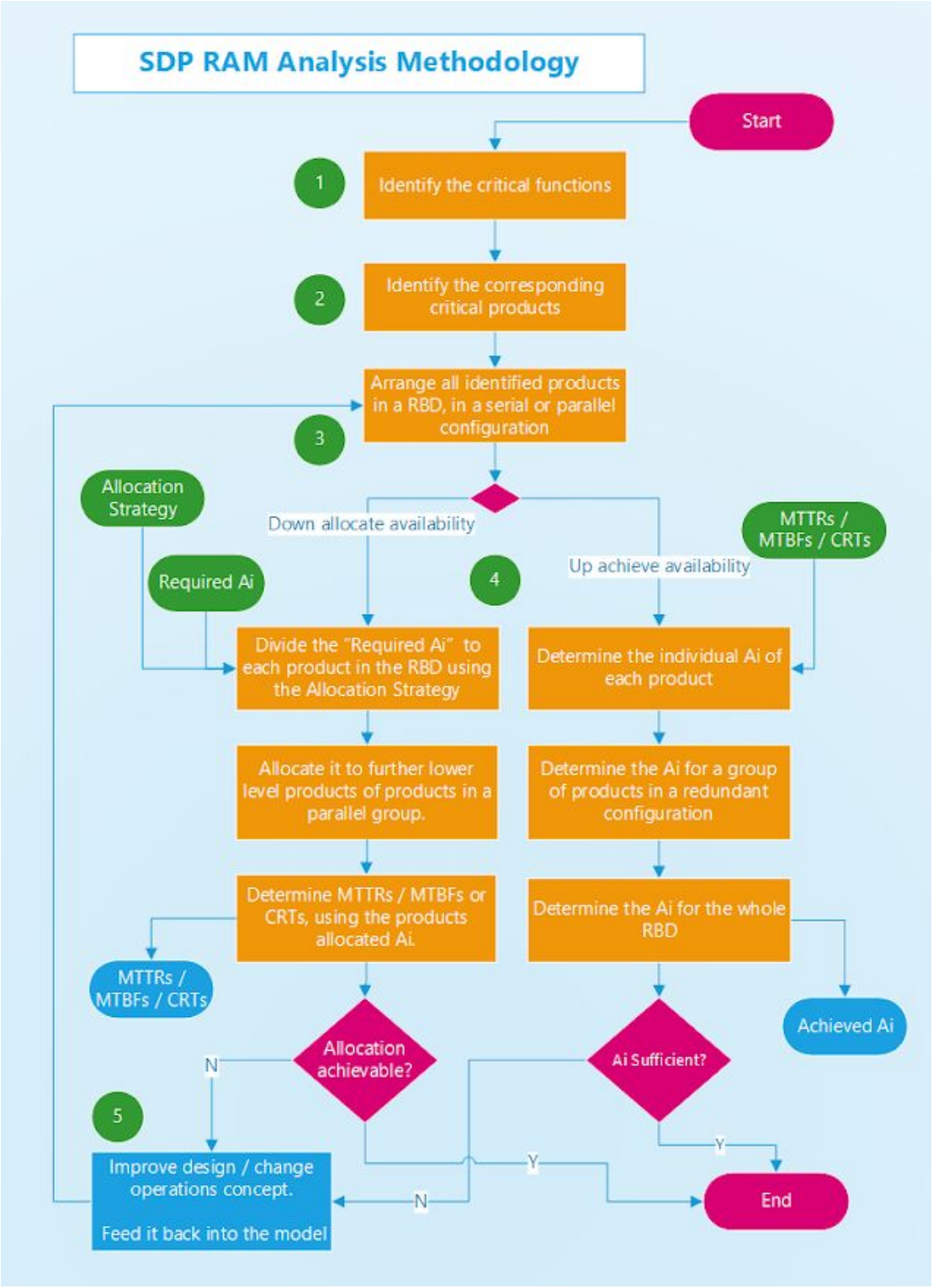

Figure 1: SDP RAM analysis methodology

3.1. RBD

A Reliability Block Diagram (RBD) is a graphical representation of the reliability characteristics of elements in a system. It defines the critical path for the system to function. If any component is in a failed state, it would mean that one or more paths are broken. Therefore in a serial configuration, each or any element can bring about system failure. In a parallel configuration there is some redundancy, as the path can continue through another element.

3.2. Equations

Equations exist for the various serial and parallel configurations [RD6,RD7]:

- Inherent Availability (Ai) using MTTR and MTBF

$$Ai = \frac{MTBF}{MTBF+MTTR} \quad \text{[Equation 1]}$$

- Inherent Availability (Ai) using Critical Repair Time (CRT), Observing (OT), Standby (ST) (not used)

$$Ai = \frac{OT+ST}{OT+ST+CRT} \quad \text{[Equation 2]}$$

- Operational Availability (Ao) using Observing (OT), Standby (ST), Engineering Maintenance and Critical Maintenance (CMT) (not used)

$$Ao = \frac{OT+ST}{OT+ST+MT+CMT} \quad \text{[Equation 3]}$$

- Availability in serial configuration

$$Ai_{tot} = \prod_{n=1}^{N} Ai_n \quad \text{[Equation 4]}$$

- Availability in parallel configuration, equal individual Ai

$$Ai_{tot} = 1 - \prod_{n=1}^{N} (1 - Ai_n) \quad \text{[Equation 5]}$$

- Availability with K out of N redundancy, equal individual Ai (probability of k or more success out of n trials)

$$Ai_{tot}(k,N,Ai) = \sum_{r=k}^{N} \binom{N}{r} Ai^r (1-Ai)^{N-r} \quad \text{[Equation 6]}$$

N is the total number of elements in parallel.
k is the minimum number of elements required for successful operation of the system.
N-k therefore translates to the number of redundant elements / spare capacity.

3.3. Ai Allocation Strategies

Various strategies can be used to determine the Ai of elements in the RBD. Some examples:

- Equal allocation to all elements in the RBD
- Estimate achieved Ai of known elements
- Minimised allocation for some elements
- Maximised allocation for some elements
- A combination of the ones listed above

These strategies can be used to assess the impact of changes on the total Ai, or to perform a sensitivity analysis to assess the impact on a specific product.

Refer to Section 4 for more details on how Ai is interpreted for SDP.

3.3.1. Chosen Allocation Strategy

For hardware it is possible to estimate an Achieved Ai, as failure rates and MTBF data are available. Therefore in 5.4. Step 4: Modelling the Ai, the Estimated Achieved Ai is calculated for all the hardware products using the assumed input variables in Table 8.

It is difficult to estimate the reliability of software modules, in particular if the software has not yet been implemented. Therefore Ai is allocated to software modules, after taking into account the estimated achieved Ai of the hardware products in the RBD. The remaining Ai is allocated to software modules, according their criticality (section 5.5.4. Software Components).

As the SDP architecture and design matures and evolves or new information becomes available, the allocation strategy will need to be re-evaluated and optimised.

3.4. Software Reliability

RAM analyses are typically applied on hardware. It is important to note that the goal with including the software in the RAM analysis, is to understand the impact the availability requirements have on the software in terms of an estimate failure rate and failure recovery times. It is not an in depth analysis of all software modules. By stating that some are more critical than others, one simply gets a grasp on ranges of failure rates. The allocation method could vary with the course of software development, changing the corresponding percentages allocated to that component. The software part of this analysis however provides us with an general understanding of failure rate and recovery times required within the total SDP context. More in depth Software Analysis could be done with Quality Attribute Scenarios and Failure Mode Analysis 5.6. Step 5: Follow-up action.

4. SDP RAM Context

4.1. L1 RAM Requirements

The following L1 RAM requirements [AD3] are allocated to SDP[2]:

Table 1: L1 Requirements

REQ ID	Name	Description
SKA1-SYS_REQ-3245	Inherent availability	The SKA1_Mid[3] shall have an Inherent Availability of more than 99%
SKA1-SYS_REQ-2716	Operational availability	The SKA1_Mid and SKA1_Low shall each have an operational availability of at least 95%.
SKA1-SYS_REQ-3247	Software updates	SKA1_Low and SKA1_Mid equipment shall facilitate updates of major software updates within the system availability allocations.
SKA1-SYS_REQ-3276	SKA1_Low maintenance hours	The SKA1_Low shall require less than 1600 Direct Maintenance Hours per month.
SKA1-SYS_REQ-3246	SKA1_Mid maintenance hours	The SKA1_Mid shall require less than 1600 Direct Maintenance Hours per month.
SKA1-SYS_REQ-3249	Testability	SKA1_Low and SKA1_Mid shall each test (for), detect, isolate and report failures to the operational and maintenance personnel. SKA1_Mid shall detect more than 99% of all Critical Failures. SKA1_Mid shall isolate and log more than 95% of all failures down to a LRU level.

4.2. SKA RAM Allocation to SDP

Along with the L1 Requirements, the RAM budget allocations performed at Telescope level [AD1], provides further context for this RAM analysis. Table 2 summarises the current allocation to SDP.

A direct Ai allocation to SDP is provided. There is no direct Ao allocation to SDP, but SDP's Maintenance Down Time (Table 2), together with all downtimes of other subsystems is constraint by the Telescope Ao of 95% (Equation 3).

[2] Other maintainability L1 requirements relating to maintenance design & installation, are addressed in ILS Document [RD3]

[3] This requirement is intended for Low as well. The text should be updated by SKAO

The work performed for this report, feeds back into the Telescope Level RAM analysis and RBD done by SKAO.

Table 2: SDP RAM Allocation

Attribute	Value	L2 Requirement & Comment
Ai	99.9 %	SDP_REQ-762: SDP Inherent Availability (Ai) The SDP shall have an Inherent Availability (Ai) higher than or equal to 99.9%.
CRT	2 hours	CRT is equal to the MTTR in the Ai equations. That MTTR therefore excludes all possible logistic support delays and therefore assumes that staff and spares is at hand. This report shows (section 5.4. Step 4: Modelling the Ai) that a CRT of 2 hours can be met for the hardware. For the Software, the total failures per year and recovery times shown in Table 14 however shows that our current model supports a Critical Repair Time of 8 hours.
Maintenance Down Time	2 hours	The System shall be designed not to require software and hardware maintenance down time, in excess of 2 hours per year (during steady state operations). L2 requirements for this allocation: **SDP_REQ-759**: SDP Software update downtime The SDP shall not require the telescope to be offline for more than 2 hours per year (during steady state operations) while performing software updates. Rationale: There is currently no planned maintenance downtime for the telescope. Software updates that require downtime, therefore needs to fit within the allocated 2 hours per year [AD1]. **SDP_REQ-874**: SDP shall not require the telescope to be offline while performing preventative maintenance on hardware. Rationale: Maintenance Down Time constraint of 2 hours per year placed on SDP hardware and Software.

4.3. L2 RAM Requirements

L2 Requirements were derived from L1 Requirements and RAM budget allocations as discussed in the sections above. The following L2 RAM requirements [AD2] are included in our current baseline. These requirements are listed below, but are discussed at the relevant places in the document:

- SDP_REQ-762: SDP Inherent Availability (Ai) - SKA RAM Allocation to SDP
- SDP_REQ-763: SDP Critical failure identification - Step 5: Follow-up action
- SDP_REQ-764: SDP Isolation of critical failures - Step 5: Follow-up action
- SDP_REQ-822: Node failures recovery - Step 5: Follow-up action

- SDP_REQ-821: Failure detection to Achieve Ai - 4.6. Step 5: Recommended action
- SDP_REQ-823: Failure Prevention - Step 5: Follow-up action
- SDP_REQ-825: Monitoring to prevent critical failures - Step 5: Follow-up action
- SDP_REQ-4: SDP Resource Reporting - SDP RAM Analysis
- SDP_REQ-810: Maintainability of Software -Step 5: Follow-up action
- SDP_REQ-814: Level of Monitoring - Step 5: Follow-up action
- SDP_REQ-811: Usability of SDP hardware - Step 5: Follow-up action
- SDP_REQ-759: SDP Software update downtime - SKA RAM Allocation to SDP
- SDP_REQ-874: SDP Hardware Maintenance Down Time - SKA RAM Allocation to SDP
- SDP_REQ-739: Pipeline maintenance usability - not addressed in this document
- SDP_REQ-818: Software Reboot Time - Software Components
- SDP_REQ-819: Software Maximum Allowable Recovery Time - Software Components

5. SDP RAM Analysis

> "Aperture synthesis telescopes are inherently robust. Their failure modes are not critical, and a large fraction of the telescope can be unavailable, yet the telescope is still capable of performing valuable observations. "

The quotation above from [AD1] aimed at Telescope Availability, holds true for the SDP, where large portions of the processing capability may be unavailable and yet the SDP can be available to the Telescope. Therefore it is important to understand the following axioms for SDP unavailability:

- SDP is considered unavailable, when SDP is unavailable to the telescope for the scheduled observation.

- SDP may have several components in a failed state, while still being available.

- An important distinction exists between SDP Availability[4] and Resource Availability[5].

  Consider these 2 scenarios:

  1. SDP has resources available, but there is a critical failure resulting in the loss of a scheduled observation.
  2. SDP has no resources available for Telescope Manager to schedule observations, yet SDP itself isn't experiencing any critical failures.

  The SDP hardware will be refreshed and upgraded regularly. When hardware changes the science capacity may change, but the definition of SDP availability and the model should still apply, just scaled for the new hardware numbers (N).

  The following[6] SDP resource pools are exposed to the Telescope Manager [SDP_REQ-4] [AD5]

  - Real-time processing Compute Capacity
  - Batch processing Compute Capacity
  - Buffer Storage Capacity
  - Long Term Storage Capacity

  Theses resources allocated to Processing Blocks are abstracted from the hardware. Failure handling of hardware, is therefore included in that abstraction to prevent handling of failed hardware resource impacting resources allocated to Processing Blocks. The detail of resource reporting is explained in [AD5].

  For these reasons, the topic of resource availability and scheduling should be seen as outside the scope of the RAM Analysis. The current concept for the SDP's response to failure vs lack of resources also differs. Failures, as discussed in this document, result in a change of SDP or

[4] A function of reliability and the means to restore SDP science capacity when failed
[5] Science capacity, a function of use
[6] note for future reference that this list is extensible

SDP sub-level state, while lack of resources results in an alarm [Operational System C&C View, AD4].

- While the SDP is available, some failures may result in degraded science output. The telescope is busy observing, but the science output is of an unacceptable quality. Science time could be lost, so by definition it leads to unavailability. Telescope Manager is however informed on quality metrics of an observation, and subsequently could cancel the observation. Reporting on quality metrics is therefore essential for reliability and availability. An unavailable component relating to Quality Assessment would however not cause the observation to be aborted immediately. This is elaborated on in Section 5.3. Step 3: Reliability Block Diagram.

The steps as described in section 3. RAM analysis methodology, were followed and discussed in the sections below.

5.1. Step 1: Failure Analysis

In order to have critical failures, it is necessary to understand what primary functions are required for a given operation. The two primary functions required of the SDP are:

1. Receive
2. Real-Time Processing

Naturally, a total failure of the SDP to provide a platform or execute control would lead to both of these functions failing. Buffer failure would also lead to the failure of these two functions.

5.2. Step 2: Critical Components

The Operational System C&C View [AD4], describes the SDP architecturally significant functionality in a run-time component view. This view is suited to analysing the availability characteristics of the architectural design. These run-time components are implemented through software modules as documented in the SDP Architecture Overview Document [AD4]. The hardware analysis is done on hardware products, but for the software analysis, it proved more valuable at this point in time to keep the analysis on the components level. The rationale for that follows:

- At the time of writing this report, the SDP design focus was on concluding the SDP Architecture for CDR.
- Components show more complex interactions, and therefore failures w.r.t those interactions can be better understood through analysing it on component level. These failures are more relevant at this point in time than for example failures in line of code (modules).
- This also supports the application of reliability tactics relevant to architecture to be employed e.g. decoupling of subcomponents [Operational System C&C View, AD4] etc.
- In understanding criticality as defined in the section above, modules may serve to be a tricky concept. Components are implemented through modules, but a module may have critical elements and non-critical elements. The same module may even be critical to some modules in some instance, while not critical to others.

- More emphases could be placed on failure analysis of modules in later phases when code construction is taking place.

Figure 2 below shows the critical run-time components of the SDP:

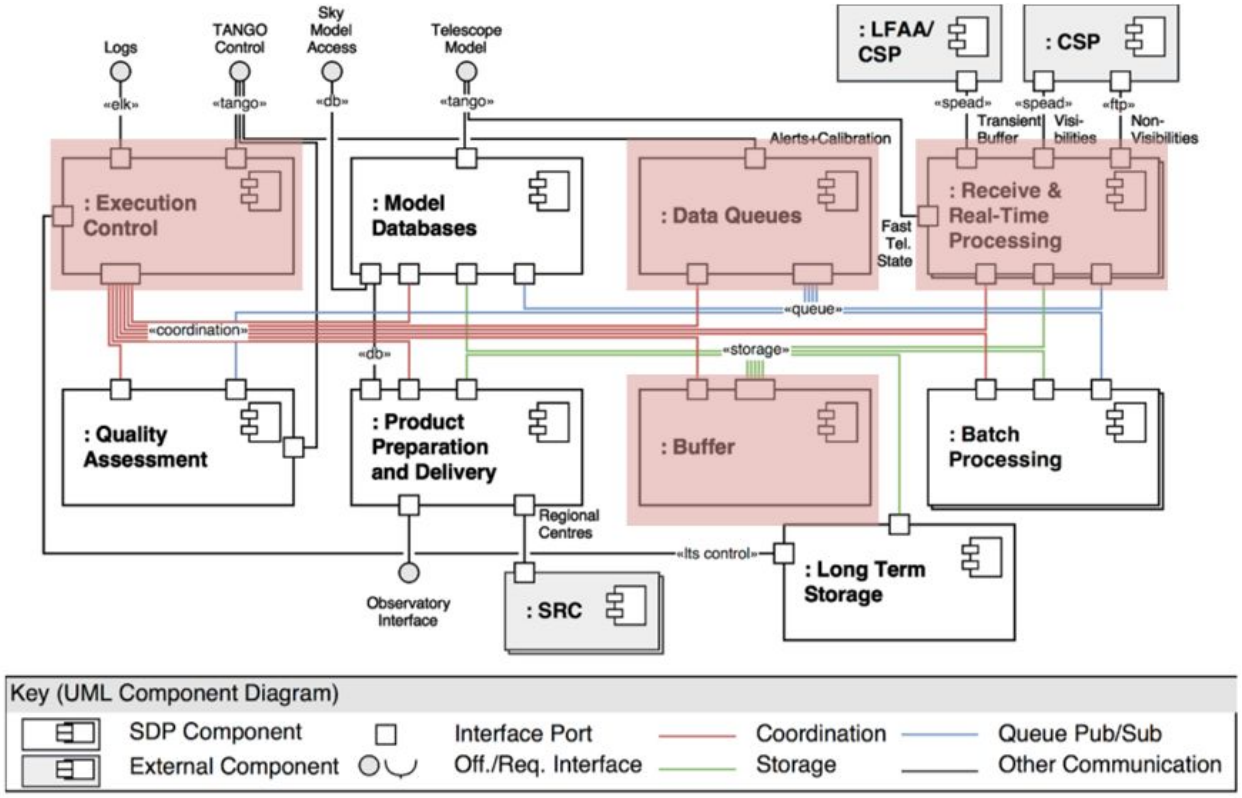

Figure 2: Most critical components of the SDP Operational System C&C View [AD4] are highlighted in red.

The following run-time components are most critical for SDP availability:

- Execution Control
- Data Queues,
- Receive & Real-time processing
- Buffer
- Platform[7]

Note: Other components have critical failures that are conditional. These are discussed in section 5.3.1.

The hardware products required for these components are depicted in Table 3.

[7] Shown in Execution Control presentation in the SDP Operational System C&C view [RD05].

Table 3: Component to Product Mapping

Hardware Products	Execution Control	Data Queues	Receive & Real-Time Processing	Buffer
Compute Rack (service Nodes)	x	x		
Compute Rack			x	x
Management Network	x	x	x	x
High throughput Ethernet Network			x	x
Low Latency Network		x	x	x

5.3. Step 3: Reliability Block Diagram

The products and components can now be arranged as elements in a Reliability Block Diagram (RBD), as shown in Figure 3 below. In Table 4, each RBD element is explained along with its availability strategy.

Intra-Rack Infrastructure and Rack Infrastructure, cabling and power distribution are assumed as Ai = 1 and not further developed under RBD elements.

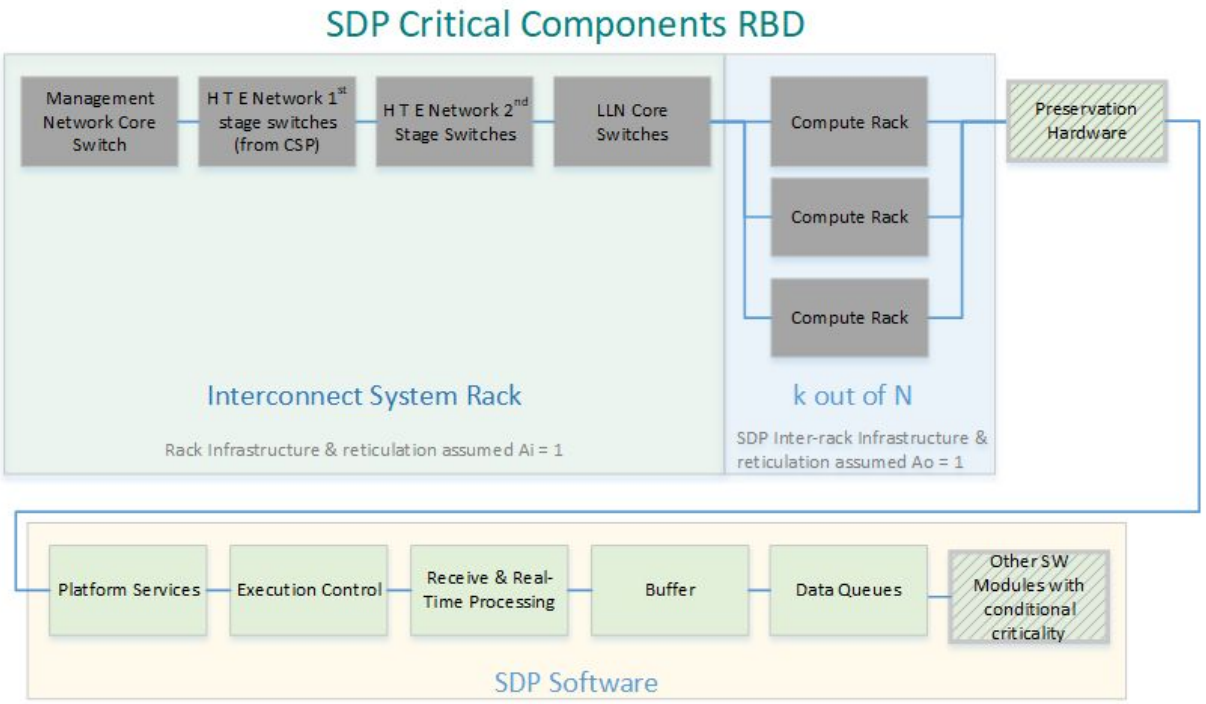

Figure 3: SDP RBD

Table 5: RBD Elements

HW Products & SW Components	Description & Availability Strategy
Management Network Core Switch	Inside the Interconnect System Rack[8], is the Management Network Core Switch. All Compute Racks are connected to the Management Network through this switch. The Top of Rack Switches, are included in the Compute Rack's Ai. The Management Network Core Switch has a redundant topology.
1st stage Switches High Throughput Ethernet Network to CSP	Each switch is a single point failure for those fixed lines from CSP. No rerouting is possible. Incoming visibility data carried by the associated SaDT links from CSP will be lost until repair. All switches are therefore in series leading to unavailability.
2nd Stage Switches for the High Throughput Ethernet Network	These switches are housed inside the Interconnect System Rack, and connect several Compute Racks to the High Throughput Ethernet Network. The Top of Rack Switches, are included in the Compute Rack's Ai. Upon failure of these switches, rerouting can take place. The network will still be available, but with reduced bandwidth.

[8] The Rack which houses all the core switches, see PBS in Hardware Decomposition View [AD4]

Low Latency Network Core Switches	These switches are housed inside the Interconnect System Rack, and connect several Compute Racks to the Low Latency Network. The Top of Rack Switches, are included in the Compute Rack's Ai. Upon failure of these switches, rerouting can take place. The network will still be available, but with reduced bandwidth.
Compute Rack	An important architectural decision [AD4] was made that different types of nodes will be differentiated on a platform level and not on hardware level. In other words, hardware can be dedicated by Platform Service (discussed on the following page) to become hardware available for a certain type of resource pool. However, at the hardware level, hardware is still procured [RD2] to be optimised for certain functions. A Compute Rack includes nodes with one or more, of 4 specific capabilities: ● Nodes optimised for low latency ● Nodes optimised for high throughput (e.g. containing GPUs) ● Service Nodes ● Storage Nodes This architecture supports a K out of N redundancy (Equation 6) for the nodes (servers), which means all nodes aren't in parallel, but SDP will still be available if some of the nodes (N-K) fail. (see section 5.5.4 for detail on Hot Spare Compute Rack calculations). The Compute Rack also includes other components such as the HTE and LLN switches. See Figure 4 for more detail.
Preservation Hardware	The Preservation Hardware is anticipated to be a COTS platform sourced separately. The Hierarchical Storage Management will provide an interface to Cold Buffer and manage the intermediate and long term storage of the Preservation Hardware which could consist of MAID and Tape. The longevity of data products on these media will be dictated by access and policy rules. [AD4]
Platform Services	Platform Services is required for application level software to execute. Platform services is responsible for presenting shared services to the wider SDP software stack. For example, the Core Infrastructure Services component presents the SDP hardware as a Software Defined Datacenter, i.e. control the compute, storage and networking hardware delivering the expected System Services as required [Platform C&C View, AD4]. Unavailability of Platform Services would result in SDP to be unavailable as defined in this analysis. This component is therefore a single point failure for the SDP and requires high availability.
Execution Control	Execution Control is expected to become active once the platform has minimal resources available to provide the necessary compute and database infrastructure.

	Although high levels of decoupling are achieved in the general architecture, the SDP Component State Dependencies diagram in the Operational System C&C View [AD4] shows that the only runtime component whose failed state would result in unavailability of all other critical components is Execution Control. Execution Control (component) is a potential Single Point Failure. This component so be designed for a higher Ai. Current architectural strategies employed include hierarchically splitting up control between different controller components, so reducing the complexity that every single controller component has to handle, limiting the extent of failures. Given current architectural solutions, the Configuration Database will remain a critical point of failure, which could result in entire system shut down. High availability off-the shelf databases are available for the Configuration Database. [Operational System C&C View, AD4]
Receive and Real-time processing	The Receive Component handles data from the Central Signal Processor and the Low Frequency Aperture Array while the observation is running. Multiple instances of Receive might be active at the same time in case multiple types of observations are running and/or the telescope is split into sub-arrays. Handling the received data consists of: 1. Writing received and possibly pre-processed measurement data to the Buffer for later Batch Processing 2. Feeding it directly to Real-Time Processing, such as a fast imaging or real-time calibration solving Science Pipeline Workflows. Real-time results of such processing pipelines are pushed out via Data Queues and might lead to e.g. alerts or calibration solutions getting published back to the Telescope Manager. [Operational System C&C View, AD4]
Buffer	The Buffer stores and makes available primary inputs and outputs of processing using a file system interface. Storage will consist of multiple tiers (such as a “Hot” and a ”Cold” Buffer storage tier) to provide storage at different sizing and performance requirements. The Buffer will have the capability to handle the data life cycle of stored objects [Operational System C&C View, AD4]. The architectural view on the buffer considers that it would comprise of a number of namespaces supporting different quality attributes of the data access patterns, performance and reliability. This implies that there will not be a single unified “filesystem” for example but a number of separate instances.

Data Queues	The Data Queues component handles medium-rate real-time information such as calibration solutions, alerts or Quality Assessment data exchanged between model databases, processing and Quality Assessment. The Data Queues component also supports sharing of intermediate processing results such as global calibration solutions between Execution Engine instances [Operational System C&C View, AD4].
Long Term Storage[9]*	Long Term Storage is used for storing Buffer data that is marked for long-term storage by the Storage Lifecycle Policy, i.e. data products. This will only happen after the associated Processing Blocks have finished. [AD4]
Product Preparation & Delivery*	The Delivery component is responsible for maintaining the Science Data Product Catalogue and for distributing Data Products to the SKA Regional Centres (SRCs). [AD4]
Batch Processing*	Batch processing runs the most demanding Science Pipeline Workflows of the SDP, both in terms of computational and scientific complexity. Multiple instances of Batch Processing can execute at the same time, thus providing an easy mechanism for scaling this part of processing. These instances can coordinate loosely by communicating via Data Queues (e.g. to exchange calibration solutions). [AD4]
Model Databases*	This component is responsible for creating Science Data Models as a Buffer object to be used in processing - and feeding back updates after processing has finished. [RD04]
Quality Assessment*	Telescope Operators monitor the real-time quality metrics fed from SDP toTM. SDP workflows determine quality metrics. The Quality Assessment component aggregates information generated at runtime by both Real-time and Batch Processing to provide an early assessment of produced science data. This will involve aggregating data published by Science Pipeline Workflows using Data Queues, as well as some analysis to make the information usable to telescope operators. [AD4]

5.3.1. Elements with conditional critical failures

It was debatable whether these items should be included in the RBD at all, as their failures do not immediately result into SDP unavailability (as defined in 5. SDP RAM Analysis). Failures of these elements could however become critical failures if certain conditions occur or enough time passes without intervention. Thinking operationally, the failures of these elements is recoverable before such conditions for critical failures would occur. The detection and compensatory measures for failure recovery is however not with the scope of this report, and therefore to keep it clear in terms of traceability of criticality, these elements are included in the RBD.

[9] * Elements with conditional critical failures

It was therefore important to clearly state the assumptions made of possible compensatory measures and their contribution to the RBD in terms of Ai. Improved assumptions could surface after further failure mode analyses (5.6. Step 5: Follow-up action), which would then allow the model (see spreadsheet submitted as part of this document), to reflect the impact of those changed assumptions. The Table below states the assumptions made:

Table 6: Conditional Critical Failures

Failure	**Impact on SDP**	**Impact on Telescope**	**Assumption in model**
Failures of the Preservation Hardware and components Long Term Storage & Product Preparation & Delivery	Buffer capacity is reduced as items are not placed onto Long Term Storage.	It is most likely that components are recovered before SDP becomes unavailable to Telescope. Observations therefore go on as scheduled.	MTTR = 0, resulting in Ai = 1
Critical failure of component Batch Processing	Multiple instances of Batch Processing is executing at the same time. A critical Failure in Batch Processing would only influence one Processing Block at a time. That PB would be terminated and restarted.	The delay would not contribute toward the Buffer becoming full, resulting in the SDP unavailable. Failure probability of this component should be included in the Resource Scheduling Model.	Not in model
Quality Assessment Component fails	SDP cannot send real-time aggregated quality metrics to the telescope operator.	The Assumption is made that the operator will not stop an observations due to quality metrics not available for a period between 10 min (most likely) and 8 hours (worst case scenario). If this assumption is incorrect it can be updated in the model.	MTTR = 0, resulting in Ai = 1.
Critical failure of Model Databases at the beginning of a scheduling block.	Since Model Databases are only required when creating processing blocks, the probability of of this component failing	The real-time processing blocks cannot be started, which will delay the Scheduling Block until recovery. Other processing continues.	MTTR = 10 min (reboot) Failure rate - see section 5.5.5

	in that time is much lower.		

For the purposes of the report, the values where MTTR is assumed to be 0, are not included in the Tables below, as there is no significance in them or their effect on the total Ai.

5.4. Step 4: Modelling the Ai

Using the logic of the RBD and the underlying equations (section 3.2. Equations), a model was developed (see spreadsheet submitted as part of this document). The idea of the model is that any variables (e.g. MTTR, MTBF, Ai allocated, N, K etc) can be changed to assess the impact on Ai or another variable under investigation. The model was built in a spreadsheet.

The spreadsheet submitted as part of this document reflects the values for SKA1_Mid and SKA1_Low (which are very similar). Only SKA1_Mid is discussed in this report.

5.4.1. List of assumptions for the Model

- Calculations were done in double precision and shown in 8 digits (0.99999999) (TBC)
- The Input Variable (green) MTBF are assumptions based on industry expert opinions
- Spares for Network components are currently estimated at 10% vs 1% for Compute Rack components [RD2].
- Rerouting of all switches is possible, except for the 1st stage High Throughput Ethernet Switches (product 2) which are Single Point Failures.
- For network switches with K out of N redundancy (Equation 6), spare capacity (N-K), was assumed to be 1 (further discussion in 5.5.1. Network Core Switches).
- N, the total number of Network Core switches (product 1, 4) is taken from the current Cost Model [RD2] for the LLN. The 1:1 oversubscriptions values are used for switches.

Table 7: Key for Table 8, Table 9, Table 10 & Table 14

Input Variable
Sliding Variable
Output Variable
Selected option in **bold**

5.4.2. MTTR in Ai

It is important to note that MTTR in the Equation 1 used for calculating Ai, relate to the Critical Repair Time. Therefore for that MTTR all logistic support is excluded. It can assume staff and spares as well as all support equipment is at hand. It therefore only required the faulty part to be identified and replaced.

5.4.3. The Model

Table 8: Selected Allocation Strategy with values (hardware)

#	Product	Allocation Strategy	Ai	MTBF (hours)	MTTR [10] (hours)	N[11]	k[12]	N-k[13]
1	Management Network Core	Estimated Achieved, Equation 1&5	0.99999999	200000	2	8	7	1
2	1st stage HT Ethernet Network (CSP)	Estimated Achieved, Equation 1	0.9999900	200000	2			
3	2nd stage HT Ethernet Network	Estimated Achieved, Equation 1&5	0.99999999	200000	2	12	11	1
4	Low-latency network core switches	Estimated Achieved, Equation 1&5	0.99999999	200000	2	8	7	1
5	Compute Racks	Estimated Achieved with 1 Hot Spare per Telescope, Equation 1, 4, 5	0.99999841	Table 10				
6	Software	Remaining Ai, Equation 4	0.99901159	Table 14				
	SDP Total Ai		0.999					

5.5. Product Discussion

5.5.1. Network Core Switches

Network Core Switches as described in Product 1, 3 & 4, have a similar topology. Although N is different for each of these networks and also Mid and Low Telescopes, what we learned from the model for these 3 sets of network switches are similar, as described below:

[10] See definition in "MTTR in Ai", excludes all Logistical support aspects
[11] N, total number of units, for k out of N redundancy
[12] k, number of units that must be available, for k out of N redundancy
[13] N-k, number of redundant units, for k out of N redundancy

- **High Availability** | Given the topology of these networks, they are inherently highly available. Even with very low numbers of K, the network will be available. Therefore for our analysis K rather relates to an acceptable bandwidth.
- **Over-design** | Cost trade-off is always a driver. As shown in (Table 9), care should be taken that networks are not over-designed. If (N-k) > 1 , i.e. the ability to have more than 2 switches in a failed state or repair it in a critical time, would result in higher cost, but would make no difference to improve availability. The model could further assist in analysing such trade-offs.
- The impact on the Logistic Support is not as critical here. A relaxed MTTR for Ao (72-168 hours), could be traded-off with a higher (N-k) & sufficient bandwidth.

Table 9: Sliding Variables for Network Switches

#	Product	Allocation Strategy	Ai	MTBF (hours)	MTTR[14] (hours)	N[15]	k[16]	N-k[17]
3	2nd stage HT Ethernet Network	Estimated	0.99999999	200000	2	12	11	1
		Achieved Equation 1&4	0.99999999				10	2
4	Low-latency network core switches	Estimated	0.99999999	200000	2	8	7	1
		Achieved Equation 1&4	0.99999999				6	2

5.5.2. Stage 1 High Throughput Ethernet Network Switches

Stage 1 High Throughput Ethernet Network Switches (Product 2) is different from the other switch configurations, in that no rerouting can take place. For the period until a switch has been repaired, the data carried by the associated SaDT links from CSP will be lost, and hence the SDP would be unavailable for that observation. Although the theoretical Ai is met in the current analysis, improvements in this area would have a positive impact on total availability.

This item could receive more attention in the LSA. It would make sense for MTTR (contributing to Ao) to be a driver for the availability of these switches. The feasibility of a lower MTTR and sufficient spares is more important for this product than for the other network products.

5.5.3. Compute Racks

Compute Racks[18] (Product 5) are in a k out of N Redundancy (Equation 6), which means that k Compute Racks must be online to meet the demand. K is fixed from the estimated design, but N will then be determined as k + hot spares. To estimate that, we need to have an estimation of an actual

[14] See definition in "MTTR in Ai", excludes all Logistical support aspects
[15] N, total number of units, for k out of N redundancy
[16] k, number of units that must be available, for k out of N redundancy
[17] N-k, number of redundant units, for k out of N redundancy
[18] Previously referred to as Compute Island

individual Compute Rack. That can be estimated by looking at the RBD for a Compute Rack. Figure 4 shows the RBD for a Compute Rack.

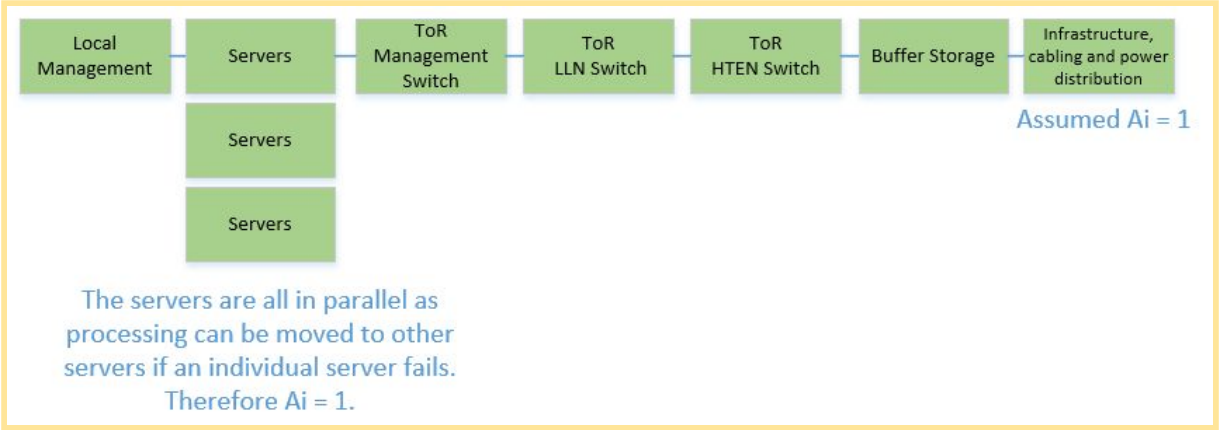

Figure 4: Compute Rack RBD

5.5.3.1. Assumptions and Rationale for Compute Rack RBD:

- There is spare capacity on the Buffer (Compute Rack). Batch processing could be delayed, while only Real-Time Processing continues.
- During processing, upon failure of a node, it is possible to carry on the processing on another node.
- The combination of the Compute Rack's infrastructure, cabling and power distribution can be assumed as Ai = 1. Redundancy exist for Power Supply.
- Computes Nodes (servers) are effectively in parallel, as processing can be allocated to any other node. This will also result in the group Compute Nodes having a Ai = 1.
- The Server element in the RBD include Service Nodes (type of server). It includes 2 Management Islands of each 28 servers. They are in a highly available setup (1:1). According to Equation 5, these service nodes has an availability of Ai = 1.
- Compute Rack ToR Switches are included in the Compute Rack's Ai.
- The MTTR for the Buffer, was an estimate. The hardware implementation is still conceptual. Therefore the general assumed MTBF for this item is left as is. Further analysis for trade-offs can be done using the model.

Table 10: Selected Allocation Strategy with values (Compute Rack)

#	Product	Allocation Strategy	Ai	MTBF (hours)	MTTR[19] (hours)	N[20]	k[21]	N-k[22]
5.1	ToR Management Switch	Equation 1	0.9999900	200000	2	1		
5.2	ToR LLN Switch	Equation 1	0.9999900	200000	2	1		

[19] See definition in "MTTR in Ai", excludes all Logistical support aspects
[20] N, total number of units, for k out of N redundancy
[21] k, minimum number of units that must be available, for k out of N redundancy
[22] N-k, number of redundant units, for k out of N redundancy

5.3	ToR HTEN Switch	Equation 1	0.9999900	200000	2	1		
5.4	Servers	Equation 5	1	100000 250000 [RD5]	2	56		
5.5	Buffer Storage (conceptual)	Equation 1	0.9999600	50000	2			
5.6	Infrastructure, Cabling, power Distribution	Assumed	1					
	Single Compute Rack Ai	Equation 4	0.9999300					
	Compute Racks Total Ai	Equation 6	0.99999841			26	25	1

5.5.4. Hot spares for the Compute Rack

SDP has employed the strategy that the minimum amount of hot spare Compute Racks shall be included while still able to meet the total Ai. The analysis have shown **1 hot spare per telescope** to be sufficient to meet the total Ai of 99.9%.

Table 11: Analysis of number of hot spares

Hot Spares	**Ai of Compute Rack**	**Impact**
0	Ai (MID) = 0.99825155	This Compute Rack Ai is already lower than the required total Ai for SDP. The Software Component would need to have a higher Ai than 1 to still meet the total Ai of 99.9%.
1	Ai (Mid) = 0.99999841	With 1 hot spare, the total SDP Ai of 99.9% can be met. The impact that this Ai have on the budget available for Software Failures is reasonable (Table 14).
2	Ai (Mid) = 0.99999999	With 2 hot spares, there was no improvement on the allowed failures per year for the Software (see Table 14). The cost trade-off would not make sense in this case.

5.5.5. Software Components

Since we have no reliability data for Software, the Ai remaining (RAi) after taking the hardware into consideration was divided between the software components. The remaining Ai was then divided according to the ranking in Table 12 and Equation 7. This weighting scheme was assumed to be a

sensible way to proceed without detailed software module specific information, according to the chosen weight (that can be varied in the spreadsheet submitted as part of this document).

Table 12 list the components in order of criticality and assigns a priority weight to it.

Table 12: Weighting for software allocation

Software module	Rationale for ranking	Ranking (W)
Platform	Without platform no higher level software components can be active.	1
Execution Control	If Execution Control is down, all other components are also unavailable.	1
Receive and real-time processing	This component is required to ingest data and perform time-critical processing.	2
Buffer	This component is required to ingest data for later processing, therefore making it possible to still observe, even though resources are unavailable or other non-critical components are down.	2
Data Queues	Data Queues is required to perform real-time calibration and it does not affect other areas of SDP in terms of Ai.	3
Model Databases	As discussed in section 5.3.1, the failure rates associated with this component would only be critical during a small time window. Therefore for the modeling purposes, a weighting of a lower order was assigned.	0.1
Total number (N)		9.1

$$Allocated\ Ai\ =\ RAi^{\frac{W}{N}} \quad \text{[Equation 7]}$$

W is the weighting according to Table 12
N is the sum of all the weightings
RAi is the remaining Ai budget to be allocated to software

$$RAi\ =\ \frac{0.999}{\prod\ Hardware\ Ai}$$

The use of the software analysis is that for a given allocated Ai, a failure rate can be obtained. This allocation may change throughout design and construction. This initial estimate is just to aid in understanding the required software availability. Allowable failure rates (per year) go hand in hand

with the selected MTTR (Equation 1). For analysis purposes and for discussion, the following initial 3 failure recover types and repair times have been defined. These 3 failure recover types and repair times have all been modelled (Table 14) for understanding the software's availability requirement.

Table 13: Types of repair times for software

Failure recovery type (SW)	**Description of recovery type**	**Repair time**
Reboot / Restart	Automated failure recovery where the node is rebooted or application software is restarted by Platform Services, e.g. when a node or software becomes unresponsive.	10 min (TBC)
Rollback or other automated recovery mechanism	Automated rollback to a previously known working version (deployment) of software.	2 hours (TBC)
Manual intervention	Manual intervention is required to roll back software to a known working state or implement a workaround to prevent the failure re-occurring while cause of the failure is resolved.	8 hours (TBC)

Table 14: Selected Allocation Strategy with values (software)

#	**Component**	Allocation Strategy	Weighting (Table 12)	Ai	MTBF (hours)	MTTR[23] (hours)	Critical Failures (per year)
6	Platform Services	Allocated with weighting, Equation 1	1	0.99989134	1564	10 min	5.6
					18403	2 h	0.5
					73613	8 h	0.1
7	Execution Control	Allocated with weighting, Equation 1	1	0.99989134	1564	10 min	5.6
					18403	2 h	0.5
					73613	8 h	0.1
8	Receive & Real-time processing	Allocated with weighting Equation 1	2	0.99978268	782	10 min	11.2
					9201	2 h	1.0
					36805	8 h	0.2
9	Buffer	Allocated with Equation 1	2	0.99978268	782	10 min	11.2
					9201	2 h	1.0
					36805	8 h	0.2

[23] See definition in "MTTR in Ai", excludes all Logistical support aspects

10	Data Queues	Allocated with Equation 1	3	0.99967404	521	10 min	16.8
					6134	2 h	1.4
					24535	8 h	0.4
11	Model Databases	Allocated with Equation 1	0.1	0.99998913	15644	10 min	0.6
					184042	2 h	0.0
					736169	8 h	0.0
Total Software		Reboot/Restart				10 min	51
		Rollback				2h	4.3
		Manual intervention				8h	1.1

Note: It is important to emphasise that the figures in Table 14 relate to critical failures which result in SDP not being available, i.e. 51 restarts per year for critical failures and not 51 restarts per year in total. The restarts of non-critical failures shall not require down time of the SDP. From Table 14 it is clear that the majority of software failures should be recoverable by the first failure recovery type (rebooting / restarting).

L2 requirements have been generated for the software recovery times. These are estimates and do not trace from L1 requirements.

Table 15: L2 requirements for software recovery times

REQ ID	Name	Description
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.
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.

Further discussion on ongoing work for software in section 5.6. Step 5: Follow-up action.

These run-time components are implemented through software modules as shown in [AD4].

6. Step 5: Follow-up action & Recommendations

Recommendations following this report, need to be considered within the context of the follow up work. The first step of the follow on work shall include an expansion of the Failure Modes and Effects Analysis (FMEA).

The FMEA can then be used to evaluate the following recommendations.

Hardware:

- Ai = 1 for elements in the Interconnect System can easily be achieved. Care should however be taken in not over designing this element from an availability point of view. Confirm what is a sufficient N-K for the Interconnect System Rack Switches [drawing in Hardware Decomposition View, AD4], w.r.t acceptable bandwidth.
- The Single Point Failures in the High Throughput Ethernet Network could be considered for improvement, at least by drawing attention to it in the LSA
- As the Compute Rack's buffer design becomes more detailed, this could be analysed in more detail and a margin for acceptable MTBFs considered.

Software:

- Software RAM work to continue, as per ILS [RD3] in the form of failure analysis (e.g. RD4), and Quality Attribute Scenarios (see section 5.6.3. Quality Attribute Scenarios). This includes failure handling mechanisms, failure recovery strategies and monitoring. The following L2 requirements (Table 16), are also applicable.
- With automatic failure handling mechanisms further developed, the related assumptions made w.r.t conditional critical items can be validated. Further constraints / requirements on failure handling can be identified (e.g. the Long Term Storage capability or Batch Processing capability must be restored after failure within TBD hours).
- Software stakeholders to confirm and finalise the software MTTRs.

Operational conditions and operator actions:

- Evaluate operational conditions, preferences, operationally appropriate actions and responses.
- With operator failure handling mechanisms further developed through the FMEA, the related assumptions made w.r.t conditional critical items can be validated e.g. the operator's response to failure of real-time Quality Assessment reporting.

The SDP L2 requirements [AD2] also contain a number of requirements meant to address architectural concerns and failure modes.

Table 16: L2 requirements for failure identification

REQ ID	Name	Description	Comment
SDP_REQ-763	SDP Critical failure identification	The SDP shall identify more than 99% of all critical failures and report them to the TM.	
SDP_REQ-764	SDP Isolation of critical failures	The SDP shall isolate 95% of all critical failures and report it to TM.	
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.	SDP_REQ-763 is about identification of critical failures and communicating it to TM.
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).	Also see QA Scenarios: SDP_REQ-825 SDP_REQ-814 SDP_REQ-822

6.1. Quality Attribute Scenarios

Availability and other quality attributes are addressed through Quality Attribute Scenarios. A Software RAM workshop was held and several Quality Attribute Scenarios were developed. A framework for classification of failures was also used. SDP Software Failure Options. The following Quality Attributes have been captured as L2 Requirements. These will continue to be used in further design and construction.

- SDP_REQ-822: Node failures recovery
- SDP_REQ-825: Monitoring to prevent critical failures
- SDP_REQ-811: Usability of SDP hardware
- SDP_REQ-810: Maintainability of Software
- SDP_REQ-814: Level of Monitoring
- SDP_REQ-739: Pipeline maintenance usability