SDP Functional View

Contributors: L. Christelis, F. Graser

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

ADD	Attribute Driven Design
C&C	Component and Connector
FFBD	Functional Flow Block Diagram
PBS	Product Breakdown Structure

1. Primary Representation

SDP Functions

F.1 Receive data

F.2 Process visibility data fast

F.3 Process visibility data

F.4 Process pulsar candidates

F.5 Process pulsar timing

F.6 Aggregate quality metrics

F.7 Preserve data

F.8 Deliver data

F.9 Monitor and Control SDP

F.10 Support Observatory Commissioning & Operations

Note this functional decomposition is focussed on primary functionality only in order to complement the SDP Architecture Views. See 4. Rationale for more detail on Primary Functionality.

The functional decomposition and its relationship to the SDP Product Breakdown Structure (as shown in Table 1) is also used to allocate SDP L2 requirements to functions and products. This is important for requirements that are not architecturally significant since the architecture design and documentation process used for SDP, Attribute Driven Design (ADD) [RD01], is focussed on mainly addressing architecturally significant requirements. Therefore allocation (and traceability) of requirements to functions and/or products allows these requirements to be addressed in the detail design and implementation process (see 4.2. Requirements Model). The allocation of requirements to functions also supports the verification process of functional requirements (as described in [RD03]).

2. Element Catalogue

2.1. Element and Their Properties

Each element shown in the Primary Presentation is a primary function. This section shows the functional decomposition (hierarchy diagrams) of each of the SDP Level functions shown in section 1. Primary Representation. The SDP Architectural Workflow Views developed the implied functionality with its behaviour and data flow.

2.1.1. Receive

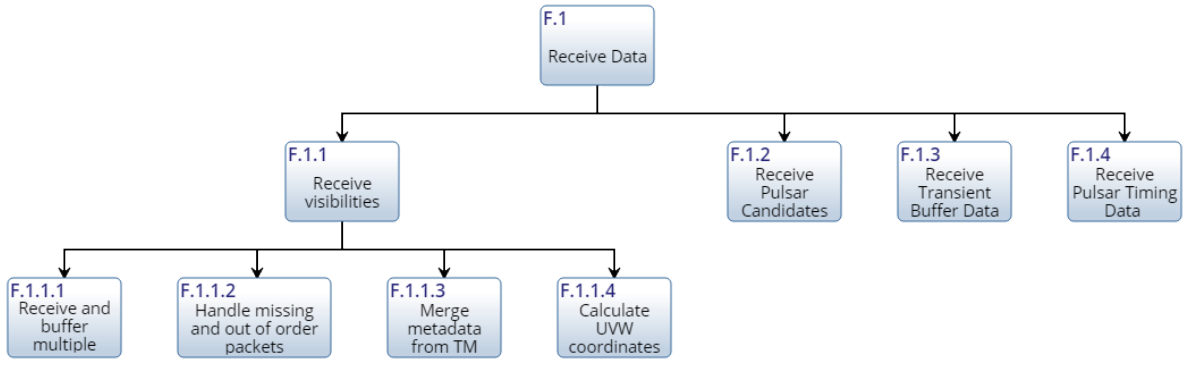


Figure 1: Functional decomposition RECEIVE

2.1.2. Process visibility data fast

Table 1: Functional Decomposition PROCESS VISIBILITY DATA FAST

F.2.1 Pre-process Fast	
	F.2.1.1 RFI flagging and excision (RT)
	F.2.1.2 Receive and apply initial Calibration Solutions
	F.2.1.3 Remove strong sources (RT)
F.2.2 Calibrate Real-time	
	F.2.2.1 Predict model visibilities (RT)
	F.2.2.2 Solve & update Calibration Solutions
	F.2.2.3 Apply Calibration Solutions
F.2.3 Pointing Calibration (MID)	
F.2.4 Image Fast	

	F.2.4.1 UV Phase Rotate
	F.2.4.2 Subtract LSM
	F.2.4.3 Grid for Fast Imaging
	F.2.4.4 FFT
F.2.5 Detect Imaging Transient Candidates	
	F.2.5.1 Source Finding
	F.2.5.3 Extract Flux
	F.2.5.4 Update Transient Source Catalogue
F.2.6 Monitor Ionosphere	

2.1.3. Process visibility data

Table 2: Functional decomposition of PROCESS VISIBILITY DATA

F.3.1 Pre-process data		
	F.3.1.1 RFI flagging and excision (Batch)	
	F.3.1.2 Receive and apply initial Calibration Solutions	
	F.3.1.3 Remove strong sources (Batch)	
F.3.2 Calibrate and Image		
	F.3.2.1 Instrument Calibration	
		F.3.2.1.1 Global antenna location and cable delay (MID)
		F.3.2.1.2 Low station beam calibration
		F.3.2.1.3 Measure antenna voltage patterns (holography or beam scans)
		F.3.2.1.4 Dish / Station Polarization Calibration
		F.3.2.1.5 Flux Scale Transfer
		F.3.2.1.6 Bandpass
		F.3.2.1.7 Polarization leakage
		F.3.2.1.8 Parallel and cross-hand delay measurement

	F.3.2.4. Calibrate auto-correlation spectra	
	F.3.2.3. Self-Calibration	
		F.3.2.3.1 Direct Fourier Transform Predict
		F.3.2.3.2 Apply Calibration Solutions
		F.3.2.3.3 Flagging (based on Calibration Solutions)
		F.3.2.3.4 Flagging (based on visibilities)
		F.3.2.3.5 Resample Visibility
		F.3.2.3.6 UV Phase Rotate
		F.3.2.3.7 UV Subtract
		F.3.2.3.8 Solve & update Calibration Solutions
		F.3.2.3.9 Resample Calibration Solution
		F.3.2.3.10 Create Kernel
		F.3.2.3.11 Grid / Degrid Visibilities
		F.3.2.3.12 Transform between uv-grid and image grid (and vice versa)
		F.3.2.3.13 Deconvolution
		F.3.2.3.14 Image arithmetic and reproject
		F.3.2.3.15 Insert source component in image
		F.3.2.3.16 Fit sky component
		F.3.2.3.17 Source find & estimate
		F.3.2.3.18 Update Sky Model
		F.3.2.3.19 Pointing Self-cal
		F.3.2.3.20 Peeling
F.3.2.4 Data Preparation		
	F.3.2.4.1 Apply Calibration Solutions (Data Prep)	
	F.3.2.4.2 Produce Continuum Image	
	F.3.2.4.3 Produce Spectral Line Image	

	F.3.2.4.3.1 Produce Spectral line emission image
	F.3.2.4.3.2 Produce Spectral line absorption Image
	F.3.2.4.4 Produce Mosaicing Image
	F.3.2.4.5 Produce Drift Scan Image
	F.3.2.4.6 Produce Gated Pulsar Images (TBC)

2.1.4. Process pulsar candidates

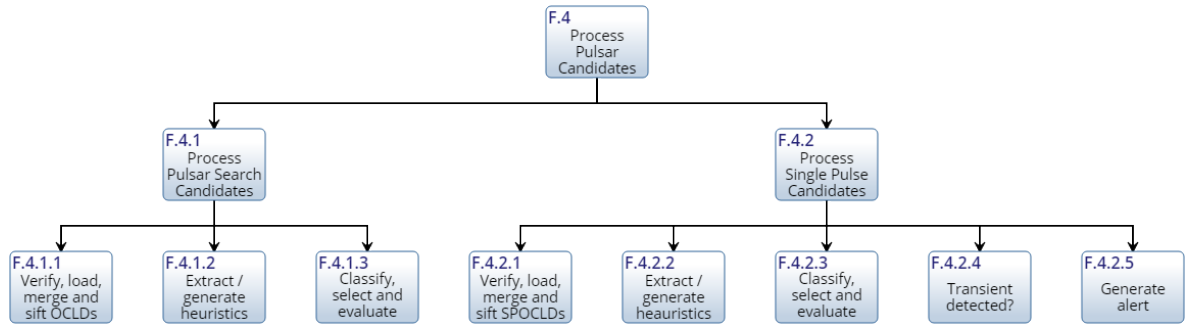


Figure 2: Functional Decomposition of PROCESS PULSAR CANDIDATES

2.1.5. Process pulsar timing

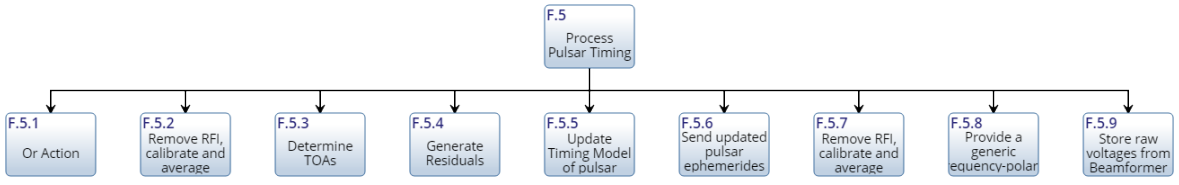


Figure 3: Functional decomposition of PROCESS PULSAR TIMING

2.1.6. Aggregate quality metrics

The metrics are obtained during processing. Metrics are aggregated for Telescope Manager.

2.1.7. Preserve data

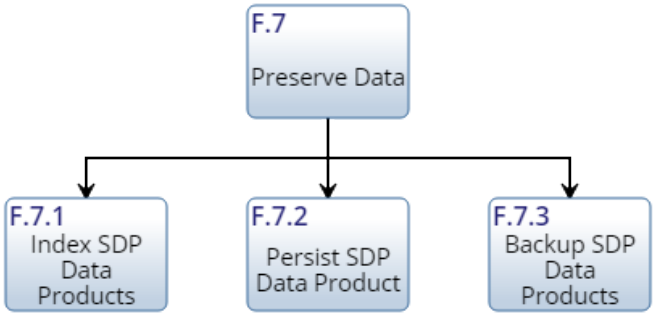


Figure 4: Functional Decomposition Preserve Data

2.1.8. Deliver data

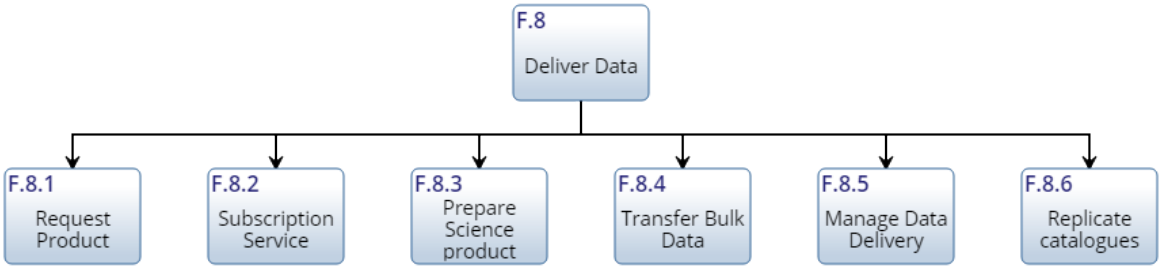


Figure 5: Functional decomposition of DELIVER DATA

2.1.9. Monitor & Control SDP

In order to support the primary functionality, SDP needs to perform monitoring and control functionality. It includes Local Monitoring & Control of SDP, responding to control information from Telescope Manager and providing Resource information and healthstate to Telescope Manager.

2.1.10. Support Observatory Commissioning & Operations

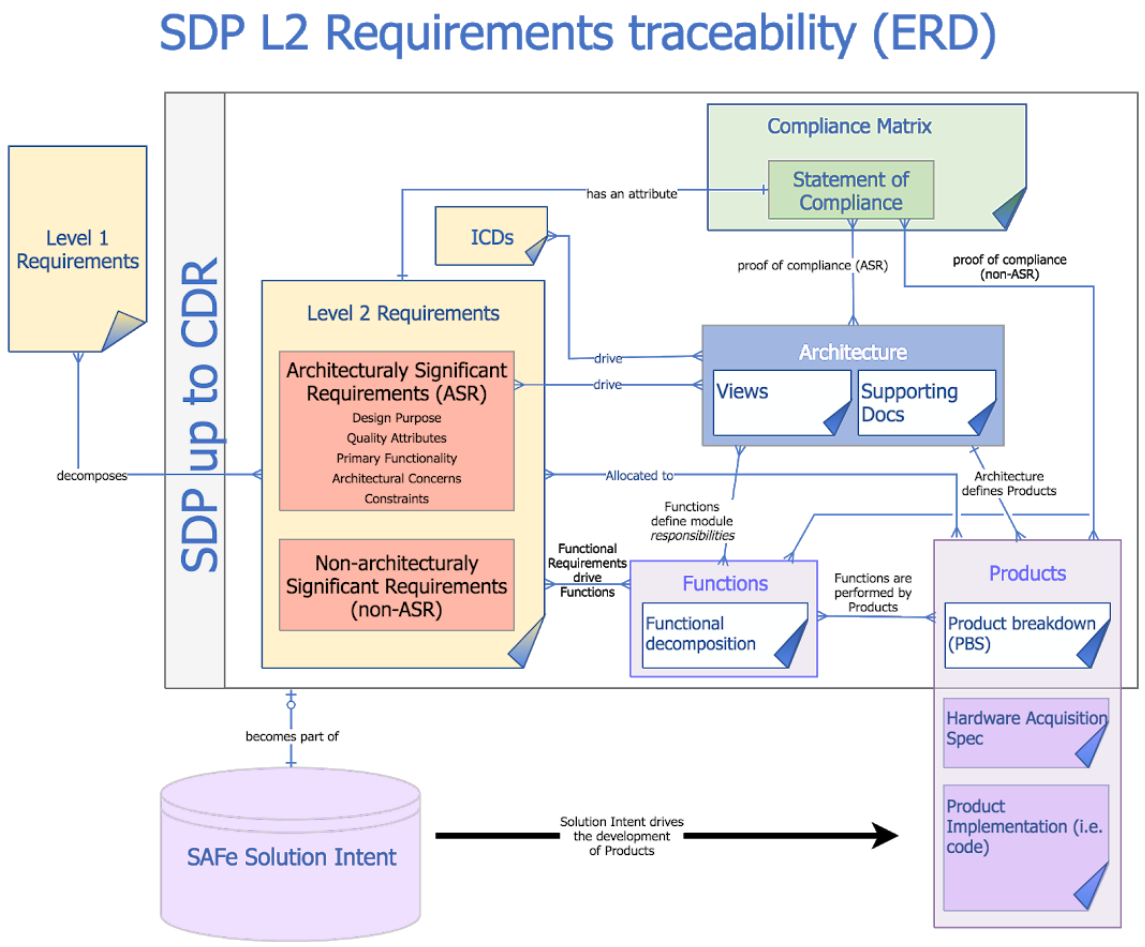
SDP shall support Observatory Commissioning & Operations. The “lower level functions” has been described as a Use Case in the Science Pipeline Management Use Case View.

2.2. Relations and Their Properties

2.2.1. Requirements Model

The role of functions is explained in Figure 6. From this diagram it is clear that functions have a mapping to

- Requirements (drive)
- Products (performed by), Processing Software Modules (define module responsibilities)
- Statement of Compliance (proof of compliance for non-ASR)



Reference Artefacts:

has an attribute, proof of compliance - compliance rationale in SKA-TEL-SDP-0000033_RSP_03_SDPL2Requirements

drive - referenced in Architectural Views and Supporting documents, which document stated in Compliance Rationale

allocated to - labels in SKA-TEL-SDP-0000033_RSP_03_SDPL2Requirements

decomposes - SKA-TEL-SDP-0000033_RSP_03_SDPL2Requirements

Functional Requirements drive Functions - Labels in SKA-TEL-SDP-0000033_RSP_03_SDPL2Requirements

Functions define module responsibilities - SKA-TEL-SDP-0000013_06_SDPArchitecture_Mapping Spreadsheet

Functions are performed by Products - mapping in SKA-TEL-SDP-0000013_06_REP_SDP Functional View

Architecture defines Products - PBS / Module View/ Hardware Allocation View

becomes part of - all the L2 Requirements, ICDS, functions, products and architecture documentation becomes part of the Solution Intent for the next phase

Solution Intent drives the development of Products - SAFe is used where development continues. This takes the place of L3 requirements and lower level specifications

Legend:

Relationships written from perspective of L2 e.g. decomposes

Verification not included for the purposes of this diagram

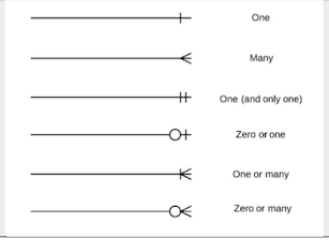


Figure 6: SDP Requirements Traceability Model

2.2.2. Relationship to requirements

The SDP L2 requirements are allocated to the functions and documented in [RD02][1].

2.2.3. Relationship to products

Each of these primary functions has a *performed by* relationship to a Product[2]. Table 1 below shows this mapping of functions to products following the *performed by* relationship. Products are documented in the SDP PBS [AD1], including Software Modules (see SDP System-level Module Decomposition and Dependency View) and Hardware Products (see Hardware Decomposition View).

		Science Data Processor														
	SDP Functions to Science Data Processor	P Science Data Processor	P.1 SDP Software	P.1.1 System Interfaces	P.1.2 Platform Services	P.1.3 Processing Functions	P.1.4 Science Pipeline Workflows	P.1.5 Execution frameworks	P.1.6 Execution Control	P.1.7 SDP Services	P.1.8 Data Models	P.1.9 Workflow Libraries	P.1.10 Quality Assessment	P.1.11 Platform Interfaces	P.2 SDP Hardware MID/LOW	P.2.1 SDP Compute Hardware	P.2.2 SDP Preservation Hardware
SDP Functions	F SDP Functions	X															
	F.1 Receive Data		X	X	X	X	X	X	X		X	X		X	X	X	
	F.2 Process Visibility Data Fast		X	X	X	X	X	X	X		X	X		X	X	X	
	F.3 Process Visibility Data		X	X	X	X	X	X	X		X	X		X	X	X	
	F.4 Process Pulsar Candidates		X	X	X	X	X	X	X		X	X		X	X	X	
	F.5 Process Pulsar Timing		X	X	X	X	X	X	X		X	X		X	X	X	
	F.6 Aggregate quality metrics		X	X	X					X			X	X	X		
	F.7 Preserve Data		X	X	X					X	X			X	X	X	X
	F.8 Deliver Data		X	X	X				X	X				X	X	X	
	F.9 Monitor & Control SDP		X	X	X				X			X		X	X	X	
	F.10 Support Observatory Commissioning & Operations		X	X	X		X							X	X	X	

Table 3: SDP High Level Function to High Level Product mapping where functions are *performed by* products.

As seen in Table 1, the processing functions (F.1 through F.5) map to most of the SDP products shown in the table and therefore the correct interpretation is that most SDP products are required to perform any of the processing functions. Although this is technically the correct way to interpret the *performed by* relationship, and this is typically the mapping used in a functional architecture, it is not a useful view onto the SDP architecture. In order to perform a useful allocation of processing functions to products (in particular software modules) the functions have been further decomposed. This View is accompanied by a Spreadsheet with filename **SKA-TEL-SDP-0000013_06_SDPArchitecture_MappingSpreadsheet.** It provides a mapping between lower level domain related functions and Processing Component Modules, where the "functions define module responsibilities" relationships is shown.

[1] Filter the labels column for a particular function to obtain a list of requirements per functions.
[2] This is modelled in Innoslate and functions (actions) are *performed by* products (assets)

	SDP Functions to Science Data Processor	Science Data Processor													
		P Science Data Processor	P.1 SDP Software	P.1.1 System Services	P.1.2 Platform Services	P.1.3 Core Processing	P.1.4 Science Pipeline Workflows	P.1.5 Execution frameworks	P.1.6 Execution Control	P.1.7 SDP Services	P.1.8 Data Models	P.1.9 Execution Framework Interface	P.2 SDP Hardware MID/LOW	P.2.1 SDP Compute Hardware	P.2.2 SDP Preservation Hardware
SDP Functions	F.6 Preserve Data		X	X	X					X	X		X	X	X
	F.6.1 Index SDP Data Products		X	X	X					X	X		X	X	
	F.6.2 Persist SDP Data Product		X	X						X			X	X	X
	F.6.3 Backup SDP Data Products		X	X						X			X	X	X
	F.7 Deliver Data		X	X	X				X	X			X	X	
	F.7.1 Request Product		X	X	X					X			X	X	
	F.7.2 Subscription Service		X	X	X					X			X	X	
	F.7.3 Prepare Science product		X	X	X	X			X	X	X		X	X	
	F.7.4 Transfer Bulk Data		X	X	X					X			X	X	
	F.7.5 Manage Data Delivery		X	X	X					X			X	X	
	F.7.6 Replicate catalogues		X	X	X					X			X	X	

Table 4: Mapping of non-domain functions

2.2.4. Relationship to Statement of Compliance

Non-ASR requirements' statement of compliance include mention of functions and products. The Statement of Compliance is documented in [RD02].

2.3. Element Interfaces & Behaviour

Initially functional flow / action diagrams were drawn up for all functions in Innoslate. The detail of behaviour and data flow mow lies in the solution space as documented in the Workflow Views. Therefore these action diagrams are not included in this view.

3. Context Diagram

Figure 3 in the SDP Architecture Overview serves as a good Context Diagram for this View.

4. Rationale

4.1. Primary Functionality

The primary functionality of the SDP is derived from the SDP L2 requirements [RD02]. This is user required functionality and therefore an implementation-free representation.

[RD1] defines Primary Functionality as follows:

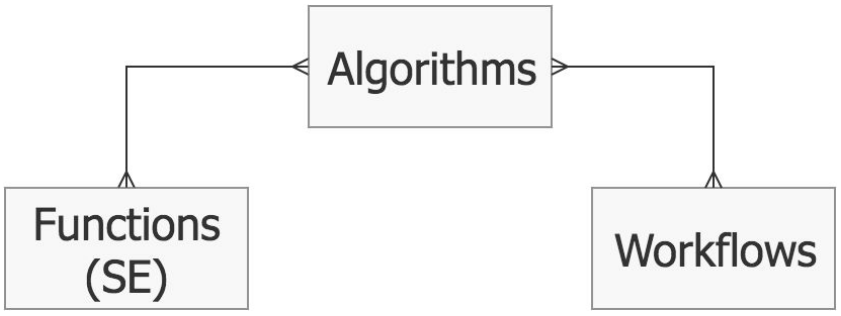
> "Functionality of the system to do the work for which it was intended. Primary functionality is usually defined as functionality that is critical to achieve the business goals that motivate the development of the system". p 272 [RD1].

The functional decomposition in this view is based on a functional model. Depending on an individual's background or expectations, he/she may have different ideas of what a functional model should look like. This model focuses on **primary functionality** and serves to complement the other SDP Architecture Views. Non-functional aspects and derived or implied functionality are not modelled here as these aspects are represented (and rationalised) more clearly in the other views, using the Views and Beyond approach. The functional model shown here is therefore not a complete or detailed functional model of the SDP system, but meant to supplement the SDP Architecture Views. This approach is significantly different from the classical FFBD style of functional modelling (where the sequential relationship of all functions that must be accomplished by a system are modelled).

SEI also advocates other definitions of functions, e.g. functions that require a high level of technical difficulty, or require interaction with several architectural elements. The focus in this case, however, is "motivating the development of the system" i.e. what the SDP is really intended to do.

4.2. Functions, Workflows and Algorithms

Defining the functions for SDP has been difficult. This was partly due to an unclear distinction between Functions, Workflows and Algorithms. This distinction brought some clarity. Functions (per Systems Engineering definition) represents the problem definition, while workflows represent the solution definition. Forcing a relationship between the two is not sensible, although both are in the behavioural domain. A way to understand this is through the relationships with algorithms. Functions required certain algorithms, while workflows implement specific algorithms.



5. Related Views

There are no parent and children views for this view.

6. References

6.1. 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.

This list of applicable documents applies to the whole of the SDP Architecture.

[AD01] SKA-TEL-SKO-0000002 SKA1 System Baseline Design V2, Rev 03

[AD02] SKA-TEL-SKO-0000008 SKA1 Phase 1 System Requirement Specification, Rev 11

[AD03] SKA-TEL-SDP-0000033 SDP Requirements Specification and Compliance Matrix, Rev 02C

[AD04] SKA-TEL-SKO-0000307 SKA1 Operational Concept Documents, Rev 02

[AD05] 000-000000-010 SKA1 Control System Guidelines, Rev 01

[AD06] 100-000000-002 SKA1 LOW SDP to CSP ICD, Rev 04A

[AD07] 100-000000-025 SKA1 LOW SDP to SaDT ICD, Rev 04

[AD08] 100-000000-029 SKA1 LOW SDP to TM ICD, Rev 03B

[AD09] 100-000000-033 SKA1 LOW SDP to LFAA Interface Control Document (ICD), Rev 01

[AD10] 300-000000-002 SKA1 MID SDP to CSP ICD, Rev 04A

[AD11] 300-000000-025 SKA1 MID SDP to SaDT ICD, Rev 04

[AD12] 300-000000-029 SKA1 MID SDP to TM ICD, Rev 03B

[AD13] SKA-TEL-SKO-0000484 SKA1 SDP to INFRA-AUS and SKA SA Interface Control Document, Rev 02

[AD14] SKA-TEL-SKO-0000661 Fundamental SKA Software and Hardware Description Language Standards

[AD15] http://www.ivoa.net/documents/TAP/

[AD16] http://www.ivoa.net/documents/latest/SIA.html

[AD17] http://www.ivoa.net/documents/DataLink/

[AD18] http://www.ivoa.net/documents/SSA/

[AD19] Memorandum of Understanding between the SKA organisation and National Radio Astronomy Observatory relating to a work package for the study and design of a new data model for the CASA software package

[AD20] MeasurementSet definition version 3.0. MSv3 team, eds. 2018. http://casacore.github.io/casacore-notes/264

[AD22] Shibboleth Authentication Service from Interenet2
https://www.internet2.edu/products-services/trust-identity/shibboleth/

[AD23] COmanage Authorization Service from Interenet2
https://www.internet2.edu/products-services/trust-identity/comanage/

[AD24] SKA-TEL-SKO-0000990 SKA Software Verification and Testing Plan

6.2. 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.

[RD01] Designing Software Architectures: A Practical Approach (SEI Series in Software Engineering) 1st Edition, by Humberto Cervantes, Rick Kazman

[RD02] SKA-TEL-SDP-0000033, SDP L2 Requirements, Rev 03

[RD03] SKA-TEL-SDP-0000047, SDP Construction and Verification Plan

[RD04] SDP Configuration Item List, SKA-TEL-SDP-0000048, Rev 03