



**SKA1\_LOW TELESCOPE  
INTERFACE CONTROL DOCUMENT  
CSP TO TM**

Document Number .....100-000000-021  
 Document Type..... ICD  
 Revision ..... 01  
 Author .....S. Vrcic  
 Date.....2016-01-19  
 Document Classification .....FOR PROJECT USE ONLY  
 Status ..... Released

Name	Designation	Affiliation	Signature	
Owned by:				
T. J. Stevenson	Chief System Engineer	SKA Office		
			Date:	
Approved by:				
M. Caiazzo	CSP Element System Engineer	SKAO		
			Date:	
J. Santander-Vela	TM Element System Engineer	SKAO		
			Date:	
R. Olguin	SaDT Element System Engineer TM	SKAO		
			Date:	

S. Vrcic	CSP Consortium System Engineer	CSP Consortium		
			Date:	
P. Swart	TM Consortium System Engineer	TM Consortium		
			Date:	
R. Gabrielczyk	SaDT Consortium System Engineer	SaDT Consortium		
			Date:	
Released by:				
A. McPherson	Head of Project	SKAO		
			Date:	

## DOCUMENT HISTORY

Revision	Date Of Issue	Engineering Change Number	Comments
A	2013-07-22	-	First draft release for internal review
B	2013-07-30	-	Updated points of contact
C	2013-08-01	-	Incorporated comments from Paul Swart
D	2013-10-17	-	Updated by S.Vrcic after the SKA Engineering Meeting.
E	2014-04-28	-	Updated by S.Vrcic as per comments by Kobus Cloete. Added requirements specified in the documents LMC Scope and Responsibilities, LMC Guidelines. And CSP specific requirements derived from SKA Level 1 requirements. This version has been released for internal review for CSP TIM#2 but has not been reviewed by TM Consortium and is still work in progress.
F	2014-06-29	-	Updated after TIM#2. Removed interface requirements. Significant additions to Overview and Interface Details. Sent to SKAO and TM Consortium for review.
G	2014-08-27	-	Updated with the comments collected from CSP Consortium at and after TIM#3 and the comments received from Juande Santander-Vela, SKAO. Resources (receptors and beams) can be added/removed to/from sub-array only at scan boundaries or when sub-array is not used for observing (i.e. is in Idle mode). Simplified the list of Administrative Modes and States. Use status is reported independently from Operational State. CSP Test Mode can be turned on/off independently from the Administrative Mode.
H	2014-09-19		Updated based on comments received from M.Caiazzo (SKAO). Sent for review before SKA Engineering Meeting in Perth, Australia (29 Sep. – 02Oct. 2014).
J	2014-10-29	-	Updated based on the discussion at the SKA Eng. Meeting in Perth, Australia., and comments received from Michael Rupen (NRC), Lize van den Heever (SA) and Paul Swart (SA). Added Pulsar Search related information provided by Carlo Baffa and Elisabetta Giani (INAF).  Most notable changes: TM shall specify Activation Time for configuration changes and commands. TM shall provide delay model polynomials.  75% draft for review. Merged health status reporting into Operational State (added operational state degraded).
K	2014-11-12		Updated as per comments by B.Carlson, NRC (editorial changes, added ISO/OSI diagram Figure 1-2, when delay models and other real-time updates are not received CSP flags data or stops generation of output products) and P.Swart (SKA SA) and A.Bridger (STFC) – CSP makes provision for TM to create a sub-array with zero receptors (see Section 7.7). Added SaDT System Engineer to ‘approved by’ list on the front page. Sent to SKAO for review.
1	2015-02-16	CSP-PDR-OAR-081	081- Updated the list of acronyms.
		CSP-PDR-OAR-082	Added section ‘extremely high message rate’ to Section 7.2.

Revision	Date Of Issue	Engineering Change Number	Comments
		CSP-PDR-OAR-083	No limit on the number of parameters in a 'SET Parameters' message at this time (Table 7-21 and Section 7.6.1).
		CSP-PDR-OAR-085	Section 7.7 added clarification re. Activation Time - to execute more than one message at the same time transmit messages in advance and specify the same Activation Time.
		CSP-PDR-OAR-086	Correct standard is: ISO 8601
		CSP-PDR-OAR-087	Requirements for precision when specifying time still TBD.
		CSP-PDR-OAR-088	Updated section 7.3 Extensibility and future compatibility to emphasize that CSP supports a single version of the protocol at a time.
		CSP-PDR-OAR-089 CSP-PDR-OAR-091	Updated Section 7.6.7 and 7.6.8. The mechanism will be defined when the protocol/framework is selected – need to decide whether the same or different data stream is used for file transmission.
		CSP-PDR-OAR-090	Table 7-18 There can be multiple instances of software per LRU (software images, processes, etc.)
		CSP-PDR-OAR-092	Table 7-18 updated to refer to Figure 7-12 Observing Mode Change – Time Sequence Diagram regarding parameters that require regular updates (for delay tracking, calibration and beamforming).
			Added to the list of TBD items (Appendix B: To Be Defined) Note: before it was signed this version was circulated as draft L.
1.A	2015-09-17	CSP-PDR-OAR-436	Updated description of modes and states.
			Updated with results of re-baselining. <b>Split material per telescope. This document contains only material that applies for SKA1_Low telescope.</b> Removed references to the CSP sub-element Clock and Timing Distribution (CLK) which is now included in the CSP_Low.CBF. Replaced terms Pulsar Search Non-Imaging Processor (NIP) and Pulsar Timing Non-Imaging Processor (NIP) with Pulsar Search Engine and Pulsar Timing Engine to match terminology used in the SKA Architecture Package released in July 2015.
1.B	2015-12-17	-	Updated parameters for all Observing Modes.
1.C	2016-01-19	-	Updated as per comments received from Paul Swart and Rodrigo Olguin: defined the 2 <sup>nd</sup> interface for the scan related information generated by CSP (auxiliary data) and updates provided by TM. Clearly stated that Layers 1, 2 and 3 are defined in SADT ICDs. CSP is not aware of the administrative state. Other minor updates and improvements.  Pulsar Search related (to be added to the next version of IICD: a) PSS-beams are moved in and out of sub-arrays in pairs. d) the next scan can start only after CSP reported that the previous Pulsar Search scan has been completed (Section 7.8.9.2).
2	2016-02-04	-	Released for CSP Delta PDR.

Revision	Date Of Issue	Engineering Change Number	Comments
01	2016-02-28	Under Change Control	<p>New number allocated as 100-000000-021 Rev 01. Supersedes:</p> <ul style="list-style-type: none"> <li>• SKA-TEL-SKO-0000143</li> <li>• SKA-TEL.CSP.SE-TEL.TM.SE-ICD-001</li> <li>• SKA-TEL.CSP.MGT-TEL.MGR.MGT-ICD-001</li> <li>• SKA-TEL-CSP-0000016</li> </ul>

## DOCUMENT SOFTWARE

	Package	Version	Filename
Word processor			100-000000-021_01_CSLOW-TM_ICD - Updated to template.docx
Block diagrams			
Other			

## ORGANISATION DETAILS

Name	SKA Organisation
Registered Address	<p>Jodrell Bank Observatory Lower Withington, Macclesfield Cheshire SK11 9DL</p> <p>Registered in England &amp; Wales Company Number: 07881918</p>
Website	<a href="http://www.skatelescope.org">www.skatelescope.org</a>

## TABLE OF CONTENTS

<b>TABLE OF CONTENTS</b> .....	<b>6</b>
<b>LIST OF FIGURES</b> .....	<b>10</b>
<b>LIST OF TABLES</b> .....	<b>11</b>
<b>LIST OF ABBREVIATIONS</b> .....	<b>13</b>
<b>LIST OF TERMS</b> .....	<b>17</b>
<b>1 INTRODUCTION</b> .....	<b>19</b>
1.1 Scope and Purpose .....	19
1.2 Interface Identification.....	19
1.3 Applicable and Reference Documents .....	19
1.3.1 Applicable Documents.....	19
1.3.2 Reference Documents .....	20
1.4 Roles and Responsibilities .....	22
1.5 Interface Scope.....	22
1.6 Interface Topology.....	23
1.7 ICD Development Stages .....	23
1.8 Status of the Document.....	24
1.9 Summary of Standards Rationales .....	24
1.10 Specification Class List and Applicable Standards .....	24
<b>2 OVERVIEW</b> .....	<b>26</b>
2.1 TM and CSP Roles and Responsibilities .....	26
2.2 LMC Interface Guidelines .....	27
2.3 Message Content - Overview .....	27
2.4 Protocol .....	29
<b>3 TEST, DIAGNOSTIC OR MAINTENANCE FEATURES</b> .....	<b>30</b>
<b>4 SAFETY ASPECTS</b> .....	<b>31</b>
<b>5 INTERFACE IMPLEMENTATION SPECIFICATION</b> .....	<b>32</b>
5.1 Data exchange specifications .....	32
5.2 Human-Machine Interfaces.....	32
<b>6 INTERFACE VERIFICATION</b> .....	<b>33</b>
6.1 Verification stages .....	33

**7 INTERFACE DETAILS .....34**

- 7.1 Components and Capabilities..... 34
  - 7.1.1 CSP\_Low Sub-elements ..... 35
  - 7.1.2 CSP\_Low Capabilities..... 36
  - 7.1.3 CSP\_Low Component and Capability Hierarchy ..... 37
- 7.2 Error Handling..... 38
  - 7.2.1 Loss of Connectivity..... 39
  - 7.2.2 Unexpected, Malformed or Invalid Message ..... 39
  - 7.2.3 Message Time-out ..... 40
  - 7.2.4 Extremely High Message Rate ..... 40
  - 7.2.5 Invalid Configuration or State Transition Request ..... 40
  - 7.2.6 Failed Re-configuration or State Transition..... 41
- 7.3 Extensibility and Future Compatibility ..... 41
- 7.4 Naming Conventions ..... 41
- 7.5 CSP\_Low Modes and States ..... 42
  - 7.5.1 Administrative Mode..... 45
  - 7.5.2 Test Mode..... 49
  - 7.5.3 CSP\_Low Observing Mode..... 50
  - 7.5.4 Control Mode ..... 51
  - 7.5.5 Simulated Mode ..... 52
  - 7.5.6 Operational State ..... 53
    - 7.5.6.1 Components That Do Not Implement OFF-DUTY State..... 57
  - 7.5.7 Health State ..... 59
  - 7.5.8 Usage State..... 60
  - 7.5.9 Redundancy State..... 61
- 7.6 Operations ..... 62
  - 7.6.1 Set CSP Parameters ..... 62
    - 7.6.1.1 Activation Time..... 63
    - 7.6.1.2 Activation Queue..... 63
  - 7.6.2 Get CSP Parameters (Query) ..... 65
  - 7.6.3 Monitor Points..... 65
  - 7.6.4 CSP Generated Auxiliary Data ..... 66
  - 7.6.5 Alarms..... 67

7.6.6	Events .....	68
7.6.7	Logging.....	69
7.6.8	Software/Firmware Upgrade.....	69
7.6.9	Reset/Restart/Reboot .....	69
7.6.10	OFF-DUTY (Low-power) State.....	70
7.6.11	Power-down .....	70
7.6.12	Power-up .....	70
7.6.13	Direct access to CSP Components (Tunnelling).....	70
7.7	Sub-array .....	71
7.7.1	End Of Scan.....	72
7.7.2	Observing Mode Change .....	73
7.8	Protocol .....	74
7.8.1	Types of messages.....	74
7.8.2	Conventions.....	75
7.8.2.1	Date and Time .....	75
7.8.3	CSP_Low Parameters.....	77
7.8.4	Parameters Common to all CSP_Low Sub-elements.....	84
7.8.5	Parameters Common to all CSP_Low LRUs .....	91
7.8.6	Parameters Common to all CSP_Low Sub-arrays.....	99
7.8.7	Parameters Common to all CSP_Low Capabilities.....	102
7.8.8	Messages .....	105
7.8.8.1	Message: Set CSP Parameters .....	105
7.8.8.2	Message: Get CSP Parameters (Query) .....	105
7.8.8.3	Message: CSP Response .....	106
7.8.8.4	Message: Alarm .....	108
7.8.8.5	Message: Event.....	109
7.8.8.6	Message: Monitor Point Report .....	109
7.8.8.7	Message: CSP Auxiliary Data Report .....	110
7.8.8.8	Sub-array and Observing Mode Related Messages.....	111
7.8.9	Observing Mode Parameters.....	112
7.8.9.1	Observing Mode Imaging .....	114
7.8.9.2	Observing Mode Pulsar Search.....	116
7.8.9.3	Observing Mode Pulsar Timing.....	120
7.8.10	CSP_Low Data to be stored by Telescope Manager.....	130

7.8.11	Required TM Service.....	130
7.8.12	Alarms.....	131
7.8.13	Events .....	132
7.8.14	Monitor Points.....	132
7.8.14.1	CSP_Low.CBF (Correlator and Beamformer) Monitor Points.....	132
7.8.14.2	CSP_Low.PSS (Pulsar Search Engine) Monitor Points.....	132
7.8.14.3	CSP_Low.PST (Pulsar Timing Engine) Monitor Points.....	133
7.8.14.4	CSP_Low.LMC Monitor Points .....	133
7.9	HMI Engineering interface specifications.....	133
<b>8</b>	<b>VERIFICATION METHODS .....</b>	<b>134</b>
8.1	Interface not made.....	134
8.1.1	Data exchange interface verification.....	134
8.1.2	HMI Engineering interface verification.....	134
8.2	Interface made .....	135
8.2.1	Data exchange interface verification.....	135
8.2.2	HMI Engineering interface verification.....	135
<b>APPENDIX A.</b>	<b>INTELLECTUAL PROPERTY DECLARATION.....</b>	<b>136</b>
<b>APPENDIX B.</b>	<b>TO BE DEFINED .....</b>	<b>137</b>
<b>APPENDIX C.</b>	<b>LMC SCOPE AND RESPONSIBILITIES REQUIREMENTS VERIFICATION MATRIX .....</b>	<b>140</b>

## LIST OF FIGURES

Figure 1-1 SKA1_Low CSP to TM Interface topology.....	23
Figure 1-2 CSP to TM interface represented using ISO/OSI Reference Model .....	25
Figure 7-1 An example for monitor and control hierarchy of Components and Capabilities .....	38
Figure 7-2 Administrative Mode Transitions.....	46
Figure 7-3 Example for the Administrative Mode transition: CSP_Low.CBF Admin Mode transition from MAINTENANCE to ENABLED does not override the higher priority Administrative Mode previously assigned to the children.....	49
Figure 7-4 Test Mode Transitions.....	50
Figure 7-5 Control Mode Transitions.....	52
Figure 7-6 Simulated Mode Transitions – applies for the entities that implement both REAL and SIMULATED mode .....	53
Figure 7-7 Operational State Machine .....	56
Figure 7-8 Operational State Machine for Components that do not implement OFF-DUTY state .....	58
Figure 7-9 Health State Transitions .....	60
Figure 7-10 Usage State Transitions.....	61
Figure 7-11 Redundancy Sate Transitions .....	61
Figure 7-12 Observing Mode Change – Time Sequence Diagram .....	74

## LIST OF TABLES

Table 1-1 Applicable Documents.....	20
Table 1-2 Reference Documents .....	20
Table 1-3 : Roles and responsibilities .....	22
Table 1-4 : Interface class specifications .....	22
Table 7-1 CSP_Low Capabilities.....	36
Table 7-2 CSP_Low Modes and States - overview. Observing Mode related Modes and States are shaded.....	43
Table 7-3 Examples of CSP_Low Mode and State reporting. Note that these examples are also applicable to the following sub-elements: Correlator-Beamformer (CBF), Pulsar Search Engine and Pulsar Timing Engine. .	44
Table 7-4 Administrative Mode Values .....	45
Table 7-5 Test Mode Values .....	50
Table 7-6 CSP_Low Observing Modes (parameter of a Subarray) .....	51
Table 7-7 Control Mode Values.....	52
Table 7-8 Simulated Mode Values.....	53
Table 7-9 Operational State Values.....	53
Table 7-10 Health State Values .....	59
Table 7-11 Usage State Values .....	60
Table 7-12 Redundancy State Values .....	61
Table 7-13 CSP Monitor Point Parameters (accessible using SET/GET Parameter messages).....	66
Table 7-14 CSP parameters implemented for every CSP Alarm (accessible using SET/GET) .....	68
Table 7-15 Messages .....	75
Table 7-16 CSP_Low Parameters used to monitor and set status the CSP_Low as a whole. ....	77
Table 7-17 Parameters implemented by all CSP_Low Sub-Elements.....	84
Table 7-18 Parameters implemented by all CSP_Low LRUs.....	91
Table 7-19 Parameters of the CSP_Low Subarrays .....	99
Table 7-20 Parameters Implemented by all CSP_Low Capabilites (PSS-Beams, PST-Beams, station-input) .....	102
Table 7-21 Set CSP Parameters – Message Content .....	105
Table 7-22 Get CSP Parameters – Message Content.....	106
Table 7-23 CSP Response – Message Content.....	106
Table 7-24 CSP Alarm – Message Content .....	108
Table 7-25 CSP Event – Message Content.....	109
Table 7-26 CSP Monitor Point Status Report – Message Content .....	110
Table 7-27 CSP_Low Create Sub-array .....	111

Table 7-28 CSP_Low Modify Sub-array .....	112
Table 7-29 CSP_Low Delete Sub-array .....	112
Table 7-30 CSP_Low Scan Configuration .....	113
Table 7-31 CSP_Low.CBF End Scan .....	114
Table 7-32 CSP_Low Parameters for the Observing Mode Imaging .....	115
Table 7-33 CSP_Low Pulsar Search Observing Mode Parameters .....	118
Table 7-34 Legend for Table 7-35.....	122
Table 7-35 CSP_Low – Pulsar Timing Observing Mode Parameters.....	122
Table 7-36 CSP_Low – preliminary list of alarms generated by the sub-element CSP_Low.PSS .....	131
Table 8-1 Summary of content to be defined .....	137
Table 8-2 LMC Scope and Responsibilities [ <a href="#">RD3</a> ] Requirements Verification Matrix.....	140

## LIST OF ABBREVIATIONS

ADD	Architecture Design Document
AIV	Assembly, Integration and Verification
ALMA	Atacama Large Millimetre Array
ASCII	American Standard Code for Information Interchange, a character-encoding scheme.
ASIC	Application Specific Integrated Circuit
ATC	Astronomy Technology Centre
CASA	Common Astronomy Software Applications
CDR	Critical Design Review
CI	Configuration Item
CIDL	Configuration Item Data List
CM	Configuration Management
COTS	Commercial Off-The-Shelf
CP	Construction Plan
CSCI	Computer Software Configuration Item
CSIRO	Commonwealth Scientific and Industrial Research Organization
CSP	Central Signal Processor
CUR	Curtin University
DD	Design Description document
DDD	Detailed Design Description document
DRD	Document Requirements Descriptions
DSH	Dish Element or Consortium
DSP	Digital Signal Processing
ECP	Engineering Change Proposal
EICD	External Interface Control Document
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interface
FAD	Feasibility Analysis Documentation
FFT	Fast Fourier Transformation
FMECA	Failure Modes, Effects and Criticality Analysis
FPGA	Field Programmable Gate Array
FTA	Fault Tree Analysis

GPU	Graphics Processing Unit
HDL	High Level Design Language
HTTP	Hypertext Transfer Protocol
ICD	Interface Control Document
IICD	Internal Interface Control Document
ILS	Integrated Logistic Support
ILSP	Integrated Logic Support Plan
INAF	National Institute for Astrophysics, Italy
INFRA	Infrastructure Element or Consortium
I/O	Input/Output
IP	Intellectual Property
IR	SKA South Africa and Australia Infrastructure Requirements
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
LFAA	Low Frequency Aperture Array Element or Consortium
LMC	Local Monitor and Control Sub-element and Work Package
LOW	Used to refer to SKA-1 LOW Telescope located in Western Australia
LRU	Line Replaceable Unit
LSA	Logistical Support Analysis
MATLAB	MATLAB simulation language and application
MID	Used to refer to SKA-1 MID Telescope located in Karoo Region in South Africa
M&C	Monitor and Control
MOD	Modelling Work Package
MTTR	Mean Time To Repair
NCRA	National Centre for Radio Astrophysics, India
NIP	Non-Imaging Processor (Term used in CSP Pulsar Search Non-Imaging Processor and CSP Pulsar Timing Non-Imaging Processor, in the SKA Architecture pack v2.0 replaced with Pulsar Search Engine and Pulsar Timing Engine).
NRC	National Research Council of Canada
NZA	New Zealand Alliance
QA	Quality Assurance
OPS	Operations Work Package
OSI	Open Systems Interconnection
OX	Oxford University

PA	Product Assurance
PDF	Portable Document Format
PDR	Preliminary Design Review
PHS&T	Packaging, Handling, Storage and Transportation
PIP	Physical Implementation Proposal
PMX	PowerMX
PPIA	Precursor and Pathfinder Interaction and Analysis (Work Package)
PSS	Pulsar Search
PST	Pulsar Timing
PTP	Prototyping Plan
QA	Quality Assurance
QAP	Quality Assurance Plan
QC	Quality Control
QP	Quality Plan
RAM	Reliability, Availability and Maintainability
RAMS	Reliability, Availability, Maintainability and Safety
REN	Requirements Engineering Work Package
REN-ADD	Requirements Engineering and Architecture Work Package
REST	Representational State Transfer
RFI	Radio Frequency Interference
RMP	Risk Management Plan
R/O	Read-only (access)
RR	Risk Register
RRS	Reutech Radar Systems
RS	Requirement Specification
R/W	Read-write (access)
SAD	System Baseline Design
SADT	Signal and Data Transport Element or Consortium
SDE	Software Development Environment (Work Package)
SDP	Science Data Processing Element and Consortium
SDRA	Site Documentation and Review Analysis
SEMP	System Engineering Management Plan
SKA	Square Kilometre Array

SKA1	Square Kilometre Array Phase 1
SKA1_Low	SKA1 LOW Telescope (located in Western Australia)
SKA1_Mid	SKA1 MID Telescope (located in Karoo region in South Africa)
SKAO	SKA Organisation (or office)
SKA-SA	SKA South Africa
SNMP	Simple Network Management Protocol
SOW	Statement of Work
SPA	Software Product Assurance
SRR	System Requirements Review
STFC	Science and Technology Facilities Council
SWIN	Swinburne University of Technology
SW	Software
SYSML	System Engineering Simulation Language and application
TBC	To be confirmed
TBD	To be defined (decided)
TDT	Time Domain Team
TM	Telescope Manager Element and Consortium
UMAN	University of Manchester
UML	Unified Modelling Language
URL	Uniform Resource Locator
UTC	Coordinate Universal Time
VPL	Verification Planning Work Package
WBS	Work Breakdown Structure
W/O	Write-only (access)
WP	Work Package
WPEP	Work Package Execution Plan
XML	Extensible Markup Language

## LIST OF TERMS

**Alarm** – The term Alarm (with capital A) refers to the CSP-generated message used to report errors and faults, and other events of interest for the operations. This document describes only those Alarms generated by the CSP; Alarms generated by the TM are sent elsewhere (beyond the scope of the CSP/TM interface).

**Active Alarm** – An Alarm which has an alarm state that has been raised, but not cleared.

**Alarm Detection Point** – The entity that detected the alarm.

**Component** – The term *Component* (with a capital C) is used to refer to a CSP hardware or software Component that can be identified, controlled and monitored via this interface. The top level CSP Component in each telescope is the CSP Element instantiation (a single instance in each telescope). Term Component is used to refer to the top-level Component, CSP sub-elements and their components, down to the level of LRU and software process. Each Component is assigned a unique identifier. For more information refer to Section 7.1.

**Capability** – The term Capability refers to representation of a CSP functionality that can be identified, controlled and monitored via this interface. CSP Capabilities provide a layer of abstraction to allow TM to set, control and monitor signal processing without being aware of the CSP implementation. This ICD identifies the set of CSP Capabilities and their parameters. The most notable CSP Capability is sub-array. For more information refer to Section 7.1.

**Event** – The term *Event* (with capital E) refers to a CSP-generated message used to report an event. An event is something that happens which may be of interest. Examples: a fault, a change of status, crossing a threshold, or an external input to the system.

**Error** – A deviation of a system from normal operation.

**Fault** – Lasting error or warning condition.

**Finite-State Machine (FSM)** – A Finite-State Machine (FSM) is defined by the list of states, allowed state transitions and triggering condition(s) for each transition. The entity is in only one state at a time. The state it is in at any given time is called the *current state*. Transition from one state to another is triggered by an event or condition (a completion of an operation, detection of error, receipt of a message or command, etc.). A Finite-State Machine (FSM) shall be defined for each state and mode indicator.

**Log** – A record generated and logged into a file in order to store (more-or-less permanently) information that can be of interest for testing and troubleshooting.

**LRU** – Line Replaceable Unit is a modular component designed to be replaced quickly at an operating location to restore the end item to an operational ready condition. (Conversely, a non-LRU is a part, component, or assembly used in the repair of an LRU, when the LRU has failed and has been removed from the end item for repair.)

**Mode** – The term ‘*mode*’ is used for parameters that indicate the intended mode of operation set by the user or the parent entity. Modes are implemented as read-write parameters. Mode is set by outside authority in order to change behaviour of the entity.

**Perceived Severity** – The severity of the alarm as determined by the Alarm Detection Point using the information it has available. Severity is also assigned to Events (TBC).

**Query** – The term *Query* is used for TM-generated messages that instruct the CSP to report values for a set of parameters. In the technical literature Query is often referred to as a GET operation. A Query or GET can be issued for all parameters (read-only and read-write).

**Response** – The term *Response* is used to describe a CSP-generated message that contains a response to a TM SET Parameters Request or Query/GET.

**Scan** – A scan is the atomic unit of observer command. Certain parameters can change only at scan boundaries, most notably the observing band and the composition of a Sub-Array. Note that, while the scan is a common concept used in most current interferometers and output data formats, it is still TBD whether “scan” will be so used by the SKA.

**State** – The term ‘**state**’ is used for parameters that indicate state of the entity as derived by the entity itself (in some cases by the entity that reports on behalf of another entity). External entity or authority cannot directly set the state of another entity, but can initiate a state transition by issuing a corresponding command. States are implemented as read-only parameters.

**Sub-array** – A collection of Capabilities used to perform an independent observing programme. CSP sub-array is an exclusive set of Capabilities. Exclusive in this context means that a Capability cannot belong to more than one sub-array at any given time.

# 1 Introduction

## 1.1 Scope and Purpose

This document defines the interface between the Central Signal Processor (CSP) and Telescope Manager (TM) for the Square Kilometre Array (SKA) Phase 1 LOW Telescope.

## 1.2 Interface Identification

This Interface Control Document (ICD) defines the requirements and implementation details of the monitor and control data exchange interfaces:

- **I.S1L.CSP\_TM.001** "SKA1\_Mid Central Signal Processing \_Telescope Manager Monitor and Control Interface".
- **I.S1L.CSP\_TM.002** "SKA1\_Mid Central Signal Processing \_Telescope Manager Scan Real-Time Monitor and Control Interface". This interface is for the CSP generated auxiliary data (Section 7.6.4) as well as for the real-time updates provided by TM for the on-going scans, as defined in Section 7.8.9 Observing Mode Parameters (e.g. delay models, weights used in beamforming, Jones Matrices). Some of the information transmitted by TM over this interface is provided by SDP.

Reason for two interfaces is that nature of content differs and implementation will most probably be different also.

*[Note: the interface identifier used is I.S1t.xxx\_yyy.nnn, where:*

*S = SKA,*

*1 = Phase 1,*

*t = Telescope: L for Low,*

*Xxx = Interfacing element 1*

*Yyy = Interfacing element 2*

*nnn = identification number]*

This interface is between the following items of the SKA1\_Low Telescope:

- Central Signal Processing Element (CI number 102-000000)
- Telescope Manager: SKA1-Mid TM (CI number 103-000000)

## 1.3 Applicable and Reference Documents

### 1.3.1 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.

**Table 1-1 Applicable Documents**

Ref No	Document/Drawing Number	Document Title	Version
AD1	SKA-TEL.SE.INTERF-SKO-MP-001	SKA Interface Management Plan	-
AD2	SKA-TEL-SKO-0000008	SKA Phase 1 System (Level 1) Requirements Specification	-
AD3	SKA-OFF.PAQA-SKO-QP-001	SKA Product Assurance & Safety Plan	-

### 1.3.2 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.

**Table 1-2 Reference Documents**

Ref No	Document/Drawing Number	Document Title	Version
RD1	International ISO/IEC Standard 7498-1	Information technology – Open Systems Interconnection – Basic reference Model: The Basic Model	Second edition 1994-11-15. Corrected and reprinted 1996-06-15
RD2	<i>ISO 8601:2004(E)</i>	<i>Data elements and interchange formats – Information interchange – Representation of dates and time</i>	2004-12-01
RD3	SKA-TEL-TM-0000030	SKA Local Monitor and Control Scope and Responsibilities	
RD4	SKA-TEL-TM-0000031	SKA1 Local Monitor and Control Interface Guidelines	
RD5	SKA-TEL.SADT.SE-TEL.CSP.SE-ICD-001	SKA1 Interface Control Document SADT to CSP	
RD6	SKA-TEL-SKO-0000153	SKA1 Interface Control Document TM to SADT	
RD7	SKA-TEL-CSP-0000159	SKA Statement of Work for Central Signal Processor	

Ref No	Document/Drawing Number	Document Title	Version
RD8	SKA-TEL-CSP-0000010	SKA CSP Element Requirement Specification (SE-2)	Rev 1
RD9	SKA-TEL-CSP-0000014	SKA CSP Element Architecture Design Document (SE-5)	-
RD10	SKA.TEL.SE.OPS-SKO-COO-001-0-A	Concept of Operations for the SKA Observatory	-
RD11	SKA-TEL-CSP-0000158	Interface Control Document CSP_Low to INFRA AU (SE-6a1)	-
RD12	N/A	PSRCHIVE and PSRFITS - An Open Approach to Radio Pulsar Data Storage and Analysis, A.W.Hotan, W.vanStraten, R.N.Manchester, Swinburne Centre for Astrophysics and Supercomputing, Australia Telescope National Facility, Netherlands Foundation for Research in Astronomy, 2004	
RD13	N/A	PSRCHIVE and PSRFITS - Definition of the Stokes parameters and Instrumental Basis Convention, W. van Straten, R. Manchester, S. Johnston, J. Raynolds, Swinburne Centre for Astrophysics and Supercomputing, CSIRO Astronomy and Space Science.	
RD14	TBD	Interface Control Document TM_Low to INFRA AU	

## 1.4 Roles and Responsibilities

Table 1-3 shows the roles and responsibilities for the relevant parties.

**Table 1-3 : Roles and responsibilities**

Role	Element	Point of contact	Responsibilities
Leading party	CSP	LMC Work Package Lead	Create and maintain this document.
Following party	TM	System Engineer	Contribute to, review and approve this document.
Owning Party	SKA Office	SKAO Chief System Engineer	Own this document.
		SKAO Head of Project	Release this document.
		SKAO TM System Engineer	Review and approve this document.
		SKAO CSP System Engineer	Review and approve this document.
		SKAO SaDT System Engineer	Review and approve this document.

## 1.5 Interface Scope

Table 1-4 lists the classes of interface specification which are included in this document.

**Table 1-4 : Interface class specifications**

Interface class	Included in document?
Mechanical	N
Fluid	N
Thermal	N
Electromagnetic	N
Optical	N
Electrical	N
Electronic	N
Electro-optical	N
Data exchange specifications	Y
Human-Machine Interface	Y

## 1.6 Interface Topology

This ICD specifies requirements and implementation for the interface between CSP and TM in the SKA1\_Low Telescope.

Signal and Data Transfer (SADT) Element provides communication link between CSP and TM, as described in the following ICDs:

1. Interface Control Document SADT to CSP, SKA-TEL.SADT.MGT-TEL.CSP.MGT-ICD-001, [\[RD5\]](#)
2. Interface Control Document SADT to TM, SKA-TEL.SADT.MGT-TEL.TM.MGT-ICD-001, [\[RD6\]](#)

The network interface and carrier protocol provided by the SADT Element transparently carries data between CSP and TM equipment.

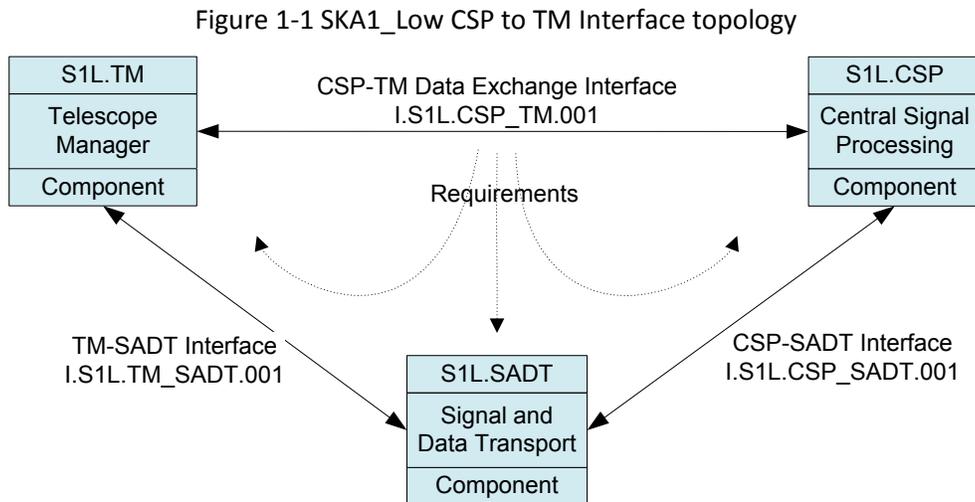


Figure 1-1 shows how interfaces are identified in the SKA1 system.

This version of the ICD assumes that there is a single CSP to TM interface per telescope.

The spatial deployment of the CSP\_Low equipment is as specified in the Interface Control Document CSP\_Low to INFRA AU [\[RD11\]](#).

The spatial deployment of the TM\_Low equipment is as specified in the Interface Control Document TM\_Low to INFRA SA [\[RD14\]](#).

## 1.7 ICD Development Stages

This ICD will be developed in stages as follows:

- a) The first signed-off ICD was published after PDR, with all requirements agreed in:
  - Section 3 Test, Diagnostic or Maintenance Features
  - Section 5 Interface Implementation Specification

➤ Section 6 Interface Verification

Some implementation details were agreed at this stage (Section 7 Interface Details), depending on the information available and on the criticality of implementation details.

- b) Preliminary agreement to use TANGO Controls framework ([www.tango-controls.org](http://www.tango-controls.org)) was achieved in March 2015; prototyping has been started. TM Consortium will define the guidelines for use of TANGO Controls in the SKA Telescopes. When guidelines are published, this ICD will be updated accordingly.
- c) The exact content and format of messages will be defined during Pre-construction Phase2 and documented in the Section 7 "Interface Details".
- d) 6 months prior to the CDR of any one of the three elements involved in this interface: Final ICD agreed, including all implementation details.

## 1.8 Status of the Document

This version of the document identifies information that will be exchanged between TM and CSP over this interface and, where applicable, in what order (i.e. the time sequences for data exchange).

Requirements for this interface have been derived from the CSP requirements [\[RD8\]](#) and functionality [\[RD9\]](#) and from the following TM documents:

- SKA1 LMC Scope and Responsibilities [\[RD3\]](#), and
- SKA1 LMC Guidelines [\[RD4\]](#).

## 1.9 Summary of Standards Rationales

As requested in the SKA Interface Management Plan [\[AD1\]](#) this document uses an OSI Model [\[RD1\]](#) to describe the interface. Protocols and standards to be used to implement this interface are TBD (see Section 1.7).

## 1.10 Specification Class List and Applicable Standards

Figure 1-2 shows CSP to TM interface using ISO/OSI Reference Model [\[RD1\]](#). As specified in the section 1.6 Interface Topology, the Physical Layer, Data Link Layer and Network Link layer are described in SaDT to CSP ICD [\[RD5\]](#) and SaDT to TM ICD [\[RD6\]](#). This ICD describes the rest of the OSI protocol stack (Transport Layer, Session Layer, Presentation Layer and Application Layer as shown in Figure 1-2). Protocols and standards to be used for this interface are TBD (see Section 1.7).

In order to achieve uniformity, Telescope Manager (TM) Consortium will define the protocol stack to be used for exchange of monitor and control data with all SKA Elements in the document SKA1 LMC Guidelines [\[RD4\]](#). The CSP to TM interface will comply with the requirements specified in [\[RD4\]](#).

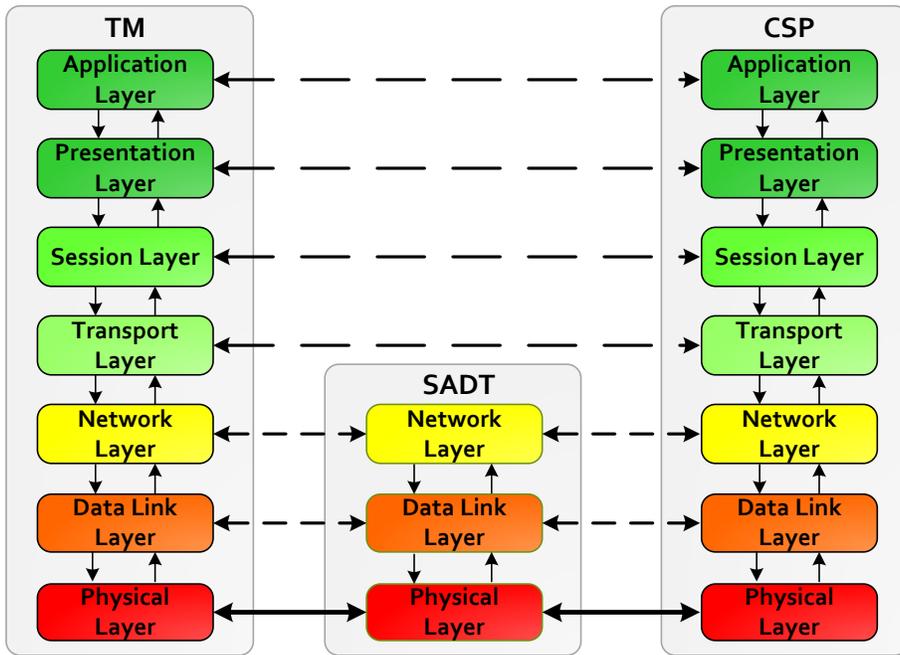
In addition to the exchange of the monitor and control messages, this ICD identifies the Human-Machine Interfaces (HMIs) to be integrated into the SKA engineering interface to be used during commissioning and normal operations for testing, troubleshooting and maintenance. Guidelines for the SKA human-machine interfaces will be defined by the TM Consortium. The CSP will follow these TM-defined guidelines.

This ICD describes the following classes of interfaces:

- Data exchange specifications, and
- Human-machine Interface.

When identified, applicable standards shall be identified here.

Figure 1-2 CSP to TM interface represented using ISO/OSI Reference Model



## 2 Overview

This Interface Control Document defines:

1. Format and content of messages that pass between the TM and the CSP.
2. Sequenced behaviour of those messages.
3. Human-Machine Interfaces to be integrated into the SKA engineering interface to be used during commissioning and normal operations for testing, troubleshooting and maintenance.

There is a single instance of the TM to CSP interface in SKA1\_Low Telescope. Messages flow to and from the CSP.LMC server(s). There is one CSP.LMC server in SKA1\_Low telescope. Redundant CSP.LMC hardware for reliability may or may not be provided.

### 2.1 TM and CSP Roles and Responsibilities

This section defines CSP and TM responsibilities specifically pertaining to this interface.

1. TM roles and responsibilities *specifically pertaining to this interface*:
  - a) Initiator (client) of control/configuration commands for normal observing, engineering test, and maintenance. TM provided configuration includes destination addresses for the CSP output products.
  - b) Acquisition and storage of monitor data, alarms, and events generated by the CSP. This includes acquisition of internal low-volume data products needed by SDP for its processing such as channel gains, RFI statistics, state counts, phase-cal measurements, etc. Also includes requesting health status of CSP resources (servers, processing nodes, etc.).
  - c) Setting of alarm/notification activity levels, i.e., instructing CSP what level of alarms and monitor data to pass across this interface.
  - d) Providing direct access to CSP equipment (components) using messages that do not necessarily comply with this interface (this may require tunnelling). This is required to allow for use of engineering interfaces developed during construction for troubleshooting, testing and maintenance. One possible mechanism is for TM to request a tunnel to an LRU (Line Replaceable Unit) and thereafter allow HMI to talk to the LRU directly using LRU-specific messages.
  - e) Provide user authentication and authorization for access to the CSP network via this message passing interface. That is, the TM ensures that no unauthorized traffic/messages pass across this interface.
2. CSP roles and responsibilities *specifically pertaining to this interface*:
  - a) Executor (server) of control/configuration commands for normal observing, engineering test, and maintenance. This includes responding within the required time with proper

acknowledge/action messages. It is likely the CSP must maintain a queue of commands to execute at specific times.

- b) Collection, filtering, and generation of monitor data, alarms, and events to report to the TM on request, including the status of CSP servers/processing nodes.
- c) Setting of alarm/notification activity levels as specified by the TM.
- d) Providing access to all CSP processing nodes' information including FMECA, so as to facilitate TM-provided HMIs.

## 2.2 LMC Interface Guidelines

TM will define standards and guidelines for the interface between the TM and other Elements in the document "SKA TM to Element LMC Interface Guidelines" [RD4]. Wherever possible, CSP will adopt and implement the standards defined by TM; any non-compliance will be specified in this ICD.

The document "SKA TM to Element LMC Interface Guidelines" [RD4] will define the following:

1. Naming conventions.
2. SKA Control Model including Modes and States, state transitions in the form of state machine, messages used to request mode and state transitions, etc.
3. Protocol stack for the interfaces between TM and other SKA Elements (including CSP).
4. Format and content of messages.
5. Component and Capability Health Reporting guidelines.
6. Guidelines for implementation of logging.
7. Guidelines for implementation of alarms, etc.

## 2.3 Message Content - Overview

This section provides an overview of the messages exchanged over TM to CSP interface.

TM is a client and initiator of requests, unless otherwise noted:

1. Request environmental (temperatures, voltages, power usage, fan speeds etc.) status of all or specific CSP LRUs.
2. Request logical/functional health (e.g., comm. status, processing status) of CSP instantiation or for a specific Component or Capability. CSP makes provision for TM to request either the full report or a very compact indication of the functional health.
3. Request power on, power off, transition to low-power mode for CSP as a whole and for a specific CSP sub-element or Component. Allows individual LRUs to be power cycled for hard resetting and

removal/replacement, as well as power down or low-power mode<sup>1</sup> of the entire sub-element for any purpose.

4. Update CSP firmware/software.
5. Set alarm/event/logging activity level. Used to throttle the number of Alarm messages delivered to the TM.
6. Define a sub-array, and tag it with a unique logical ID. Precedes definition of what to do with that sub-array, i.e. how to configure it at any point in the future. Message includes the list of stations and other generic sub-array specific information (TBD).
7. Add/remove station to/from a sub-array. Message includes sub-array logical ID and the list of stations to be added or removed. Stations can be added and removed to/from an idle sub-array and at scan boundaries (i.e. after one scan ends and before the other scans begins).
8. Declare station as available or unavailable for use. This message is used to tell the CSP that a particular station is available or unavailable for use but does not otherwise affect any sub-array composition. This is not a mechanism to add or remove stations from sub-arrays.
9. Update/change sub-array parameters such as phase centre.
10. Observation specific configuration messages for a pre-specified sub-array:
  - a) Imaging observation configuration message as specified in Section 7.8.9.1.
  - b) Pulsar Search configuration message as specified in Section 7.8.9.2.
  - c) Pulsar Timing configuration message as specified in Section 7.8.9.3.
15. Update time-dependent parameters as applicable to each telescope:
  - a) On-sky calibration residuals sourcing from the SDP via the TM for central beamforming (i.e. station-based delay corrections), as applicable.
  - b) Real-time round-trip LO corrections sourcing from the SDP via the TM. These are delay correction messages sent periodically to the CSP.
  - c) Jones matrices and weights to be used in creating the tied-array beams.
  - d) Bandpass corrections and RFI information from SDP.
17. Request direct access to a particular LRU. This may be implemented as tunnelling, so that TM defines LRU-specific tunnel which can be used for exchange of messages that are LRU-specific (not defined by this ICD) without CSP.LMC vetting, filtering, or interaction. This allows engineering interfaces that do not conform to this interface to be used for testing and troubleshooting.
18. Set a CSP sub-element or portion of a sub-element in test mode. Typically a sub-element or part of a sub-element is put into test mode and generates data; analysis of that data is performed outside the scope of the CSP. Exact requirements TBD. Additional parameters may be defined to be used specifically in test mode.

CSP responds the above listed requests with appropriate action and acknowledgement. In addition CSP:

---

<sup>1</sup> For FPGA-based systems this means de-configuring the bitstream; for GPU and ASIC systems action is TBD.

1. Reports detected errors and faults in the form of Alarm messages.
2. Reports other events of interest in the form of Event messages. Events of interest will be identified in advance and documented in this ICD.
3. Logs all CSP generated Alarms and Events in the Central CSP logging file. (CSP makes provision for TM to access, copy and search the Central CSP logging file.)

## 2.4 Protocol

This interface will use the protocol/message passing platform common to all TM to SKA1 Element interfaces defined in [\[RD4\]](#). In general, that will be a high-level message passing protocol easy to parse, robust, and easy to augment/upgrade message content.

### 3 Test, Diagnostic or Maintenance Features

The following design features are provided at the CSP to TM interface exclusively for testing, diagnosis or maintenance procedures, for the interface itself:

- 1) Both CSP and TM provide for optional logging of the messages received and transmitted via this interface.
- 2) CSP makes provision for TM to remotely activate/deactivate logging of the messages received and transmitted on this interface by CSP.
- 3) Both TM and CSP detect and report errors on this interface, such as time outs, CRC errors, checksum errors, dropped packets, etc. Details will be provided when the protocol stack is defined.
- 4) Both TM and CSP maintain connectivity at all times. TM and CSP monitor activity on this interface; in the case that TM or CSP detects that it has not transmitted a message for longer than TBD seconds, TM/CSP generates and transmits a keep-alive message. Note: The need for keep-alive messages may depend on the underlining protocols and will be re-examined when the protocol stack is defined.
- 5) TM shall define a coherent purpose, method, and look-and-feel for human-machine interface. It is envisioned that these human-machine interfaces will be built and used for low-level unit testing. Such engineering interfaces will be developed for all CSP components and will be essential during integration, delivery & acceptance, commissioning, and operations. For each component, engineering interface will provide access to registers, parameters, LEDs, status information, monitor points and such. These engineering interfaces will be organized hierarchically to allow for drill-down access. For example, for a printed circuit board a Graphical User Interface (GUI) should be developed to display board status, versions of hardware and software, and rolled-up status of the components (FPGAs/ASICs). A click on an FPGA or ASIC should open a GUI to display detailed information for that device: firmware version, content of the registers, status indicators. Where applicable a higher-level representation of the status and setup should be provided. For example, for an FPGA/ASIC that implements a correlator, the display should include a list of baselines, output products, number of spectral channels per products and destination address for products. Hyperlinks to relevant documentation (user manuals, board schematics, etc.) could be also included. Detailed specification for engineering interfaces is still to be developed (TBD); the purpose of the description provided here is to clarify the intended scope and purpose of the planned HMIs.
- 6) Additional Element and sub-element level human-machine interfaces will be built and used for testing during integration, acceptance, commissioning, and operations. These HMIs will be used for unit testing, to emulate or replace the functionality of other CSP components and other Elements. They will be used to generate test vectors (test patterns), send commands, intercept, save, analyse and/or display output data, meta-data, responses to commands and queries, monitor data, alarms, etc. These interfaces may be useful even after commissioning is completed, and shall be delivered together with other software and test tools on an as-is basis. There is no plan to integrate these interfaces and tools into a single framework. HMIs that belong to this group are in fact test tools and are not described in this document.
- 7) It is to be determined (TBD) whether CSP should make provision for TM to verify, without affecting the current observations and CSP setup, whether a particular request (SET Parameters) can be executed at the specified time in the future.
- 8) CSP shall make provision for TM to request CSP to generate and transmit to the specified destination a stream of packets (defined either by the time duration, number and size of packets, pattern etc.). This will be used to test the CSP-SDP link at time of installation or troubleshooting.

## 4 Safety Aspects

CSP consists mainly of computers and digital hardware; safety concerns are overheating and power surges. CSP implements sensors that detect temperature and power surges. Temperature and voltage measurements are monitored on the regular bases (periodically). Measurements are reported to TM periodically and when the value crosses one of the pre-configured thresholds.

In addition, CSP digital hardware, i.e. Printed Circuit Boards, are equipped with ‘dead-man’ protection mechanism able to autonomously switch-off supply of power when a hazardous situation (overheating, power surge) is detected.

It is to be defined (TBD) whether CSP, in the case of prolonged loss of communication with TM, should stop on-going signal processing and transmission of output data and set the CSP equipment in the low-power mode.

## 5 Interface Implementation Specification

Interface Implementation Specification is described in the following documents:

1. Network interface and carrier protocol are defined in the “Interface Control Document SADT to CSP” [\[RD5\]](#).
2. Network interface and carrier protocol are defined in the “Interface Control Document SADT to TM” [\[RD6\]](#).
3. Communication protocols for OSI/ISO Layers 4 to 7 are defined in this document.
4. Format and content of messages passed across this interface are defined in this document.
5. Time-sequenced behaviour of these messages is defined in this document.

Required hardware infrastructure and support:

1. CSP connects to the network interface port as described in the SADT to CSP ICD [\[RD5\]](#). (CSP-internal servers and networking equipment are provided by CSP.)
2. TM connects to the network interface port as described in the SADT to TM ICD [\[RD6\]](#).
3. The CSP network interface port must be protected by TM and/or SADT from any unauthorized external access (i.e. the CSP network contains neither firewalls nor any other protection).

### 5.1 Data exchange specifications

Protocol to be used for data exchange between TM and CSP has not been identified (TBD).

Using a model of process interactions, such as the ISO OSI standard [\[RD1\]](#), the data exchange between interfacing entities will be fully described in this section.

Communication between TM and CSP is asymmetrical. TM generates commands and queries, while CSP acts upon them and generates responses and reports. In addition, CSP reports errors, faults and other significant events using Alarm and Event messages. List of messages is provided in Section 7.8.

### 5.2 Human-Machine Interfaces

Human-machine interfaces (HMIs) to be included in the SKA engineering interface will be described here. As already stated in section 1.10, CSP will follow the guidelines for the SKA human-machine interfaces defined by the TM Consortium.

## 6 Interface Verification

Interface verification is carried out to demonstrate that the design and implementation conform to the ICD, and it will be described in formal procedures.

### 6.1 Verification stages

The Interface validation and verification process occurs in at least three stages:

- a) **Stage 1:** Validation and verification of the Interface Control Document (ICD) through formal reviews during the design phase. This stage is the responsibility of the leading party, using models and analysis methods which themselves have been validated and verified.
- b) **Stage 2:** Verification of the interface implementation by individual parties, i.e. suppliers of CSP and TM, during their design qualification phase by means of exercising the interface with the use of test equipment and/or simulators. This verification is responsibility of the individual Elements.
- c) **Stage 3:** Verification of the interface during system integration (for each telescope). This verification is the joint responsibility of the SKA AIV (Assembly, Integration and Verification), CSP team and TM team. This stage has two phases – a formal acceptance for each Element (TM and CSP) in isolation, followed by system integration and formal verification of the CSP to TM interface.

Interface verification occurs at many stages during the integration of the Elements. It is carried out to demonstrate that the design and implementation conform to the ICD, and it will be described in formal procedures. A key distinction is made between verification tests which require the interfacing Element and those that do not (using a simulator or standard test equipment).

The ICD shall contain a section on the verification methods to be used to:

1. Verify the design and implementation of interfacing hardware and software without the interface being made, i.e. without connection between TM and CSP.
2. Verify the interface for integration and acceptance purposes (involving the interface being made). 'Interface made' in this context means that TM is connected to CSP and that CSP can receive TM generated messages and vice versa.

## 7 Interface Details

This chapter contains data exchange specifications for the TM to CSP interface including:

- Definition (format and content) of the messages and
- Definition of the time sequenced behaviour of these messages.

General requirements for interfaces between TM and other SKA Elements will be defined in the document “SKA1 Element LMC Interface Guidelines” [RD4]. At the time when this version of the ICD is being released the document “SKA1 Element LMC Interface Guidelines” [RD4] is not yet complete; when complete [RD4] will define communication infrastructure (protocol stack), format and content of the messages, message types, state machine, naming conventions, etc.

Protocol stack and message encoding and format are still to be defined (TBD) - this version of the document defines only information (content) to be exchanged between TM and CSP.

Whenever possible CSP will adhere to the guidelines specified in [RD4]. Any non-compliance will be specified in this ICD.

This Chapter will contain detailed description of the following:

1. CSP naming conventions
2. Content and format of messages
3. Valid message sequences
4. List of CSP Alarms
5. List of CSP Events
6. List of CSP Monitor points
7. CSP generated information (data) to be stored in the Telescope Model.

Throughout the document references are made to the TM Element and CSP Element, not to sub-elements and Components within the Elements.

Exception to this rule may be use of CSP engineering interfaces that will be developed during construction. Use of such interfaces may require both TM and CSP to provide for tunnelling, i.e. to transparently carry messages from TM to a particular CSP component and vice versa.

### 7.1 Components and Capabilities

Objects that can be managed (identified, monitored and controlled) via this interface can be classified in two groups:

1. **CSP hardware and software Components**, including CSP Element instantiation, CSP sub-elements and their Components (LRUs and software process). Some LRUs make provision for the individual sub-components to be monitored and controlled via this interface. The exact definition and list of Components that can be identified, controlled and monitored via this interface is implementation-dependent. Each Component is assigned a unique identifier based on its physical location and role. A Component may comprise several other Components, in which case it owns and controls those Components, such Components may be referred to as sub-components or subordinate Components. Terms sub-component and subordinate Component are used only in relative terms, e.g. Component B is a sub-component of Component A and Component C is a sub-Component of Component B. To allow

comparison of the intended configuration of the telescope, stored in the SKA Configuration Data Base (maintained using eB) with the reported configuration, in addition to the identifiers that reflect the role and physical location of the components, upon request, CSP shall report serial numbers and versions of software, firmware and hardware according to conventions used in the product breakdown structure.

2. **CSP Capabilities.** A Capability is a representation of a CSP functionality that can be identified, controlled and monitored via this interface. CSP Capabilities provide a layer of abstraction to allow TM to set, control and monitor signal processing and other CSP functions without being aware of the details of the CSP implementation. The most notable CSP Capability is sub-array. Sub-arrays are created on TM request; all other Capabilities are created automatically, by CSP, based on the available hardware and software Components. For each Capability, CSP can report a set of associated parameters (including status and health indicators) and the list of Components and Capabilities owned and/or used by a Capability. In some cases a Capability has exclusive use and control of a Component, while in others two or more Capabilities use the same Component. For this reason, state and mode transitions at the Capability level do not necessarily translate into state and mode transitions for the Components used by the Capability. Allocation of hardware and software Components to Capabilities is implementation-dependent and will be defined for each telescope when the CSP design is defined to a sufficient detail. A Capability can own and control other Capabilities (e.g. a sub-array can own a PSS beam).

CSP implements a self-describing interface that allows TM to 'discover' CSP Components and Capabilities via interrogation. CSP Components and Capabilities are organized in a hierarchical order, as a tree. The top level of the hierarchy (i.e. the root of the tree) is the CSP Element. For each object in the hierarchy (each Component and Capability) TM can request the list of parameters and their values, and the list of sub-ordinate Components and Capabilities.

CSP makes provision for TM to obtain the list of:

- All Capabilities that use a particular Component.
- All Components used by a particular Capability.
- All Components and Capabilities in use.
- All Components and Capabilities not in use.

### 7.1.1 CSP\_Low Sub-elements

CSP\_Low Sub-elements are 2<sup>nd</sup> tier Components in the CSP\_Low monitor and control hierarchy.

CSP\_Low sub-elements are:

1. CSP\_Low Correlator and Beamformer (CSP\_Low.CBF)
2. CSP\_Low Pulsar Search Engine (CSP\_Low.PSS)
3. CSP\_Low Pulsar Timing Engine (CSP\_Low.PST)
4. CSP\_Low Local Monitor and Control (CSP\_Low.LMC)

All CSP\_Low sub-elements implement a common set of parameters as defined in section 7.8.4.

### 7.1.2 CSP\_Low Capabilities

Table 7-1 lists Capabilities implemented by CSP\_Low.

Capabilities station-beam, PSS-beam and PST-beam are created during CSP initialization or when new equipment is installed and powered-up. At creation, a set of hardware and software Components are assigned to the Capability.

Sub-arrays are created and destroyed dynamically, on TM request.

Capability baseline exists only in the context of a sub-array. Capability baseline has been defined to allow TM to query status of the output products for a particular baseline – it is to be confirmed (TBC) whether such functionality is required.

Capabilities station-input, CSP-PSS-beams and CSP-PST-beams can be moved in and out of sub-arrays on TM request (at scan boundaries).

**Table 7-1 CSP\_Low Capabilities**

CSP_Low Capability	Max. Number of Instances	M&C Functionality Supported
<b>sub-array</b>	16	List availability (number of sub-arrays and their status) Create Delete Add/remove stations Add/remove PSS-beams and/or PST-beams Set Observing Mode and parameters Stop signal processing and generation of output products (set Observing Mode IDLE) Real-time updates for Observing Mode parameters (delay models and similar) Query parameters & status
<b>station-input</b>	512	Station is not part of CSP; this Capability has been defined to allow TM to query status of input signal and status of the CSP Components that process input from a station. Input from each station consists of 384 coarse channels. Number of station-inputs and status Disable (set 'not available for use') Enable (set 'available for use') Query parameters & status.
<b>CSP-PSS-beam</b>	500	List availability (number of beams and status) Disable / Enable Set parameters Get parameters & status
<b>CSP-PST-beam</b>	16	List availability (number of beams and status) Disable / Enable Set parameters Get parameters & status

CSP_Low Capability	Max. Number of Instances	M&C Functionality Supported
baseline	131,328 $n*(n-1)/2 + n$ n=512 number of stations	Get status Disable/enable generation (or transmission) of the products. Auto-correlations (correlation of station with itself) are also represented as baselines.

### 7.1.3 CSP\_Low Component and Capability Hierarchy

Query at the top level of the CSP instantiation in the SKA1\_Low telescope returns:

1. Status and other parameters of the top level component (CSP\_Low instantiation), ad listed in Table 7-16 on page 77.
2. List of sub-elements. CSP\_Survey has one instance of each sub-element:
  - CSP\_Low.CBF,
  - CSP\_Low.PSS,
  - CSP\_Low.PST,
  - CSP\_Low.LMC
3. List of Capabilities that are 'visible' at the top level:
  - sub-array
  - station-input
  - CSP-PSS-beam
  - CSP-PST-beam

Figure 7-1 is an example of the possible hierarchy of Components and Capabilities; it does not represent the actual hierarchy of Components.

All CSP\_Low sub-elements consist of a number of Components (LRUs, PCBs, servers, compute nodes, GPUs, software processes). Some LRUs have sub-components (e.g. FPGAs, software processes) that can be monitored and controlled via this interface. The LRUs and/or servers will be assigned names according to their location (in the cabinets) and/or role; naming conventions for Components and Capabilities are still to be defined (TBD). Component hierarchy is fairly stable, once hardware and software are installed, configuration rarely changes.

TM can use queries (Get Parameters) to 'discover' hierarchy of CSP Components and Capabilities.

CSP makes provision for TM to request a specific logical group of CSP parameters at various level of hierarchy. TM queries can also specify the level of detail and/or 'depth' of hierarchy to be reported in the response (format of queries is TBD).

#### **Example**

The following is an example of the information provided as a brief response to a query at the top level of CSP\_Low hierarchy; it includes the overall Administrative Mode (enabled) and Operational State (ready) of the CSP\_Low, the list of sub-elements, and an overview of the Capability status:

CSP\_Low, enabled, ready, degraded

CSP\_Low.CBF: enabled, ready, ok  
 CSP\_Low.PSS: enabled, ready, ok  
 CSP\_Low.PST: enabled, ready, ok  
 CSP\_Low.LMC: enabled, ready, ok

station-input: 500 enabled, ready, ok | 8 maintenance, ready, ok | 4 maintenance, ready, degraded

CSP-PSS-beams: 400 enabled, ready, ok |100 disabled, unknown

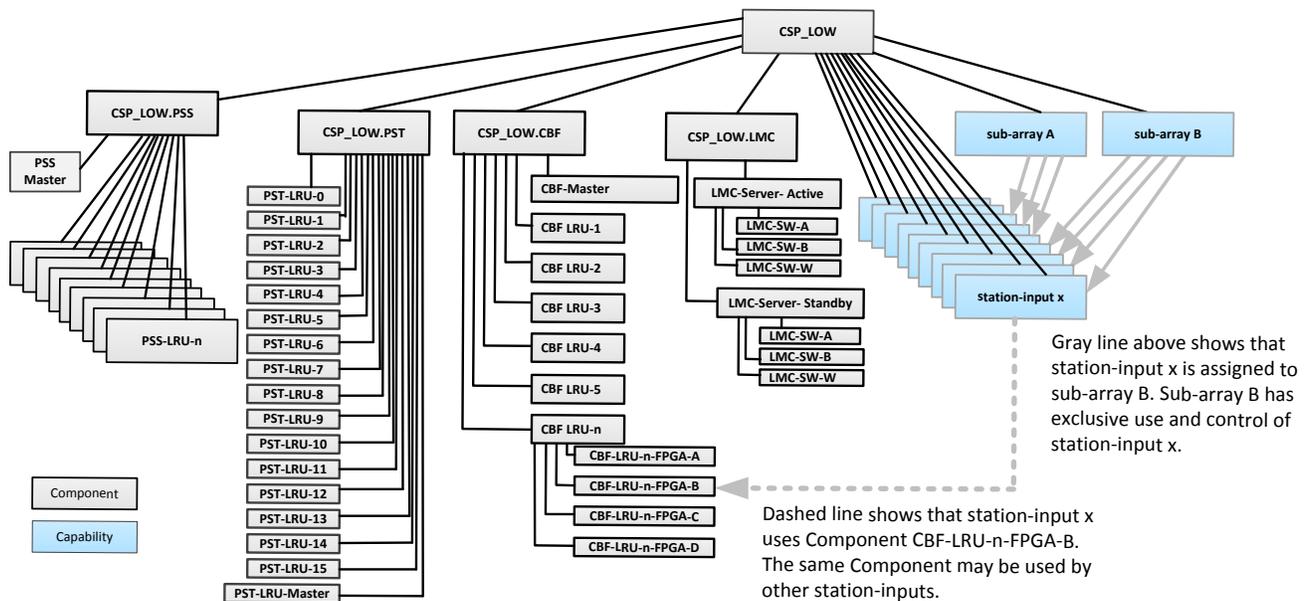
CSP-PST-beams: 16 maintenance, ready, ok

sub-array, id=xa-1, enabled, ready, ok, imaging, 10 stations

sub-array, id=test-pst, maintenance, ready, ok, idle, 250 stations, 16 PST-beams

sub-array, id=test-IMG, maintenance, ready, degraded, imaging, 100 stations

**Figure 7-1 An example for monitor and control hierarchy of Components and Capabilities**



## 7.2 Error Handling

This section defines how TM and CSP handle errors and failures related to this interface.

This version of the document contains a preliminary list of errors and failures that can be detected by CSP and TM on this interface. Error handling will be analysed and defined more precisely when the underlining protocol stack and protocol between CSP and TM are defined (i.e. during the Pre-construction Phase 2).

### 7.2.1 Loss of Connectivity

When TM detects loss of connectivity<sup>1</sup> with CSP, TM generates Alarm to notify operators and keeps trying to establish communication with CSP.

In the case when CSP detects loss of connectivity with TM:

1. CSP generates and logs Alarm message (Note: Alarm message cannot be delivered to TM, due to the loss of connectivity).
2. If due to loss of connectivity, CSP does not receive delay model polynomials and other real-time updates, CSP either flags output data or stops generation of output products (TBD for each Observing Mode individually). Any on-going signal processing should continue using previously provided models.
3. Ancillary data collected by CSP (e.g. input counts) cannot be delivered to TM.
4. If communication is not established after prolonged period of time, CSP assumes that it is no longer useful to keep generating output data and sets all sub-arrays to idle mode. It is to be defined (TBD) what does 'prolonged period of time mean' (how many hours or days).

### 7.2.2 Unexpected, Malformed or Invalid Message

Both TM and CSP perform syntactic and semantic checks for received messages.

If CSP receives an unexpected, malformed or invalid message:

- CSP reports the event to TM. Depending on the protocol and point where the error was detected, receipt of unexpected, malformed or erroneous message can be reported as an Alarm and/or in the message generated as a direct response to the received message.
- CSP logs the event.
- After the error has been reported and logged, the unexpected, malformed and invalid messages are discarded.

If TM receives an unexpected, malformed or invalid message from CSP, TM logs the event, generates an Alarm to notify the operator and discards the message. It is not necessary to notify CSP – unless CSP implements automatic corrective action.

Functionality related to bad message counters is still to be defined (pending the selection of the protocol stack). Proposed functionality (TBC): *Both CSP and TM implement bad message counters and use those counters to throttle logging and alarm reporting. The bad message counters are reset periodically; before the reset, the content is logged in the local log file and optionally sent to TM, either in the form of an alarm or as monitor data (TBD). At any time, a counter indicates the number of bad messages received during the most recent period. Both CSP and TM could use the bad message counters to temporarily disable the interface when excessive number of bad messages is received over a short period of time – the exact functionality requires more consideration; analysis will be performed during the Pre-construction Phase 2.*

---

<sup>1</sup> What exactly 'loss of connectivity' means on this interface, and how it is detected, will be defined when the protocol stack is selected (TBD).

If connectivity with TM is lost, CSP is isolated and therefore unable to report alarms; errors and alarms are logged locally, so that they can be analysed when connectivity is established.

### 7.2.3 Message Time-out

Both TM and CSP expect the other side to confirm receipt of messages within pre-defined time period. Detection, handling and reporting of message time-outs, will be defined and documented when the protocols stack is defined.

Both TM and CSP:

1. report time-out using Alarm messages,
2. re-transmit the messages that were not confirmed in expected time frame,
3. maintain re-transmission counter, and
4. use the re-transmission counter to throttle Alarm generation.

### 7.2.4 Extremely High Message Rate

TBD: A possibility of (intentional or unintentional) starvation of TM and CSP resources due to an extremely high message rate will be analysed when the protocol and underlying communication infrastructure are selected. Both TM and CSP should implement techniques to prevent starvation of resources.

### 7.2.5 Invalid Configuration or State Transition Request

TM shall verify that only valid and viable configuration requests are sent to CSP.

When a SET Parameters message (i.e. a request to re-configure CSP or perform state transition) is received, CSP checks whether a request can be executed at the specified Activation Time. If a request cannot be executed, CSP discards<sup>1</sup> the received message and generates Response and/or Alarm message (or both) to inform TM that the request has been rejected. Note: It is TBD whether TM wants CSP to generate both Reject and Alarm message.

Examples:

- CSP rejects “create sub-array” request if a sub-array with the same sub-array ID already exists.<sup>2</sup>
- CSP rejects a request to add stations and/or tied-array beams to a sub-array if those stations and/or tied-array beams are already assigned to another sub-array. Resources (stations and tied-array beams) must be explicitly removed from one sub-array before they can be added to another sub-array.<sup>2</sup>
- CSP rejects a request to transition to the low-power state if the specified Component does not implement low-power state.

---

<sup>1</sup> Discarded message is removed from the CSP Input Queue (or any other CSP queue), discarded message cannot be displayed, revoked or deleted.

<sup>2</sup> These rules have been introduced to avoid accidental overwrite of the previously transmitted configuration (it could be removed if deemed unnecessary or undesirable).

Note: TM sends SET Parameters messages in advance of the specified Activation Time, which requires CSP to maintain a queue of future configurations (Activation Queue). A received SET Parameters message is verified taking in consideration requests already placed in the Activation Queue (that are waiting to be activated). For more information related to Activation Queue refer to section 7.6.1.2.

### 7.2.6 Failed Re-configuration or State Transition

If a SET Parameters message (i.e. configuration request or state transition) fails at Activation Time, CSP generates an Alarm message to notify TM.

In the case when a re-configuration affects two or more CSP Components, if one Component fails to re-configure, CSP continues with re-configuration of other Components.

Examples:

- If during power-down of the CSP equipment, one Component does not complete shut-down procedure as expected, CSP reports the problem (generates Alarm) and continues with the power-down procedure.
- If during re-configuration of a sub-array, one of the LRUs fails (e.g. becomes unresponsive or does not activate new configuration as expected), CSP informs TM (generates Alarm), and continues with re-configuration of other LRUs.

If one or more Components affected by the configuration change fail in advance of the Activation Time, other Components are re-configured as scheduled.

*TBD: CSP generates Alarm message for each failure. In addition, CSP could generate a configuration report to report status for all Components and Capabilities affected by the configuration change. It is TBD (by TM Consortium) whether CSP should generate such report (TM may prefer to query status).*

## 7.3 Extensibility and Future Compatibility

In order to allow for extensions and upgrades, version of the protocol is specified in each message generated by CSP.

CSP implements self-describing interface which allows TM to obtain the list of Components and Capabilities, and the list of parameters for each Component and Capability. This allows TM to ‘discover’ newly added parameters, Components and Capabilities.

As shown in Table 7-16 CSP reports to TM the version of the protocol it implements. There is a single instance of the CSP to TM interface per telescope. So far, the need for TM to report the version of the protocol to CSP has not been identified; CSP implements only one version of the protocol at any given time and expects TM to handle that version. It is up to TM to allow for CSP software and interface upgrades.

When a new version of CSP hardware and/or software, which includes a new version of the CSP to TM interface, is deployed, TM will instruct CSP to start using the new version, which may include a request to power-cycle (turn off-on) or reboot a particular piece of equipment (e.g. CSP LMC server) or to restart a software process. The exact sequence of commands will be defined by CSP. When the upgrade procedures are defined they will be described in the Section 7.6.8.

## 7.4 Naming Conventions

TBD: Naming conventions to be used on this interface will be derived from the SKA naming conventions to be defined in the document “SKA TM to Element LMC Interface Guidelines” [\[RD4\]](#).

The following naming conventions will be defined in this ICD:

1. CSP Component IDs (hardware and software Components),
2. CSP Capability names and IDs,
3. CSP Parameter names (for common parameters),
4. CSP Modes and states,
5. Message types and names,
6. Names and IDs for CSP monitoring points ,
7. Alarm severity
8. Software and firmware versions, etc.

## 7.5 CSP\_Low Modes and States

Document “SKA TM to Element LMC Interface Guidelines” [\[RD4\]](#) defines the set of modes and states used in the SKA1 telescopes. CSP adopts and implements the SKA1 standards wherever possible and practical. At this time the definition of the common SKA modes and states is not finalised. When the definition of the SKA modes and states is finalized, any non-compliance with the SKA standards will be clearly indicated in this ICD and justification will be provided (for example if a particular state does not apply for a particular CSP Component or Capability).

In the absence of the system-level documentation the definition of mode and state is provided here:

- **Mode** - The term ‘mode’ is used for parameters that indicate the intended mode of operation set by the user or the parent entity. Modes are implemented as read-write parameters. Mode is set by outside authority in order to change behaviour of the entity.
- **State** - The term ‘state’ is used for parameters that indicate state of the entity as derived by the entity itself (in some cases by the entity that reports on behalf of another entity). External entity or authority cannot directly set the state of another entity, but can initiate a state transition by issuing a corresponding command. States are implemented as read-only parameters.

Description of the modes and states in this version of the ICD has been provided with the intention to define a comprehensive set of modes and states that can be used to monitor and control the overall status of the CSP and the status of the entities that CSP consists of, as follows:

1. Sub-elements (Correlator/Beamformer, Pulsar Search Engine, Pulsar Timing Engine and LMC),
2. LRUs (PCBs, computers, compute nodes, GPUs),
3. Other hardware and software components (software processes and applications, FPGAs, microprocessors, etc.),
4. Sub-arrays, and
5. Capabilities (PSS and PST beams).

Table 7-2 provides an overview of the CSP Modes and States. Table 7-3 provides several examples for the CSP Modes, States and status indicators. Following sections provide description for each Mode and State listed in Table 7-2.

Not all entities implement all modes and states.

Table 7-2 CSP\_Low Modes and States - overview. Observing Mode related Modes and States are shaded.

Mode / State + values	Element	Sub-element	LRU	s/w and h/w components	Sub-array	Capabilities
<b>Administrative Mode</b> ENABLED, DISABLED, MAINTENANCE, NOT-FITTED	<sup>1</sup>	<b>Error! Bookmark not defined.</b>	R/W <b>Error! Bookmark not defined.</b>	R/W <b>Error! Bookmark not defined.</b>	<b>Error! Bookmark not defined.</b>	R/W <b>Error! Bookmark not defined.</b>
<b>Observing Mode</b> IMAGING, PSS, PST, IDLE	-	-	-	-	R/W	R/O
<b>Control Mode<sup>2</sup></b> CENTRAL, LOCAL	R/W Optional	R/W Optional	R/W Optional	R/W Optional	-	-
<b>Simulated Mode<sup>2</sup></b> REAL, SIMULATED	R/W Optional	R/W Optional	R/W Optional	R/W Optional	-	-
<b>Test Mode<sup>2</sup></b> NORMAL, TEST	R/W Optional	R/W Optional	R/W Optional	R/W Optional	-	-
<b>Operational State</b> INITIALIZING, READY, FAILED, OFF_DUTY, GOING_OFF_DUTY, GETTING- READY, SHUTTING-DOWN, OFF, UNKNOWN	R/O	R/O	R/O	R/O	-	-
<b>Health State</b> OK, DEGRADED, NOT_AVAILABLE	R/O	R/O	R/O	R/O	R/O	R/O
<b>Usage State</b> IDLE, USED	R/O	R/O	R/O	R/O	-	-
<b>Redundancy State</b> ACTIVE, STANDBY	R/O	R/O	R/O	R/O	-	-

<sup>1</sup> It seems that Administrative Mode at Element and sub-element level will be implemented by TM only, i.e. that CSP is not aware of the Administrative Mode. CSP may make provision for TM to enable/disable individual components and capabilities, the need for such parameter will be considered during detailed design phase.

<sup>2</sup> Not all entities implement this as read-write parameter, but all entities shall implement at least read-only parameter. Not all entities implement LOCAL Control Mode and TEST mode. Simulators implements only SIMULATED mode (and shall be able to report mode as r/o parameter). Some real devices may implement SIMULATED mode and allow user to set either REAL or SIMULATED mode of operation.

**Table 7-3 Examples of CSP\_Low Mode and State reporting. Note that these examples are also applicable to the following sub-elements: Correlator-Beamformer (CBF), Pulsar Search Engine and Pulsar Timing Engine.**

	CSP is initializing	CSP is in the normal operational mode, no active observations	CSP is in the normal operational mode, at least one observation is in progress	CSP Health is degraded	CSP is off-duty
<b>Administrative Mode</b>	n/a <sup>1</sup>	n/a	n/a	n/a	n/a
<b>Observing Mode</b>	No sub-arrays	Observing mode for all sub-arrays is IDLE. Health for all sub-arrays is OK.	List of sub-arrays with health and observing mode for each sub-array. Health for all sub-arrays is OK. At least one sub-array is not IDLE.	List of sub-arrays, with health and observing mode for each sub-array; one of subarrays is DEGRADED.	No sub-arrays
<b>Control Mode</b>	CENTRAL	CENTRAL	CENTRAL	CENTRAL	CENTRAL
<b>Simulated Mode</b>	REAL	REAL	REAL	REAL	REAL
<b>Test Mode</b>	NORMAL	NORMAL	NORMAL	NORMAL	NORMAL
<b>Operational State</b>	<b>INITIALIZING</b>	<b>READY</b>	<b>READY</b>	READY	<b>OFF-DUTY</b>
<b>Health State</b>	NOT-AVAILABLE	OK	OK	<b>DEGRADED</b>	NOT-AVAILABLE
<b>Usage State</b>	IDLE	<b>IDLE</b>	<b>USED</b>	USED	IDLE
<b>Redundancy State</b>	ACTIVE	ACTIVE	ACTIVE	ACTIVE	ACTIVE

<sup>1</sup> Most probably CSP is not aware of the Administrative Mode, this row is provided for completeness and may be removed in the next releases.

### 7.5.1 Administrative Mode

The concept of administratively assigned mode vs. Operational State is often used in industry, in particular in telecommunications and networking. However, in telecommunications and networking the intent is to change the behaviour of the interface or device, while in the SKA telescopes, at least as defined so far, this parameter is used to disable or limit the use of the Element (or a sub-set of an Element) for normal observing. For example:

- SKA operations personnel (in the further text referred to as operator) may declare a sub-set of CSP equipment as DISABLED in order to indicate that it should not be used for observing.
- Operator may set a sub-set of the CSP equipment in MAINTENANCE mode to indicate that the equipment should not be used for normal observing but can be used for observing in maintenance sub-array.

**Table 7-4 Administrative Mode Values**

Administrative Mode Value	Priority	Description
<b>ENABLED</b>	1	CSP (entity) can be used for observing. During normal operations CSP (entities) shall be in this mode.
<b>MAINTENANCE</b>	2	CSP (entity) is reserved for maintenance and cannot be part of 'normal' observations, but can be used for observing if part of the 'Maintenance Sub-array'. TM may define a sub-set of commands that operator and TM can issue for entity in MAINTENANCE mode. The restriction is enforced by TM. <i>TBD: Is it a requirement for the CSP (and its sub-elements and Components) to enforce such restriction? Should the CSP discard commands 'not allowed' in MAINTENANCE mode? Is it a requirement that the CSP (entity) should change its behaviour when in MAINTENANCE mode?</i>
<b>DISABLED</b>	3	CSP (entity) cannot be used for observing.
<b>NOT-FITTED</b>	4	Administrator (operator) declared the CSP (entity) as NOT-FITTED. TM shall not send commands or queries to the CSP (entity) while in this mode. (This mode has been added for historical reasons – the first version of SKA Level 1 requirements included a requirement to be able to declare equipment as disabled.)

Figure 7-2 Administrative Mode Transitions

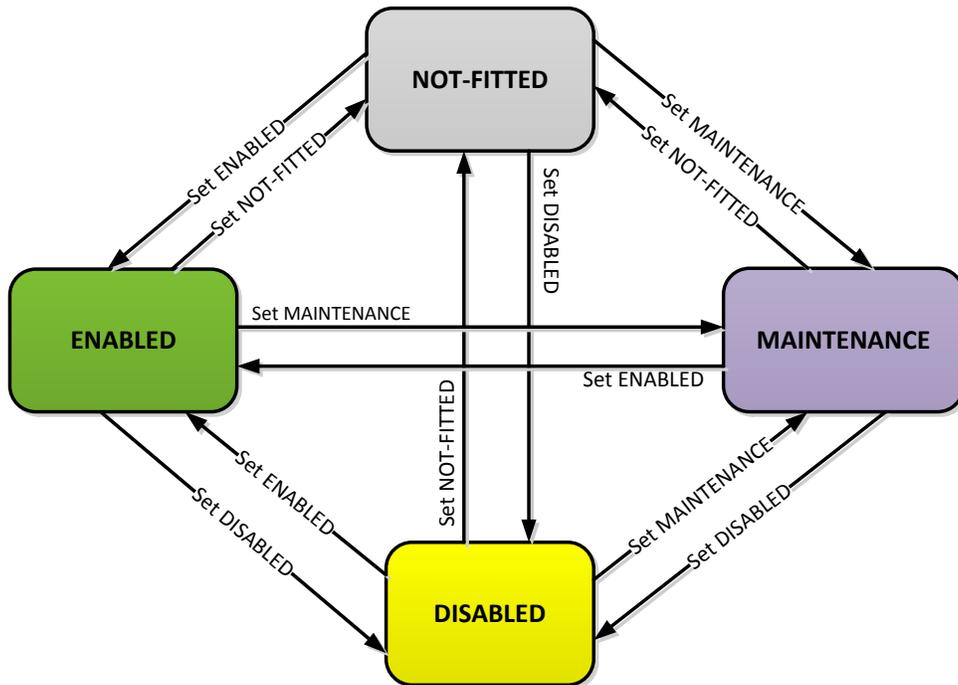


Figure 7-2 shows that there are no restrictions on Administrative Mode transitions. Operator may request transition from any Administrative Mode to any other Administrative Mode.

Administrative Mode set for an entity also applies for all the children (in the monitor & control hierarchy). Figure 7-3 provides an example for an Administrative Mode transition for entities that have children.

As shown in Figure 7-2, the change does not apply to a child, if the Administrative Mode of the child has a higher priority than the new Administrative Mode assigned to the parent.

Table 7-4 above lists the values for the Administrative Mode and assigns a priority to each value.

TM shall allow operator to set the Administrative Mode for:

- a) CSP\_Low as a whole,
- b) CSP\_Low sub-elements (CBF, PSS, PST and LMC),
- c) LRUs,
- d) software and hardware components (granularity TBD based on the design and functionality provided).

It is to be determined (TBD) whether Administrative Mode can be assigned to Sub-arrays and Capabilities.

***At this time it seems that the Administrative Mode, if defined for SKA Telescopes at all, will be implemented by TM only and that CSP will not be required to implement and report Administrative Mode. The content of this chapter (and the whole document) will be updated accordingly when the system-wide states and modes are defined with more certainty and the requirements for CSP are defined.***

To be determined (TBD):

- a) *Is the purpose of the Administrative Mode to define behaviour of TM or of CSP (and its Sub-elements, Components, and Capabilities)? Are there requirements that CSP (and its Sub-elements, Components, and Capabilities) have to change behaviour in different Administrative Modes?*
- b) *Is it a requirement that the CSP and its Sub-elements, Components and Capabilities have to be aware of the Administrative Mode?*
- c) *Should the Administrative Mode be forwarded to Elements at all, or should the Administrative Mode be implemented and maintained by TM only?*

*TM could implement the following functionality:*

- *TM shall not send commands to the entity (CSP or parts of the CSP) if the Administrative Mode of the entity is NOT-FITTED.*
- *TM shall NOT use for observing entity (CSP, or a sub-set of the CSP) if the Administrative Mode of the entity is DISABLED.*
- *TM shall not use for normal observing entities (CSP or parts of the CSP) where Administrative Mode is MAINTENANCE (they can be used only in maintenance sub-array).*

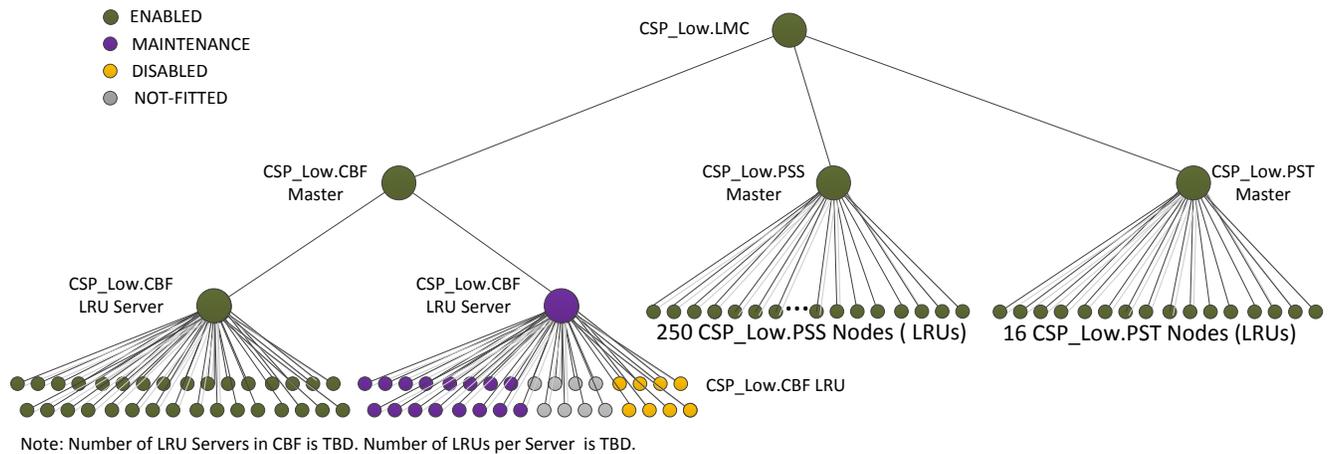
*If it is a requirement that the CSP (or any of its parts) has to be aware of the Administrative Mode and has to change its behaviour as a consequence of the Administrative Mode change, the following shall apply:*

- 1) *CSP (and its Sub-elements and Components) shall implement Administrative Mode as a read-write parameter.*
- 2) *Change of the Administrative Mode shall not cause transition of the Operational State.*
- 3) *In any of the Administrative Modes listed in Table 7-4, CSP can report any Operational State.*
- 4) *CSP shall persist the Administrative Mode over shut-down/restart/reboot (This would require that each entity remembers and restores its Administrative Mode after power-up/restart/reboot).*
- 5) *The default Administrative Mode assigned to equipment in production shall be MAINTENANCE. When first deployed, all equipment comes up in MAINTENANCE mode and shall be set in ENABLED mode after commissioning.*
- 6) *Entities that implement Administrative Mode as a read-write parameter shall store and upon request report the timestamp for the last Administrative Mode change, along with the identifier of the entity which requested the change and the reason or other textual description (if it was provided).*
- 7) *Command 'Set Administrative Mode' shall be used to set Administrative Mode for a specific Component and its sub-Components; however, sub-Components already set to a higher-priority Administrative Mode shall not be affected. See Figure 7-3 for examples.*
- 8) *A child (in the monitor and control hierarchy) shall not be assigned Administrative Mode with a lower priority than its parent. For example, if a sub-element is in MIANTENANCE mode its Components cannot be ENABLED. See Figure 7-3 provides an example.*

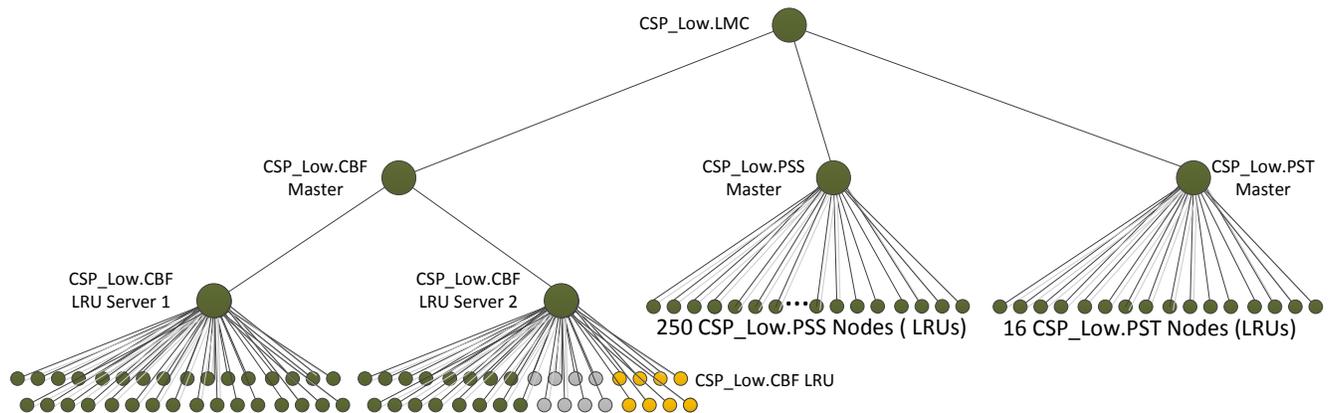
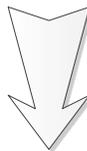
*Further limitations may be imposed by the CSP design. Details will be defined as the work on the CSP design progresses. When defined, all rules and limitations will be documented in this ICD.*

*If it is a requirement that CSP has to be aware of the Administrative Mode, further consideration is required to determine if the Administrative Mode can be supported for sub-arrays and other Capabilities. In CSP a single hardware component (e.g. FPGA) may be used by two or more Capabilities (sub-arrays or beams); a change of the Administrative Mode at Capability level does not necessarily result in the Administrative Mode change for a hardware or software Component used by that Capability.*

**Figure 7-3 Example for the Administrative Mode transition: CSP\_Low.CBF Admin Mode transition from MAINTENANCE to ENABLED does not override the higher priority Administrative Mode previously assigned to the children.**



**Received command:**  
**CBF LRU Server 2 Administrative Mode= ENABLED**  
**Note that Admin. Status of DISABLED and NOT-FITTED LRUs does not change.**



## 7.5.2 Test Mode

CSP, or some of its parts, may implement TEST mode that implements a different set of commands and different behaviour than the NORMAL operating mode. An entity that implements Test Mode shall make provision for TM (and the engineering interface) to request transition from NORMAL to TEST mode and vice versa.

A set of commands available in the NORMAL and TEST mode may not be the same; CSP self-description shall report the set of commands supported in the current mode. CSP shall provide documentation that clearly describes functionality and purpose of the TEST mode.

TM may request CSP (entity) to transfer to TEST mode or to NORMAL operating mode, but TM shall not assume that the transition has occurred until the desired state is reported by the CSP (entity); the command may be delayed or may fail.

Test Mode setup shall be preserved over shut-down, restart and reboot.

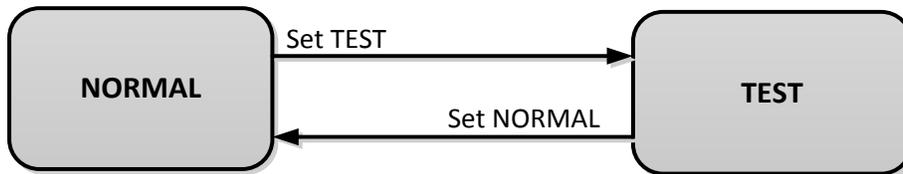
CSP shall report Test Mode for:

- a) CSP as a whole,
- b) sub-elements,
- c) LRUs.

**Table 7-5 Test Mode Values**

Test Mode Value	Description
<b>NORMAL</b>	Normal mode of operation to be used for normal observing.
<b>TEST</b>	CSP (entity) behaviour and/or set of commands differ for the NORMAL operating mode. The Element documentation shall provide detailed description.

**Figure 7-4 Test Mode Transitions**



An example for the Test Mode that may be implemented is a mode where CSP processes input data generated by the in-built test vector generator (instead of input received from antennas). When set in Test Mode, CSP reports all mode and state indicators and other parameters as in the NORMAL Mode (operating state, health state, observing mode, etc.). Different CSP sub-elements, LRUs and Capabilities may implement different type of Test Mode; some components and Capabilities may implement more than one Test Mode, in which case CSP must be able to report more than one Test Mode (e.g. as TEST1, TEST2). CSP documentation shall clearly describe how CSP functionality in Test Mode differs from the Normal Mode and when and why the Test Mode may be used (purpose, parameters, output products).

### 7.5.3 CSP\_Low Observing Mode

1. Observing Mode is a parameter of a sub-array and cannot be directly assigned to other Capabilities (e.g. beams). TM sets the Observing Mode for a sub-array; CSP selects the equipment to be used for the particular Observing Mode (from the resources previously assigned to the sub-array).
2. CSP shall be able to report the Observing Mode(s) and relevant parameters for each subarray.

3. CSP shall be able to report Observing Mode(s) and relevant parameters for each entity that performs signal processing or other function required for execution of an Observing Mode.
4. Observing Mode IDLE is the default mode in which a sub-array is placed when created.
5. When an observation ends, the sub-array is placed in IDLE mode.
6. If an Observing Mode uses only a subset of Capabilities that belong to a sub-array, unused Capabilities are placed in IDLE mode.
7. The exact setup for the CSP Capabilities and Components in IDLE mode is implementation-dependent and will be defined during detailed design phase.

**Table 7-6 CSP\_Low Observing Modes (parameter of a Subarray)**

Sub-array Observing Mode	Concurrency Any or all Observing Modes marked with X can be executed concurrently in the same sub-array.
<b>IMAGING</b>	X
<b>PULSAR SEARCH</b>	X
<b>PULSAR TIMING</b>	X
<b>IDLE</b>	

Section 7.7.1 provides the list of parameters for each Observing Mode.

A sub-array shall be able to transit from any Observing Mode to any other Observing Mode.

Note: Unlike other modes defined in this document (Administrative Mode, Control Mode, Simulated Mode, Test Mode), where an entity can be assigned only one value for each mode at a time, a sub-array may be in more than one Observing Mode at the same time. Some Observing Modes can be operated concurrently in the same sub-array; concurrency is indicated in Table 7-6 above. For this reason mode transition diagram for Observing Mode is not provided.

#### **7.5.4 Control Mode**

Control Mode can be administratively set via TM and, where supported, via CSP engineering interface.

CSP and all its sub-elements and Components shall implement CENTRAL Control Mode.

Optionally, entities may implement LOCAL Control Mode. When in LOCAL Control Mode, an entity cannot be controlled from TM. When in LOCAL Control Mode, the entity can be controlled only via the ‘local’ engineering interface provided by the CSP.

CSP and its entities may allow TM and CSP engineering interface to set Control Mode for:

- a) The CSP as a whole,
- b) sub-elements,

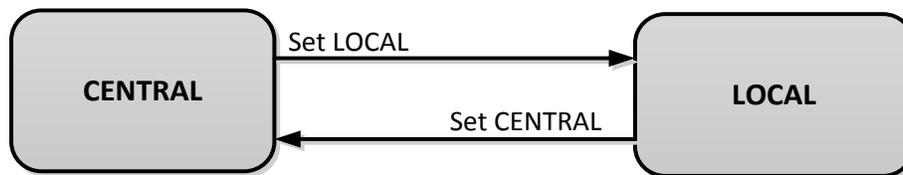
- c) sub-systems (depending on the design),
- d) LRUs.

The Control Mode (CENTRAL vs. LOCAL) setting shall be preserved over shut-down, restart and reboot.

**Table 7-7 Control Mode Values**

Control Mode	Description
<b>CENTRAL</b>	The entity is under TM control.
<b>LOCAL</b>	The entity is under local/manual control and ignores commands and queries received from TM. CSP and its entities shall provide a method for transition from LOCAL to CENTRAL Control Mode which can be triggered from TM in order to override LOCAL setup.

**Figure 7-5 Control Mode Transitions**



### 7.5.5 Simulated Mode

A number of simulators will be developed to be used during testing and integration.

CSP and its sub-elements and Components shall be able to report whether they are operating in REAL or SIMULATED mode.

An entity which operates only as a simulator shall always report its mode as SIMULATED.

Some entities may implement this parameter as a read-write parameter to allow TM (and the local operator via the engineering interface) to set the real CSP (entity) to act as a simulator. If that is the case, TM or local operator may issue a command to cause transition from REAL to SIMULATED mode of operation, and vice versa. TM shall not assume that transition has been completed until the Element itself reports the change (the command may be delayed or may fail).

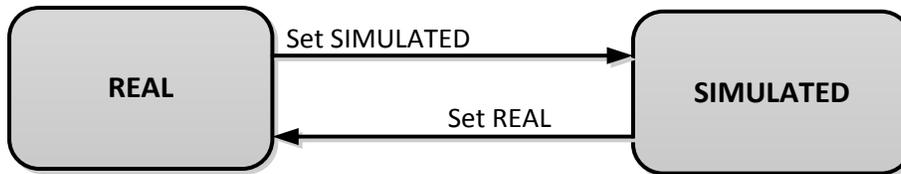
The Simulated Mode shall be reported for:

- a) CSP as a whole,
- b) sub-elements,
- c) LRUs.

**Table 7-8 Simulated Mode Values**

Type Indicator	Description
<b>REAL</b>	A real CSP (entity) is being monitored and controlled by TM.
<b>SIMULATED</b>	A simulator is being monitored and controlled by TM. (Or the real entity which acts as a simulator.)

**Figure 7-6 Simulated Mode Transitions – applies for the entities that implement both REAL and SIMULATED mode**



### 7.5.6 Operational State

**Table 7-9 Operational State Values**

Operational State	Description
<b>INITIALIZING</b>	This is a transient state. Entity (CSP, sub-element, LRU, Component) enters this state upon power-up, reboot and reset. SKA1-CSP_REQ-2478-02 requires CSP to transit to low-power OFF-DUTY state upon start-up. When initialization is successfully completed the entity transfers to OFF-DUTY state. If initialization fails or if a failure is detected that causes the entity to be unusable for all its core functions, the entity shall transfer to FAILED state.
<b>OFF-DUTY</b>	There is a requirement for CSP to enter, upon start-up, a low power mode where power consumption is limited to less than 10% (TBC) of the maximum sustained full load power consumption (see SKA1-CSP_REQ-2478-02). CSP low power mode corresponds to OFF-DUTY state as defined in [RD4]. Exact implementation of OFF-DUTY state is yet to be defined for CSP, its sub-elements and LRUs. For example, for an FPGA based LRU OFF-DUTY may be implemented as a state when the FPGAs are de-programmed. Not all CSP Components implement OFF-DUTY state; for example, CSP. LMC upon initialization transits into READY state, some software components may be launched only when transferring into READY state, etc. Definition of OFF-DUTY state is implementation specific. CSP documentation shall precisely define OFF-DUTY state and related transitions for each entity (CSP/sub-element/LRU/Component).

Operational State	Description
<b>GETTING-READY</b>	This is a transient state. Entity (CSP/sub-element/LRU, Component) enters this state when a command to transfer to READY state has been received. In this state the entity performs necessary steps to bring equipment to READY state. Transition is performed gradually to ensure conformance with SKA1-CSP_REQ-2478-01 [The instantaneous value AC start-up inrush current of the CSP shall not exceed 200% (TBC) of the maximum sustained full load root mean square (rms) current.] Entity (CSP/sub-element/LRU, Component) leaves this state when transition to READY state has been completed. CSP documentation shall clearly define the criteria for transition to READY state for each entity.
<b>READY</b>	This is normal operational state in which the entity (Element/sub-element/LRU/Component) is available for use. An entity may be in READY state although some functions are compromised or not-available; in such case Health is reported as DEGRADED. For each entity documentation shall clearly define when (under which circumstances) the Operational State is reported as READY.
<b>GOING-OFF-DUTY</b>	This is a transient state. An entity (CSP/sub-element/LRU/Component) enters this state when a command to transit to OFF-DUTY state has been received. The entity leaves this state when the transition to OFF-DUTY has been successfully completed.
<b>FAILED</b>	Entity (Element/sub-element/LRU/Component) cannot be used for its core functionality but is able to report its status. <i>In the case when an entity is not able to report its status, the entity that reports on its behalf shall report the Operational State as UNKNOWN.</i> When Operational Status is reported as FAILED, Health Status must be reported as FAILED, and vice versa. For each entity documentation shall clearly define under which circumstances the Operational State is reported as FAILED.
<b>SHUTTING-DOWN</b>	This is a transient state. An entity (CSP/sub-element/LRU/Component) enters this state on TM request, i.e. upon receipt of the command to shut-down. Entity exits this state when the shut-down procedure has been completed.
<b>OFF</b>	Entity (Element/sub-element/LRU/Component) is able to report its status but the parts of the entity that perform its core functionality are OFF. This presumes that the shut-down command is implemented to shut down everything but leave the monitor & control functionality running. Not all entities implement this state.
<b>UNKNOWN</b>	This state is reported by an entity that reports on behalf of another entity (Element/sub-element/LRU/Component). The Operational State is reported as UNKNOWN for a non-responsive entity. For example, TM may report status of CSP as UNKNOWN when connectivity with CSP.LMC is lost.

The CSP shall be able report Operational State for:

- a) the CSP as a whole,
- a) sub-elements (CBF, PSS Engine, PST Engine, LMC),
- b) LRUs,
- c) software and hardware components (granularity TBD based on the design and functionality provided).

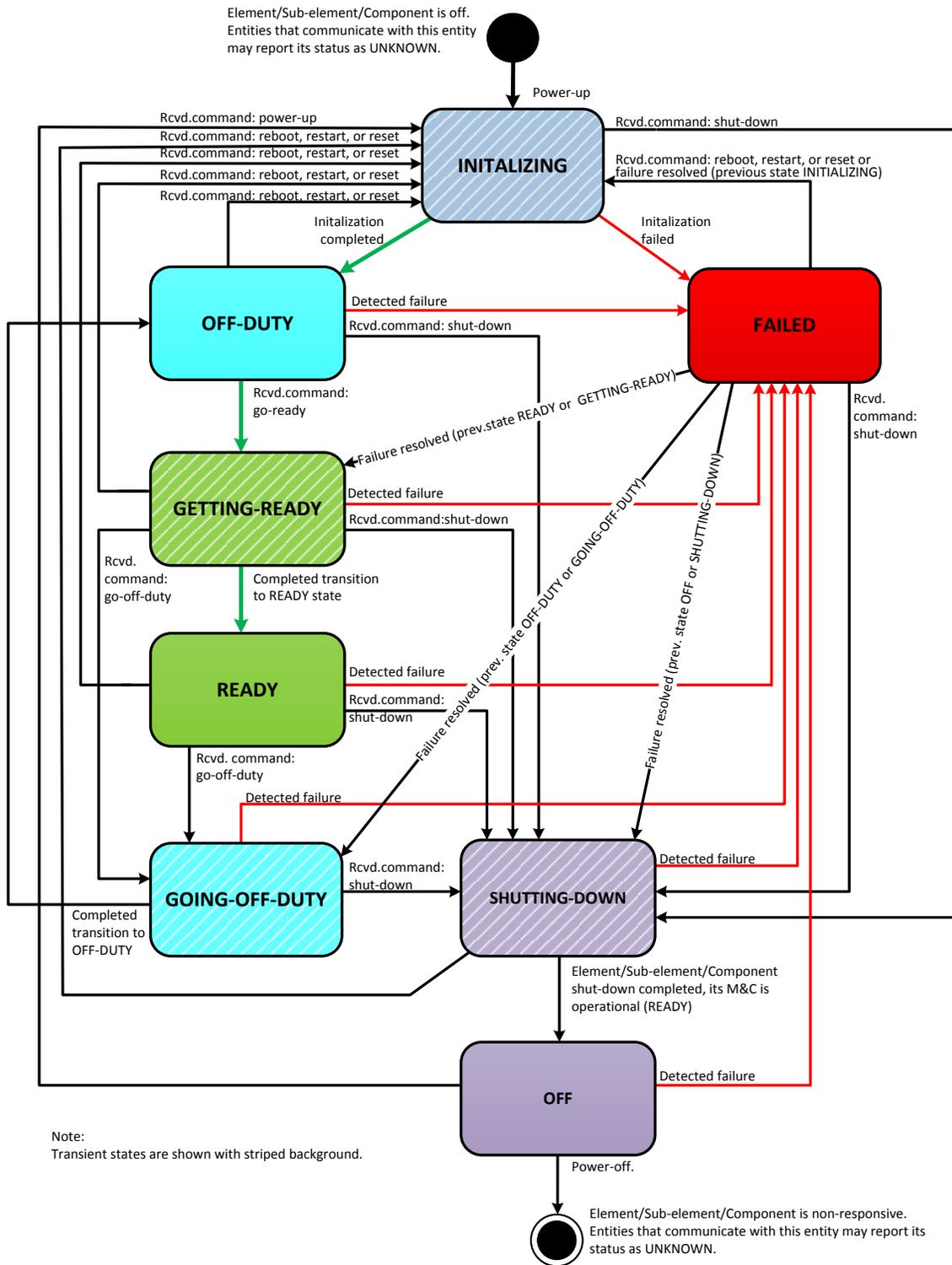
TM cannot directly set the Operational State of an entity, but can issue commands to request state transitions.

Discussion:

a) Transient states may be more accurately reported if both the start and target states are initiated, as in: transition-offDuty-to-ready, transition-ready-to-offDuty, transition-ready-to-off, transition-offDuty-to-off.

b) CSP will consist of many components; need to define the criteria for ending transition states, i.e. when to declare that transition failed. For example: for a transition from off-duty to ready, if after TBD seconds some of the components do not enter ready state, report CSP state as ready but degraded (TBD).

**Figure 7-7 Operational State Machine**



### 7.5.6.1 Components That Do Not Implement OFF-DUTY State

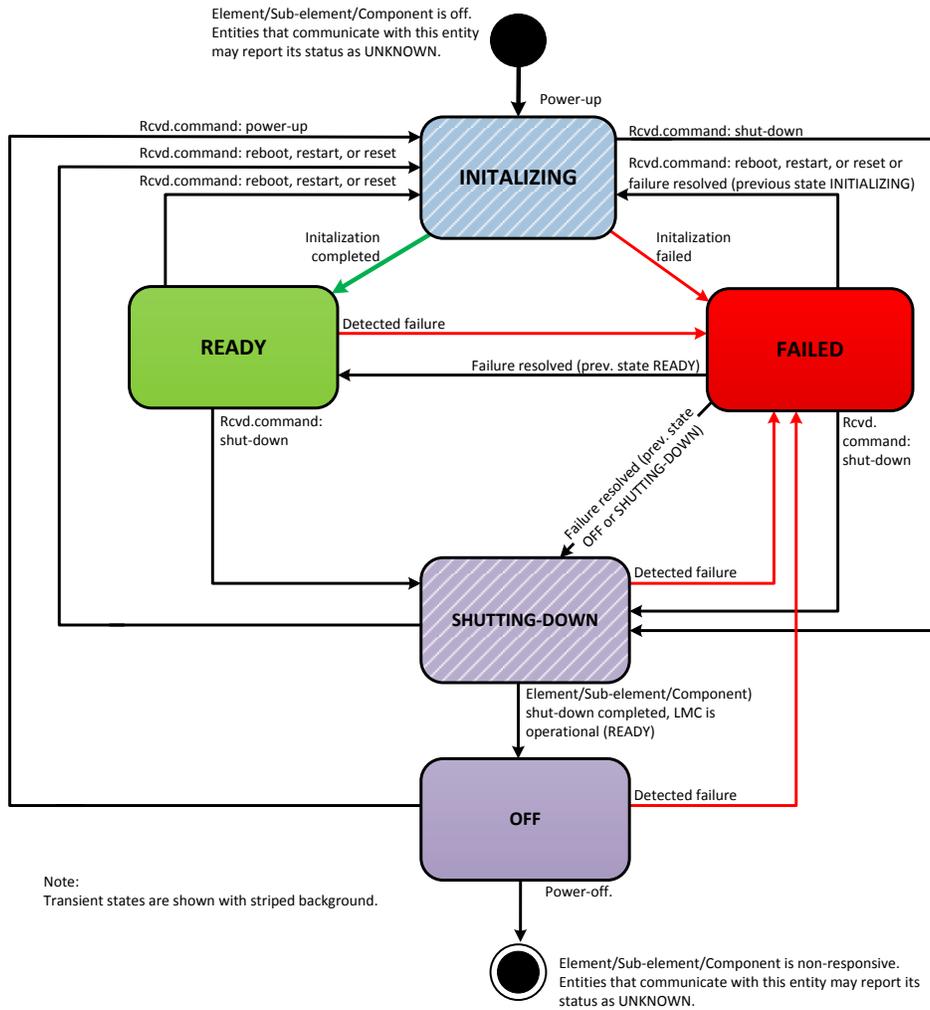
Some CSP components do not implement OFF-DUTY (low-power) state. In particular, components that implement monitor and control functionality may after INITIALIZATION become fully operational and enter READY state.

- Upon power-up CSP.LMC performs initialization, becomes fully operational and enters READY state, so that it can monitor and control state of the CSP equipment. While CSP.LMC itself becomes READY, it keeps the rest of CSP in OFF-DUTY state, until the command is received from TM to transition to READY state. Upon successful INITIALIZATION Operational Status of CSP.LMC is reported as READY and the overall CSP Operational Status is OFF-DUTY.
- The same applies for the monitor and control components of the CSP sub-elements. Upon power-up CSP Sub-element Master initializes and transitions to READY state, but keeps the rest of the sub-element OFF-DUTY until a command to transition to READY state is received from CSP.LMC. Upon successful INITIALIZATION, Operational Status of the Sub-element Master is READY and Operational Status of the sub-element is OFF-DUTY.
- Custom made LRUs developed for the FPGA based CBF (PCBs with FPGAs) may implement OFF-DUTY state so that upon power-up on-board micro-processor performs initialization and becomes ready to report board status, while daughter boards are not powered up and/or FPGAs are not programmed. Transition to READY state is performed when a command is received from the Sub-Element Master.

The staged transition to READY state will be implemented by CSP.LMC and CSP Sub-element Masters, as follows:

- 1) CSP.LMC sends 'go-ready' command to the sub-elements in sequential order, when one sub-elements reports that transition to READY state has been completed, CSP.LMC triggers transition for the next sub-element by transmitting the appropriate command.
- 2) Each Sub-element Master performs transition to READY state in compliance with SKA1-CSP\_REQ-2478-01 [[RD8](#)], which defines the limit on inrush current).

**Figure 7-8 Operational State Machine for Components that do not implement OFF-DUTY state**



### 7.5.7 Health State

Health State indicates availability of the CSP (entity) to produce output products.

If reported for the entities that do not produce output products, Health State shall describe their ability to perform the core functionality.

The CSP shall maintain and report the Health State for:

- a) the CSP as a whole,
- b) Sub-elements (sub-systems)
- c) LRUs
- d) s/w and h/w components (as applicable)
- e) Sub-arrays and
- f) Capabilities.

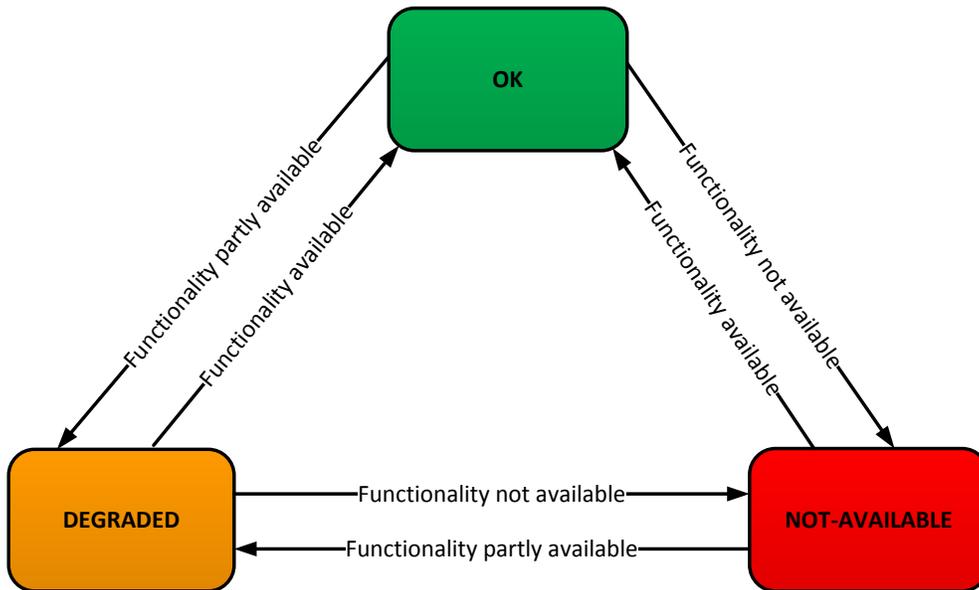
The CSP (entities) shall implement Health State as monitoring point; changes shall be reported to TM.

**Note: Although the words used to describe Health Status are the same as those used to report CSP availability, the CSP reported Health State should not be confused with the CSP availability. Health State reports the status of CSP equipment as detected and perceived by CSP, while the actual availability is determined based on many more parameters and cannot be determined in real-time.**

Table 7-10 Health State Values

Health Status Indicator	Description
<b>OK</b>	CSP (entity) is available for use and able to produce output products.
<b>DEGRADED</b>	CSP (entity) is partly available. For example, a sub-array Health State should be reported as DEGRADED if in the IMAGING mode output products can be provided for a sub-set of baselines.
<b>NOT-AVAILABLE</b>	<p>Element (entity) is completely unable to produce valid output.</p> <p>Example: A compute node that produces output (performs search) for a particular Pulsar Search beam is switched-off, consequently the Health State for that beam shall be reported as NOT-AVAILABLE. The Health State for the sub-array to which the failed beam belongs is reported as DEGRADED. (Unless the failed beam is the only PSS beam in the sub-array and the Observing Mode is Pulsar Search, in which case the Health Sate is reported as NOT-AVAILABLE).</p> <p><i>If the Operational State (Table 7-9) of the Element is not READY, Health State must be reported as NOT-AVAILABLE.</i></p>

Figure 7-9 Health State Transitions



### 7.5.8 Usage State

Usage State indicates whether CSP (entity) is used for observing (i.e. for its core functionality).

It is to be determined whether the entities that are not directly involved in the observing shall report Usage State (for instance, LMC is always used).

The CSP shall be able to report the Usage State for:

- a) the CSP as a whole,
- b) sub-systems (depending on the design),
- c) LRUs,
- d) software and hardware components (granularity TBD based on the design and functionality provided).

Note that Usage State is not reported for subarrays and capabilities. Usage State for subarrays is reported via Observing Mode. If the Observing Mode is IDLE, subarray is not used, if Observing Mode is anything but IDLE, subarray is used.

The same applies for Capabilities. TM cannot assign Observing Mode directly to a Capability; TM assigns Capability to a subarray and then sets Observing Mode for the subarray. CSP assigns Observing Mode for the Capabilities based on the TM instructions.

Table 7-11 Usage State Values

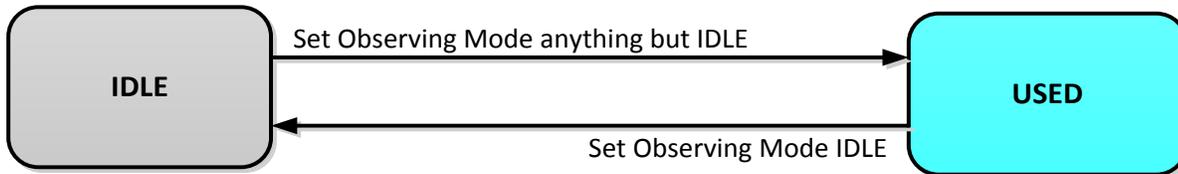
Usage Indicator	Description
USED	CSP (entity) is used for observing.
IDLE	CSP (entity) is not used for observing.

If the Usage Status is reported as USED, CSP (entity) shall be able to report the list of sub-arrays that use the CSP (entity).

Status may be reported as USED even for a FAILED Element or Component to indicate that the failed equipment should be used for current observation(s).

Component or Capability that belongs to a USED sub-array may be IDLE if the current Observing Mode uses only a sub-set of equipment and Capabilities assigned to the sub-array.

**Figure 7-10 Usage State Transitions**



### 7.5.9 Redundancy State

CSP, and all its sub-elements and Components shall report the Redundancy State.

Entities that do not provide redundancy shall always report Redundancy State as ACTIVE.

The entities that implement redundancy shall provide detailed documentation regarding the type of redundancy, transition mechanism, election mechanism (how does entity decide whether it is ACTIVE or STANDBY), etc.

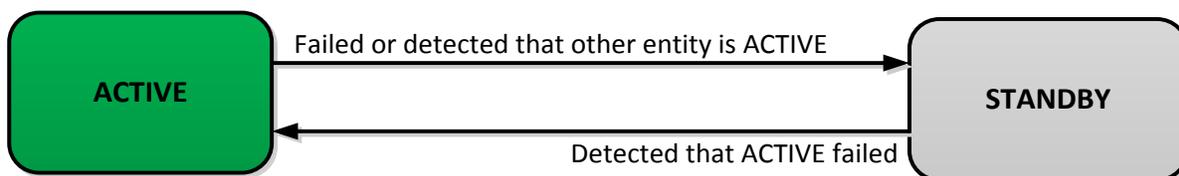
Transition from ACTIVE to STANDBY usually occurs when the ACTIVE fails. TM may trigger the transition by temporarily disabling the ACTIVE entity (e.g. TM can request restart or reboot of the ACTIVE entity).

Note: Operational State (see Table 7-3) shall be reported both for the ACTIVE and STANDBY. An entity can report state as READY & ACTIVE or READY & STANDBY. In the latter case, READY means that the STANDBY is ready to take over if the ACTIVE fails.

**Table 7-12 Redundancy State Values**

Redundancy Indicator	Description
<b>ACTIVE</b>	CSP (entity) is active.
<b>STANDBY</b>	CSP (entity) is providing backup capacity to be used when the ACTIVE fails.

**Figure 7-11 Redundancy State Transitions**



Redundancy can be implemented as cold, warm or hot.

- a) Cold redundancy means that a spare Component is in place and can take over functionality of the active component when turned on. If ACTIVE Component fails the service is interrupted until STANDBY is turned on, initialized and configured. Usually cold standby component is not able to report its state (is not on).
- b) Warm redundancy means that a STANDBY component has been installed, is up and running, and ready to take over function of the ACTIVE component. A warm STANDBY is not fully aware of the configuration and mode of operation of the ACTIVE Component. In general, when a warm STANDBY becomes ACTIVE there is an interruption of service until the new ACTIVE receives configuration parameters. A warm STANDBY shall be able to report its status.
- c) Hot standby means that STANDBY is ready to become ACTIVE with minimal interruption of service. A hot STANDBY shall be able to report its status.

Components that require redundancy will be identified during detailed design phase based on RAMS analysis.

*So far it seems that redundancy will be required only for components that provide monitor & control, namely for the CSP.LMC and Sub-element Masters. If CSP.LMC is down, the whole CSP is down. (Please note this is a memo and not a requirements specification, this statement should not be considered as final and binding).*

## 7.6 Operations

This section describes operational concepts, i.e. how this interface is used.

Details of the protocol are defined in the section 7.8.

### 7.6.1 Set CSP Parameters

CSP makes provision for TM to control CSP (and its Components and Capabilities) by setting (changing the values) of CSP parameters.

Protocol between CSP and TM is still to be defined (TBD), for simplicity, this version of the ICD, assumes that:

1. A single message type (SET Parameters) is used to set any CSP parameter that allows for write access.
2. Write-only parameters are used to request actions, such as reboot and restart.
3. Write-only parameters are also used to request state transitions, for example: power-down, go-to-ready (to request transition to READY state) and go-off-duty (to request transition to OFF-DUTY low-power state).

SET Parameters message contains:

1. Activation Time (optional),
2. ID of the entity (Component or Capability) to be configured,
3. Parameter name and value.

A single message can be used to set two or more parameters.

A single message may be used to set parameters for more than one Component and/or Capability.

This version of the document does not impose a limit on the number of Components, Capabilities and parameters that can be set using a single message.

*It is to be defined (TBD) by the TM Consortium whether CSP should generate Response or Alarm message (or both) to report that SET Parameter message has been rejected.*

#### 7.6.1.1 Activation Time

The scope of the message SET Parameters may vary from a request to change the value of a single parameter to a massive re-configuration, required at the beginning of a scheduling block, in a sub-array that consists of hundreds of stations, and requires synchronized re-configuration of the stations, correlator, beam-formers, pulsar search engine, pulsar timing engine and science data processor. To expedite configuration changes, TM distributes Observing Mode changes in advance.

CSP makes provision for TM to specify the Activation Time in the SET Parameters message. The Activation Time can be specified in any SET Parameters message, but is of particular importance for the Observing Mode changes, as they require synchronization of the stations, CSP and SDP.

The Activation Time is the time at which CSP should start re-configuration of CSP equipment.

A SET Parameters message that does not specify the Activation Time is executed as soon as possible.

A SET Parameters message received past its Activation Time, is executed as soon as possible.

CSP does not impose limit on how much in advance a SET Parameter message can be accepted. (However, *CSP might warn TM, i.e. generate a low severity Alarm, when a SET Parameters message is received with the Activation Time in distant future (e.g. in a month), suspecting that the message was transmitted due to an error.*

CSP distributes (forwards) SET Parameter message to all affected Components (servers, software processes, LRUs) and Capabilities in advance. At the Activation Time all affected Components and Capabilities perform re-configuration independently and simultaneously. Note that after Observing Mode changes, different CSP Components may start signal processing and generation of output data at different times, which may result in incomplete data sets received by SDP at the beginning of a scan.

SKA system requirements allocate 30 seconds for the telescope Observing Mode change. It is a CSP requirement that the Observing Mode change shall be completed within 25 seconds of the receipt of TM message, which leaves 5 seconds for TM to forward the message to CSP.

*The time required for updating parameters/settings other than the Observing Mode is not specified in the requirements, and is TBD. The agreed-upon timings will be documented in this ICD.*

#### 7.6.1.2 Activation Queue

CSP maintains a queue where SET Parameter messages are waiting to be activated.

CSP makes provision for TM to:

1. List content of the CSP Activation Queue,
2. Remove a particular message from the CSP Activation Queue (revoke a message) , and
3. Flush the CSP Activation Queue (remove all messages).

In the Activation Queue, messages are sorted in chronological order, starting from the configuration to be activated first.

*The maximum supported 'depth' of the CSP Activation Queue will be determined empirically. In general, CSP will have capacity to handle many future configurations for each sub-array. However, as discussed below, adding many future configurations to the Activation Queue may affect CSP performance, in particular ability to perform rapid configuration changes. Adding entries at the front or middle of the Activation Queue requires CSP to re-examine all previously received messages scheduled after the newly added configuration; this can slow down re-configuration and result in rejection of previously accepted configurations.*

When a SET Parameter message is received:

- a. Based on the content of the Activation Queue, CSP verifies that the parameters can be set as requested at the specified Activation Time.
- b. If the received SET Parameters request is valid and can be activated at the Activation Time, it is added to the CSP Activation Queue.
  - If the new message has been added to the end of the Activation Queue, no other action is required.
  - If the message has been added in front of previously received (and accepted) messages, CSP must verify that those messages can be executed as specified. This may result in rejection (discard) of previously received and accepted SET Parameters messages.

**Example: Create sub-array messages received out-of-order:**

At 08:10:

CSP Activation Queue is empty, there are no messages waiting to be activated.

CSP receives SET Parameters message:

Activation Time=10:15 Create sub-array A, add stations 1 to 30 to sub-array A, set Observing Mode=imaging.

- CSP verifies that at 10:15 stations 1 to 30 will not be used by another sub-array, accepts the message and adds it to the Activation Queue.

At 09:30:

CSP receives SET Parameters message:

Activation Time= 10:00 Create sub-array B, add stations 1 to 40 to sub-array B, set Observing Mode = imaging.

- CSP verifies that at 10:00 stations 1 to 40 will not be used by any other sub-array, accepts the message and adds it to the Activation Queue.
- This message is placed in front of the message scheduled at 10:15. CSP must re-examine previously received message with Activation Time = 10:15.
- CSP finds that at 10:15 stations 1 to 30 will be used by sub-array B, and must reject a request to add those stations to sub-array A.
- CSP removes 10:15 message from the Activation Queue, discards the message and notifies TM.

## 7.6.2 Get CSP Parameters (Query)

TM uses a Query (GET Parameters message) to obtain CSP configuration parameters and status. A Query identifies an entity (Element instantiation, sub-element, Component or Capability) and, optionally, the parameters to be reported by the CSP. The default set of parameters, to be reported if the entity and/or parameters are not identified, will be defined for each CSP entity.

Upon receipt of a Query message, CSP generates a Response message that contains the list of parameters and their values, and sends Response to the originator of the Query.

In the response to a Query, CSP reports status both for the read-write parameters (which can be set using a SET Request) and for the read-only parameters (which cannot be set using a SET Request).

CSP allows for interrogation of CSP parameters at all levels of the CSP Component and Capability hierarchy.

Queries can be used to 'discover' the CSP Component and Capability hierarchy as well as configurable parameters and their status.

## 7.6.3 Monitor Points

CSP monitor points are parameters (attributes) that are periodically monitored.

CSP reports status of monitor points:

- on request, i.e. in response to TM Query (Get Parameters) message,
- periodically and/or
- when the monitored value crosses a predefined threshold.

Typical monitor points for the CSP equipment are:

1. Internal temperature,
2. Voltage,
3. Power,
4. Communication path status,
5. Health Status derived based on the built-in FMECA analysis.

Table 7-13 lists the parameters assigned to a CSP Monitor Point.

Table 7-13 Table 7-13 does not define the destination address for the Monitor Point reports as a parameter of an individual Monitor Point. All CSP Monitor Point Reports are sent to the same destination (address); the destination for Monitor Point Reports is defined (set) at the CSP Element level (for each CSP instantiation independently) and applies for all Monitor Points in a particular CSP instantiation. The default destination address is loaded from the configuration file during CSP start-up. This allows Monitor Point reporting to resume as soon as a CSP Element instantiation comes up after reset, restart or power-down.

Monitor Points Reports are generated in all modes and states, as long as CSP is able to report status.

**Table 7-13 CSP Monitor Point Parameters (accessible using SET/GET Parameter messages)**

Monitor Point Parameter	Access	Type	Range	Description
Monitor Point ID	r/o	ASCII String	Unique MP ID	Unique MP ID is assigned to each CSP monitor point.
Object ID	r/o	ASCII String	Object ID	Object ID for the Component being monitored.
Parameter Name	r/o	ASCII String	ASCII String	Name and the last detected/derived value of the parameter being monitored.
Parameter Value	r/o	MP dependent	MP dependent	
Monitoring frequency	r/w	Integer	TBD	How often CSP checks the MP value. If 0 monitoring of the parameter is OFF.
Reporting frequency	r/w	Integer	TBD	How often CSP reports the MP status (value). If 0 periodic reporting is OFF.
Threshold	r/w	MP dependent	MP dependent	Min/Max threshold. There can be more than one threshold per monitor point. For instance, two thresholds can be defined for temperature MPs, so that crossing different thresholds triggers generation of alarms with different severity indicators. When a parameter value crosses a threshold, MP status is reported and/or alarm message is generated.
Threshold-Action	r/o	ASCII String	CSP Alarm ID GenerateReport DoNotReport	For each threshold TM can define what CSP should do when the value crosses the threshold. TM can specify: a) Alarm ID and severity for the alarm to be generated; b) 'generate report' or c) string 'do not report'. Details TBD – this is just a preliminary idea.
Threshold-Alarm ID	r/w	ASCII String	TBD	Alarm to be generated when the parameter value crosses the threshold.
Threshold-Severity	r/w	Enumeration	As per <a href="#">[RD4]</a>	Severity level assigned to a threshold – CSP generates alarms with this severity level when the parameter value crosses the threshold.
Description	r/o	ASCII Text	TBD	Text to be added to the Alarm message, if not provided to TM in advance. Description provides information related to failure and identifies potential causes and effects the failure can have on the telescope functionality and output.

#### 7.6.4 CSP Generated Auxiliary Data

CSP will collect auxiliary data to be stored by Telescope Manager or forwarded to SDP.

The list of data collected and reported by CSP is TBD.

For each type of auxiliary data TM will be able to set the following parameters:

1. Reporting cadence (frequency)
2. Destination for CSP reports (messages)
3. Status (reporting on/off).

## 7.6.5 Alarms

The document “SKA TM to Element LMC Interface Guidelines” [RD4 ] will define requirements and guidelines for reporting errors, failures and safety threats. It is expected that guidelines will specify the format and content for the alarm messages, including the alarm severity levels. The list of events that should be reported as alarms depends on the physical implementation; more details will be provided in future releases.

CSP generates Alarm messages to report errors, faults and transient events that should be reported to TM. TM makes Alarms visible to the operators and other members of the operations team.

Alarm is an unsolicited message generated by CSP.

Each Alarm message contains the following:

- 1) Alarm Identifier
- 2) Alarm Severity Level - Perceived Severity as assessed at the Alarm Detection Point – Alarms levels as defined in [RD4].
- 3) Alarm State (e.g. raise, clear, transient – Alarm states are defined in [RD4]).
- 4) Time stamp (indicates when the alarm was detected).
- 5) Failed Component or Capability – may or may not be the same as the Alarm Detection Point.
- 6) Alarm Detection Point - entity that detected and reported alarm (Component or Capability ID).
- 7) Description - description provides information related to failure and identifies potential causes and effects the failure can have on the functionality and output.
- 8) Suggested Action.

Content of the Alarm messages is defined in Section 7.8.8.4.

The description and suggested action may be provided either in the message itself or in advance, i.e. before commissioning, so that they can be stored in the TM maintained data base and retrieved when TM receives an Alarm.

For a persistent fault, the Alarm State RAISE is assigned when that Alarm is generated by CSP. When CSP detects that the fault condition ended, CSP generates an Alarm message with the same Alarm ID and Alarm State CLEAR.

For transient events, CSP assigns Alarm State TRANSIENT. Examples of transient events reported to TM:

- Receipt of a bad or malformed message,
- Software exception.

Transient Alarms are NOT reported in the CSP list of Active Alarms.

For Components and Capabilities with Administrative Mode 'NOT-FITTED' all alarms, except the highest priority alarms that report security threats (e.g. overheating) are suppressed.

**Table 7-14 CSP parameters implemented for every CSP Alarm (accessible using SET/GET)**

CSP Alarm Parameter	Access	Type	Range	Description
<b>Alarm ID</b>	r/o	ASCII String	Unique Alarm ID	Unique Alarm ID is assigned to each CSP alarm.
<b>Supressed</b>	r/w	ASCII String	no/yes	Allows TM to supress Alarm.
<b>Severity level</b>	r/w	ASCII String	Enumeration	As defined in <a href="#">RD4</a> .
<b>Alarm Sate</b>	r/w	ASCII String	RAISE CLEAR TRANSIENT	Values as specified in <a href="#">RD4</a> . Enumeration specified here is just an example. TM can clear alarm by setting state=clear.
<b>Alarm Timestamp</b>	r/o	Date & Time	UTC	Timestamp for the last status change. For transient alarms, time at which this alarm was generated last time.
<b>Object ID</b>	r/o	ASCII String	Object ID	ID for the Component or Capability being monitored.
<b>Parameter Name</b>	r/o	Date & Time	UTC	The name of the parameter being monitored – if applicable. And, if status is 'raised', the value of the parameter when the alarm was raised.
<b>Parameter Value</b>	r/o	ASCII String	TBD	
<b>Description</b>	r/o	ASCII Text	text	Text describing the failure/event, to be added to the Alarm message, if not provided to TM in advance. Description provides information related to failure and identifies potential causes and effects the failure can have on the functionality and output.

### 7.6.6 Events

CSP\_Low uses Event messages to report detected events that may be of interest to TM. Event messages may be used by CSP\_Low to report progress, for example, upon successful reconfiguration CSP\_Low could use an Event message to report that it is ready to start processing observed data.

TBD: Events that CSP\_Low can and should report will be specified for each telescope in this ICD. (Again, depending on the protocol selected, Events may or may not be used.)

## 7.6.7 Logging

CSP\_Low maintains two types of Log Files:

1. CSP\_Low Central Log which contains logs for Alarms and Events generated by all CSP\_Low Components and Capabilities. CSP\_Low Central Log contains at least 10.000 most recent log records. Log records are added to the CSP\_Low Central Log File in order in which Alarms and Events (messages) are received by the CSP\_Low sub-element that maintains Central Log File (CSP\_Low LMC). In addition to all information contained in the Alarm/Event message, each log record contains the timestamp when the log was generated (and by which entity (Component or Capability)).
2. Each CSP\_Low Component that has a capacity, maintains own log file. In the same fashion as in the CSP\_Low Central Log File, when the capacity is reached, new log record overwrites the oldest log record. Log records are added to the Component log file in order in which they are generated. Each Log record contains timestamp generated by the entity that generated the log.

The content and format of the log records is defined in [\[RD4\]](#).

CSP\_Low implements logging levels as defined in [\[RD4\]](#).

CSP\_Low makes provision for TM to:

1. Change the logging level for the Central CSP\_Low Log File and for the individual Components.
2. Upload the CSP\_Low Central Error Log File and Log Files generated by individual CSP\_Low Components.
3. Search Log Files based on sub-array ID, Capability ID, Component ID, severity, Scan ID, time, etc.

Depending on the choice of the protocol, different data streams may be used for exchange of messages (Get/Set/Response/Alarms/Event) and for file transmission (TBD).

## 7.6.8 Software/Firmware Upgrade

CSP\_Low makes provision for TM to deploy new versions of software and firmware, and to trigger restart so that a Component initializes using newly deployed version.

On request, CSP\_Low is able to report the currently used version of software and firmware and the date & time at which the currently used software and firmware were installed.

Software/firmware upgrade may be requested for individual Component or for all Components that use software or firmware package (file) in question. In the request TM can identify the sub-element or the Component to be upgraded.

Depending on the choice of the protocol, different data streams may be used for exchange of messages (Get/Set/Response/Alarms/Event) and for file transmission (TBD).

## 7.6.9 Reset/Restart/Reboot

CSP\_Low makes provision for TM to request reset, restart and/or reboot for Components where such functionality is supported. Details are implementation dependent and will be specified for each Component individually (and documented in the ICD).

### 7.6.10 OFF-DUTY (Low-power) State

It is a requirement for CSP\_Low to enter, upon start-up, a low power mode where power consumption is limited to less than 10% (TBC) of the maximum sustained full load power consumption (requirement SKA1-CSP\_REQ-2478-02 [RD8]).

The CSP\_Low low-power state corresponds to the SKA Operational State OFF-DUTY.

OFF-DUTY state is an Operational State that applies for CSP equipment and should not be confused with the IDLE Observing Mode which applies for sub-arrays.

CSP\_Low makes provision for TM to request transition to OFF-DUTY (low-power) state for the following entities:

- CSP\_Low as a whole,
- Individual Sub-element(s), or
- Individual Component.

Note that implementation of the OFF-DUTY (low-power) state is implementation dependent, not all Components (and perhaps not all sub-elements) implement OFF-DUTY state.

Whether or not sub-arrays can be placed in the OFF-DUTY state is implementation dependent and will be determined and document in the future releases of this document (TBD). If the same hardware or software Component can be used by two or more sub-arrays, a sub-array cannot be placed in OFF-DUTY state. The same applies for other Capabilities (e.g. beams).

### 7.6.11 Power-down

CSP\_Low makes provision for TM to request:

1. Staged power-down of CSP\_Low equipment. (Staged power-down is required in order to prevent power surges.)
2. Delayed (staged) power-down of CSP\_Low equipment. TM may specify the Activation Time in the shut-down request message.
3. Where possible, CSP\_Low makes provision for TM to request power-down for individual LRU and/or sub-element.

### 7.6.12 Power-up

CSP\_Low makes provision for TM to request staged power-up of CSP\_Low equipment.

### 7.6.13 Direct access to CSP Components (Tunnelling)

TM shall make provision for direct access to individual CSP\_Low components via engineering interfaces implemented by CSP\_Low that do not conform to this ICD. The exact method used for such access is still to be defined; it may or may not require TM and CSP\_Low can create a 'tunnel' that can be used to exchange messages that do not conform to this ICD. Details are TBD.

## 7.7 Sub-array

The CSP\_Low Capability *sub-array* is defined as an exclusive set of stations in SKA1\_Low. Exclusive in this context means that a station cannot belong to two (or more) sub-arrays at the same time. Apart from the restrictions due to finite hardware, the way in which one sub-array is used shall have no effect on the use of, or observations with, any other sub-array. In the context of the CSP\_Low, the term sub-array is also used to refer to cross-section of the CSP\_Low equipment used to process data from the stations that belong to the sub-array.

CSP\_Low sub-arrays are created and deleted on TM request.

CSP\_Low makes provision for TM to request the following:

1. Create (define) a sub-array, assign a specific ID to newly created sub-array, specify stations to be included, and other parameters. Creation of a sub-array precedes definition of what to do with the sub-array, i.e. which observing mode to set. When a command 'create sub-array' is received, at the specified 'Activation Time' (see section 7.6.1.1) CSP\_Low assigns the resources (station-beam and the corresponding cross-section of the correlator, tied-array beams), and sets the sub-array Observing Mode as Idle until further instructions are received from TM. Once created, a sub-array is referred to using the sub-array Identifier assigned by TM. At any given time, a sub-array ID must be unique per telescope.
2. CSP\_Low makes provision for TM to create an 'empty' sub-array, i.e. to create a sub-array that does not contain any stations (and assign Sub-array ID).
3. Add station to a sub-array. Stations can be added only to an unused sub-array or at the scan boundaries (when ending one scan and starting the next). Note: by adding station(s) to the sub-array, user/operator, via TM, implicitly add a cross-section of the correlator to the sub-array.
4. Remove stations from a sub-array. Stations can be removed only from an unused sub-array or at the scan boundaries (when ending one scan and starting the next).
5. Assign the CSP\_Low resources (Components) for Pulsar Search and Pulsar Timing beam-forming to a particular sub-array. PSS and PST resources are assigned to the sub-array by adding the corresponding Capabilities: CSP-PSS-beam and CSP-PST-beam. Adding a Capability CSP-PSS-beam to a sub-array reserves the resources in the beamformer and PSS Engine for that particular sub-array. The same applies for CSP-PST-beams. Resources (beams) may be added only to idle sub-array or at the scan boundaries.
6. Release PSS and PST resources (beams) allocated to a sub-array. Resources (beams) may be removed from an unused sub-array or at the scan boundaries.
7. Set Observing Mode. For each Observing Mode this ICD specifies a set of parameters that must be provided. Some of the Observing Mode parameters can be changed during an observation (scan); in the future versions this ICD will indicate which CSP\_Low parameters can be changed (set) during an observation (scan) and which cannot.
8. Stop generation and transmission of output products, which is implemented as transition to Idle Mode. 'End Observing Mode' or 'End Scan', if implemented as a distinct command, will have the same effect as 'Set Observing Mode Idle'. CSP\_Low re-configuration occurs either immediately (if Activation Time is not specified) or at the specified Activation Time.

9. Set individual sub-array parameters. The list of parameters that can be changed during an observation (scan) includes a subset of the Observing Mode parameters and common Capability management parameters and is specified in Section 7.8.9 for each Observing Mode.
10. Query sub-array configuration (parameters) and status.
11. Delete (destroy) sub-array – this also implies that CSP\_Low should release the resources assigned to the sub-array. TBC: Released resources are set in IDLE mode.
12. Report sub-array availability, i.e. how many sub-arrays are used and how many can be created.
13. Report resource availability, i.e. how many stations, beams, etc. is available (not assigned to sub-arrays).

Note: It should not be assumed that each ‘command’ in the above list will be implemented as a different message. At this time, the format of messages and the actual protocol have not been defined.

Note that TM can choose to assign the same Activation Time to several messages. For example, TM can instruct CSP\_Low to end the on-going scan in a particular sub-array, add stations to the sub-array and start the next scan in the same sub-array by assigning the same Activation Time to the messages: 'end scan in sub-array A', 'add stations to subarray A' and 'start scan Y in sub-array A' (start scan will probably be implemented as 'set Observing Mode = z'). Refer to Section 7.6.1.1 for more information regarding Activation Time.

Once created, a sub-array is not destroyed until one of the following occurs:

1. The sub-array is deleted on TM request.
2. CSP\_Low restart, transition to low-power state or power-down.

Once added to a sub-array, a station remains part of the sub-array (and cannot be added to another sub-array) until one of the following occurs:

1. Station is explicitly removed from the sub-array on TM request.
2. The sub-array is deleted on TM request.
3. CSP\_Low restart, transition to low-power state or power-down.

Once added to a sub-array, Capabilities CSP-PSS-beam and CSP-PST-beam cannot be used by another sub-array until one of the following occurs:

1. Capability is explicitly removed from the sub-array on TM request.
2. The sub-array is deleted on TM request.
3. CSP\_Low restart, transition to low-power state or power-down.

### 7.7.1 End Of Scan

For the CSP\_Low subarrays the end of a scan is implemented as a transition to the Observing Mode IDLE.

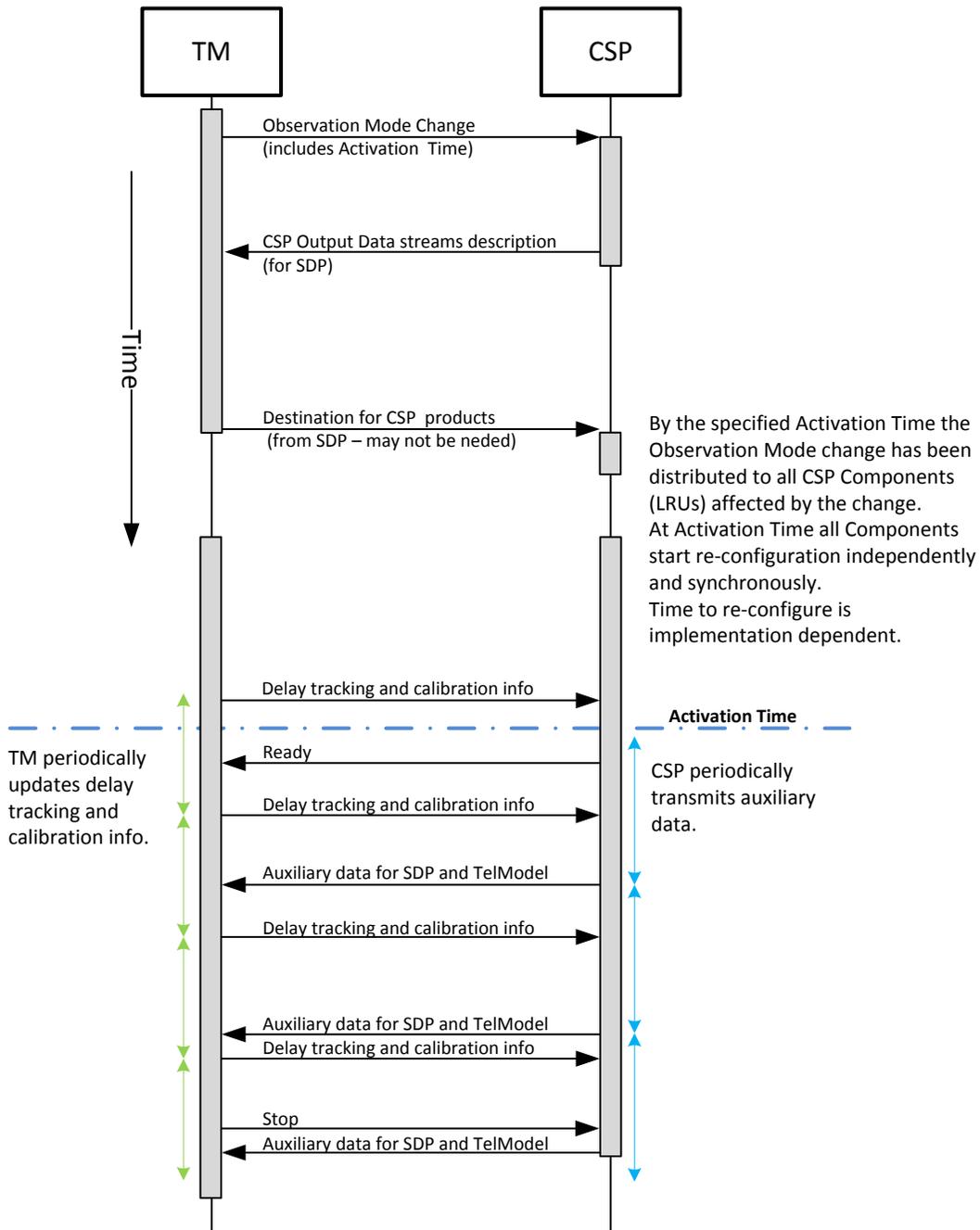
It is to be defined (TBD) whether TM has to explicitly end an on-going observation (scan), before the sub-array can be re-configured for another observation.

## 7.7.2 Observing Mode Change

Figure 7-12 is an example for the exchange of messages between TM and CSP\_Low for the case where the Observing Mode Change message contains an Activation Time:

1. TM transmits an Observing Mode change request including: sub-array ID, Activation Time, description of input data (band), description of the CSP\_Low output products, unique ID to be inserted in the CSP\_Low output products, and other relevant parameters (for the complete list of Observing Mode parameters refer to Chapter 7 Interface Details).
2. CSP\_Low informs TM that the Observing Mode change has been accepted and provides information regarding the output streams that will be used to transmit CSP\_Low output products to SDP (e.g. identify the spigots that will be used, distribution of output data, and similar). TM forwards that information to SDP.
3. SDP, via TM, sends to CSP\_Low, destination parameters (addresses) for the CSP\_Low products.
4. The CSP\_Low forwards the configuration to all affected Capabilities and Components in advance, so that re-configuration can start simultaneously for all affected Capabilities and Components.
5. TM provides delay and phase tracking information, and other real-time updates, before the Activation Time, so that, at the Activation Time, CSP\_Low already has the delay tracking and other real-time control information in place. If such information is not available at the Activation Time, CSP\_Low generates an alarm and flags output data.
6. At the specified Activation Time, the CSP\_Low begins re-configuration of all affected Components simultaneously.
7. Individual CSP\_Low Components (LRUs, FPGAs, servers, software processes) start generating output products as soon as they become ready. Time required for re-configuration of individual LRUs and other Components is implementation dependent. CSP\_Low Components re-configure independently from each other and start generating output as soon as re-configuration is complete, CSP\_Low output products may be incomplete for the first several integrations at the beginning of a scan.
8. CSP shall periodically report status of the on-going observations/scans to TM; more precisely CSP shall periodically report status of the CSP generated products and shall report anything that may impact quality and status of the CSP generated products. Frequency of reports is TBD (may be configurable parameter).
9. During the scan, TM periodically generates time-dependent updates required for delay correction, beamforming and RFI mitigation. The frequency for the updates is specified in Section 7.8.9 for each Observing Mode.
10. During the scan, CSP\_Low collects and sends to TM auxiliary data. Content of auxiliary data will be defined (TBD) for each Observing Mode.
11. TM uses the Stop (end scan) message to instruct CSP\_Low to stop generating and transmitting output products. Upon receipt of the Stop message all affected CSP\_Low Capabilities and Components return to Idle Mode. The 'Stop Observing Mode' request has the same effect as the request to change Observing Mode to 'idle'. In order to avoid an accidental over-write of on-going observations, TM may be required to explicitly 'stop' an on-going observation (Observing Mode) before starting new Observing Mode (TBC). Unnecessary gap between the scans can be avoided by transmitting 'stop' command followed by an 'Observing Mode change' command with the same Activation Time.

Figure 7-12 Observing Mode Change – Time Sequence Diagram



## 7.8 Protocol

### 7.8.1 Types of messages

Message encoding and format will be defined in the document “SKA1 LMC Interface Guidelines” [RD4].

Table 7-15 below contains the preliminary list of messages exchanged over this interface; the exact list of messages will be defined when underlying protocols, message encoding and model of interaction with TM are defined (TBD).

**Table 7-15 Messages**

Message	Generated by	Description
SET Parameters	TM	Set value of CSP parameters. Also used to request mode and state transitions. Requires write access to CSP parameters). Note: In the absence of the protocol definition, for simplicity, this version of the ICD assumes that TM will use a single command 'SET Parameters' to request configuration changes and state transitions.
GET Parameters (query)	TM	Get status of CSP parameters. In the absence of the protocol definition, for simplicity, this version of the ICD assumes that all Queries will use the same format.
Response	CSP	Generated by CSP in response to TM commands and queries ('SET Parameters' or 'GET Parameters').
Alarm	CSP	Report errors and faults.
Event	CSP	Report events that may be of interest to TM or operators. Set of events to be reported will be agreed upon in advance and documented in this ICD.
Monitor Point Reports	CSP	CSP periodically reports status for a selected subset of CSP parameters (so called monitor points). The set of CSP Monitor Points will be determined based on FMECA. CSP makes provision for TM to set frequency, status (enabled/disabled) and other parameters (using SET Parameters).
Auxiliary Data	CSP	CSP generates auxiliary data. The content and cadence will be determined for each Observing Mode. (TBD).

## 7.8.2 Conventions

This section defines conventions used in this protocol specification and in the messages.

### 7.8.2.1 Date and Time

Date and time in the messages is specified using the following format:

- UTC (Coordinated Universal Time) displayed as ISO 8601 [RD2].

Requirements for precision when specifying time have not been identified yet. Guidance is expected from the “SKA Concept of Operations” [RD10], “SKA1 Level 1 Requirements” [AD2] and CSP sub-elements.

For the FPGA/ASIC based solutions, re-configuration occurs on system tick (TBD), therefore the start of a scan or other re-configuration requests could be specified with TBD precision. Better precision could be achieved if required, as FPGAs could be programmed to start product generation at exact time.

Note, however, that the Activation Time is the time when re-configuration of the CSP equipment should start; depending on the design, up to 25 seconds may be needed to re-configure CSP when switching Observing Modes (exact duration is implementation dependent and is TBD). Therefore, beginning of a scan and other massive configuration changes need not be specified with high precision.

Higher precision is required when specifying the epoch – exact requirements TBD.

### 7.8.3 CSP\_Low Parameters

Table 7-16 lists parameters that apply to the CSP\_Low as whole, these parameters are assigned to the top-level entity in the CSP\_Low monitor & control hierarchy. These parameters allow TM to monitor and control CSP as a single entity.

Table 7-16 CSP\_Low Parameters used to monitor and set status the CSP\_Low as a whole.

CSP_Low Parameter	Access <sup>1</sup>	Type	Range	Description
<b>Reporting Entity</b>	r/o	ASCII String	Object ID and/or address	Identifier and address of the entity (sub-element, Component or Capability) reporting on behalf of CSP (i.e. CSP.LMC).
<b>Protocol version</b>	r/o	ASCII String	TBD	Version of the CSP-to-TM protocol implemented by the CSP instantiation.
<b>Administrative Mode</b>	r/w	Enumeration	ENABLED, DISABLED, MAINTENANCE, NOT-FITTED	Administrative Mode as assigned by TM for the CSP as a whole. See Section 7.5.1.
<b>Administrative Mode Timestamp</b>	r/o	Date & Time	UTC	Timestamp for the last Administrative Mode change at CSP level.
<b>Administrative Mode Info</b>	r/w	ASCII Text	-	Information provided by TM, e.g. reason or purpose for mode transition.
<b>Administrative Mode Originator</b>	r/w	ASCII String	ID	ID of the person or entity who requested the last Administrative Mode change.
<b>Control Mode</b>	r/w	Enumeration	CENTRAL, LOCAL	Control Mode as assigned by TM for CSP as a whole. See Section 7.5.4.
<b>Control Mode Timestamp</b>	r/o	Date & Time	UTC	Timestamp for the last Control Mode change at CSP level.

<sup>1</sup> Access can be defined as: read-only (r/o), read-write (r/w) or write-only (w/o).

CSP_Low Parameter	Access <sup>1</sup>	Type	Range	Description
<b>Control Mode Info</b>	r/w	ASCII Text	-	Information provided by TM, e.g. reason or purpose for mode transition.
<b>Control Mode Originator</b>	r/w	ASCII String	ID	ID of the person or entity who requested the last Control Mode change. If CSP implements Control Mode as a read-only parameter, CSP itself is the originator.
<b>Simulated Mode</b>	r/w	Enumeration	REAL, SIMULATED	Simulated Mode as assigned by TM for CSP as a whole. See Section 7.5.1. It is TBC whether CSP will make provision to set Simulated Mode (which would allow using the real CSP in simulated mode).
<b>Simulated Mode Timestamp</b>	r/o	Date & Time	UTC	Timestamp for the last Simulated Mode change at CSP level.
<b>Simulated Mode Info</b>	r/w	ASCII Text	-	Information provided by TM, e.g. reason or purpose for mode transition.
<b>Simulated Mode Originator</b>	r/w	ASCII String	ID	ID of the person or entity who requested the last Simulated Mode change. If CSP implements Simulated Mode as a read-only parameter, CSP itself is the originator.
<b>Test Mode</b>	r/w	Enumeration	NORMAL, TEST	Test Mode as assigned by TM for the CSP as a whole. See Section 7.5.1. CSP may implement more than one TEST mode or no TEST mode at all – TBD.
<b>Test Mode Timestamp</b>	r/o	Date & Time	UTC	Timestamp for the last Test Mode change at CSP level.
<b>Test Mode Info</b>	r/w	ASCII Text	-	Information provided by TM, e.g. reason or purpose for mode transition.

CSP_Low Parameter	Access <sup>1</sup>	Type	Range	Description
<b>Test Mode Originator</b>	r/w	ASCII String	ID	ID of the person or entity who requested the last Test Mode change. If CSP implements the Test Mode as a read-only parameter, CSP itself is the originator.
<b>Operational State</b>	r/o	Enumeration	INITIALIZING, OFF_DUTY, READY, FAILED, GOING_OFF_DUTY, GETTING_READY, SHUTTING_DOWN, OFF, UNKNOWN	Operational State as derived/detected by CSP (element level). See Section 7.5.6.
<b>Operational State Timestamp</b>	r/o	Date & Time	UTC	Timestamp for the last Operational State change at CSP level.
<b>Operational State Info</b>	r/o	ASCII Text	-	Generated by CSP – reason or cause for state transition or for inability to progress to the next state.
<b>Health State</b>	r/o	Enumeration	OK, DEGRADED NOT-AVAILABLE	Health State as derived/detected for CSP instantiation (element level). See Section 7.5.7.
<b>Health State Timestamp</b>	r/o	Date & Time	UTC	Timestamp for the last Health State change at CSP level.
<b>Health State Info</b>	r/o	ASCII Text	-	Generated by CSP – optional additional information, for instance CSP may provide additional information if CSP health is DEGRADED.
<b>Usage State</b>	r/o	Enumeration	USED, IDLE	Usage State as derived by for CSP instantiation (element level) – USED is reported if at least one sub-array is not IDLE (or in other words if at least one scan is in progress). See Section 7.5.8.
<b>Usage State Timestamp</b>	r/o	Date & Time	UTC	Timestamp for the last Health State change at CSP level.

CSP_Low Parameter	Access <sup>1</sup>	Type	Range	Description
<b>Usage State Info</b>	r/o	ASCII Text	-	Generated by CSP – optional additional information, for instance CSP may indicate sub-arrays that are used for observing.
<b>Redundancy State</b>	r/o	ASCII String	ACTIVE	Redundancy State as derived by for CSP instantiation (element level). Redundancy is not planned for the CSP as a whole; the Redundancy State for the CSP as a whole shall be always reported as ACTIVE. See Section 7.5.9.
<b>Redundancy State Timestamp</b>	r/o	Date & Time	UTC	Redundancy State at CSP level is always ACTIVE - this parameter is not applicable but shall be implemented for consistency and report the date and time when the CSP has become ACTIVE (during INITIALIZATION).
<b>Redundancy State Info</b>	r/o	ASCII Text	-	Generated by CSP – optional additional information, for example CSP may report ‘redundancy not provided’.
<b>External interfaces</b>	r/o	ASCII String	CSP-TM CSP-SADT CSP-SDP CSP-INFRA AU	List of all CSP external interfaces (between CSP and other Elements), their version and status.
<b>Internal interfaces</b>	r/o	ASCII String	LMC-CBF, LMC-PSS, LMC-PST, CBF-PSS, CBF-PST	List of internal interfaces (between CSP sub-elements), their version and status.
<b>List of sub-elements</b>	r/o	TBD	CBF, PSS,PST, LMC	List of the CSP_LOW sub-elements (status may be included in the reply).
<b>List of Capabilities</b>	r/o	TBD	Sub-array Station-input CSP-PSS-beam CSP-PST-beam	Capabilities implemented by CSP_Low.
<b>Capability status</b>	r/o	TBD	TBD	For each supported Capability type the list of Capabilities and their availability and status.

CSP_Low Parameter	Access <sup>1</sup>	Type	Range	Description
Capability availability	r/o	Integer	-	Number of available (unused) Capabilities for each type (including sub-array).
Alarm reporting level	r/w	Enumeration	TBD	Alarm reporting level can be set at element level. if set at element level applies for all sub-elements and LRUs
Alarm destination	r/w	TBD	-	Destination address for alarm messages can be set at element level. if set at element level applies for all sub-elements and LRUs
List of active Alarms	r/o	TBD	TBD	Alarm ID, timestamp and brief description for all active alarms.
List of all implemented Alarms	r/o	TBD	TBD	List of all implemented alarms.
Logging level	r/w	Enumeration	TBD	If set at element level, logging level applies for all sub-elements and LRUs
CSP Logging File URL	r/o	ASCII String	URL	URL for the central CSP logging file. (so that TM can query the location and upload the logging file).
Logging of messages	r/w	Enumeration	ENABLE DISABLE	Enable/disable logging of messages received and transmitted (supported only for monitor and control interfaces).
Disable transmission of messages	r/w	Enumeration	ENABLE DISABLE	Enable/disable transmission of messages for each CSP interface individually.
Destination for MP reports.	r/w	TBD	TBD	Destination address for Monitoring Point reports. If set at element level applies for all sub-elements and LRUs.
<p><b>The following parameters/commands are related to the scheduled execution of commands. These commands will be used for troubleshooting.</b></p>				

CSP_Low Parameter	Access <sup>1</sup>	Type	Range	Description
List of received messages	r/o	ASCII Text	List of TM messages	List received TM messages (input queue) that are waiting to be executed. TM may request short overview or full content.
List of messages in progress	r/o	ASCII Text	List of TM commands in progress and their status.	List of TM commands (SET, GET) in progress and their status. For example 'waiting sub-element' response.
List of messages waiting to be executed	r/o	ASCII Text	List of TM SET requests waiting to be executed.	List of TM commands waiting to be executed (waiting for the Activation Time).
Revoke transmitted message	w/o	TBD	Revoke + identifier of the message to be revoked or revoke + all + status	Depending on the message status (received, in-progress, scheduled): a) remove a message from the input queue; b) cancel message if in progress; c) discard command waiting to be executed. Revoke may be specified for a particular message or for all messages with the specified status. Details TBD.
<b>Commands / state transitions</b> (Depending on the choice of the protocol, commands may be implemented either as dedicated messages or as write-only parameters transmitted in SET Parameter messages).				
Power-down	w/o	ASCII String	power-down	Staged Power-down of CSP equipment TM can request delayed power-down i.e. TM can specify wait period or delay period in seconds.
Power-up	w/o	ASCII String	power-up	Staged power-up of CSP equipment.
Transition to OFF-DUTY state	w/o	ASCII String	go-off-duty	This command instructs CSP to transition in Operational State OFF-DUTY, i.e. to set CSP equipment in the low-power mode. See Section 7.5.6).

CSP_Low Parameter	Access <sup>1</sup>	Type	Range	Description
<b>Transition to READY state</b>	w/o	ASCII String	go-ready	Examples: This command can be used to request transition from OFF-DUTY to READY state. This command may be also used to force CSP equipment to re-try transition to READY state in the case when reboot or restart was not completed as expected.
<b>Firmware upgrade</b>	w/o	TBD	TBD	This command will be probably supported only at the sub-element and LRU level.
<b>Software upgrade</b>	w/o	TBD	TBD	This command will be probably supported only at the sub-element and LRU level.

## 7.8.4 Parameters Common to all CSP\_Low Sub-elements

Table 7-17 Parameters implemented by all CSP\_Low Sub-Elements

CSP_Low Sub-element Parameter	Access <sup>1</sup>	Type	Range	Description
<b>Reporting Entity</b>	r/o	ASCII String	Object ID and/or address	Identifier and address of the entity reporting on behalf of the sub-element.
<b>Administrative Mode</b>	r/w	Enumeration	ENABLED, DISABLED, MAINTENANCE, NOT-FITTED	Sub-element Administrative Mode as set by TM. See Section 7.5.1.
<b>Administrative Mode Timestamp</b>	r/o	Date & Time	UTC	Timestamp for the last Administrative Mode change at sub-element level.
<b>Administrative Mode Info</b>	r/w	ASCII Text	-	Information provided by TM, e.g. reason or purpose.
<b>Administrative Mode Originator</b>	r/w	ASCII String	ID	ID of the person or entity who requested the last Admin. Mode change.
<b>Control Mode</b>	r/w	Enumeration	CENTRAL, LOCAL	Sub-element Control Mode as assigned by TM. See Section 7.5.4.
<b>Control Mode Timestamp</b>	r/o	Date & Time	UTC	Timestamp for the last sub-element Control Mode change.
<b>Control Mode Info</b>	r/w	ASCII Text	-	Information provided by TM, e.g. reason or purpose for mode transition.
<b>Control Mode Originator</b>	r/w	ASCII String	ID	ID of the person or entity who requested the last sub-element Control Mode change.

<sup>1</sup> Access can be defined as: read-only (r/o), read-write (r/w) or write-only (w/o).

CSP_Low Sub-element Parameter	Access <sup>1</sup>	Type	Range	Description
<b>Simulated Mode</b>	r/w	Enumeration	REAL, SIMULATED	Sub-element Simulated Mode as assigned by TM. See Section 7.5.5. Not all sub-elements make provision for TM to set Simulated Mode. All sub-elements shall report Simulated Mode (as REAL or SIMULATED).
<b>Simulated Mode Timestamp</b>	r/o	Date & Time	UTC	Timestamp for the last sub-element Simulated Mode change, if applicable (i.e. if both modes are supported). A sub-element that implements only one value (REAL or SIMULATED) shall report date and time when the sub-element entered that state (during INITIALIZATION).
<b>Simulated Mode Info</b>	r/w	ASCII Text	-	Information provided by TM, e.g. reason or purpose for mode transition. Sub-element that implements Simulated Mode as read-only parameter may report CSP generated information, e.g. 'sub-element simulator'.
<b>Simulated Mode Originator</b>	r/w	ASCII String	ID	ID of the person or entity who requested the last sub-element Simulated Mode change. For sub-elements that implement Simulated Mode as read-only parameter, sub-element itself is the originator.
<b>Test Mode</b>	r/w	Enumeration	NORMAL, TEST	Test Mode as assigned by TM for sub-element as a whole. See Section 7.5.1. Sub-element may implement more than one TEST mode or no TEST mode at all – to be defined for each sub-element (implementation dependent).
<b>Test Mode Timestamp</b>	r/o	Date & Time	UTC	Timestamp for the last sub-element Test Mode change.
<b>Test Mode Info</b>	r/w	ASCII Text	-	Information provided by TM, e.g. reason or purpose for mode transition.

CSP_Low Sub-element Parameter	Access <sup>1</sup>	Type	Range	Description
<b>Test Mode Originator</b>	r/w	ASCII String	ID	ID of the person or entity who requested the last Test Mode change.
<b>Operational State</b>	r/o	Enumeration	INITIALIZING, OFF_DUTY, READY, FAILED, GOING_OFF_DUTY, GETTING_READY, SHUTTING_DOWN, OFF, UNKNOWN	Sub-element Operational State as derived/detected and reported by the CSP. See Section 7.5.6.
<b>Operational State Timestamp</b>	r/o	Date & Time	UTC	Timestamp for the last sub-element Operational State change.
<b>Operational State Info</b>	r/o	ASCII Text	-	Generated by CSP – reason or cause for state transition or for inability to progress to the next state.
<b>Health State</b>	r/o	Enumeration	OK, DEGRADED. NOT-AVAILABLE	Sub-element Health State as derived/detected and reported by the CSP. See Section 7.5.7.
<b>Health State Timestamp</b>	r/o	Date & Time	UTC	Timestamp for the last Health State change.
<b>Health State Info</b>	r/o	ASCII Text	-	Generated by CSP – optional additional information, for instance CSP may provide additional information if sub-element health is DEGRADED.
<b>Usage State</b>	r/o	Enumeration	USED, IDLE	Sub-element Usage State as derived by CSP – USED is reported if at least one non IDLE sub-array is using this sub-element. See Section 7.5.8.
<b>Usage State Timestamp</b>	r/o	Date & Time	UTC	Timestamp for the last sub-element Usage State change.
<b>Usage State Info</b>	r/o	ASCII Text	-	Optional additional information generated by CSP; for instance CSP may indicate which sub-arrays are using sub-element.

CSP_Low Sub-element Parameter	Access <sup>1</sup>	Type	Range	Description
<b>Redundancy State</b>	r/o	Enumeration	ACTIVE, STANDBY	Sub-element Redundancy State as derived by CSP (more precisely by the entity that reports on behalf of the sub-element). If redundancy is not provided the sub-element Redundancy State shall be always reported as ACTIVE. See Section 7.5.9.
<b>Redundancy State Timestamp</b>	r/o	Date & Time	UTC	Date & time for the last Redundancy State transition. If redundancy is not provided, this parameter is not applicable, but shall be implemented as read-only for consistency. CSP shall report the date and time when the sub-element has become ACTIVE (during INITIALIZATION).
<b>Redundancy State Info</b>	r/o	ASCII Text	-	Optional additional information generated by CSP, for example CSP may report 'redundancy not provided'.
<b>Used by sub-arrays</b>	r/o	ASCII String , list	List of sub-arrays	List of sub-arrays that use this sub-element. For each sub-array: Observing Mode, rolled-up status for the part of a sub-array implemented by this sub-element.
<b>Used by Capabilities</b>	r/o	ASCII String , list	List of Capability IDs	List of Capabilities (other than sub-array) that use this sub-element (if applicable). For each Capability: rolled-up status of the part of the Capability implemented by this sub-element.
<b>Interfaces</b>	r/o	TBD	TBD	List of all sub-element interfaces, internal and external, their status, version of the protocol, addresses, etc.
<b>List of Components</b>	r/o	TBD	sub-element dependent	List of Components (LRUs, servers, etc.) this sub-element consists of. A digest status report for each Component (TBC).

CSP_Low Sub-element Parameter	Access <sup>1</sup>	Type	Range	Description
List of Capabilities	r/o	TBD	TBD	List of capability types implemented (supported) by this sub-element (if any).
Capability status	r/o	TBD	TBD	For each supported Capability type, the list of Capabilities, their availability and status.
Capability availability	r/o	Integer	Depends on Capability type and telescope	Number of available (unused) Capabilities for each type (including sub-array).
Alarm reporting level	r/w	Enumeration	TBD	When set at sub-element level applies for all Components and LRUs.
Alarm destination	r/w	TBD	-	Destination address for alarm messages. When set at sub-element level applies for all Components and LRUs.
List of active Alarms	r/o	TBD	TBD	Alarm ID, timestamp and brief description for all active alarms.
List of all implemented Alarms	r/o	TBD	TBD	List of all implemented alarms.
Logging level	r/w	Enumeration	TBD	Logging level when set at sub-element level applies for all Components and LRUs.
Logging of messages	r/w	Enumeration	ENABLE DISABLE	Enable/disable logging of messages received and transmitted on a particular interface (supported only for monitor and control interfaces).
Transmission of messages	r/w	Enumeration	ENABLE DISABLE	Enable/disable transmission of messages for a particular interface.
Destination for MP reports	r/w	TBD	TBD	Destination address for Monitoring Point reports; when set at sub-element level applies for all Components and LRUs.

CSP_Low Sub-element Parameter	Access <sup>1</sup>	Type	Range	Description
<p>The following parameters/commands are related to the scheduled execution of commands. Similar commands are available at the CSP Element (top) level. It may be useful for TM to have access to the same commands at sub-element level.</p>				
List of received messages	r/o	ASCII Text	List of messages	List of received messages (input queue) that are waiting to be executed. May provide short overview or full content.
List of messages in progress	r/o	ASCII Text	List of commands in progress and their status	List of commands in progress and their status. For example 'LRU' response.
List of messages waiting to be executed	r/o	ASCII Text	List of commands waiting to be executed.	List of commands waiting to be executed (waiting for the Activation Time).
Revoke transmitted message/command	w/o	TBD	Revoke + identifier of the message to be revoked or revoke + all + status	Depending on the message/command status (received, in-progress, scheduled): a) remove a message from the input queue; b) cancel a message if in progress; c) discard a message waiting to be executed.  Revoke may be specified for a particular message or for all messages with the specified status.
<p><b>Commands / state transitions</b> (Depending on the choice of the protocol, commands may be implemented either as dedicated messages or as write-only parameters transmitted in SET Parameter messages).</p>				
Power-down	w/o	ASCII String	power-down	Staged power-down of CSP equipment. TM can request delayed power-down i.e. TM can specify wait period or delay period in seconds.
Power-up	w/o	ASCII String	power-up	Staged power-up of CSP equipment.
Transition to OFF-DUTY state	w/o	ASCII String	go-off-duty	Set sub-element in low-power state.

CSP_Low Sub-element Parameter	Access <sup>1</sup>	Type	Range	Description
<b>Transition to READY state</b>	w/o	ASCII String	go-ready	This command can be used to request transition from OFF-DUTY to READY state. This command may be also used to force CSP equipment to re-try transition to READY state in the case when reboot or restart was not completed as expected.
<b>Firmware upgrade</b>	w/o	TBD	TBD	Firmware upgrade for all Components of the specified type. TM could query sub-element to obtain the location, deploy a new version and instruct sub-element which LRU type (and Component) should be updated.
<b>Software upgrade</b>	w/o	TBD	TBD	Software upgrade for all Components of the specified type. Implementation TBD (i.e. TM may be required to use a sequence of commands – see comment for firmware upgrade).

## 7.8.5 Parameters Common to all CSP\_Low LRUs

Table 7-18 Parameters implemented by all CSP\_Low LRUs

LRU Parameter	Access <sup>1</sup>	Type	Range	Description
<b>Reporting Entity</b>	r/o	ASCII String	Object ID and/or address	Identifier and address of the entity reporting on behalf of the LRU.
<b>Type</b>	r/o	Enumeration	TBD	LRU type.
<b>h/w version</b>	r/o	ASCII String	TBD	As applicable to LRU type. There can be more than one instance (if LRU consists of several h/w components).
<b>f/w version</b>	r/o	ASCII String	TBD	As applicable to LRU type. There can be more than one instance.
<b>s/w version</b>	r/o	ASCII String	TBD	As applicable to LRU type. There can be more than one instance.
<b>Serial number</b>	r/o	ASCII String	TBD	As applicable to LRU type.
<b>Interface info</b>	r/o	ASCII String	TBD	List of interfaces, their status and version of the protocol.
<b>Administrative Mode</b>	r/w	Enumeration	ENABLED, DISABLED, MAINTENANCE, NOT-FITTED	LRU Administrative Mode as set by TM. See Section 7.5.1.
<b>Administrative Mode Timestamp</b>	r/o	Date & Time	UTC	Timestamp for the last Administrative Mode change.
<b>Administrative Mode Info</b>	r/w	ASCII Text	-	Information provided by TM, e.g. reason or purpose.
<b>Administrative Mode Originator</b>	r/w	ASCII String	ID	ID of the person or entity who requested the last Admin. Mode change.

<sup>1</sup> Access can be defined as: read-only (r/o), read-write (r/w) or write-only (w/o).

LRU Parameter	Access <sup>1</sup>	Type	Range	Description
<b>Control Mode</b>	r/w	Enumeration	CENTRAL, LOCAL	LRU Control Mode as assigned by TM. See Section 7.5.4.
<b>Control Mode Timestamp</b>	r/o	Date & Time	UTC	Timestamp for the last LRU Control Mode change.
<b>Control Mode Info</b>	r/w	ASCII Text	-	Information provided by TM, e.g. reason or purpose for mode transition.
<b>Control Mode Originator</b>	r/w	ASCII String	ID	ID of the person or entity who requested the last Control Mode change.
<b>Simulated Mode</b>	r/w	Enumeration	REAL, SIMULATED	LRU Simulated Mode as assigned by TM. See Section 7.5.5. Not all LRU make provision for TM to set Simulated Mode. All LRU shall implement Simulated Mode as read-only parameter and shall report Simulated Mode (as REAL or SIMULATED).
<b>Simulated Mode Timestamp</b>	r/o	Date & Time	UTC	Timestamp for the last LRU Simulated Mode change, if applicable (i.e. if both modes are supported). An LRU that implements only one value (REAL or SIMULATED) shall report date and time when the sub-element entered that state (during INITIALIZATION).
<b>Simulated Mode Info</b>	r/w	ASCII Text	-	Information provided by TM, e.g. reason or purpose for mode transition. LRU that implements Simulated Mode as read-only parameter may report CSP generated information, e.g. 'LRU simulator'.
<b>Simulated Mode Originator</b>	r/w	ASCII String	ID	ID of the person or entity who requested the last LRU Simulated Mode change. For LRUs that implement Simulated Mode as read-only parameter, LRU itself or other CSP entity is the originator.

LRU Parameter	Access <sup>1</sup>	Type	Range	Description
<b>Test Mode</b>	r/w	Enumeration	NORMAL, TEST	Test Mode as assigned by TM. See Section 7.5.1. LRU may implement more than one TEST mode or no TEST mode at all – TBD for each LRU.
<b>Test Mode Timestamp</b>	r/o	Date & Time	UTC	Timestamp for the last Test Mode change.
<b>Test Mode Info</b>	r/w	ASCII Text	-	Information provided by TM, e.g. reason or purpose for mode transition.
<b>Test Mode Originator</b>	r/w	ASCII String	ID	ID of the person or entity who requested the last Test Mode change.
<b>Operational State</b>	r/o	Enumeration	INITIALIZING, OFF_DUTY, READY, FAILED, GOING_OFF_DUTY, GETTING_READY, SHUTTING_DOWN, OFF, UNKNOWN	LRU Operational State as derived/detected and reported by the LRU itself or by the entity that reports on behalf of the LRU. See Section 7.5.6.
<b>Operational State Timestamp</b>	r/o	Date & Time	UTC	Timestamp for the last Operational State change.
<b>Operational State Info</b>	r/o	ASCII Text	-	Generated by CSP – reason or cause for the last state transition or for inability to progress to the next state.
<b>Health State</b>	r/o	Enumeration	OK, DEGRADED, NOT-AVAILABLE	LRU Health State as derived/detected and reported by the LRU itself or by the entity that reports on behalf of the LRU. See Section 7.5.7.
<b>Health State Timestamp</b>	r/o	Date & Time	UTC	Timestamp for the last Health State change.
<b>Health State Info</b>	r/o	ASCII Text	-	Optional additional information generated by CSP, for instance CSP may provide additional information when health is DEGRADED.

LRU Parameter	Access <sup>1</sup>	Type	Range	Description
<b>Usage State</b>	r/o	Enumeration	USED, IDLE	LRU Usage State as derived/detected and reported by the LRU itself or by the entity that reports on behalf of the LRU. USED is reported if at least one non IDLE sub-array is using this LRU. See Section 7.5.8.
<b>Usage State Timestamp</b>	r/o	Date & Time	UTC	Timestamp for the LRU Usage State change.
<b>Usage State Info</b>	r/o	ASCII Text	-	Optional additional information generated by CSP; for instance CSP may indicate which sub-arrays are using the LRU.
<b>Redundancy State</b>	r/o	Enumeration	ACTIVE	LRU Redundancy State as derived/detected and reported by the LRU itself or by the entity that reports on behalf of the LRU. If redundancy is not provided the LRU Redundancy State shall be always reported as ACTIVE. See Section 7.5.9.
<b>Redundancy State Timestamp</b>	r/o	Date & Time	UTC	Date & time for the last Redundancy State transition. If redundancy is not provided, this parameter is not applicable, but shall be implemented as read-only for consistency. In such case, CSP shall report the date and time when the LRU has become ACTIVE (during INITIALIZATION).
<b>Redundancy State Info</b>	r/o	ASCII Text	-	Optional additional information generated by CSP, for example 'redundancy not provided'.
<b>Used by sub-arrays</b>	r/o	ASCII String , list	TBD	List of sub-arrays that use this LRU. For each sub-array: Observing Mode and rolled-up status for the part of the sub-array implemented by this LRU.
<b>Used by Capabilities</b>	r/o	ASCII String , list	TBD	List of other Capabilities that use this LRU. For each Capability: Observing Mode and rolled-up status for the part of the Capability implemented by this LRU.

LRU Parameter	Access <sup>1</sup>	Type	Range	Description
<b>Interface status</b>	r/o	Enumeration	OK COMMUNICATION-FAILURE	List of interfaces/network connections, their status, version of the protocol, time stamp for the last interface status change, re-transmission counters, etc.
<b>List of Components</b>	r/o	TBD	List of Component IDs	List of Components and their status, where applicable, i.e. if the LRU consists of other Components that can be individually monitored and controller via this interface.
<b>List of Capabilities</b>	r/o	TBD	List of Capability IDs	List of Capabilities implemented by this LRU (e.g. beams, station inputs, output products), if applicable. Note: The list of products per LRU may be extremely long. If that is the case, abbreviated form should be available and the complete list should be provided only when explicitly requested. Capability availability and rolled-up status provided on request.
<b>Alarm level</b>	r/w	Enumeration	TBD	Alarm reporting level.
<b>Alarm destination</b>	r/w	TBD	-	Destination address for alarm messages.
<b>List of active Alarms</b>	r/o	TBD	TBD	For each active alarm: Alarm ID, timestamp and brief description.
<b>List of all implemented Alarms</b>	r/o	TBD	TBD	List of all alarms implemented by LRU may be available – TBD.
<b>Logging level</b>	r/w	Enumeration	TBD	Level for local logging (if implemented by a sub-element)
<b>Logging of messages</b>	r/w	Enumeration	ENABLED DISABLED	Enable/disable logging of messages received and transmitted (supported only for monitor and control interfaces).
<b>Transmission of messages</b>	r/w	Enumeration	ENABLED DISABLED	Enable/disable transmission of messages for each interface individually.
<b>Destination for MP reports.</b>	r/w	TBD	TBD	Destination address for Monitoring Point reports.

LRU Parameter	Access <sup>1</sup>	Type	Range	Description
List of MPs and status	r/o	-	-	List of all Monitoring Points and their status.
Running since timestamp	r/o	Date & Time	UTC	Timestamp after last reboot/restart (independent of Administrative Mode changes).
Software location	r/o	ASCII String	URL	Location from which the software is loaded. (Consider implementing as r/w parameter, so that TM can easily switch between the versions).
Firmware location	r/o	ASCII String	URL	Location from which firmware (e.g. FPGA bitstreams) is loaded. Consider implementing as r/w parameter, so that TM can easily switch between the versions.
Log file location	r/o	ASCII String	URL	Location of the logging file.
<p><b>The following parameters/commands are related to the scheduled execution of commands. These commands will be used for debugging and troubleshooting. Similar commands are available at the CSP Element (top) level. It may be useful for TM to have access to the same commands at LRU level.</b></p>				
List received messages	r/o	ASCII Text	List of messages	List of received messages (input queue) that are waiting to be executed. Two options may be provided: short overview and full content.
List messages/commands in progress	r/o	ASCII Text	List of commands in progress and their status	List of commands in progress and their status.
List messages/commands waiting to be executed	r/o	ASCII Text	List of commands waiting to be executed.	List of commands waiting to be executed (waiting for the Activation Time).

LRU Parameter	Access <sup>1</sup>	Type	Range	Description
<b>Revoke transmitted message/command</b>	w/o	TBD	Revoke + identifier of the message to be revoked or revoke + all + status	Depending on the message/command status (received, in-progress, scheduled): a) remove a message from the input queue; b) cancel a message if in progress; c) discard a message waiting to be executed. Revoke may be specified for a particular message or for all messages with the specified status. Details TBD.
<b>Commands / state transitions</b> (Depending on the choice of the protocol, commands may be implemented either as dedicated messages or as write-only parameters transmitted in SET Parameter messages).				
<b>Power-down</b>	w/o	ASCII String	power-down	Power-down.
<b>Power-up</b>	w/o	ASCII String	power-up	Power-up.
<b>Restart</b>	w/o	ASCII String	restart	Hard restart.
<b>Reboot</b>	w/o	ASCII String	reboot	Reboot processor and reload software.
<b>Reset</b>	w/o	ASCII String	reset	TBD
<b>Transit to OFF-DUTY state</b>	w/o	ASCII String	go-off-duty	Set LRU in low-power state (Observing Mode OFF-DUTY). Not all LRUs implement low-power mode.
<b>Transit to READY state</b>	w/o	ASCII String	go-ready	This command can be used to request transition from low-power to ok state. This command may be also used to force CSP equipment to re-try transition to 'ok' state in the case when reboot or restart was not completed as expected.
<b>Firmware upgrade</b>	w/o	TBD	TBD	TM can query the location where the FPGA bitstream file is loaded from, deploy a new version and instruct CSP to re-program FPGA(s), i.e. TM may be required to use a sequence of commands.

LRU Parameter	Access <sup>1</sup>	Type	Range	Description
<b>Software upgrade</b>	w/o	TBD	TBD	TM can query location where the s/w is loaded from, deploy a new version of software and restart it, i.e. TM may be required to use a sequence of commands.

## 7.8.6 Parameters Common to all CSP\_Low Sub-arrays

Table 7-19 Parameters of the CSP\_Low Subarrays

Subarray Parameter	Access <sup>1</sup>	Type	Range	Description
<b>Reporting Entity</b>	r/o	ASCII String	Object ID and/or address	Identifier and address of the entity reporting on behalf of the sub-array (e.g. LMC).
<b>Version</b>	r/o	ASCII String	TBD	Version, If applicable.
<b>List of stations that belong to the sub-array</b>	r/w	List of Station IDs	TBD	List of stations assigned to this sub-array.
<p>Note: RFI related parameters that apply for all Observing Modes can be specified as parameters of a subarray. Parameters applicable for a single Observing Mode shall be specified as parameters of the Observing Mode. This is partly implementation dependent and the list will be finalized as design progresses. CSP_Low may implement additional parameters for dynamic control of RFI excision, for example, a choice of RFI excision algorithms.</p>				
<b>RFI mask</b>	R/W	TBD	TBD	Mask for RFI excision. Specified per station. See SKA1-CSP_REQ-2474-00 <a href="#">[RD8]</a> .
<b>RFI blanking level</b>	R/W	TBD	TBD	When to start RFI excision and for how long to perform excision. See SKA1-CSP_REQ-2473-00 <a href="#">[RD8]</a> .
<b>RFI blanking period</b>	R/W	TBD	TBD	
<b>Number or list of the Pulsar Search beams assigned to this sub-array</b>	r/w	Number of CSP-PSS-beams or List of PSS-Beams IDs	1..500	TM can specify either the number of PSS-beams or the list of PSS-Beam IDs.
<b>Number or list of the Pulsar Timing beams assigned to this sub-array</b>	r/w	Number of beams or List of CSP-PST-Beams IDs	1..16	TM can specify either the number of PST-beams or the list of PST-Beam IDs.
<b>Administrative Mode</b>	r/w	Enumeration	ENABLED, DISABLED, MAINTENANCE, NOT-FITTED	Sub-array Administrative Mode as set by TM. See Section 7.5.1.
<b>Administrative Mode Timestamp</b>	r/o	Date & Time	UTC	Timestamp for the last Administrative Mode change (for the sub-array).
<b>Administrative Mode Info</b>	r/w	ASCII Text	TBD	Information provided by TM, e.g. reason or purpose.

<sup>1</sup> Access can be defined as: read-only (r/o), read-write (r/w) or write-only (w/o).

Subarray Parameter	Access <sup>1</sup>	Type	Range	Description
<b>Administrative Mode Originator</b>	r/w	ASCII String	ID	ID of the person or entity who requested the last Admin. Mode change.
<b>Observing Mode</b>	r/w	Enumeration	IDLE IMAGING PULSAR-SEARCH PULSAR-TIMING	Observing Mode assigned by TM. Two or more Observing Modes can be set concurrently in the same sub-array. Observing Mode specific parameters listed in Section 7.8.9 shall be specified for each Observing Mode. See Sections 7.5.2 and 7.8.9 for more information.
<b>Observing Mode Timestamp</b>	r/o	Date & Time	UTC	Timestamp for the last Observing Mode change.
<b>Observing Mode Info</b>	r/w	ASCII Text	TBD	Additional information provided by TM.
<b>Observing Mode Originator</b>	r/w	ASCII String	ID	ID of the person or entity who requested the last Observing Mode change.
<b>Health State</b>	r/o	Enumeration	OK, DEGRADED, NOT-AVAILABLE	Sub-array Health State as derived/detected by CSP. See Section 7.5.7.
<b>Health State Timestamp</b>	r/o	Date & Time	UTC	Timestamp for the last Health State change.
<b>Health State Info</b>	r/o	ASCII Text	-	Optional additional information generated by CSP, for instance CSP may provide additional information if health is DEGRADED.
<b>Timestamp for the last change in the list of stations that 'belong' to the sub-array.</b>	r/o	Date & time	UTC	Timestamp for the last change of sub-array configuration – list of stations that belong to the sub-array
<b>Timestamp for the PSS-beam configuration change (for this subarray).</b>	r/o	Date & time	UTC	Timestamp for the last change of sub-array configuration – number and list of PSS-beams that are assigned to the sub-array.
<b>Timestamp for the PST-beam configuration change (for this subarray).</b>	r/o	Date & time	UTC	Timestamp for the last change of sub-array configuration – number and list of PST-beams that are assigned to the sub-array.
<b>Originator of the last station configuration change (for this subarray).</b>	r/o	ASCII String	ID	ID of the person or entity who requested the last station configuration change (added or removed stations to/from the sub-array).

Subarray Parameter	Access <sup>1</sup>	Type	Range	Description
<b>Originator of the last PSS-Beam configuration change (for this subarray).</b>	r/o	ASCII String	ID	ID of the person or entity who requested the last PSS-beam configuration change (added or removed PSS-beams to/from the sub-array).
<b>Originator of the last PST-Beam configuration change (for this subarray).</b>	r/o	ASCII String	ID	ID of the person or entity who requested the last PST-beam configuration change (added or removed PSS-beams to/from the sub-array).
<b>List of Components</b>	r/o	TBD	TBD	List of Components used by this sub-array and their status.
<b>List of active Alarms</b>	r/o	TBD	TBD	List of active alarms for the components and capabilities used by this sub-array.
<b>List of all implemented Alarms</b>	r/o	TBD	TBD	List of alarms implemented for the sub-array.
<b>List of MPs and status</b>	r/o	TBD	TBD	List of Monitoring Points for the sub-array and their status.
<b>Created at (timestamp)</b>	r/o	Date & Time	UTC	Timestamp when the sub-array was created.
<b>Originator</b>	r/w	ASCII String	ID	ID of the person or entity who requested creation of the sub-array.
<p><b>The following parameters/commands are related to the scheduled execution of commands. These commands will be used for debugging and troubleshooting. Similar commands are available at the CSP Element (top) level. It may be useful for TM to have access to the same commands at Capability level. May or may not be supported for all Capabilities, if at all.</b></p>				
<b>List messages/commands in progress</b>	r/o	ASCII Text	List of commands in progress and their status	List of commands in progress and their status (if applicable).
<b>List messages/commands waiting to be executed</b>	r/o	ASCII Text	List of commands waiting to be executed.	List of commands waiting to be executed (waiting for the Activation Time) – if applicable.
<b>Revoke transmitted message/command</b>	w/o	TBD	Revoke + identifier of the message to be revoked or revoke + all + status	Depending on the message/command status (received, in-progress, scheduled): a) remove a message from the input queue; b) cancel a message if in progress; c) discard a message waiting to be executed.  Revoke may be specified for a particular message or for all messages with the specified status.

## 7.8.7 Parameters Common to all CSP\_Low Capabilities

Table 7-20 Parameters Implemented by all CSP\_Low Capabilities (PSS-Beams, PST-Beams, station-input)

Capability Parameter	Access <sup>1</sup>	Type	Range	Description
<b>Reporting Entity</b>	r/o	ASCII String	Object ID and/or address	Identifier and address of the entity reporting on behalf of the Capability.
<b>Type</b>	r/o	Enumeration	PSS-Beam PST-Beam station-input	Capability type
<b>Version</b>	r/o	ASCII String	TBD	Capability version, If applicable.
<b>ID</b>	r/o	TBD	TBD	Identifier – ASCII string, naming conventions TBD.
<b>Administrative Mode</b>	r/w	Enumeration	ENABLED, DISABLED, MAINTENANCE, NOT-FITTED	Administrative Mode as set by TM. See Section 7.5.1.
<b>Administrative Mode Timestamp</b>	r/o	Date & Time	UTC	Timestamp for the last Administrative Mode change (for the sub-array).
<b>Administrative Mode Info</b>	r/w	ASCII Text	TBD	Information provided by TM, e.g. reason or purpose.
<b>Administrative Mode Originator</b>	r/w	ASCII String	ID	ID of the person or entity who requested the last Admin. Mode change.
<b>Observing Mode</b>	r/o	Enumeration	IDLE IMAGING PULSAR-SEARCH PULSAR-TIMING	TM assigns Observing Mode to a sub-array and in some cases specified a subset of Capabilities to be used.
<b>Observing Mode Timestamp</b>	r/o	Date & Time	UTC	Timestamp for the last Observing Mode change.
<b>Health State</b>	r/o	Enumeration	OK, DEGRADED, NOT-AVAILABLE	Health State of the Capability as derived/detected by CSP. See Section 7.5.7.
<b>Health State Timestamp</b>	r/o	Date & Time	UTC	Timestamp for the last Health State change.

<sup>1</sup> Access can be defined as: read-only (r/o), read-write (r/w) or write-only (w/o).

Capability Parameter	Access <sup>1</sup>	Type	Range	Description
Health State Info	r/o	ASCII Text	-	Optional additional information generated by CSP, for instance CSP may provide additional information if health is DEGRADED.
Belongs to sub-array	r/o	Sub-array ID	TBD	ID of the sub-array to which the Capability belongs.
Used by Capabilities	r/o	TBD	List of Capability IDs	In the case when the Capability is used by other Capabilities, CSP shall report the list of other Capabilities that use this Capability (so far such case has not been identified).
List of Components	r/o	TBD	TBD	List of Components used by this Capability and their status.
List of Capabilities	r/o	TBD	TBD	List of Capabilities used by this Capability and their status.
List of active Alarms	r/o	TBD	TBD	List of active alarms for the Capability and for the Components and Capabilities (if any) used by this Capability.
List of all implemented Alarms	r/o	TBD	TBD	List of all alarms implemented for the Capability.
Logging level	r/w	Enumeration	TBD	It is TBD whether this parameter is read-only or read-write.
Destination for MP reports.	r/w	TBD	TBD	Destination for Monitoring Point reports.
List of MPs and status	r/o	-	-	List of all Monitoring Points and their status.
Created at (timestamp)	r/o	Date & Time	UTC	Timestamp when the Capability was created
<p><b>The following parameters/commands are related to the scheduled execution of commands. These commands will be used for debugging and troubleshooting. Similar commands are available at the CSP Element (top) level. It may be useful for TM to have access to the same commands at Capability level. May or may not be supported for all Capabilities, if at all.</b></p>				
List messages/commands in progress	r/o	ASCII Text	List of commands in progress and their status	List of commands in progress and their status (if applicable).
List messages/commands waiting to be executed	r/o	ASCII Text	List of commands waiting to be executed.	List of commands waiting to be executed (waiting for the Activation Time) – if applicable.

Capability Parameter	Access <sup>1</sup>	Type	Range	Description
Revoke transmitted message/command	w/o	TBD	Revoke + identifier of the message to be revoked or revoke + all + status	Depending on the message/command status (received, in-progress, scheduled): a) remove a message from the input queue; b) cancel a message if in progress; c) discard a message waiting to be executed.  Revoke may be specified for a particular message or for all messages with the specified status. Details TBD.

## 7.8.8 Messages

### 7.8.8.1 Message: Set CSP Parameters

Table 7-21 Set CSP Parameters – Message Content

SET Message Parameters	Type	Range	Description
Message type	Enumeration	SET	Identifies type of message
Message ID	Integer	TBD	Unique Message Identifier. Used as a reference in responses, useful for testing and troubleshooting.
Time stamp	Date & Time	UTC	Time at which message was generated. Displayed as defined in ISO8601.
Activation Time	Date & Time	UTC	Time at which parameter change should take effect. Displayed as defined in ISO8601.
Object ID	ASCII String	Valid CSP managed object ID	Identifier of the object (CSP top level, Sub-element, Component, Capability, LRU). Naming conventions TBD.
Parameter Name	Enumeration	Any supported read-write or write-only parameter	Parameter name and value to be set at the specified activation time. Name and value can be specified more than once in the same message, i.e. a single message may be used to set two or more parameters. There is no limit on the number of parameters that can be set using the same message.
Parameter Value	Any supported type	As defined for the specified parameter	

CSP response to the SET Parameters message is still to be defined (TBD); CSP could generate any (or all) of the following:

1. Immediately upon receipt of a SET Parameters message, CSP could generate Acknowledgement (ACK) to inform TM that the message was received and will be processed. Negative Acknowledgement (NACK) could be used to inform TM that CSP received invalid or unexpected message.
2. After CSP has examined the message and determined whether or not the message can be executed at the specified Activation Time, CSP could inform TM using messages Accept or Reject.
3. After the Activation Time, CSP could report whether command was successfully executed.

### 7.8.8.2 Message: Get CSP Parameters (Query)

TM uses GET message to query CSP parameters. Upon receipt of the GET message, CSP collects parameter setup/status and responds with Response message.

CSP makes provision for TM to specify the level of detail and depth of hierarchy to be reported.

Logical and functional grouping of CSP parameters will be defined to provide tailored view of the CSP for various purposes. This version of the document describes only a basic GET message.

**Table 7-22 Get CSP Parameters – Message Content**

GET Message Parameter	Type	Range	Description
Message type	Enumeration	GET	Identifies type of message.
Message ID	Integer	TBD	Unique Message Identifier.
Time stamp	Date & Time	UTC	Time at which message was generated. Displayed as defined in ISO8601.
Object ID	ASCII String	Valid CSP managed object ID	Identifier of the object (CSP top level, sub-element, Component, Capability, LRU). Naming conventions TBD.
Parameter Name	Enumeration	Any supported parameter	Parameter name. More than one parameter can be listed. When parameters are not specified in a GET message, CSP reports all parameters for a given object, or to avoid generation of extremely large messages, CSP may report a pre-defined sub-set of parameters (TBD).

### 7.8.8.3 Message: CSP Response

CSP generates response messages to reply to messages SET Parameters and GET Parameters.

**Table 7-23 CSP Response – Message Content**

Response Message Parameter	Type	Range	Description
Message type	Enumeration	RESPONSE	Identifies type of message.
Message ID	Integer	TBD	Unique Message Identifier.
Time stamp	Date & Time	UTC	Time at which the message was generated. Displayed as defined in ISO8601.
Reference	Integer	Message ID	Identifier of the SET or GET message that this message responds to.

<b>Response Message Parameter</b>	<b>Type</b>	<b>Range</b>	<b>Description</b>
<b>Object ID</b>	ASCII String	Valid CSP managed object ID	Identifier of the Component or Capability (as specified in SET/GET).
<b>Parameter Name</b>	ASCII String	Any supported parameter	Parameters specified/requested in GET/SET message. Parameter name and value can be specified more than once in the same message. I.e. value for more than one parameter may be reported in the same message.
<b>Parameter Value</b>	Any supported type	As defined for the specified parameter	

7.8.8.4 Message: Alarm

Table 7-24 CSP Alarm – Message Content

Alarm Message Parameter	Type	Range	Description
Message Type	ASCII String	ALARM	Identifies type of message.
Message ID	Integer	TBD	Unique Message Identifier.
Time stamp	Date & Time	UTC	Time at which the message was generated. Displayed as defined in ISO8601.
Alarm ID	Enumeration	TBD	Unique Alarm Identifier.
Severity level	Enumeration	TBD	As defined in <a href="#">[RD4]</a> .
Alarm State	Enumeration	RAISE CLEAR TRANSIENT	Values as specified in <a href="#">[RD4]</a> . Enumeration specified here is just an example.
Alarm Timestamp	Date & Time	UTC	Date and time at which failure/error was detected.
Failed Object ID	Object ID	TBD	Identifier of a CSP Component or Capability.
Alarm Detection Point	Object ID	TBD	Entity that detected and reported alarm (if not the same as failed entity).
Parameter Name	ASCII String	Enumeration	Parameter name and value can be specified more than once in the same message. I.e. values for more than one parameter may be reported in the same message.
Parameter Value	Any supported type	As defined for the specified parameter	
Description	ASCII Text	-	Text describing the failure/event, if not provided to TM in advance. Description provides information related to failure and identifies potential causes and effects the failure can have on the functionality and output.
Suggested Action	ASCII Text	-	Text describing suggested action if not provided to TM in advance.

### 7.8.8.5 Message: Event

An Event message is generated by CSP to report events which require TM attention.

For example an Event message may be used to report that CSP has completed re-configuration and is ready to start generating output for a scan or an Observing Mode.

**Table 7-25 CSP Event – Message Content**

Event Message Parameter	Type	Range	Description
Message type	ASCII String	EVENT	Identifies type of message.
Message ID	Integer	TBD	Unique Message Identifier.
Time stamp	Date & Time	UTC	Time at which the message was generated. Displayed as defined in ISO8601.
Event ID	Enumeration	TBD	Unique Event Identifier.
Event Timestamp	Date & Time	UTC	Date and time at which event was detected.
Event Detection Point	Object ID	TBD	Entity that detected and reported event (if not same as failed entity).
Object ID	ASCII String	Valid CSP managed object ID	Optionally, event message may include values of the parameters of interest (e.g. status) for the Component or Capability for which the event applies/relates.
Parameter Name	Enumeration	TBD	Parameter name and value can be specified more than once in the same message. I.e. values for more than one parameter may be reported in the same message.
Parameter Value	Any supported type	As defined for the specified parameter	
Description	ASCII Text	-	Text describing the event if not provided to TM in advance. Text describing the failure/event if not provided to TM in advance. Additional information and/or potential causes and effects the event may have on the system functionality and output.

### 7.8.8.6 Message: Monitor Point Report

CSP will report status of the Monitor Points periodically or on request.

**Table 7-26 CSP Monitor Point Status Report – Message Content**

<b>Monitor Point Report Parameter</b>	<b>Type</b>	<b>Range</b>	<b>Description</b>
<b>Message type</b>	ASCII String	MP-REPORT	Identifies type of message.
<b>Message ID</b>	Integer	TBD	Unique Message Identifier.
<b>Time stamp</b>	Date & Time	UTC	Time at which message was generated. Displayed as defined in ISO8601.
<b>Object ID</b>	ASCII String	Valid CSP managed object ID	Identifier of the object for which monitor points are being reported.
<b>Parameter Name</b>	Enumeration	TBD	Parameter name and value can be specified more than once in the same message. I.e. values for more than one monitor point may be reported in the same message.
<b>Parameter Value</b>	Any supported type	As defined for the specified parameter	

*7.8.8.7 Message: CSP Auxiliary Data Report*

The format for the Auxiliary Data Reports is TBD.

### 7.8.8.8 Sub-array and Observing Mode Related Messages

This section defines the content of the sub-array and Observing Mode related messages.

CSP supports the following operations:

1. Create sub-array.
2. Add / remove stations to/from a sub-array.
3. Add/remove PSS beams and/or PST beams to/from a sub-array
4. Set Observing Mode for a sub-array. Start scan.
5. Stop scan (Observing Mode).
6. Time-Dependent updates for Observing Mode.
7. Delete sub-array.
8. Set engineering parameters for a sub-array.
9. Query parameters and status, including status and setup of Capabilities.

Table 7-27 lists content of the message ‘create subarray’, the table does not show the Message ID, timestamp and other parameters common to all messages.

Table 7-28 shows content of the message ‘modify sub-array’, this message may be used to add/remove stations, CSP-PSS-Beams and CSP-PST-Beams to/from sub-array. In addition, TM will be able to specify engineering parameters as defined in Table 7-19 and Observing mode parameters as specified in Section 7.8.9.

Table 7-29 shows content of the message Delete Sub-array.

Date & time when to start sub-array re-configuration is indicated in the message. If the message does not specify date & time, CSP shall execute command (and start re-configuration) as soon as possible.

**Table 7-27 CSP\_Low Create Sub-array**

Parameter	Type	Range	Description
Sub-array ID	ASCII String	-	Sub-array ID – must be unique.
Action	Enumeration	Create	Create a new sub-array.
Activation time	Date & time	UTC	Date and time when to configure new sub-array.

**Table 7-28 CSP\_Low Modify Sub-array**

Modify Sub-array Parameter	Type	Range	Description
Sub-array ID	ASCII String	-	Sub-array ID
Action	Enumeration	Set (TBC)	Modify existing sub-array – add/remove stations, PSS, PST and/or VLBI beams. In addition to the parameters specified here, any of the sub-array parameters listed in Table 7-19 can be specified in this message
Activation time	Date & time	UTC	Date and time when to start re-configuration of the sub-array.
List of stations	List	Station IDs	List of stations that belong to the sub-array.
Array delay center	TBD	TBD	Geographic coordinates of the sub-array delay centre.
CSP-PSS-Beams	Integer or List of Capability IDs	Up to 500	TM may specify either the number or the list of CSP-PSS-Beams to be assigned to the sub-array. The latter may be useful for testing and commissioning.
CSP-PST-Beams	Integer or List of Capability IDs	Up to 16	TM may specify either the number or the list of CSP-PST-Beams to be assigned to the sub-array. The latter may be useful for testing and commissioning.

**Table 7-29 CSP\_Low Delete Sub-array**

Parameter	Type	Range	Description
Sub-array ID	ASCII String	-	Sub-array ID
Action	Enumeration	Delete	Delete existing sub-array.
Activation time	Date & time	UTC	Date and time when to execute command delete sub-array.

### 7.8.9 Observing Mode Parameters

Table 7-30 lists parameters to be specified by TM to set CSP Sub-array for a particular Observing Mode and start scan.

If the message does not specify date & time when to start re-configuration of the sub-array, CSP\_Low shall start re-configuration as soon as possible.

CSP\_Low shall start to generate output data as soon as re-configuration is completed.

Note: In general, a sub-array will use more than one, and in most cases many, output spigots to transmit generated output products; not all CSP\_Low components begin generating output data at the same time. At the beginning of the scan CSP\_Low may generate incomplete data sets; the recipient of the CSP\_Low output (SDP) must be able to handle incomplete data sets.

Once configured, CSP\_Low keeps the sub-array configuration until one of the following happens:

- CSP\_Low receives a command to re-configure the sub-array (as specified in Table 7-30 and Table 7-31).
- CSP\_Low receives a command to delete the sub-array.
- CSP\_Low receives a command to transit in low-power mode.
- CSP\_Low receives a command to shut-down.
- CSP\_Low monitor and control function fails so that the configuration is lost.

**Table 7-30 CSP\_Low Scan Configuration**

CSP_Low Scan Configuration	Type	Range	Description
<b>Sub-array ID</b>	ASCII String	-	Sub-array ID
<b>Action</b>	Enumeration	Set	Set sub-array parameters
<b>Activation Time</b>	Date & time	UTC	Date and time when to start re-configuration of the sub-array
<b>Scan ID</b>	Integer	64-bit	Scan ID to be inserted in the CSP_Low output products.
Any combination of the following Observing Modes can be specified (in the same message):			
<b>Observing Mode Imaging</b>	Enumeration	IMAGING	Observing Mode parameters as specified in Section 7.8.9.1
<b>Observing Mode Pulsar Search</b>	Enumeration	PULSAR SEARCH	Observing Mode parameters as specified in Section 7.8.9.2
<b>Observing Mode Pulsar Timing</b>	Enumeration	PULSAR TIMING	Observing Mode parameters as specified in Section 7.8.9.3

Table 7-31 CSP\_Low.CBF End Scan

CSP_Low End Scan	Type	Range	Description
Sub-array ID	ASCII String	-	Sub-array ID
Action	Enumeration	Set	Set sub-array parameters.
Activation Time	Date & time	UTC	Date and time when to start re-configuration of the sub-array.
Observing Mode	Enumeration	IDLE	Set the sub-array in IDLE mode.

### 7.8.9.1 Observing Mode Imaging

CSP\_Low configuration and output products are the same for Spectral-line Imaging and Continuum Imaging; CSP\_Low implements a single Observing Mode Imaging.

Observing Mode Imaging can be started, operated and stopped independently and concurrently in up to 16 sub-arrays.

*TBD: Input from each station may consist of up to 16 beams. It is to be defined whether the station-beam setup is configurable, and should it be specified as a part of a scan configuration.*

For each sub-array in Imaging mode, CSP\_Low makes provision for TM to configure up to four zoom windows, each with bandwidth selected independently from values within 10% of 4 MHz, 8 MHz, 16 MHz, and 32 MHz; each independently tuned, with frequency granularity better than 1.1 MHz, such that the entire zoom window lies anywhere within the subarray's digitized observing band. CSP\_Low is able to produce output products (correlated visibilities and autocorrelations for all polarization products) concurrently for four zoom windows and over the entire digitized band (300MHz, 384 coarse channels), with an evenly spaced frequency resolution of 1 MHz +/- 10%.

TBD:

- Does CSP\_Low allow for total of up to 4 zoom windows per sub-array, or up to 4 zoom windows per beam per sub-array?
- Are frequency resolution, integration time, number of phase bins and zoom window configuration same for all the beams in the sub-array? Or, are some, or all, of these parameters specified for each beam independently?

CSP\_Low can perform Imaging concurrently in the same sub-array with the following Observing Modes:

1. Pulsar Search
2. Pulsar Timing

Before CSP\_Low can begin to generate output products, the CSP\_Low and SDP shall exchange the following information via TM:

1. After allocating resources to be used for the specified sub-array and Observing Mode, CSP\_Low shall notify SDP, via TM, regarding the number of output streams (spigots) to be used to transmit CSP\_Low products (visibilities) to SDP. Distribution of baselines/products per stream shall be also indicated.
2. SDP shall send to CSP\_Low, via TM, destination addresses (MAC and IP addresses) for the CSP\_Low products.

For each sub-array in Imaging mode, TM shall provide delay tracking and calibration in advance of the specified configuration Activation Time, and at regular intervals during the scan, as follows:

1. Delay models per beam per station. (There can be up to 16 beams per station.)
2. On-sky calibration residuals sourcing from the SDP (i.e. station-based delay corrections).

To reduce the number of messages, delay correction derived from calibration (and provided by SDP) shall be rolled-up (by TM) with the geometric delay and delivered as a single delay model.

For each delay model TM shall specify the time when the model becomes valid (i.e. when CSP\_Low should start using the model).

TBD:

- How much in advance a delay model has to be delivered to CSP\_Low in order to be applied at the time specified in the message.
- Required cadence and number of coefficients for the delay models provided by TM.

**Table 7-32 CSP\_Low Parameters for the Observing Mode Imaging**

CSP_Low Imaging Mode Parameter	Type	Range	Description
Observing Mode	Enumeration	IMAGING	Observing Mode
Frequency resolution	TBD	TBD	
Integration time	Integer	TBD	Integration time for correlation products, specified in milliseconds. SKA1-CSP_REQ-2150-0 [RD8]: For each subarray, CSP_Low shall have independently configurable visibility integration periods which are within 2% of 0.9s, and within 10% of (2, 3, 5, 10) times 0.9s.
Number of phase bins	Integer	1..256 (TBC)	By default, the number of phase bins = 1, i.e. no binning. Number of phase bins * number of channels * 4 <= 262,144. Number of phase bins may be defined as enumeration, i.e. only a limited choice may be available (TBD).
Pulsar ephemeris	TBD	TBD	Parameters required for pulsar phase binning, i.e. required

CSP_Low Imaging Mode Parameter	Type	Range	Description
The physical location of the centre of the sub-array	TBD	TBD	when the number of phase bins > 1.
Optionally, TM may specify up to 4 zoom windows. TBC: The parameters specified below are specified for each zoom window independently.			
Zoom window ID	Integer	1..4	Identifier used to refer to the zoom window in commands, queries and reports (including alarms). Must be unique within a sub-array.
Bandwidth	Enumeration	TBD	Defined CSP RS Rev1 [RD8]: 4 MHz, 8 MHz, 16 MHz, and 32 MHz ( $\pm 10\%$ ). TBD: is the zoom window bandwidth specified as a multiple of the coarse channel bandwidth?
Position of the zoom window within the input band	TBD	TBD	Zoom window may be placed anywhere within the digitized input band (300MHz), but quantized to the coarse channel frequencies. TBD: Is a zoom window defined by the list of coarse channels or by the central frequency?
Number of spectral channels	Enumeration	TBD	Most likely a user (via TM) will be able to choose the number of channels from the pre-defined list of values.

### 7.8.9.2 Observing Mode Pulsar Search

Overview of the CSP\_Low functionality related to the Observing Mode Pulsar Search:

1. CSP\_Low can form up to 500 Pulsar Search beams and concurrently perform Pulsar Search in all 500 beams.
2. CSP-PSS-Beams (Capabilities) must be assigned to a sub-array in advance of the Observing Mode configuration.
3. Restriction: CSP-PSS-Beams are assigned to sub-arrays in pairs, CSP-PSS-Beams  $n$  and  $n+1$ , where  $n$  is an even number, must belong to the same sub-array. This restriction is imposed by implementation: each PSS LRU can perform pulsar search for two PSS beams; two PSS beams processed on the same LRU cannot be used for different observations. Other than this restriction, any (even) number of CSP-PSS-Beams can be assigned to any sub-array.
4. Pulsar Search can be started, operated and stopped independently and concurrently in up to 16 sub-arrays.
5. Note, however, that the next scan cannot be started in the same sub-array before pulsar search has been completed for the previous pulsar search scan. More precisely, TM should wait for CSP notification that pulsar search has been completed before starting the next scan in the same sub-array. During construction, CSP will provide an estimate for the delay required between the end of data acquisition from antennas and the end of pulsar search so that this can be taken in consideration when scheduling

observations. TM shall allow for this parameter to be updated (it may change when hardware and software are updated).

For the Observing Mode Pulsar Search to be accepted, the sub-array must contain at least one station and at least one CSP-PSS-beam. In other words, stations and CSP-PSS-beams must be assigned to a sub-array in advance of Observing Mode configuration. Re-configuration may be scheduled to start at the same time for 'add receptor', 'add CSP-PSS-beam' and 'set Observing Mode' requests.

Destination addresses for CSP\_Low output data shall be provided by TM before the scan Activation Time. If some of the destination addresses change during the scan (due to failure of SDP equipment or other event) CSP\_Low accepts the new destination addresses as provided by TM and starts using them as soon as possible.

CSP\_Low implements two sub-modes:

1. Acceleration Search (a.k.a. Pulsar search), and
2. Single Pulse Search (a.k.a. Transient Search)

Acceleration Search and Transient Search can be performed concurrently on the same set of CSP-PSS-Beams. Table 7-33 lists parameters for both sub-modes.

CSP\_Low can perform Pulsar Search concurrently in the same sub-array with the following Observing Modes:

1. Imaging
2. Pulsar Timing

Note1: Pulsar Search Engine requires that an entire observation be completed before it can begin the required acceleration processing modules; there is a lag between the end of an observation and the end of processing. This places some constraints on scheduling of observations as scheduling a short observation after a long observation may result in the CSP\_Low not being ready for the next observation. The CSP\_Low uses an Event message to report to TM when the processing has been completed.

Note2: In addition to Pulsar Search output data, for every sub-array in Pulsar Search Mode, CSP\_Low also generates visibilities. It is still to be defined (TBD) whether CSP\_Low will define a default set of Imaging parameters to be used for a sub-array in Pulsar Search mode when Imaging is not explicitly required.

In addition to the parameters specified in Table 7-33, in advance of the scan Activation Time and at regular intervals during the scan, TM shall provide the following parameters:

1. Delay models for each station that belongs to the sub-array, as specified in Section 7.8.9.1.
2. On-sky calibration residuals sourcing from the SDP (i.e. station-based delay corrections), as specified in Section 7.8.9.1.
3. Per-station, per PSS beam delay+cal models relative to the primary delay center. Up to 500\*512 models. However, these models are relative to the delay center and should have a low update rate.
4. Weights to be used in beamforming (per station per beam).
5. Jones Matrix per coarse channel (1MHz channel) per PSS beam; up to 384x500=192,000 Jones Matrices.
6. Pulsar ephemeris models (polynomials) at a regular rate.
7. Diurnal variations of ionospheric parameters (that affect Doppler Frequency Shift). (TBC)

Messages that contain above listed parameters shall indicate the time when CSP\_Low should start using new values.

To be defined (TBD):

- Cadence for the updates.
- How much in advance a message must be delivered to CSP\_Low in order to be applied at the time specified in the message.

**Table 7-33 CSP\_Low Pulsar Search Observing Mode Parameters**

<b>CSP_Low Pulsar Search Mode Parameter</b>	<b>Type</b>	<b>Range</b>	<b>Description</b>
<b>Observing Mode</b>	ASCII String	Pulsar Search	Observing Mode
<b>Stations to be used in beamforming</b>	List of station IDs	Up to 512 stations	Optionally user may specify a subset of stations that belong to the sub-array to be used in tied-array beamforming. If not specified, all stations that belong to the sub-array are used in beamforming.
<b>Station-beam to be used in beamforming</b>	List	1..16	Input from each station may consist of up to 16 beams. This parameter identifies the station-beam to be used in PSS-beamforming.
<b>Phase centre of the sub-array</b>	TBD	TBD	
<b>Number of bits per sample</b>	Integer	TBD	Number of bits per sample in the beamformer output (for the data passed internally from CSP_Low.CBF to CSP_Low.PSS).
<b>Number of channels per beam</b>	Integer	4096	TBD: Is this a configurable parameter?
<b>PSS Beam BW</b>	Integer	TBD	TBD: Is PSS-beam bandwidth specified as a multiple of the coarse channel bandwidth? TBD: User (via TM) can trade the number of PSS beams for bandwidth – details TBD.
<b>Pointing Name</b>	TBD	TBD	Station-beam pointing symbolic name and coordinates (observed source name and co-ordinates).
<b>Pointing coordinates</b>	TBD	TBD	
<b>Pointing switching schedule</b>	TBD	TBD	Schedule for switching between observed sources - if CSP_Low is required to handle source switching autonomously (requirements TBD).
<b>Acceleration Search</b>	Boolean	Enable Disable	Observing Mode: Acceleration Search (a.k.a. Pulsar Search) and Single Pulse Search (a.k.a. Transient

CSP_Low Pulsar Search Mode Parameter	Type	Range	Description
Single Pulse Search	Boolean	Enable Disable	Search) can be performed concurrently.
Integration time	Integer	Up to 1,800 seconds	The maximum length of a data chunk on which search can be performed.
Acceleration Range	Integer	0-350 m/s <sup>2</sup> Default=0	Range in source acceleration to be searched. Specified as integer number of milliseconds.
Number of Trials	Integer	TBD	Number of trials to be performed.
Time Resolution	Integer	50-800 $\mu$ s ( $2n * 50 \mu$ s)	Time resolution of input data. Values in increments ( $2n * 50 \mu$ s).
Dispersion Measure	Integer	0-3000 pc cm <sup>-3</sup>	Dispersion correction. TBD: Do we need a separate DM search range for single-pulsar sources (transients).
Number of Frequency Channels	Integer	1,000-4,096	Multiple of 1,000 or of 1,024 (NIP.PSS.009) - TBD
Time Sample per Block	Integer	Up to (integration time) / (time resolution)	Number of time samples in each block of data.
Sub-bands	Integer	Up to 64	Number of frequency band groups summed up during folding.
Input Buffer Size	Integer	2 <sup>18</sup> -2 <sup>24</sup>	Size of the buffer receiving raw data.
Harmonic summing control parameter	Integer	1-32	Number of the "harmonic folds" on the initial Fourier power-spectrum summed up.
Complex FFT Control Parameters	TBD	TBD	CXFT control parameters.
Candidate Sifting Parameters	TBD	TBD	Constraints on matches between candidates.
Candidates Output Parameters	TBD	TBD	Define data sinks and subscriber to be notified (TBC).
Single Pulse Threshold	Integer	TBD	Threshold for a single pulse trigger.
Single Pulse Optimization Parameters	TBD	TBD	Single pulse optimization parameters.
RFI Mask list	TBD	TBD	Interference Mask to be used and updated by CSP_Low PSS Engine.

CSP_Low Pulsar Search Mode Parameter	Type	Range	Description
<b>DREAD Beam Statistics</b>	Table of floats	TBD	DREAD: statistics of spectra to derive the normalization factors.
<b>CDOS Control Parameters</b>	TBD	TBD	CDOS: control parameters and related statistical data. Exact list of parameters TBD.
<b>RFIM Control Parameters</b>	TBD	TBD	RFIM: control parameters. Exact list of parameters TBD.
<b>Parameters specified for each PSS beam (for up to 500 PSS-beams):</b>			
<b>PSS Beam ID</b>	Integer	TBD	PSS Beam ID as specified by TM. Note: Beam ID specified by TM is not the same as CSP Capability ID. Internally CSP_Low must perform mapping of TM assigned IDs into CSP IDs. TBD: CSP_Low could make provision for user to choose PSS beam configuration/packing from one of the pre-defined beam configurations or to specify parameters for each PSS beam independently.
<b>PSS Beam pointing co-ordinates</b>	TBD	TBD	Pointing co-ordinates of the beam.
<b>PSS Beam centre frequency</b>	Integer	TBD	Beam centre frequency and beam delay centre, relative to the array delay center.
<b>PSS Beam delay centre</b>	TBD	TBD	
<b>Destination address</b>	TBD	TBD	Per beam destination address, for CSP_Low output, must be provided by TM before CSP can start generating output products.

### 7.8.9.3 Observing Mode Pulsar Timing

Overview of the CSP\_Low functionality related to the Observing Mode Pulsar Timing:

1. CSP\_Low can form up to 16 Pulsar Timing beams and concurrently perform Pulsar Timing in all 16 beams.
2. Pulsar Timing can be started, operated and stopped independently and concurrently in up to 16 sub-arrays.
3. CSP-PST-Beams (Capabilities) must be assigned to a sub-array in advance of the Observing Mode configuration.

4. Any number of CSP-PST-Beams can be assigned to any sub-array.

For the Observing Mode Pulsar Timing to be accepted by CSP\_Low, the sub-array must contain at least one station and at least one CSP-PST-Beam. In other words, stations and CSP-PST-Beams must be assigned to a sub-array in advance of Observing Mode configuration. Re-configuration may be scheduled to start at the same time for 'add station', 'add CSP-PSS-Beam' and 'set Observing Mode' requests.

CSP\_Low performs mapping between PST-beam IDs assigned by TM (in the configuration message) and CSP\_Low Capabilities CSP-PST-Beam, in other words CSP\_Low selects which hardware and software components to use for which PST-Beam.

As shown in Table 7-35, TM shall provide destination addresses for the CSP output data before the scan Activation Time. If, due to SDP failure or other event, some of the destination addresses change during the scan, CSP\_Low accepts the new destination addresses as provided by TM and starts using them as soon as possible.

TM shall provide a mechanism for CSP\_Low to notify TM that Pulsar Timing for a particular beam has been completed. Upon receipt of such notification, TM shall stop Pulsar Timing for that particular pulsar (PST-Beam).

CSP\_Low shall make provision for TM to stop Pulsar Timing for individual beam (i.e. pulsar). When such request is received, CSP\_Low shall end timing operation for the specified pulsar and send results to SDP as appropriate. Pulsar Timing in other beams that belong to the same sub-array shall not be affected.

Note1: As specified in Table 7-35, SDP, via TM, shall provide pulsar phase/longitude predictors as generated from the pulsar ephemerides (either managed by the SDP or TM). Alternatively, SDP could provide and manage the software stack for predictor generation which is installed as a part of CSP\_Low and used by CSP\_Low to generate pulsar phase/longitude predictors with ephemerides provided by TM (via updates from SDP). The size of predictor files scales with the number of frequency and time coefficients and the length of the validity period. For a 12 time, a 12 frequency coefficient predictor that lasts 3 hours, the file size is 15 kB. Increasing the validity period by a factor of two increases the file size by a factor of 2 and vice versa. The largest predictor file that may be considered for SKA1 would be of the order 60 kB. Polyco files with the same number of coefficients are smaller than this; only predictor file sizes need to be considered when calculating maximum data rates.

Note2: In addition to Pulsar Timing output data, for every sub-array in Pulsar Timing Mode, CSP\_Low also generates visibilities. In addition to parameters listed in Table 7-35, TM has to provide parameters required for Imaging mode, as specified in Section 7.8.9.1. It is still to be defined (TBD) whether CSP\_Low will implement a default set of Imaging parameters to be used for a sub-array in Pulsar Timing mode where Imaging is not explicitly required.

TBD: CSP Consortium anticipates that the human telescope operators will need to see various forms of diagnostic information during the observation. Content and format for plots will be specified here in this ICD. TM Consortium will select the format for diagnostic plots.

TBD: CSP could send to TM images (e.g. PNG or JIF) or, if the TM expects to be able to query the data in multiple ways, CSP might send TM a copy (possibly reduced in dimension) of the information that is sent to the SDP.

**Table 7-34 Legend for Table 7-35**

Mode	
All	All
PT	Pulsar Timing
DS	Dynamic Spectrum (proposed - TBC)
FT	Flow Through (proposed - TBC)
When is the parameter required (When):	
SCfg	Scan Configuration - to be received in advance of the 'activation time' specified in the message.
SStart	At the beginning of the scan.
Intg	To be received after each N-second integration.
MC	May change, i.e. may be updated during an observation (scan).
Implementation (support) Priority (P) :	
H	High
M	Medium
L	Low
C	Not known if required until calibration model of the telescope is defined.
D	To be defined (TBD)

**Table 7-35 CSP\_Low – Pulsar Timing Observing Mode Parameters**

CSP_Low Parameter (Keyword)	Type	Units, valid values or range (all TBD)	Mode	When	Description	P
The following parameters are specified for each of up to 16 PST beams:						
PST Beam ID	TBD	TBD	All	SCfg	Identifier assigned by TM and used to identify beam configuration. CSP_Low selects which Capability CSP-PST-Beam to use for each configuration.	H
PST Beam BW	Integer	300MHz	All	SCfg	TBD: Is this configurable parameter?	H

CSP_Low Parameter (Keyword)	Type	Units, valid values or range (all TBD)	Mode	When	Description	P
Type of observation to be performed (OBSMODE)	string	PULSAR_TIMING	All	SCfg	The observing mode that the given PST server should operate in. In the future two more Observing Modes may be added (an ECP has been proposed to add Observing Modes): Dynamic Spectrum and Flow Through.	H
Observer ID (OBSERVER)	string		All	SCfg	Observer in charge of observations.	L
Project ID (PROJID)	string		All	SCfg	Project that the observations are for.	L
Pointing ID (PNT_ID)	string		All	SCfg	ID for sub-array pointing.	H
Source (SRC_NAME)	string		All	SCfg	Name or ID of source	H
ITRF (ITRF)	double [3]	[m,m,m]	All	SCfg	ITRF coordinates of the telescope delay centre.	H
Receiver ID (FRONTEND)	string		All	SCfg	Receiver name or ID (instrument).	H
Number of polarization channels (NRCVR)	int	[1, 2]	All	SCfg		H
Native polarization of feed (FD_POLN)	string	[LIN, CIRC]	All	SCfg		H

CSP_Low Parameter (Keyword)	Type	Units, valid values or range (all TBD)	Mode	When	Description	P
Feed handedness (FD_HAND)	int	[-1,+1]	All	SCfg	Code for sense of feed. FD_HAND = +1 for XYZ forming RH set with Z in the direction of propagation. Looking up into the feed of a prime-focus receiver (or at the sky), for FD_HAND = +1, the rotation from A (or X) to B (or Y) is counter clockwise or in the direction of increasing Feed Angle (FA) or Position Angle (PA). For circular feeds, FD_HAND = +1 for IEEE LCP on the A (or X) probe. See van Straten et al. (2010), PASA, 27, 104 for a full description of the polarisation conventions adopted for PSRFITS and PSRCHIVE [RD12] and [RD13].	H
Feed angle (FD_SANG)	double	Degrees [-180,180]	All	SCfg	Feed angle of the E-vector for an equal in-phase response from the A(X) and B(Y) probes, measured in the direction of increasing FA or PA (clockwise when looking down on a prime focus receiver) and in the range +/- 180 deg.	H
Feed tracking mode (FD_MODE)	string	[FA, CPA, SPA, TPA]	All	SCfg	During an observation: 'FA' means constant FA, i.e., that the feed stays fixed with respect to the telescope reference frame; 'CPA' means that the feed rotates to maintain a constant PA, i.e., it tracks the variation of parallactic angle. Note that for COORD_MD = 'GALACTIC' PA is with respect to Galactic north and similarly, for COORD_MD = 'ECLIPTIC' PA is with respect to Ecliptic north. For 'SPA' the FA is held fixed at an angle such that the requested PA is obtained at the mid-point of the observation; 'TPA' is relevant only for scan observations - the feed is rotated to maintain a constant angle with respect to the scan direction.	H
Feed position angle (FA_REQ)	double	Deg. [-180,180]	All	SCfg	The requested angle of the feed reference, for FD_MODE = 'FA', with respect to the telescope reference frame (FA = 0), and for FD_MODE = 'CPA', with respect to celestial north (PA = 0) or with respect to Galactic north for COORD_MD = 'GALACTIC'.	H
Centre frequency (OBSFREQ)	double	MHz	All	SCfg	Centre frequency of observation.	H
Total (critical) bandwidth (OBSBW)	double	MHz	All	SCfg	Total (critical) bandwidth of observation.	H

CSP_Low Parameter (Keyword)	Type	Units, valid values or range (all TBD)	Mode	When	Description	P
Number of frequency channels (OBSNCHAN)	double		All	SCfg		H
Oversampling ratio (OVSAMP)	int[2]		All	SCfg	Numerator and denominator for the oversampling ratio.	H
Beam major axis (BMAJ)	double	Deg.	All	SCfg		L
Beam minor axis (BMIN)	double	Deg.	All	SCfg		L
Beam position angle (BPA)	double	Deg.	All	SCfg		L
Frame of coordinates (COORD_MD)	string	[GALACTIC, EQUATORIAL, ECLIPTIC]	All	SCfg	Frame of coordinates.	H
Coordinate epoch (EQUINOX)	double	JD or MJD	All	SCfg		H
STT_CTD1	string	hh:mm:ss.sss or ddd.ddd	All	SStart	X component of starting coordinates in COORD_MD frame.	H
STT_CRD2	string	hh:mm:ss.sss or ddd.ddd	All	SStart	Y component of starting coordinates in COORD_MD frame.	H

CSP_Low Parameter (Keyword)	Type	Units, valid values or range (all TBD)	Mode	When	Description	P
Track mode (TRK_MODE)	string	TRACK, SCANGC, SCANLA	All	SCfg	For 'TRACK' the beam axis tracks a fixed point on the sky; for 'SCANGC' the beam axis tracks at a uniform rate along a great circle on the sky; for 'SCANLAT' the beam axis tracks at a uniform rate along a line of constant latitude or declination (depending on COORD_MD).	H
STP_CRD1	string	hh:mm:ss.sss or ddd.ddd	All	Intg	X component of the final coordinates in COORD_MD frame. Required at the end of each sub-integration (not the end of each observation). Required for non-tracking observations only. This could be every 1 second to every hour, but would only be needed in the case of non-tracked scans. Currently there is no use case for non-tracked scans with the PST, so this is unlikely to be used.	H
STP_CRD2	string	hh:mm:ss.sss or ddd.ddd	All	Intg	Y component of the final coordinates in COORD_MD frame. Required the end of each sub-integration (not the end of each observation). See STP_CRD1 for more comments.	H
De-dispersion measure (DM)	double	pc cm <sup>-3</sup> [0:10,000]	PT, DS	SCfg	De-dispersion measure for coherent/incoherent de-dispersion.	H
Rotation measure (RM)	double	rads m <sup>-2</sup> [-?:+?]		SCfg	Rotation measure for coherent de-faraday rotation. To de-faraday rotate low frequency signals while maintaining high time resolution may imply that we need to perform coherent rotation measure correction. This is not a requirement at the moment but may be implicitly derived from existing requirements - To Be Defined (TBD).	D
Maximum length of observation (SCANLEN_MAX)	float	Seconds [30:43200]	All	SCfg	Maximum length of observation.	H
Pulsar ephemeris (EPHEMERIS)	ASCII text		PT	SCfg	Pulsar ephemeris for pulsar being observed. The ephemeris files should be of the order a few kilobytes.	H

CSP_Low Parameter (Keyword)	Type	Units, valid values or range (all TBD)	Mode	When	Description	P
Pulsar phase predictor (PREDICTOR)	ASCII text		PT	SCfg	Pulsar phase predictor generated from ephemeris (T2 format). The predictor received from the SDP must have enough coefficients to represent the phase of the pulsar to better than 1 ns RMS over the full bandwidth of the scan and up to SCANLEN_MAX. More information related to predictors is provided below this table.	H
Output frequency channels (OUTNCHAN)	int	[1:2250]	PT, DS	SCfg	The number of output frequency channels.	H
Output phase bins (OUTNBIN)	int	[64:2048]	PT	SCfg	The number of phase bins in output.	H
The integration time for each output bin (OUTTBIN)	double	Seconds [0.00032:60]	DS	SCfg	The integration time for each output bin.	H
The duration of each output sub-integration (OUTSUBMIN)	double	Seconds [1:60]	All	SCfg	The size of each output sub-integration.	H
Stokes parameters (OUTSTOKES)	string	combination of I,Q,U,V	DS	SCfg	The Stokes parameters to output when in Dynamic Spectrum mode.	H
The number of bits per output sample (OUTNBIT)	int	[1,2,4,8,16,32]	DS, FT	SCfg	The number of bits per output sample.	H

CSP_Low Parameter (Keyword)	Type	Units, valid values or range (all TBD)	Mode	When	Description	P
The number of samples integrated in SK statistics (SK_INT)	int	[?:?]	PT, DS	SCfg	The number of samples integrated in SK statistics.	H
SK Excision limits (SK_EXCIS)	double	Standard deviation [?:?]	PT, DS	SCfg	SK Excision limits (RFI threshold).	H
The number of channels in FT output (OUTNCHAN_FT)	int	<=OBSNCHAN	FT	SCfg	Number of channels to propagate to SDP in Flow Through mode.	H
Channels to be passed to SDP (OUTCHAN_FT)	Int	[OUTNCHAN_FT]	FT	SCfg	List of channel numbers to be passed to SDP in Flow Through mode (of length OUTNCHAN_FT).	H
Number of stations (NANTS)	int	[1..512]	All	SCfg, Intg	If not provided, CSP_Low uses all stations that belong to the sub-array.  Required at the end of each N-second sub-integration (average weights for previous N seconds)	M
List of stations (ANTENNAS)	int	Station IDs	All	SCfg, Intg	If not provided, CSP_Low uses all stations that belong to the sub-array.  Required at the end of each N-second sub-integration (average weights for previous N seconds)	M
Station weights (ANT_WEIGHTS)	double	TBD	All	SCfg, Intg	The average weight for each station over the course of a sub-integration. These are the same weights that are delivered to the CBF.  Required at the end of each N-second sub-integration (average weights for previous N seconds).	M

CSP_Low Parameter (Keyword)	Type	Units, valid values or range (all TBD)	Mode	When	Description	P
RFI frequencies (FREQ_MASK)	ASCII text	TBD	All	SCfg	<p>A text file containing a list of known RFI frequencies to excise from the data - the PST has the capability to identify very narrow band signals.</p> <p>This would be an ASCII text file containing a list of frequencies and bandwidths for known RFI not excised by the CBF.</p> <p>Might change during course of an observation, but does not change over the course of a sub-integration.</p> <p>Suggestion would be that RFI channel lists would be updated on intervals of 10 seconds or more (most likely more). There is no specification on this at the moment, but it seems like a reasonable precaution to ensure data quality.</p> <p>RFI update should be delivered by TM every 10 or more seconds. The exact format is TBD, probably a text file, 2-column, tab-delimited.</p> <p>Size of the RFI files should be of the order 10s of kilobytes.</p>	M
Cal mode (CAL_MODE)	string	OFF, SYNC, EXT1, EXT2	TBD	SCfg	Operation mode for the injected calibration: for 'OFF' there is no injected calibration; for 'SYNC' the calibration is pulsed synchronously with the folding frequency; 'EXT1' and 'EXT2' indicate that the calibration is driven by one of two possible user-defined external signals.	C
Calibration modulation frequency (E) (CAL_FREQ)	double	HZ	TBD	SCfg	Modulation frequency for the injected calibration signal.	C
Cal duty cycle (E) (CAL_DCYC)	double		TBD	SCfg	Duty cycle for the injected calibration signal.	C
Cal phase (E) (CAL_PHS)	double		TBD	SCfg	<p>Cal phase (wrt start time).</p> <p>Phase of the leading edge of the injected calibration signal in SYNC mode.</p>	C

CSP_Low Parameter (Keyword)	Type	Units, valid values or range (all TBD)	Mode	When	Description	P
Number of states in cal pulse (I) (CAL_NPHS)	double		TBD	SCfg	Number of pulses across one period of calibration phase.	C
Destination address(es)	MAC, IP, port	TBD	All	SCfg	Destination addresses for the CSP_Low output data. To be provided by SDP, via TM, before the start of the scan. TBD: Destination addresses may be provided in the same message with other configuration parameters or (more likely) after the SDP configuration and preparation for the scan has been completed (but shall be received before the scan Activation Time).	H

### 7.8.10 CSP\_Low Data to be stored by Telescope Manager

This section lists CSP generated data that should be stored by Telescope Manager (in the Telescope Model or other repository).

List may include:

1. Configuration data:
  - Version of hardware, firmware and software for all Components
  - Serial numbers (where available)
2. Setup (configuration parameters)
3. Status reports
4. Alarms
5. Monitor data

Data generated by PSS:

1. Spectral statistics.
2. Data quality and other meta data.
3. Spectral statistics.

### 7.8.11 Required TM Service

TM provides updates for Pulsar Search based on the following:

Document No.: 100-000000-021  
Revision: 01  
Date: 2016-01-19

FOR PROJECT USE ONLY  
Author: Sonja Vrcic  
Page 130 of 148

1. BRDZ list
2. Known pulsars
3. Long term RFI
4. Short term RFI
5. Satellite positions

RFI information is provided to CSP in the form of RFI masks.

BRDZ list can be forwarded to CSP upon initialization. Updates may be forwarded to CSP as required.

### 7.8.12 Alarms

List of CSP\_Low alarms will be provided in this section

**Table 7-36 CSP\_Low – preliminary list of alarms generated by the sub-element CSP\_Low.PSS**

Alarm ID	Fault Type	Maintenance task
1	Failure of compute accelerators (FPGA, GPU) in CSP PSS Compute node	Soft/hard reboot of the concerned compute node with an interruption to the observation; replace faulty accelerator cards
2	Failure of power supply in CSP PSS Compute node	Replace the faulty power supply of the concerned node
3	Failure of CSP PSS Compute host	Soft/hard reboot of the concerned compute node with an interruption to the observation; replace faulty host
4	Failure of CSP PSS Compute node storage	Replace faulty storage disk.
5	CPU and system temperature	Soft/hard reboot of the concerned compute node with an interruption to the observation; replace faulty host
6	CPU and system status	Soft/hard reboot of the concerned compute node with an interruption to the observation; replace faulty host
7	CPU and system voltages	Soft/hard reboot of the concerned compute node with an interruption to the observation; replace faulty host
8	Fan speeds	Soft/hard reboot of the concerned compute node with an interruption to the observation; replace faulty host
9	Power supply status	Soft/hard reboot of the concerned compute node with an interruption to the observation; replace faulty host

Alarm ID	Fault Type	Maintenance task
10	Bad formatted data: network stream does not comply with the specification	Stop observation.
11	A node is not meeting the real-time requirement	Soft/hard reboot of the concerned compute node with an interruption to the observation; reprogram faulty host
12	A node report a fault	Soft/hard reboot of the concerned compute node with an interruption to the observation; replace faulty host
13	Data don't meet the specified standards for RFI interference	Stop observation.
14	Received data without a previous start signal.	No action. Data are discarded until buffer becomes empty.
15	Runtime failure of any functional blocks due to unmolded software errors	Soft/hard reboot of the concerned compute node with an interruption to the observation; replace faulty host.

### 7.8.13 Events

List of Events generated by CSP\_Low:

1. Recommendation to stop on-going pulsar timing for a particular pulsar (see Section 7.8.9.3 for more information)
2. Pulsar search completed (for a particular sub-array). Rational: As described in Section 7.8.9.2, CSP\_Low.PSS requires an entire observation to be completed before it can begin the required acceleration processing; there is a lag between the end of an observation and the end of processing. This places constraints on scheduling of observations, as the next Pulsar Search observation should not start before the processing for the previous observation has been completed. CSP\_Low generates an Event message to inform TM when the processing has been completed.

### 7.8.14 Monitor Points

This section will specify the list of Monitor Points for each CSP sub-element.

#### 7.8.14.1 CSP\_Low.CBF (Correlator and Beamformer) Monitor Points

Estimate for the number of monitor points not available.

#### 7.8.14.2 CSP\_Low.PSS (Pulsar Search Engine) Monitor Points

Estimate for the number of monitor points:

For each of 250 servers: ~10 monitor points (status, >3xtemperatures, running time, data chunk processed).

For each of 25 racks : ~ 10 monitor points (status, number of servers running, power, >2 temperatures)  
For each of 500 beams: ~16 indicators.  
Total: ~ 10,750 monitor points.  
Auxiliary data (observation monitoring):  $20 * 500 = 10,000$ .

#### 7.8.14.3 *CSP\_Low.PST (Pulsar Timing Engine) Monitor Points*

Estimate for the number of monitor points: ~ 300 Monitor Points.

#### 7.8.14.4 *CSP\_Low.LMC Monitor Points*

Estimate for the number of monitor points: ~ 20 Monitor Points (two servers, 10 monitor points per server).

### **7.9 HMI Engineering interface specifications**

TM shall define a coherent purpose, method, and look-and-feel for human-machine interface which will then be designed and implemented in the CSP\_Low (TBD). This section will specify engineering interfaces to be provided by CSP\_Low.

CSP engineering interfaces will be integrated in the SKA engineering interface.

CSP\_Low engineering interfaces will provide graphical representation of CSP\_Low, its implementation, status and configuration. HMI will be organized hierarchically to allow for drill-down access.

The top level CSP HMI will provide overall status and overview of the CSP\_Low equipment, e.g. racks/ cabinets with temperature indicators, fan speed and such. Top level CSP\_Low interface will be designed to provide status of the CSP\_Low equipment in a glance. The same or complimentary screen will display CSP\_Low use indicators, i.e. which equipment is used and by which sub-array, and similar. Similar displays will be provided for all CSP\_Low sub-elements. A click on a Component should open a display with detailed information for a particular Component (server, compute node or LRU). For custom made hardware and software such displays will be provided down to the level of FPGA/ASIC and software process.

## 8 Verification Methods

### 8.1 Interface not made

In the absence of the real interface with TM, CSP will implement a software tool that can transmit commands, requests and queries. The simplest version of such a tool will be able to load a message from a file, transmit the message and display received response(s). Such a tool could be used by the CSP design and development team to verify that CSP.LMC hardware and software is able to receive and interpret TM commands, requests and queries, and generate responses.

In order to verify that CSP\_Low is able to correctly interpret not only the syntax but also the meaning (semantics) of the TM message a more elaborate test tool would be required and, at least partly implemented CSP.LMC and other CSP sub-elements.

A more elaborate version of the CSP\_Low test tool should be able to:

- compare received CSP response to the expected response.
- compare resulting CSP equipment setup with the expected setup (saved during earlier testing session).
- compare output products with the expected set of output products (saved during earlier test session).
- query and save CSP sub-element setup and save it for later use.
- save products generated during the test so that they can be used later for analysis and/or regression testing.

The test tool may be further upgraded to execute a set of test cases, automatically compare results (CSP responses, setup of CSP equipment and output products) and generate report. Such a tool would be very useful for regression testing and would allow the CSP Consortium to perform testing in the absence of the TM.

A similar tool, able to emulate CSP behaviour, i.e. receive TM commands, requests and queries, and generate (CSP) responses, alarms, events and monitor point reports will be developed and used for testing of TM.

#### 8.1.1 Data exchange interface verification

This section will list verification methods. Required functionality and degree of automation will be determined as design progresses (TBD).

#### 8.1.2 HMI Engineering interface verification

This section describes verification techniques for the engineering interfaces for CSP sub-elements. Required functionality will be determined for each CSP sub-element individually as design progresses.

In this context (CSP to TM interface, engineering interfaces) 'interface not made' means that a particular engineering interface has been implemented but has not been integrated into TM framework.

When it receives command or query from an engineering interface, CSP will not be aware whether a particular interface is integrated into the TM framework or launched as a standalone tool. (TM framework has not yet been defined; this statement may be altered later.)

In order to verify an engineering interface one has to prove that:

- reported/displayed status reflects the actual status of hardware and software Components,
- changes of the status and setup of the h/w and s/w occur as expected,

- logs/events/alarms are generated as expected.

## 8.2 Interface made

Interface made means that communication between TM and CSP has been established. One of the generic requirements for the SKA Element Local Monitor and Control is to allow TM to query the list of implemented commands and parameters. TM may use this feature to generate automatic test cases. More elaborate test suits must be created for verification of CSP functionality similar to the one described in section 8.1 above.

### 8.2.1 Data exchange interface verification

This section describes verification techniques for data exchange interface in all SKA1 telescopes. Required functionality and degree of automation will be determined as design progresses.

### 8.2.2 HMI Engineering interface verification

Engineering interfaces are human-machine interfaces developed for testing of the CSP sub-elements. Eventually, these interfaces will be integrated into SKA engineering interface. This section discusses verification for the CSP engineering interfaces when interface is made.

In this context (CSP to TM interface, engineering interfaces) 'interface made' means that a particular engineering interface has been *integrated* into the TM framework and that CSP.LMC and other CSP sub-elements have been integrated into the system.

During CSP to TM interface verification we shall verify that:

- 1) CSP engineering interfaces can be accessed via TM engineering interface framework.
- 2) CSP engineering interfaces, when accessed via TM engineering interface framework, can be used to display and alter status (parameters) of all installed CSP Components; i.e. that access rights, addresses, etc. are propagated as expected.
- 3) CSP engineering interfaces are designed according to guidelines defined by TM.

## Appendix A. Intellectual Property Declaration

There is no IP to declare in this document.

## Appendix B. To Be Defined

Summary of the content marked as 'to be defined' in this version of the document.

Most items that are still to be defined depend on the further work on the document [\[RD4\]](#) and CSP design.

**Table 8-1 Summary of content to be defined**

To be defined	Depends on
Protocols and standards to be used to implement this interface.	<a href="#">[RD4]</a>
Naming conventions for Components, Capabilities, Alarms, Events, Monitor Points, Messages, Parameters, etc.	<a href="#">[RD4]</a>
Modes and States.	<a href="#">[RD4]</a>
Format and content of messages.	<a href="#">[RD4]</a>
Component and Capability Health Reporting guidelines.	<a href="#">[RD4]</a>
Guidelines for implementation of logging.	<a href="#">[RD4]</a>
Guidelines for implementation of alarms and events. Alarm severity levels.	<a href="#">[RD4]</a>
Messages generated by CSP to acknowledge/accept/report receipt and activation of SET Parameters message (and other commands).	TM design, <a href="#">[RD4]</a>
CSP generated auxiliary data for each telescope.	CSP design
Further work on refinement of Observing Modes and their parameters is required.	CSP design, System-level requirements <a href="#">[AD2]</a>
Requirements for precision when specifying time for activation of commands, in particular scan start, scan end (or duration), epoch for parameters related to delay tracking, calibration and beamforming, epoch for pulsars, etc.	SKA Concept of Operations and CSP sub-elements
Depending on the choice of the protocol, different data streams may be used for exchange of messages (Get/Set/Response/Alarms/Event) and for file transmission.	<a href="#">[RD4]</a>
Framework and guidelines for engineering interfaces (Chapter 3).	TM Design
In the case of loss of communication with TM for longer than TBD seconds, CSP stops on-going signal processing and transmission of output data. After additional TBD time period, CSP sets the CSP equipment in the low-power mode (Chapter 4 and 7.2.1). Is this a requirement? Is this desired behaviour?	Concept of Operations
It is to be defined (TBD) by the TM Consortium whether CSP should generate Response or Alarm message (or both) to report that SET Parameter message has been rejected. (7.6.1.)	TM design, <a href="#">[RD4]</a>

To be defined	Depends on
SKA system requirements allocate 30 seconds for the telescope Observing Mode change. Allocation (budget) for the CSP Observing Mode changes is still to be defined. (7.6.1.1)	SKA Design
Pulsar Timing (7.8.8.7): CSP Consortium anticipates that the human telescope operators will need to see various forms of diagnostic information during the observation. Content and format for plots will be specified here in this ICD. TM Consortium will select the format for diagnostic plots.	TM design
Pulsar Timing (7.8.8.7): CSP could send to TM images (e.g. PNG or JIF) or, if the TM expects to be able to query the data in multiple ways, CSP might send TM a copy (possibly reduced in dimension) of the information that is sent to the SDP.	TM design
Verification methods (see Chapter 8)	Interface definition and design.
TM shall define a coherent purpose, method, and look-and-feel for human-machine interfaces (engineering interfaces).	TM Design
Requirements for zoom mode. -Does CSP_Low allow for total of up to 4 zoom windows per sub-array, or up to 4 zoom windows per beam per sub-array? -Are frequency resolution, integration time, number of phase bins and zoom window configuration same for all the beams in the sub-array? Or, are some, or all, of these parameters specified for each beam independently?	SKA1 LOW Level1 Requirements <a href="#">[AD2]</a>
Input from each station may consist of up to 16 beams. It is to be defined whether station-beam setup is configurable and should it be specified as a part of a scan configuration.	SKA1 LOW design
-How much in advance a delay model has to be delivered to CSP_Low in order to be applied at the time specified in the message. -Required cadence and number of coefficients for the delay models provided by TM. -Required cadence and content for all parameters that require regular updates.	SKA1 LOW design , CSP_Low design
Pulsar Search output data, for every sub-array in Pulsar Search Mode, CSP_Low also generates visibilities. It is still to be defined (TBD) whether CSP_Low will define a default set of Imaging parameters to be used for a sub-array in Pulsar Search mode (and or Pulsar Timing Mode) when Imaging is not explicitly required.	SKA1 LOW design , CSP_Low design
Estimate for the Number of Monitor Points for CSP_Low.CBF.	SKA1_CSP.CBF Design
What exactly 'loss of connectivity' means on this interface, and how it is detected, will be defined when the protocol stack is selected (TBD).	<a href="#">[RD4]</a> , CSP and TM design

To be defined	Depends on
If due to loss of connectivity, CSP does not receive delay model polynomials and other real-time updates, CSP either flags output data or stops generation of output products (TBD for each Observing Mode individually). Need to define required behaviour.	CSP Design, SKA Concept of Operations
ned (TBD) whether CSP, in the case of prolonged loss of communication with TM, should stop on-going signal processing and transmission of output data and set the CSP equipment in the low-power mode.	SKA Concept of operations
Is it a requirement that CSP should implement maintenance mode. If so, need to define requirements for maintenance mode (for CSP behaviour/functionality).	SKA System Level requirements and/or Concept of Operations

## Appendix C. LMC Scope and Responsibilities Requirements Verification Matrix

This chapter is not part of the interface specification, it is provided for reference only. This chapter provides the requirement verification matrix for the requirements defined in the document “SKA1 LMC Scope and Responsibilities” [RD3].

Table 8-2 LMC Scope and Responsibilities [RD3] Requirements Verification Matrix

Requirement	Section	Comment
<p><b>LMC-REQ-0010 Maintain internal connectivity Element LMC shall at all-time maintains connectivity with all internal components.</b>  <i>[Note: Maintaining connectivity implies that keep-alive packets will be sent and maintained, even if the other side of the connection fails. It also implies that either side of the connection cannot have long blocking type calls that make it non-responsive for longer periods of time. Both TM and element LMC should provide provision to stop transmission of keep alive messages for elements and components declared as not-fitted (or not-present).]</i></p>	n/a	Requirement for CSP implementation, does not apply for this interface.
<p><b>LMC-REQ-0011 Maintain connectivity with TM.</b> Element LMC shall at all-time maintain connectivity with TM.  <i>[Note: Maintaining connectivity implies that “keep-alive” packets will be sent and maintained, even if the other side of the connection fails. It also implies that either side of the connection cannot have “long blocking type calls” that makes it non-responsive for longer periods of time. Both TM and Element LMC should provide provision to stop transmission of ‘keep alive’ messages for Elements and components declared as “not-fitted” (or “not-present”).]</i></p>	Chapter 3 bullet 4)	
<p><b>LMC-REQ-0020 Adhere to LMC Interface Guidelines.</b> Element LMC shall provide an interface for TM to configure, control and monitor the Element and its components adhering to the “LMC Interface Guidelines”.</p>	Chapter 5 Chapter 7	
<p><b>LMC-REQ-0030 Component/Part Naming Convention.</b> Element LMC shall identify all hardware and software components/parts in accordance with the naming conventions in the “LMC Interface Guidelines”. It is envisioned that component/part name will reflect the role and location in the Element (Element/sub-Element/component/LRU).</p>	Section 7.4	

Requirement	Section	Comment
<p><b>LMC-REQ-0040 Identify Capabilities</b> Element LMC shall provide for functional grouping of Element functionality and/or products as <b>capabilities</b>. For each Element capabilities will be defined by the Element during the design phase in collaboration with TM and captured in the ICD.</p>	Table 7-1	
<p><b>LMC-REQ-0060 Capability Naming Convention</b> Element LMC shall identify capabilities in accordance with the naming conventions in the “LMC T0000-0000-LSR-001-B</p>	TBD	Naming conventions not defined in <a href="#">[RD4]</a>
<p><b>LMC-REQ-0070 Self-describing interface</b> Element LMC shall implement self-describing interface that, upon request, provides the list of monitoring points, commands, components, capabilities, configurable parameters, alarms, events, etc. <i>[Note: The details shall be fully defined in the “LMC Interface Guidelines”. The List of commands, shall for example provide parameters and a description of the command, while monitoring points may include ranges for nominal, warn and error conditions, unit of measure and a description.]</i></p>	Section 7.1 Section 7.6.2	
<p><b>LMC-REQ-0080 Report versions</b> Element LMC shall, upon request, report software, firmware and hardware version for all components and LRUs.</p>	Section 7.8.3 Section 7.8.4 Section 7.8.5	
<p><b>LMC-REQ-1010 Interface to configure Element</b> Element LMC shall implement interface that allows users to set and get configurable parameters of sub-Elements, components and capabilities.</p>	Section 7.8.8.1	
<p><b>LMC-REQ-1030 Identify “not-fitted” components/capabilities</b> Element LMC shall make provision for TM to indicate components/capabilities as “not-fitted”. <i>[Note: All states, including “not-fitted”, will be fully defined in the “LMC Interface Guidelines”.]</i></p>	Section 7.5	
<p><b>LMC-REQ-1040 Persist “not-fitted” status</b> If the connectivity with TM is lost, Element LMC shall keep previously set “not-fitted” status as is (when connection is established TM should re-transmit administrative status for all components).</p>	Section 7.5	
<p><b>LMC-REQ-1050 Identify “in-maintenance” components/capabilities</b> Element LMC shall make provision for TM to indicate components/capabilities to be “in-maintenance”. <i>[Note: all states, including “in-maintenance” will be fully described in the “LMC Interface Guidelines”.]</i></p>	Section 7.5	

Requirement	Section	Comment
<p><b>LMC-REQ-1060 Persist “in-maintenance” status</b> If the connectivity with TM is lost, Element LMC shall keep previously set “in-maintenance” status as is (when connection with TM is established, TM will re-transmit administrative status for all components).</p>	Section 7.5	
<p><b>LMC-REQ-1070 Use spare/backup components</b> Element LMC shall automatically use spare/backup/redundant components to replace faulty ones where possible</p>	n/a	Supported by CSP design.
<p><b>LMC-REQ-1080 Implement capabilities</b> Element LMC shall implement capabilities that expose higher level abstractions to TM (e.g. subarrays, products, frequency ranges).</p>	Section 7.1	
<p><b>LMC-REQ-1090 Configure and control capabilities</b> Element LMC shall be able to receive configuration and control from the TM at the capability level. Element LMC shall select components to be used to provide capability and configure selected components to achieve capability.</p>	Section 7.1	
<p><b>LMC-REQ-1100 Set up capabilities</b> Element LMC shall set up the components selected to form a capability as per received instructions, so that TM does not have to address individual components within the Element.</p>	Section 7.1	
<p><b>LMC-REQ-1110 Exclude faulty components</b> Element LMC shall automatically exclude faulty components when configuring capabilities (where applicable).</p>		LMC Design.
<p><b>LMC-REQ-1120 Exclude “not-fitted” components</b> Element LMC shall automatically exclude components that are “not-fitted” when configuring capabilities.</p>		LMC Design.
<p><b>LMC-REQ-1130 Exclude “in-maintenance” components</b> Element LMC shall automatically exclude components that are “in-maintenance” from regular observation capabilities.</p>		LMC Design.
<p><b>LMC-REQ-1140 Report achievable capabilities</b> Element LMC shall report achievable capabilities to TM to ensure effective scheduling of observations (initially at least provide an interrogation of whether a specific capability is achievable).</p>	Section 7.1	

Requirement	Section	Comment
<p><b>LMC-REQ-1150 Persistence of status over power-down</b> TBD: Careful consideration should be given to persistence of not-fitted and in-maintenance status over Element power-down. Suggestion is for TM to inform Element LMC of “not-fitted” and “in-maintenance” components from a central configuration when connection is established, and Element LMC to inform TM of faulty LRUs when connection is established.</p>	Section 7.5	
<p><b>LMC-REQ-2010 Coordinated control</b> Element LMC shall implement coordinated control of its components/capabilities - at a level to be negotiated with TM and each Element LMC.</p>		
<p><b>LMC-REQ-2020 Provide observation and maintenance level control</b> Element LMC shall provide abstraction levels for observation control and maintenance control to TM in a consistent manner.</p>	Section 7.1	
<p><b>LMC-REQ-2030 Standardized global control model</b> Element LMC shall implement a standardized “global control model” as negotiated with TM and Element LMC leads as far as possible. <i>[Note: This global control model, including all global states, will be fully described in the “LMC Interface Guidelines”.]</i></p>	Chapter 7	SKA Control Model will be defined in <a href="#">[RD4]</a>
<p><b>LMC-REQ-2040 Interrogate control commands</b> Element LMC shall make provision for interrogating the list of Control commands and Maintenance commands supported by the Element, with parameters and return status of each command.</p>	Section 7.6.2 Section 7.8.3 Section 7.8.4 Section 7.8.5	
<p><b>LMC-REQ-2050 No tight feedback loops</b> Element LMC shall avoid introduction of tight feedback loops (&lt;500ms) from TM as much as possible.</p>	-	No tight feedback loops have been identified so far.
<p><b>LMC-REQ-2010 Monitor Element components</b> Element LMC shall monitor status, performance, behavioural parameters and configuration of all sub-Elements, components and LRU.</p>	n/a	Supported by CSP design.
<p><b>LMC-REQ-2020 Report monitoring points</b> Element LMC shall monitor status of monitor points and report status to TM in a standardized way.</p>	Section 7.6.3	
<p><b>LMC-REQ-2030 Allow configuration of monitoring point reporting</b> Element LMC shall allow configuration of monitoring and reporting model for individual monitor points, individual components, and based on the component and/or monitor point type.</p>	Section 7.6.3	

Requirement	Section	Comment
<b>LMC-REQ-2040 Required monitoring point reporting models</b> Element LMC shall implement at least the following reporting models for monitor points (not all models are applicable for all monitor points):	Section 7.6.3	
<b>LMC-REQ-2050 Default monitoring point reporting</b> Element LMC shall define the default configuration for each monitor point, so that reporting of T0000-0000-LSR-001-B monitor point status can begin as soon as Element/component becomes active. Default reporting model for each monitor point (or class of monitor points) will be documented in the ICD.	Section 7.6.3	
<b>LMC-REQ-2060 Interrogate list of monitoring points</b> Element LMC shall provide for interrogation of the complete list of monitoring points (for that Element) including current status and setup.	Section 7.6.2 Section 7.8.3 Section 7.8.4 Section 7.8.5	
<b>LMC-REQ-2070 Provide hierarchical reporting</b> Element LMC shall provide rolled-up and drill-down views of the health status and performance parameters, to enable hierarchical reporting and diagnostics. Exact requirements TBD.	Section 7.6.2 Section 7.8.3 Section 7.8.4 Section 7.8.5	
<b>LMC-REQ-2071 Provide rolled-up Element health</b> Element LMC shall provide a rolled-up Element / sub-Element health to TM in a standardized way (using standard naming conventions and health status levels).	Section 7.8.3	
<b>LMC-REQ-2080 Provide rolled-up LRU/component health.</b> Element LMC shall be able to provide a rolled-up component / LRU health to TM in a standardized way (using standard naming conventions and health status levels).	Section 7.6.2 Section 7.8.3 Section 7.8.4 Section 7.8.5	
<b>LMC-REQ-2090 Provide rolled-up capability health.</b> The Element LMC shall report rolled-up capability health for each configured capability to TM in a standardized way (using standard naming conventions and health status levels).	Section 7.8.7	
<b>LMC-REQ-2110 Diagnose faults and failures.</b> Element LMC shall implement procedures to diagnose faults and failures.	n/a	Supported by CSP design.
<b>LMC-REQ-2115 Take corrective action on faults and failures.</b> Element LMC shall take corrective action on diagnosed faults and failures, when possible.	n/a	Supported by CSP design.

Requirement	Section	Comment
<p><b>LMC-REQ-0130 Produce failure detection reports.</b> Element LMCs shall produce failure detection reports indicating faulty LRUs and FMECA failure detections to the TM to trigger maintenance and corrective action.</p>	Section 7.6.5	
<p><b>LMC-REQ-2120 Report events and failure detections.</b> Element LMC shall report significant events and failure detections to TM in a standardized way. List of significant events and failure detections shall be determined by the Element in consultation with TM, and document in the ICD.</p>	Section 7.6.5 Section 7.6.6	
<p><b>LMC-REQ-2130 Provide advisory inputs on failure detections.</b> When reporting failure detections, Element LMC shall provide advisory inputs to TM including recommended operator actions, potential threats, impact on the quality of observed data, potential long term effects and so on. To reduce amount of information transmitted, most of such information shall be made available to TM in advance (during design and commissioning) and referred to when reporting failure detections.</p>	Section 7.6.5	
<p><b>LMC-REQ-2140 Exclude not-fitted items from failure reporting.</b> Element LMC shall intelligently exclude items not-fitted from failure reporting.</p>	Section 7.6.5	
<p><b>LMC-REQ-2150 Provide for interrogation of alarms/events</b> Element LMC shall provide for interrogation of alarms and events on the LMC interface (self-describing interface).</p>	Section 7.6.2 Section 7.8.3 Section 7.8.4 Section 7.8.5	
<p><b>LMC-REQ-2160 Report active alarms/events.</b> On TM request Element LMC shall report all active events/alarms and known faults (even if they were previously reported, loss of connectivity may cause TM to lose all information regarding Element status).</p>	Section 7.6.2 Section 7.8.3 Section 7.8.4 Section 7.8.5	
<p><b>LMC-REQ-3010 Report Safety Critical Events/Alarms.</b> Element LMC shall report Safety and Critical Events/Alarms to TM in a consistent way and shall allow configuration of such events/alarms (including naming conventions and defined event/alarm levels, and other configuration which may include the monitoring point(s) to evaluate, for which ranges, levels/severity/criticality of events/alarms, frequency of sampling and reporting)</p>	Section 7.6.2 Section 7.8.3 Section 7.8.4 Section 7.8.5	

Requirement	Section	Comment
<b>LMC-REQ-3010 Provide advisory inputs.</b> Element LMC shall provide advisory inputs to TM on higher-level detection and response actions, including recommended operator actions.	Section 7.6.2	
<b>LMC-REQ-3030 Interrogate list of active events/alarms.</b> Element LMC shall make provision for interrogation of the list of all active events/alarms via the LMC interface.	Section 7.6.2 Section 7.8.3 Section 7.8.4 Section 7.8.5	
<b>LMC-REQ-3040 Suppress events/alarms on not-fitted and in-maintenance components</b> Element LMC shall intelligently suppress events/alarms on components that are “not-fitted” or “in-maintenance” (‘intelligently’ means that during design and construction each event/alarm will be carefully considered, for example if component “in-maintenance” is getting overheated, the event/alarm shall not be suppressed.)	Section 7.6.2	
<b>LMC-REQ-3050 Override event/alarm.</b> Element LMC shall make provision for an override to suppress a specific event/alarm.	Section 7.6.2	
<b>LMC-REQ-3060 Define additional events/alarms.</b> Element LMC shall make provision for defining additional events/alarms over the LMC interface.	TBD	Requires more consideration.
<b>LMC-REQ-3070 Report progress events.</b> Element LMC shall report Events to TM to indicate processing that has completed and TM will react to those as appropriate.	Section 7.6.6	Events to be defined.
<b>LMC-REQ-4010 Identify and report faulty LRUs.</b> The Element LMC shall identify faulty LRUs and report those to TM.	Section 7.6.2	
<b>LMC-REQ-4020 Identify maintenance activities.</b> The Element LMC shall identify LRUs that require maintenance and report those to TM.	n/a	To be considered in CSP design.
<b>LMC-REQ-4030 Report failure detections.</b> Element LMC shall report Failure Detections at the sub-element/component/LRU level as per the Element FMECA (Failure Modes Effects and Criticality Analysis) to TM in a standardized way.	Section 7.6.2	
<b>LMC-REQ-4050 Copy logging files.</b> Element LMC shall allow TM to access and copy local (to the Element) logging files (where applicable).	Section 7.6.7	

Requirement	Section	Comment
<b>LMC-REQ-4060 Standard logging.</b> Element LMC (and all its components) shall use standard format, content and logging level as will be defined in the “LMC Interface Guidelines”.	Section 7.6.7	
<b>LMC-REQ-4070 Remote logging.</b> Element LMC shall report log messages to Telescope Manager (i.e. shall provide support for remote logging).	Not addressed	Requirement for remote logging to be justified (TBJ).
<b>LMC-REQ-4080 Control logging.</b> Element LMC shall make provision for Telescope Manager to control logging, including: a) destination for logging messages, b) logging level.	Section 7.6.7	Control for logging level supported. Remote logging TBJ
<b>LMC-REQ-5010 Locally safe equipment.</b> The Element LMC shall be responsible for putting the Element’s equipment into a safe state or powering down the Element’s equipment autonomously when its own sensors detect that a condition exists that is hazardous to the Element’s equipment (e.g. ambient temperature is too high).	Chapter 4	
<b>LMC-REQ-5020 Remote power-down for safety</b> The Element LMC is responsible for powering down the Element’s equipment when it receives a power-down command from the TM.	Section 7.6.11	
<b>LMC-REQ-5030 Local personnel safety</b> The Element LMC shall not rely on the TM for equipment or personnel safety.	Chapter 4	
<b>LMC-REQ-6010 Remote Update and Upgrade</b> At least DSH LMC and LFAA LMC shall make provision for remote update/upgrade of all software and firmware at the Element. Other Elements may support this functionality where appropriate.	Section 7.6.8	
<b>LMC-REQ-6020 Remote Restart.</b> When possible, Element LMC shall make provision for remote restart of sub-Element, component and/or LRU.	Section 7.6.9	
<b>LMC-REQ-6030 Define “safe state”.</b> Where applicable, the “safe state” for the Element will be defined in the Element LMC ICD and implemented by the Element LMC on command from the TM. This may include a stow position for dishes, a low power mode for receivers/digitizers, etc.	Section 7.6.10	

Requirement	Section	Comment
<p><b>LMC-REQ-6040 Remote power-down for engineering.</b> Element LMC shall implement a remote power-down for the Element when it receives a power-down command from TM. Power down of the INFRA element will be handled by itself.</p>	Section 7.6.11	
<p><b>LMC-REQ-6050 Remote power-up (TBJ)</b> Element LMC shall implement a remote power-up capability if applicable. <i>[This requirement may have implications for remote Elements and needs careful consideration. Must be clarified during ICD discussions with the different elements.]</i></p>	Section 7.6.12	