Computing & Software Architecture: Reference Design

020.50.00.00.01-0002-REP-A-COMPUTING_SOFTWARE_ARCHITECTURE_REF_DSN

Status: **RELEASED**

PREPARED BY	**ORGANIZATION**	**DATE**
R. Hiriart, J. Robnett, M. Pokorny	Computing & Software Div., NRAO	2019-07-19

APPROVALS (Name and Signature)	**ORGANIZATION**	**DATE**
R. Selina, Project Engineer 2019.07.30 14:35:32 -06'00'	Electronics Division, NRAO	2019-07-30
M. McKinnon, Project Director Mark McKinnon Digitally signed by Mark McKinnon Date: 2019.07.30 14:46:03 -06'00'	Asst. Director, NM-Operations, NRAO	2019-07-30

RELEASED BY (Name and Signature)	**ORGANIZATION**	**DATE**
M. McKinnon, Project Director Mark McKinnon Digitally signed by Mark McKinnon Date: 2019.07.30 14:46:20 -06'00'	Asst. Director, NM-Operations, NRAO	2019-07-30

Change Record

Version	Date	Author	Affected Section(s)	Reason
01	2018-10-09	Hiriart	All	Initial Draft.
02	2019-07-19	Hiriart	All	Release for reference design package.
03	2019-07-23	Selina	All	Minor copyediting.
A	2019-07-30	Lear	All	Prepare PDF for signatures and release.

Table of Contents

1 Introduction ... 5
1.1 Purpose ... 5
1.2 Scope ... 5
2 Related Documents ... 5
2.1 Applicable Documents ... 5
2.2 Reference Documents ... 5
3 General Architecture ... 6
3.1 Subsystems ... 6
3.2 Data Stores ... 7
3.3 System Actors ... 8
4 Proposal Management Subsystem ... 10
4.1 Subsystem Function: Proposal Submission ... 10
4.2 Subsystem Function: Proposal Evaluation ... 10
4.3 Subsystem Function: Telescope Time Allocation ... 10
4.4 Subsystem Function: Scheduling Block Generation ... 10
4.5 Subsystem Function: Scheduling Block Definition ... 10
4.6 Subsystem Function: Security ... 11
4.7 Interfaces and Dataflows ... 11
4.7.1 Proposal Submission Tool Interfaces ... 11
4.7.2 Proposal Handling Tool Interfaces ... 11
4.7.3 Proposal Builder Tool Interfaces ... 11
4.7.4 Observation Preparation Tool Interfaces ... 11
5 Online Subsystem ... 12
5.1 Subsystem Function: Scheduling Block Selection ... 12
5.2 Subsystem Function: Sub-Array Management ... 13
5.3 Subsystem Function: Telescope System Configuration ... 13
5.4 Subsystem Function: Array System Calibration ... 13
5.5 Subsystem Function: Observation Control ... 14
5.5.1 Antenna Control Functions ... 15
5.5.2 Antenna Data Transmission Functions ... 16
5.5.3 Central Signal Processor Control Functions ... 17
5.5.4 Correlator Back End Processing Functions ... 17
5.5.5 Array Observation Control Functions ... 18
5.6 Subsystem Function: Archive Ingestion ... 19
5.7 Subsystem Function: Metadata Capture ... 19
5.8 Subsystem Function: Online Calibration ... 19
5.9 Subsystem Function: Online Quality Assurance ... 20
5.10 Subsystem Function: Operator Shift Logging ... 20
5.11 Subsystem Function: Monitoring ... 20
5.12 Subsystem Function: Error Handling ... 20
5.13 Subsystem Function: Security ... 20
5.14 Interfaces and Dataflows ... 20
5.14.1 Scheduling Module Interfaces ... 22
5.14.2 Operation Module Interfaces ... 22
5.14.3 Observation Module Interfaces ... 23

5.14.4 Antenna Supervisory Control Module Interfaces 23
5.14.5 LO and Timing Supervisory Control Module Interface 23
5.14.6 CSP Supervisory Control Module Interfaces 23
5.14.7 Metadata Capture Module Interfaces 23
5.14.8 Configuration Module Interfaces 24
5.14.9 Quick-Look Module Interfaces 24
5.14.10 Calibration Module Interfaces 24
5.14.11 Monitoring Module Interfaces 24
5.14.12 Data Streams 24
5.14.13 Controller Board Interfaces 25
5.15 Physical Architecture 25
6 Offline Subsystem 28
6.1 Subsystem Function: Raw Data Calibration 29
6.2 Subsystem Function: Imaging Product Generation 30
6.3 Subsystem Function: Source Catalog Product Generation 30
6.4 Subsystem Function: Project Tracking 30
6.5 Subsystem Function: Data Product Discovery 31
6.6 Subsystem Function: Data Product Retrieval 31
6.7 Subsystem Function: Data Product On-Demand Post-Processing 31
6.8 Subsystem Function: Data Product Analysis 31
6.9 Subsystem Function: Security 31
6.10 Interfaces and Data Flows 31
6.10.1 Science Archive Interfaces 32
6.10.2 Observatory Interfaces 32
6.10.3 Data Processing Management Interfaces 33
6.10.4 Processing Resource Interfaces 33
6.10.5 Data Processing Pipeline Interfaces 33
6.11 Quality Assurance Interfaces 33
6.12 Data Analysis Package Interfaces 33
6.13 Data Streams 33
6.14 Physical Architecture 34
6.14.1 Data Reduction Performance Requirements 35
7 Maintenance and Support Subsystem 39
7.1 Subsystem Function: Preventive Maintenance Analysis 39
7.2 Subsystem Function: Maintenance Schedule Management 39
7.3 Subsystem Function: Inventory Management 39
7.4 Subsystem Function: Troubleshooting Support 39
7.5 Subsystem Function: Security 39
7.6 Interfaces and Dataflows 39
7.7 Physical Architecture 40
8 Development Support Subsystem 40
8.1 Physical Architecture 40
9 Appendix 41
9.1 Abbreviations and Acronyms 41

1 Introduction

1.1 Purpose

This document describes the computing and software architecture for the ngVLA reference design. The architecture is defined by decomposing systems functions progressively in higher levels of detail, and defining the necessary structural elements and interfaces accordingly. First, the system structure is partitioned into subsystems based on high-level system functions. Next, the subsystems are broken down into modules, and the high-level functions are decomposed into subsystem functions. The functionality required by these functions is allocated between the modules, along with the necessary interfaces.

The physical architecture, i.e. the computing and networking equipment necessary to implement the required functions, is defined for each subsystem. This architecture will be refined in subsequent iterations during the project's Conceptual Design phase. Modules will be decomposed into software components, and their requirements and interfaces will be defined in detail.

1.2 Scope

This document covers all computing and software systems required for ngVLA as specified in the reference system design [AD04]. The detail provided in this document has been defined to the level necessary to enable the process of estimating development and construction costs. This document avoids the definition of design features when these do not significantly impact the cost or constrain the design.

2 Related Documents

2.1 Applicable Documents

The following documents may not be directly referenced herein but provide necessary context and requirements applicable to this architecture.

Ref. No.	Document Title	Rev / Doc. No.
AD01	ngVLA Science Requirements	020.10.15.00.00-0001-REQ
AD02	System Requirements	020.10.15.10.00-0003-SPE
AD03	Operations Concept	020.10.05.00.00-0002-PLA
AD04	ngVLA System Reference Design	020.10.20.00.00-0001-REP
AD05	SRDP System Concept	530-SRDP-014-MGMT
AD06	M&C Reference Design Concept	020.50.25.00.00-0002-DSN
AD07	Reference Observing Program	020.10.15.05.10-0001-REP

2.2 Reference Documents

The following documents are referenced within this text or provide supporting context.

Ref. No.	Document Title	Rev / Doc. No.
RD01	Digital Back End & Data Transmission System: Reference Design	020.30.25.00.00-0002-DSN
RD02	LO Reference and Timing: Reference Design	020.35.00.00.00-0002-DSN
RD03	Central Signal Processor: Reference Design	020.40.00.00.00-0002-DSN

3 General Architecture

In general, all functions under the Computing and Software IPT fall in the following main categories:

1. Manage the proposal submission process and schedule observations.
2. Perform observations.
3. Monitor telescope status and support maintenance activities.
4. Generate derived data products.
5. Provide access to data products.
6. Support commissioning, development, and other operational activities.

The expected data rates for raw data (average 7.46 GB/s, peak of ~132 GB/s for the most stringent use cases; Table 3) are high but expected to be technically manageable and economically feasible in 2035, given recent advances in computing and storage systems. For perspective, this is 1,000 times the current VLA data rate but roughly 10,000 times less than the SKA (expected to be several Petabytes per second). Differently from SKA, for ngVLA it will be possible to store the raw data and decouple data acquisition/correlation from generation of calibration and imaging products. Consequently, the system architecture has been decomposed structurally into five major subsystems and four data stores, as shown in Figure 1 (next page).

3.1 Subsystems

The subsystems that define this architecture are detailed in Figure 1 and include the following:

- **Proposal Management Subsystem:** This subsystem receives proposal submissions from PIs and provides software to support review, scoring, and allocation in observing sessions. It also provides software to simulate several weather scenarios and how they affect the schedule of observations on several timescales. Proposals accepted for observation are transformed into one or more scheduling blocks. This process transforms the proposal's science requirements into the technical configurations necessary to perform the observation with the telescope.
- **Online Subsystem:** The online subsystem selects scheduling blocks and executes them in a sub-array. It coordinates all operations necessary to perform observations, resulting in archived science data. The system reads configurations and calibrations from the telescope configuration database and applies them in the corresponding telescope systems. It continuously monitors the state of the telescope, saving the monitoring data and alarms in the Engineering database. This database also includes integrated logs generated by all software components as they perform their operations.
- **Offline Subsystem:** This subsystem is responsible for all post-observation operations performed on the collected science data, including the generation of derived data products, support for quality assurance activities, and the provision of interfaces to search for and retrieve data products.
- **Maintenance and Support Subsystem:** This subsystem incorporates several modules that support engineering diagnostic and maintenance activities.
- **Development Operations Subsystem:** This subsystem integrates common software components and provides supporting infrastructure for simulation and testing.

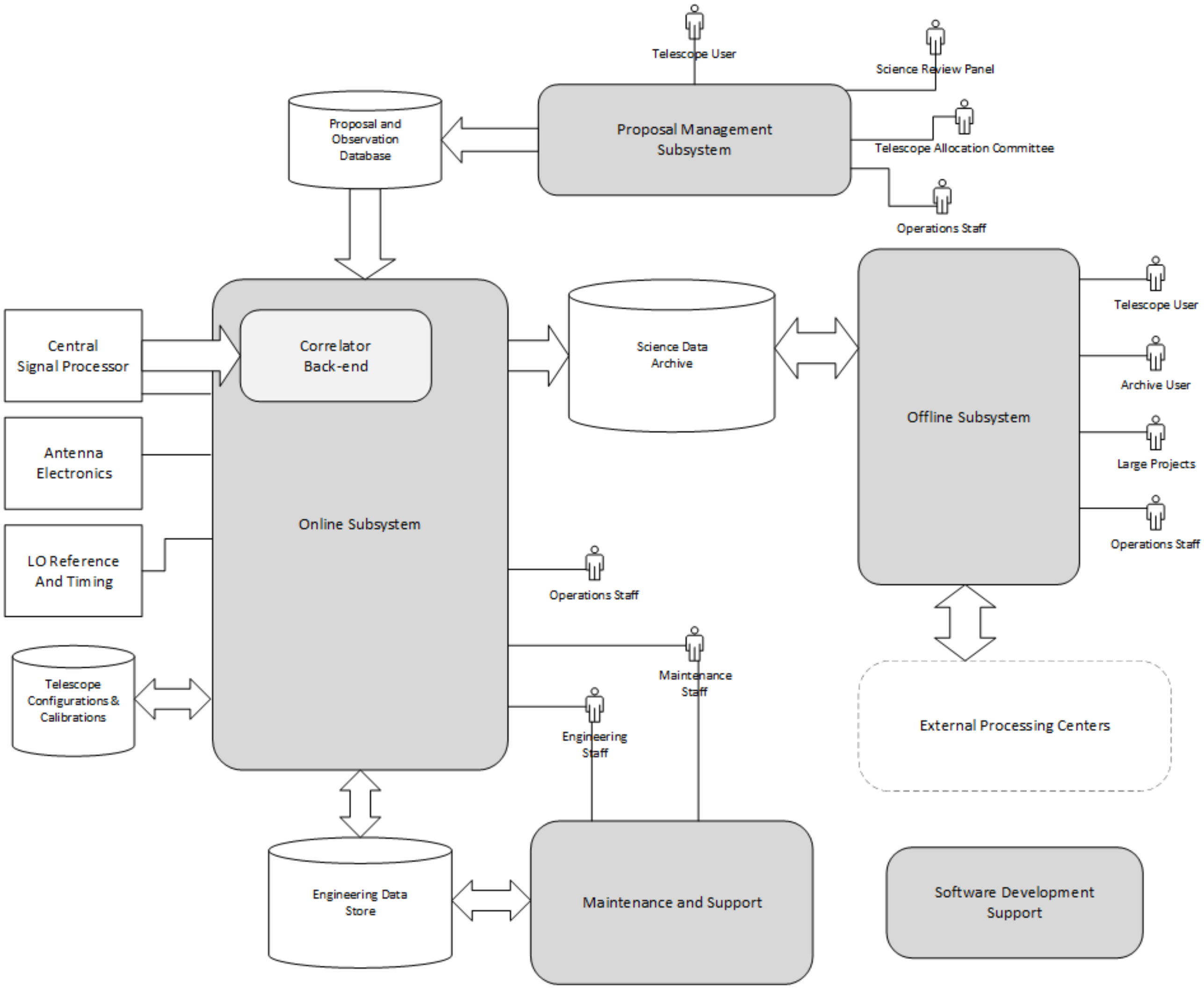

Figure 1 - General architecture for ngVLA. The major subsystems are shown as gray boxes. Also shown are the interfaces and data flows between subsystems, data stores, main actors, and other external system elements.

3.2 Data Stores

The data stores shown in Figure 1 are logical and are deployed into physical storage systems and databases depending on their requirements. They may be consolidated if necessary. The data stores include:

- **Proposal and Observation Database:** This contains science proposals and scheduling blocks along with associated data structures such as source catalogs, user information, and telescope capabilities.
- **Telescope Configurations and Calibrations Database:** This database contains all configuration data necessary to bring the telescope to an operational state, including calibrations such as antenna position and the relative offsets to focal plane, antenna pointing parameters, delay models, etc.
- **Science Data Archive:** The science data includes visibility files, calibration tables, images, catalogs, and their associated metadata.
- **Engineering Data Store:** This mainly includes monitoring data, alarms, and system logs.

3.3 System Actors

Figure 1 also shows the main system actors, which include:

- **Telescope Users:** These users enter the system through a well-defined proposal process. The Observatory has well established processes for adjudicating proposals' relative merits and allocating telescope time based on this process. Telescope users design observations to address specific scientific questions and envision the data products that will allow them to address these concerns.
- **Archive Users:** Archive users enter the system through the archive interface and may be anonymous. These users seek to reuse existing data (and products) to answer scientific questions, which may be unrelated to the initial science case.
- **Large Projects:** These projects can be both telescope and archive based, and represent a limited number of significant investments on the part of both the PI and the Observatory.
- **Operations Staff:** This category primarily comprises data analysts, telescope operators, scientific staff, and any other NRAO staff with functional effort allocated to supporting observatory operations.
- **Maintenance Staff:** These NRAO staff members are responsible for the preventive and corrective maintenance activities of the array.
- **Engineering Staff:** NRAO Engineering Division staff members support array operations, maintenance, and development activities.
- **Science Review Panel:** This panel of scientists evaluates and scores proposal according to their scientific merit.
- **Telescope Allocation Committee:** This committee is responsible for ranking proposals and allocating the available observing time among the accepted proposals. This committee incorporates the chair of each Science Review Panel.

Table 1 shows system function allocations to subsystems and their decomposition in subsystem functions.

System Function	Allocated Subsystems	Subsystem Functions
1. Manage proposal submission process and schedule observations	Proposal Management Subsystem	Proposal Submission Proposal Evaluation Telescope Time Allocation Scheduling Block Generation Scheduling Block Definition
2. Perform observations	Online Subsystem	Scheduling Block Selection Sub-Array Management Telescope System Configuration Array System Calibration Observation Control Archive Ingestion Metadata Capture Online Calibration Online Quality Assurance Operator Shift Logging
3. Monitor telescope status and support maintenance activities	Online Subsystem	Monitoring Error Handling
	Maintenance & Support Subsystem	Engineering Data Analysis Preventive Maintenance Management Inventory Management

System Function	Allocated Subsystems	Subsystem Functions
4. Generate derived data products	Offline Subsystem	Raw Data Calibration Imaging Product Generation Source Catalog Product Generation
5. Provide access to data products	Offline Subsystem	Project Tracking Data Product Discovery Data Product Retrieval Data Product On-Demand Post-Processing Data Product Analysis
6. Support commissioning, development. and other operational activities	Development Operations Subsystem	E.g., offline development and testing of new observing modes.
	(Entire system)	E.g., operations reporting, testing.

Table 1 - Allocation of system functions in subsystems.

Supporting integration and verification, commissioning, and development activities may require provision of special paths and extended APIs across the system. For example, running observations from scheduling blocks generated from proposal information imposes unnecessary overhead on these activities. Commanding the array directly from observation scripts is preferred.

It is also usually required to provide low-level access to hardware interfaces, while normal observations use higher-level science-oriented interfaces. These concerns will not be introduced as separate functions but rather as architectural features across the system.

In addition to the subsystem functions defined above, each subsystem must implement an adequate level of security. This is defined as an additional function, which will be specified in more detail during the Conceptual Design phase of the project.

The following sections describe each subsystem in detail.

4 Proposal Management Subsystem

The Proposal Management Subsystem consists of the same system that currently handles the NRAO proposal process, extended to accommodate ngVLA requirements. This system is expected to be modernized and improved before the ngVLA construction phase begins. Although its architecture may change as a result of these improvements, at the present time its modules include:

- **Proposal Submission Tool (PST):** This Web application lets Telescope Users submit an observing proposal for any NRAO research facility: VLA, VLBA, or GBT. This application will be extended to support ngVLA. This application also supports the proposal evaluation process, allowing members of the Science Review Panel to score the proposals.
- **Proposal Handling Tool (PHT):** This Web application supports the time allocation process. It facilitates the process of ranking and assigning time for each accepted proposal, taking into account its observing requirements, the available telescope time, predicted weather patterns, etc.
- **Proposal Builder Tool (PBT):** This utility generates Scheduling Blocks from the Proposal.
- **Observation Preparation Tool (OPT):** This Web application allows users to manage Scheduling Blocks and associated structures.

These applications require several modifications for ngVLA. Some of these modifications are general and can be introduced before the ngVLA starts its construction phase. These include changes necessary for SRDP, such as the introduction of the observing modes that are supported for automated imaging, and the generation of observable Scheduling Blocks (currently the PBT creates only a skeleton Scheduling Block, which needs to be edited by the user to make it observable). Another change that may be necessary is the introduction of grouping structures similar to ALMA *ObsUnitSet* and *ScienceGoal*.

4.1 Subsystem Function: Proposal Submission

This process allows a Telescope User to create and submit a Proposal requesting ngVLA observing time. This function is allocated in the PST.

4.2 Subsystem Function: Proposal Evaluation

This process allows the Science Review Panel members to score each proposal. This function is allocated in the PST.

4.3 Subsystem Function: Telescope Time Allocation

This is the process through which the Telescope Allocation Committee assigns telescope time to each accepted proposal. This function is allocated in the PHT.

4.4 Subsystem Function: Scheduling Block Generation

This function refers to the generation of Scheduling Blocks from the Proposal. It is allocated in the PBT.

4.5 Subsystem Function: Scheduling Block Definition

This function allows the creation, modification, or deletion of Scheduling Blocks, and it is allocated in the OPT. Manual creation of Scheduling Blocks is necessary to support Integration & Verification and Commissioning activities.

4.6 Subsystem Function: Security

This function refers to the implementation of strategies and countermeasures to deal with internal and external security threats.

4.7 Interfaces and Dataflows

4.7.1 Proposal Submission Tool Interfaces

- **Telescope User UI:** Through this user interface (UI), Telescope Users define, validate, and submit proposals.
- **Science Review Panel UI:** The Science Review Panel uses this interface to score proposals.
- **Operations Staff Interface UI:** This interface lets Operations Staff manage the proposal process. Constructed around other interfaces in this module, it provides extended privileges and capabilities.
- **Proposal Submission Tool Interface:** This is a REST interface used by the UIs in this module.

4.7.2 Proposal Handling Tool Interfaces

- **Telescope Allocation Committee UI:** This interface allows the TAC to manage the telescope allocation time process.
- **Proposal Handling Tool Interface:** This is a REST interface used by the UIs in this module.

4.7.3 Proposal Builder Tool Interfaces

- **Operation Staff UI:** The Operations Staff generates scheduling blocks from the proposal using this interface.
- **Proposal Builder Interface:** This is a REST interface used by the UIs in this module.

4.7.4 Observation Preparation Tool Interfaces

- **Telescope User UI:** The Telescope User uses this to review and edit scheduling blocks and associated structures.
- **Operations Staff Interface UI:** This interface lets Operations Staff manage scheduling blocks and associated structures. It is usually constructed around other interfaces in this module but provides extended privileges and capabilities.
- **Observation Preparation Tool Interface:** This is a REST interface used by the UIs in this module.

5 Online Subsystem

The Online Subsystem includes the following modules:

- **Scheduling:** This module reads the set of scheduling blocks to be observed from the Proposal and Observation Database and constructs an observation schedule based on current weather conditions and short time prediction. The observation schedule defines a program of sub-arrays and their corresponding queues of scheduling blocks.
- **Observation:** The Observation module exposes interfaces to create and destroy sub-arrays, and execute observations on them. This module implements the supported observing modes, which represent different ways to use telescope hardware to perform observations. The execution of an observation results in several commands sent to the telescope hardware, which trigger parallel and sequential operations that are coordinated by the Observation module. This module sends metadata to the Metadata Capture component as the observation proceeds.
- **Control:** This module uses software components to control telescope hardware elements, organized into different hierarchies for Antenna Electronics, Local Oscillator, and Timing equipment.
- **Correlator:** Correlator module components control the Central Signal Processor.
- **Correlator Back End**: The Correlator Back End receives visibility data from correlator hardware and performs a series of post-correlation operations before saving the files in their final format. It also receives, formats, and saves pulsar search and timing data.
- **Metadata Capture:** This module receives data from multiple sources and integrates it all in a series of tables that are saved along with the visibility data.
- **Telescope Configuration:** This module provides interfaces for other components to query telescope configuration and calibration data.
- **Calibration:** This module reads the visibility data saved during calibration scans and computes several calibration tables. These are saved along with other observation metadata tables, and in some cases are applied on the telescope instrumentation.
- **Quick-Look:** This module computes observation quality assurance information and provides interfaces to present this data to the Astronomer on Duty and the Operators.
- **Monitoring** This module provides several components that collect monitoring data from hardware controller and supervisor components, and archives the data into the Engineering Database. This module also contains interfaces to query and present results from this database.

The next sections describe the subsystem functions.

5.1 Subsystem Function: Scheduling Block Selection

This function permits selecting one or more Scheduling Blocks for execution in a specific sub-array execution queue. It is allocated entirely in the Schedule module. It provides three operation modes:

1. Manual mode: The operator manually selects the Scheduling Block to be executed.
2. Advisory mode: The system presents a selection of recommended Scheduling Blocks to be executed, but the operator must manually insert them into the execution queue.
3. Fully automatic: The system selects automatically the Scheduling Blocks to be executed.

The first manual mode requires querying the pool of "observable" Scheduling Blocks and presenting the set of attributes that the operator needs to perform a selection. The advisory and fully automatic modes require the current weather data and usually perform a short-term weather prediction to decide the next Scheduling Block to observe.

For VLBI observations, VEX files will be converted into Scheduling Blocks. These Scheduling Blocks must be executed at a precise time along with other observatories, so their handling in the execution queue have special requirements.

5.2 Subsystem Function: Sub-Array Management

This function refers to the allocation of disjointed sets of antennas in sub-arrays. It is implemented as an interaction of the Array Operator, the Scheduling module, and the Observation module. The Operator decides how the telescope antennas should be partitioned in sub-arrays, and creates sub-arrays by means of a UI in the Scheduling module, which in turn invokes functions in the Observation module for allocating hardware resources to the sub-array and configuring it.

An input for the sub-array allocation algorithm is the availability of antenna capabilities. For example, some antennas will have all bands available, but others could have only some in an operational state. Although some of the information can be retrieved directly from the online system status, it may be necessary to retrieve data from the maintenance database to have a complete picture of each antenna's availability.

Allocating a set of antennas in a sub-array involves several operations in the systems electronics:

1. The Local Oscillator reference and central timing signal generators need to be controlled for each sub-array. Each antenna LO reference is independently tunable, so this function mostly involves setting the required values of each LO reference for each scan.
2. The data transmitted from the antennas to the central signal processor (CSP) is routed to two of three CSP Tridents. This distribution operation is managed by the software, and involves issuing control commands to the digital backend (DBE) and the CSP. These operations may be different for the direct-connected antennas and the antennas connected through the public network.
3. Configure the CSP. This involves configuring the very coarse channelizers (VCCs), mapping the VCCs to common array frequency slice processors (FSPs), and configuring the FSPs.
4. Configure the data transmission between the CSP and CBE, and the CBE resources to process the sub-array data stream from the CSP.

5.3 Subsystem Function: Telescope System Configuration

The telescope systems (electronic, computing, and software) require a set of configuration data that is loaded during the system initialization. This data includes attenuator values, antenna positions, delay models, high/low alarm thresholds, software version information, etc. This data is kept in the Telescope Configuration database, under version control. This subsystem function includes both the provision of interfaces to maintain this data and procedures for retrieving and loading data on corresponding devices.

For hardware devices, this data will be loaded by the Supervisory Layer if the Telescope Configuration database can be accessed. If not, the devices will load default values from internal memory. The system should support updating and loading configuration data with minimal re-initializations.

5.4 Subsystem Function: Array System Calibration

This function refers to the acquisition of array calibration data. This includes measurement of

1. the antenna positions,
2. the delays in fiber optic cables and electronics,
3. the antenna pointing coefficients,
4. the antenna focus coefficients, and
5. the antenna surface deviations.

These measurements involve the execution of specialized procedures and observations. The acquired data will be stored in the Configuration database, along with the time the measurement was performed. After QA, this data is applied on the system and is then available to be read and loaded on the required hardware and software systems.

Note that this is not the only calibration function in the system. Calibration data are also acquired during an observation (usually by observing calibrator sources of the required characteristics), and calibration tables are derived during post-processing. These are described separately.

5.5 Subsystem Function: Observation Control

This section describes how the system issues control commands to perform astronomical observations. The system must support the following observing modes:

1. Continuum interferometry
2. Spectral line interferometry
3. Total Power (auto-correlation products)
4. Pulsar Timing (phased sum mode)
5. Pulsar Search (phased sum mode)
6. Very Long Baseline Interferometry (phased sum mode)
7. On-the-Fly Mosaicking
8. Solar Observing

An observing mode is defined as a way of using sub-array resources to meet a specific science objective. Although observation parameters (number of spectral channels, integration time, field of view, etc.) vary, all science use cases fall in one of the above categories.

An observation is divided in time intervals called *scans*, which have associated *intents*. The intent differentiates scans performed for the purpose of calibration from those performed over the astronomical objects of scientific interest. The observing mode concept also encompasses the calibration strategies that define the required duration and frequency of the calibration scans.

In practice, observing modes require high modifiability and are typically developed and maintained by scientific staff. Because of this, observing modes are implemented in scripting languages (e.g., Python). The execution of an observation script calls an API that exposes high-level functions of the telescope system. These functions encapsulate the technical complexities of controlling the telescope hardware, incorporating timing and concurrency considerations. The ngVLA system will follow a similar architecture. The scripted observing mode and high-level system API will be implemented in the Observation module.

The execution of an observing mode script in the Observation module results in several commands directed to components in the Supervisory layer. These commands include the timestamp of execution, calculated sometime in the future to allow for propagation and processing delays. The supervisory components execute the commands, which in turn results in several operations sent to the Controller boards. The Observation module waits for response messages from the Supervisory components, acknowledging reception of the command, start of the execution, and end of the execution. The Observation module waits for these messages with a timeout, and follows error procedures in case of failures or missing responses. This typically involves sending flagging commands to the Metadata Capture module or aborting the observation in case of critical problems.

Because of the timing unpredictability in the transmission of commands sent to remote station antennas, several commands will be aggregated in a single message, encompassing a larger time interval. This mode of operation is similar to the execution of VLBI observations, where a script for the entire observation is generated and executed in isolation by each station.

Controlling and observation functions can be classified by the scope of involved subsystems. The following sections describe functions specific for Antenna Electronics, Data Transmission, CSP, and correlator Back End systems, and functions encompassing more than one subsystem, i.e. array observation functions.

5.5.1 Antenna Control Functions

The control software must perform the following functions with antenna electronics.

5.5.1.1 Antenna Pointing

For each scan, the antennas are pointed to a point in the sky. As the antennas have alt-azimuth mounts, their movement is controlled by the Antenna Control Unit (ACU) in horizon coordinates (azimuth and elevation). The sky coordinates, on the other hand, are specified in celestial coordinate systems such as equatorial (right ascension and declination) or galactic coordinates. The supervisor software will perform the necessary transformations and command the ACU. These commands form a data stream when the antennas track a sky position as the Earth rotates.

The ACU incorporates a pointing model consisting of several coefficients to account for antenna deformations and other effects. The software uses a calibration program (e.g. TPOINT) to calculate these coefficients from data gathered by special pointing scans. The pointing model includes a static or main model, loaded from the Configuration Database; and an auxiliary or secondary model consisting of coefficient offsets determined by the calibration software from pointing scans and loaded into the ACU before the next scan.

Besides pointing the telescope to fixed positions in the sky, the software should support variable objects or Ephemeris (usually planets and satellites). A list of Ephemeris objects will be supported.

The antenna pointing data are included in the observation metadata recorded by the Metadata Capture module. Because this pointing data includes both the commanded coordinates and the coordinates where the antenna was actually pointed (read from the axis encoders), the pointing data needs to be retrieved from the ACU by the Antenna Control Supervisor module—which is local to the ACU—and sent to the Metadata Capture module to be included in the Science Data Model (SDM) tables.

5.5.1.2 Frequency Tuning

Configuring a sub-array to receive a selected range of frequencies involves setting up the LO frequency to the specified value. Besides this, other functions (fringe rotation and LO offsetting) require passing the LO frequencies to the CSP. This function will be performed by the Control and Correlator modules.

The central LO system distributes the LO signal to all the antennas. This is a 7 GHz reference signal shifted differently for each antenna by an offset frequency. This reference frequency is used at each antenna to synthesize the final down-conversion frequency used in the IRD. Therefore, the software system is required to set the antenna-dependent LO offset frequencies in the central LO system, and perform band-selection and spectral window selection in the antenna electronics.

Each antenna sends a total bandwidth of 20 GHz to the CSP, channelized in 200 MHz spectral windows. The band is selected by controlling the IRD input that will be processed in the DBE. Channelization is performed in the DBE.

5.5.1.3 Doppler Tracking

Because of the relative velocity of a source with respect to the telescope antennas, the observed frequency will appear shifted by a Doppler offset. The system will have the capability to calculate the frequency offsets at the beginning of an observation and apply them when tuning the frequency in the electronics, when necessary.

The Proposal Management subsystem will convert the specified user frequencies, which the PI can specify in different reference frameworks, to a common framework (e.g., the center of the solar system).

5.5.1.4 Signal Path Attenuator Management

This function involves determining and setting gain values in the attenuators installed in the signal path. The values are typically calculated from measurements of the total power, from where the attenuations are determined in such a way to optimize the output power for the antenna digitizer statistics. The sampling of the total power and setting of the gains are exposed as high-level observing script functions.

5.5.1.5 On-the-Fly Mosaicking

In this particular way of observing, an extended area of the sky is observed by moving the antennas continuously in sky coordinates. This mode is usually more efficient than forming mosaics by combining static observations of a spaced grid in the sky, a mode known as "pointed mosaics". The accurate recording of the antenna pointing positions is necessary for synthesizing images in post-processing. This mode imposes constraints in the delay correction.

5.5.1.6 Sideband Separation

After LO mixing in the IRD module, the signal received from the sky is transformed into an IQ pair that contains both upper and lower sidebands. The sidebands are separated digitally in the DBE and the FSP. Frequencies far from the LO frequency are separated in the DBE using coarse frequency resolution, while frequencies close to the LO are separated in the FSP with high spectral resolution. The data is transmitted from the antennas as a set of IQ pairs, both for the frequencies already separated and those not yet separated.

The software is required to store a set of sideband separation coefficients in the Configuration Database. These coefficients will be loaded in the DBE and FSP before execution of an observation.

5.5.1.7 Cryogenic Management

This function refers to the control and monitoring of the Front End cryostats. This function may incorporate a power saving mode.

5.5.1.8 Band Power Management

This function refers to controlling Front End bands powered up during an observation. To save power, not all available bands may be powered up at the same time.

5.5.1.9 Digitizer Offset Correction

The digitizer clock in the antennas is generated from the LO reference signal. However, as this signal includes an offset, the generated signal does not have the correct frequency and must be corrected in the DBE.

5.5.2 Antenna Data Transmission Functions

It is assumed that the data transmission from the connected antennas to the CSP is largely automatic and does not require active control, besides normal monitoring, during an observation. They are assumed to be transmitting continuously to the CSP. On the other hand, the long-haul antennas will transmit data through the public network only while observing. Therefore, the Execution module will need to issue commands to start and stop the data packet transmissions.

5.5.3 Central Signal Processor Control Functions

The CSP can be configured to work in the following observing modes for each sub-array:

- Synthesis Imaging
- (Sparse/Dense) Pulsar Timing
- Offline Pulsar Search
- VLBI

The observing modes defined at the beginning of this section are mapped into one of these correlator observing modes. The CSP observing modes differ from the observation observing modes in that the latter incorporate concerns that do not affect the CSP, such as differences in calibration strategies or antenna movement patterns.

The CSP is further divided in the Correlator and Beamformer (CBF), the Pulsar Engine (PE), and a Monitoring and Control Module that receives control and monitoring commands from the CSP Supervisor server. The CBF is further divided in the VCC-part, which divides received spectra into frequency slices, and the FSP-part, which processes each frequency slice. The computing resources allocated in the CBF and PE must be configured for each sub-array and scan during an observation. The commands necessary to accomplish this are TBD but at a minimum must include a spectral configuration data structure, directing how the spectra should be divided in slices, and how each one should be processed.

For Pulsar modes, which involve simultaneous observation of several beams inside the antenna primary beam, the CSP Supervisor computer will compute beam-forming weights and pass them on to the CSP. The CSP will also receive the necessary data for delay compensation, fringe tracking, and LO offsetting.

The data from each FSP will be directed to a computer node in the correlator backend. The details of this interface are TBD but will probably involve passing the receiving computer node IP addresses to the CSP.

5.5.4 Correlator Back End Processing Functions

The Correlator Back End (CBE) receives data for the low-level products generated in the CSP and produces the final (raw) data products. Depending on the observing modes, the final data products are:

- Visibility data stored in a format compatible with the interfaces used by data processing systems (currently the Measurement Set (MS or Multi-MS) format)
- Pulsar Timing Profiles in PSRFITS format
- Offline Pulsar Search data in PSRFITS format
- VLBI data recorded in VDIF format

The CBE may also generate additional files to store auxiliary data.

The CBE collaborates with the Metadata Capture module. While the Metadata Capture module gathers observation metadata, the CBE processes binary data, performing operations such as band-stitching, averaging, and formatting. In general, all data necessary to form a binary file should be routed to the same CBE computer node, but depending on the processing requirements, other load balancing strategies could be implemented. CBE operations will most probably be parallelized with MPI and MPI-IO, following a design similar to the VLA.

5.5.5 Array Observation Control Functions

5.5.5.1 Delay Tracking

To avoid loss of coherence and sensitivity, signals from each antenna must be in-phase before computing correlation products. This is accomplished by introducing a delay in the signal path for each antenna. The Control module will calculate the delays periodically in advance and send them to the Correlator Supervisor computer, which will compute a polynomial fitting for each time interval and apply it in the CSP. All delay compensations will be applied in the CSP.

5.5.5.2 Fringe Rotation

As the delay compensation is applied after the received signal has been down-converted in frequency, a fringe term appears in the correlator output. This effect can be compensated by introducing a rotating phase in the local oscillator signal or by removing it in the correlator. For ngVLA this operation will be performed in the CSP. The Control module sends the delay commands to the CSP along with LO frequencies.

5.5.5.3 LO Offsetting

To suppress spurious signals and other effects (such as frequency aliasing in the CSP frequency slicing), a distinct offset is introduced in the LO signal for each antenna and is removed before correlation in the CSP. The Control module incorporates these LO offsets when tuning the receiver frequencies and sends them to the CSP.

5.5.5.4 Timing Synchronization

An observation depends on application of several operations at precise, absolute times. Electronic controller cards will be time-synchronized with the Supervisor server by means of NTP. An NTP server will run in the Supervisor server, which will receive the 1 PPS and timing signals from the Reference Receiver module (L503). These signals permit assigning an absolute time (from the central GNSS module) to each edge of the 1 PPS signal. The system will use UT as the time standard across software systems. In addition, the controller cards will receive a 20 Hz clock signal from the L503 module.

The central LO and timing system will be replicated in remote centers connected with the ISP-connected antennas. To account for the communications unpredictability to remote station antennas, it is assumed that the packets containing the digitized samples of the received signal at each antenna will be timestamped. This operation is performed in the DBE, which receives the 1 PPS and timing signals, without involvement of control software.

There are two central time references: a Maser and a GNSS timing system. The former is more stable for relatively short spans of time ($<10^4$ s), while the latter provides a better long-term time reference. Both systems output 1 PPS and 10 MHz clocks. These are compared and their relative error is logged. The system is not required to apply a dynamic phase correction on the maser. Corrections are applied offline.

The central timing equipment will be replicated in remote centers that will provide time synchronization signals for the long-haul, ISP-connected antennas.

5.5.5.5 RFI Excision

This function is implemented in different stages: at the antenna-based voltage streams; after correlation either in the CSP or the CBE; or after formatting and storage in the post-processing systems. As the interface between CSP and CBE has been scoped to support the data rates shown in Table 3, any RFI excision operation that involves high time resolution visibilities will need to be implemented in the CSP.

This function also requires the development of an RFI monitoring database, which will be integrated with the Configuration Database.

5.5.5.6 Return to Phase

This system function refers to the avoidance of phase jumps when switching frequencies during an observation. All the LO mixing operations performed in the signal path need to be considered, returning the phase to the value it would have had if the frequency wouldn't have been changed. Besides the LO signal transmitted to the antennas for down-conversion, it is necessary to control the phase introduced in the digital LO mixing in the correlator. Returning to phase is necessary to be able to jointly calibrate scans observed at different times, without bracketing them with additional phase calibration scans.

5.6 Subsystem Function: Archive Ingestion

As an observation proceeds, the Correlator Back End and the Metadata Capturer write the raw data products in a staging storage area. The system ingests these files into the Science Archive, which is assumed to be collocated with the data processing cluster, in the Science Operations and Data Center. As the data arrives, a process collects the metadata and stores it in the Archive Metadata Database. The raw data files are stored into the permanent Science Data storage.

In order to avoid unnecessary copy operations, the reception staging area in the Science Operations and Data Center will be the cache space used by the processing cluster. As the data is saved directly in the format required by the data reduction software, it is immediately available for the calibration pipeline.

Pulsar timing and search data is assumed to be ingested into the archive as well. Storing the data into disk-based VLBI data systems such as Mark 6, to be sent to other observatories is not a requirement, although spare capacity will be allocated in the CBE switch (see Section 5.15) to allow the connection of the necessary equipment in the future, if necessary.

5.7 Subsystem Function: Metadata Capture

During the execution of an observation, several software modules send metadata structures to the Metadata Capture module, which parses them and writes the Measurement Set sub-tables (the MS Main table is written by the Correlator Back-end). For the non-visibility data products, the Metadata Capture module fulfills a similar function.

5.8 Subsystem Function: Online Calibration

As the observation is executed, the Calibration module receives messages from the Metadata Capture module announcing the availability of calibration scans to be processed. The Calibration module reads the corresponding scans from the raw data files and computes the calibration tables, which are sent back to the Metadata Capture component to be integrated into the metadata tables (or alternatively they are written directly to disk by the Calibration module).

In some cases, the calibration results are loaded into the Online Subsystem components to be applied on the instrumentation. This is typically the case of Pointing and Focus models, and when the array data needs to be "phased" (all relative phase differences between the data from different antennas are corrected and the time-series summed up). These calibration models are sent to the Supervisor computers to be applied in the instrumentation.

The system needs to compute and apply the attenuator gains in the antenna. This will be performed by the science script. Functions will be implemented in the script API for measuring the power after pointing the antenna to the desired target and setting other aspects of the instrumentation (frequency tuning, etc.). This measurement is used to calculate the desired attenuator setting for optimizing the output power. Currently the antenna electronics doesn't incorporate square law detectors (like ALMA), but the received power can be measured from the IRD ADC.

5.9 Subsystem Function: Online Quality Assurance

The system provides interfaces for the Science Staff and Operator to assess the quality of the observation while the observation is being executed. This is sometimes referred as QA0. (Regarding observatory quality control functions, QA0 refers to the quality control during the observation; QA1 refers to the quality control of the telescope, including antenna position models, pointing models, etc.; and QA2 refers to quality control functions after post-processing).

5.10 Subsystem Function: Operator Shift Logging

This function refers to the annotation of several events that happen during an operation shift. The system provides an application that automatically annotates events such as sub-array creation or destruction, observation beginning and end, etc., and allows the operator to annotate other relevant information.

5.11 Subsystem Function: Monitoring

This function refers to the acquisition, transmission, archival and use of monitoring data from the telescope hardware and software systems.

5.12 Subsystem Function: Error Handling

This function refers to the detection and handling of system faults and errors.

5.13 Subsystem Function: Security

This function refers to the implementation of strategies and countermeasures to deal with internal and external security threats.

5.14 Interfaces and Dataflows

Figure 2 (next page) and Figure 3 (p. 22) show the main connections and data flows involved in controlling the execution of an observation and monitoring the state of the array, respectively. The diagram shows the control and monitoring modules distributed in the layers specified in the M&C architecture (see [AD06]). These are:

- **Operation Layer:** Operational interfaces for the Array Operations Staff (e.g., Operator and Astronomer-on-Duty).
- **Execution Layer:** Coordinates the execution of observations.
- **Supervisory Control Layer:** Includes real-time computers for controlling the antenna electronics, LO reference, timing and synchronization equipment, and the CSP.
- **Hardware Controller Layer:** Includes the Hardware Controller Boards. These are interface boards between the supervisory computers and electronic equipment, translating Ethernet commands to low-level electronic protocols such as GPIO, SPI, I2C, etc.

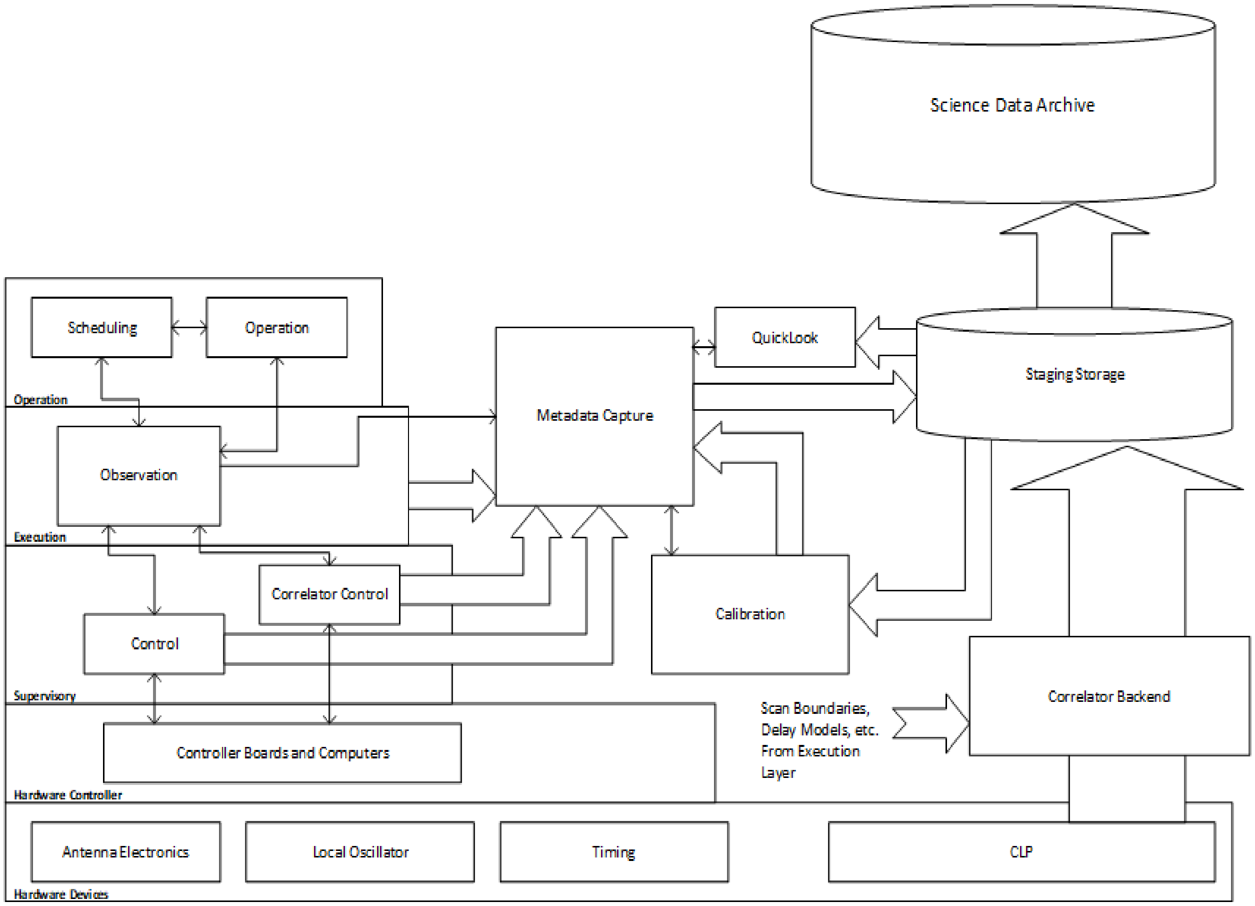

Figure 2 - Online subsystem observation-related data flows. Data streams are shown as wide arrows, while single line arrows denote control paths.

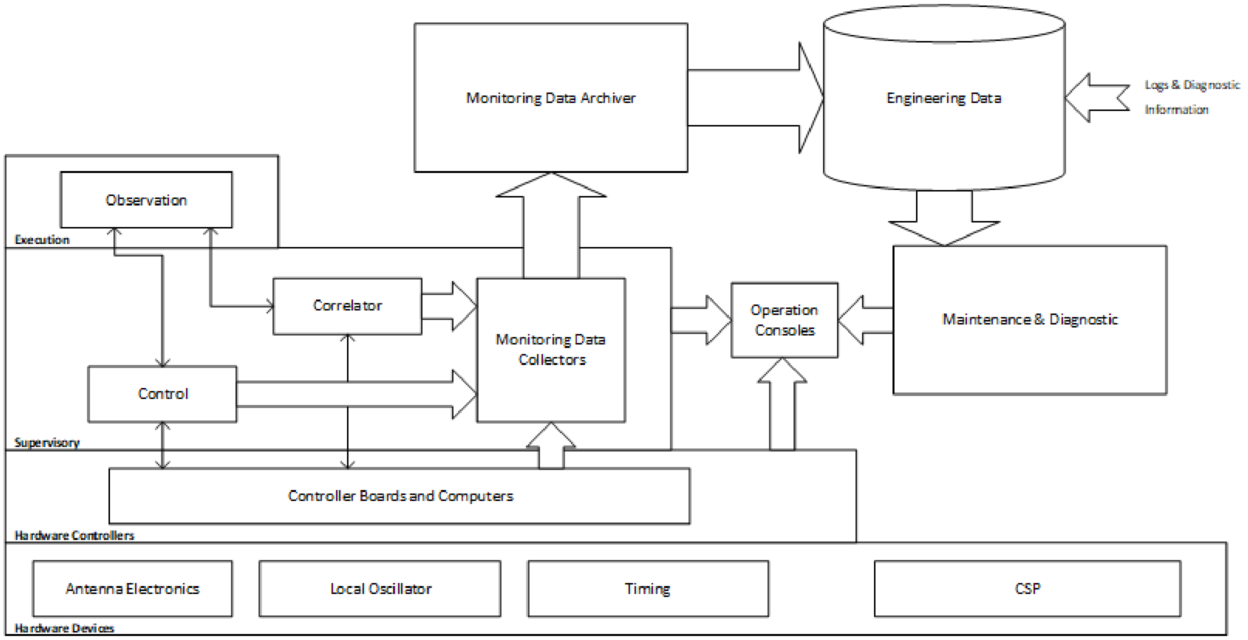

Figure 3 - Online subsystem monitoring-related data flows. Data streams are shown as wide arrows, while single line arrows denote control paths.

5.14.1 Scheduling Module Interfaces

- **Sub-Array Management UI:** This is the user interface that the Operator uses to create, destroy and query sub-arrays, query the list of observable Scheduling Blocks, select them to be executed, and monitor the status of the execution. This interface aggregates information about the status of the antennas, allowing the Operator to visualize the sets of antennas that can be used in specific sub-arrays.
- **Scheduling Block Interface:** This is a software interface that allows one to query the Scheduling Blocks stored in the Proposal Database.
- **Weather Prediction Interface:** This software interface allows one to execute the weather forecasting process, for different time scales. This process uses the weather historic database.

5.14.2 Operation Module Interfaces

- **Alarm & Notifications UI:** This user interface allows the operator to see a list of alarms and other notifications, ordered by priority. The error handling system incorporates logic to detect the root cause of problems, filtering secondary alarms. This interface allows the operator to filter alarms using several criteria, provides documentation about the recovery procedures to follow for known problems, and allows one to acknowledge alarms once they have been dealt with.
- **Logging UI:** This user interface allows the operator to see the system logs, and filter by log priority and other criteria.
- **Hardware Status UI:** This user interface allows the operator to visualize the status of the telescope systems. It allows the operator to browse this information at several levels of detail, through tabular and graphical representations (mimics).
- **Console Interfaces:** This is a low-level console interface that allows the operator to connect to different components of software and hardware (software components, supervisory servers, and controller cards) and issue monitoring and control commands.

5.14.3 Observation Module Interfaces

- **Sub-array Management Interface:** Interface used to create and destroy sub-arrays, and query information about the existing sub-arrays.
- **Observation Interface:** Interface that allows one to start observations and abort observations, passing a Scheduling Block identifier and a time in UT.
- **Observing Mode Interface:** This interface provides a science-oriented API to execute an observation. It is used by standard observation scripts and commissioning scripts.

5.14.4 Antenna Supervisory Control Module Interfaces

- **Antenna Pointing Interface:** Software interface to command the antenna to point in the supported coordinate systems. Includes support for Ephemeris objects.
- **Frequency Tuning Interface:** Software interface to command the antenna electronics to tune to specified observing frequencies. This involves selecting the observing bands in the antenna. (Besides this, frequency tuning involves tuning the LO offset in the LO reference system).
- **Attenuator Control Interface:** Software interface to retrieve and set the value of the IRD attenuator gains.

5.14.5 LO and Timing Supervisory Control Module Interface

- **Local Oscillator Reference Interface:** This interface allows one to control the Local Oscillator reference hardware. An important function in this interface is setting the offsets for each antenna. These offsets carry both the fine tuning and an LO offset that is afterwards removed in the CSP (see LO Offsetting function).
- **Time Synchronization Interface:** This interface allows one to control the timing synchronization hardware.

5.14.6 CSP Supervisory Control Module Interfaces

- **Correlator and Beam-former Interface:** This interface allows one to control the Correlator and Beam-former hardware.
- **Pulsar Engine Interface:** This interface allows one to control the Pulsar Engine hardware.

5.14.7 Metadata Capture Module Interfaces

- **Observation Metadata Interface:** The functions in this interface are called during an observation to transmit observation metadata. These are typically when the observation starts/ends, when a scan start/ends, etc.
- **Pointing Data Interface:** This interface is used to receive the Pointing Data streams from the antennas.
- **Weather Data Interface:** This interface is used to receive the Weather Data streams from the Weather Stations.
- **WVR Data Interface:** This interface is used to receive data from the Water Vapor Radiometer data and store it as an auxiliary table along with the MS tables.
- **Configuration Data Interface:** This interface is used to send the system configuration (e.g., pointing model, focus model, antenna positions, etc.) at the beginning of an observation, and any changes that may have occurred while observing.
- **Flagging Interface:** This interface is used to report flagging commands, usually as a result of a hardware fault during an observation.

- **Calibration Data Interface:** This interface is used to send calibration tables from the Calibration module, to be stored in the MS or as auxiliary tables.

5.14.8 Configuration Module Interfaces

- **Configuration Interface:** This interface is used to set and retrieve configuration data structures.
- **Configuration History Interface:** This interface allows the query of the history of changes applied in the configuration data structures.

5.14.9 Quick-Look Module Interfaces

- **Quick-Look UI:** This user interface allows one to visualize QA0 (observation quality assurance) data as the observation proceeds. This usually consists of plots showing results from the Calibration module.

5.14.10 Calibration Module Interfaces

- **Telescope Calibration Interface UI:** This interface consists of several utilities that read data collected from performing telescope calibration observations, and derive calibration models. The interface allows one to commit the calibration models into the configuration database once they have been validated.
- **Online Calibration Interface:** This is a software interface through which the Calibration Module is notified of the existence of new calibration scans to be processed. The Calibration Module computes calibration tables and sends them to the Metadata Capture Module.

5.14.11 Monitoring Module Interfaces

- **Monitoring Control Interface:** This interface allows one to control what monitor points will be sampled, along with their frequency.

5.14.12 Data Streams

- **Pointing Data Stream:** This data stream contains pointing data from the Antenna Supervisor modules to the Metadata Capture module. The pointing data includes the antenna encoder pointing data.
  Assuming that the need to record pointing data comes from the single dish and OTFM observing modes, a sample rate can be derived from the antenna scan speed requirements and the primary beam width. The scan speed has not been specified yet for ngVLA, but as a reference, the highest RF frequency is equivalent to Band 3 in ALMA, where a sampling rate of 48 ms is used. This results roughly in a 90 KB/s aggregated data rate for all antennas.
- **Delay Model Data Stream:** This data stream contains the delay compensation data, which includes the geometrical delay for all antennas combined with the cable delay and other instrumentation delays. It also includes the addition of a "causality delay" to make the delay compensation for all antennas positive. It is sent from the Observation module to CSP module.
  Although the details may change, a message containing the delay models for all antennas for some interval of time in the future (typically 1 minute) is sent in advance to allow the CSP to receive it and process it before the corresponding antenna data packet arrive.
- **Water Vapor Radiometer Data Stream:** This data stream contains Water Vapor Radiometer data, from the WVR devices to the Calibration module.
- **Weather Data Stream:** This data stream contains weather data, from the weather stations to the Observation Module (weather data is necessary for calculating delays models), to the Metadata Capture Module and to the Calibration Module.

- **Monitoring Data Stream:** This data stream contains monitoring data, from the Monitoring Data Collectors to the Monitoring Archiver. The number of monitor points per antenna for both ALMA and VLA is around 5000. Assuming each monitor value is transmitted in 16 bytes (8 bytes timestamp, 4 bytes identifier, and 4 bytes value) and an average monitoring rate of 1 Hz, this data stream should be around 80 KB/s per antenna and 21 MB/s for all 263 antennas. This data stream will require approximately 0.6 PB of storage per year. In practice, a few years will be kept in an online database, and older data will be stored offline. This is a small fraction of the storage needed for the science data.
- **Alarm and Notification Stream:** This data stream consist of alarms and notifications, broadcasted from the Supervisor modules.
- **CSP Output Data Stream:** This data stream contains low-level science data products, from the CSP to the Correlator Back End.
- **Science Staging Data Stream:** This data stream contains the final science data products, from the Correlator Back End to a disk staging area.
- **Science Archive Ingestion Data Stream:** This data stream transmits the staged science data from the disk staging area in the Array Operations and Repair Center to the Science Operations and Data Center.

5.14.13 Controller Board Interfaces

- **Generic M&C Interface:** This is an interface that all controller boards implement, and includes basic status information, the firmware version, a reset command, and other general operations.
- **Control Interface:** This interface implements the control points for each type of controller board. The execution of control commands can be performed at a specified timestamp in the future, or as soon as possible. A protocol will be defined for issuing commands and receiving return codes.
- **Monitoring Interface:** This interface allows the Data Collectors to collect monitoring data from each one of its connected controller boards.

5.15 Physical Architecture

The Online Subsystem physical architecture is shown in Figure 4 (next page). The logical modules, interfaces, and data streams listed in the previous sections are mapped into the computing, storage and network equipment presented in the diagram.

Antenna data is streamed from the antennas to the CSP. This data stream contains voltage data for two polarizations, channelized in 200 MHz spectral windows to form a total bandwidth of 20 GHz. This data is encoded in 4 bits, resulting in 320 Gbps per antenna (2 polarizations × 40 GHz Nyquist rate × 4 bits/sample).

In terms of the nature of the connections, the antennas are divided in three classes: the directly connected antennas; the antennas connected through commercial dark fiber; and the antennas that will be connected through the public packet switching network. The antenna data is sent in 4 x 100 GbE interfaces to the CSP. The CSP is composed of three Tridents, each capable of processing 2 x100 GbE streams. The software controls how these 200 Gbps data streams are distributed to each Trident.

The CSP produces visibility and pulsar data, which is sent to Correlator Back-end. This system produces the final raw data product in a Staging Storage area, from where it is read by the online computers to be sent to the Science Operations and Data Center through the ISP network. The Correlator Back-end also provides computing resources for the Calibration Module software.

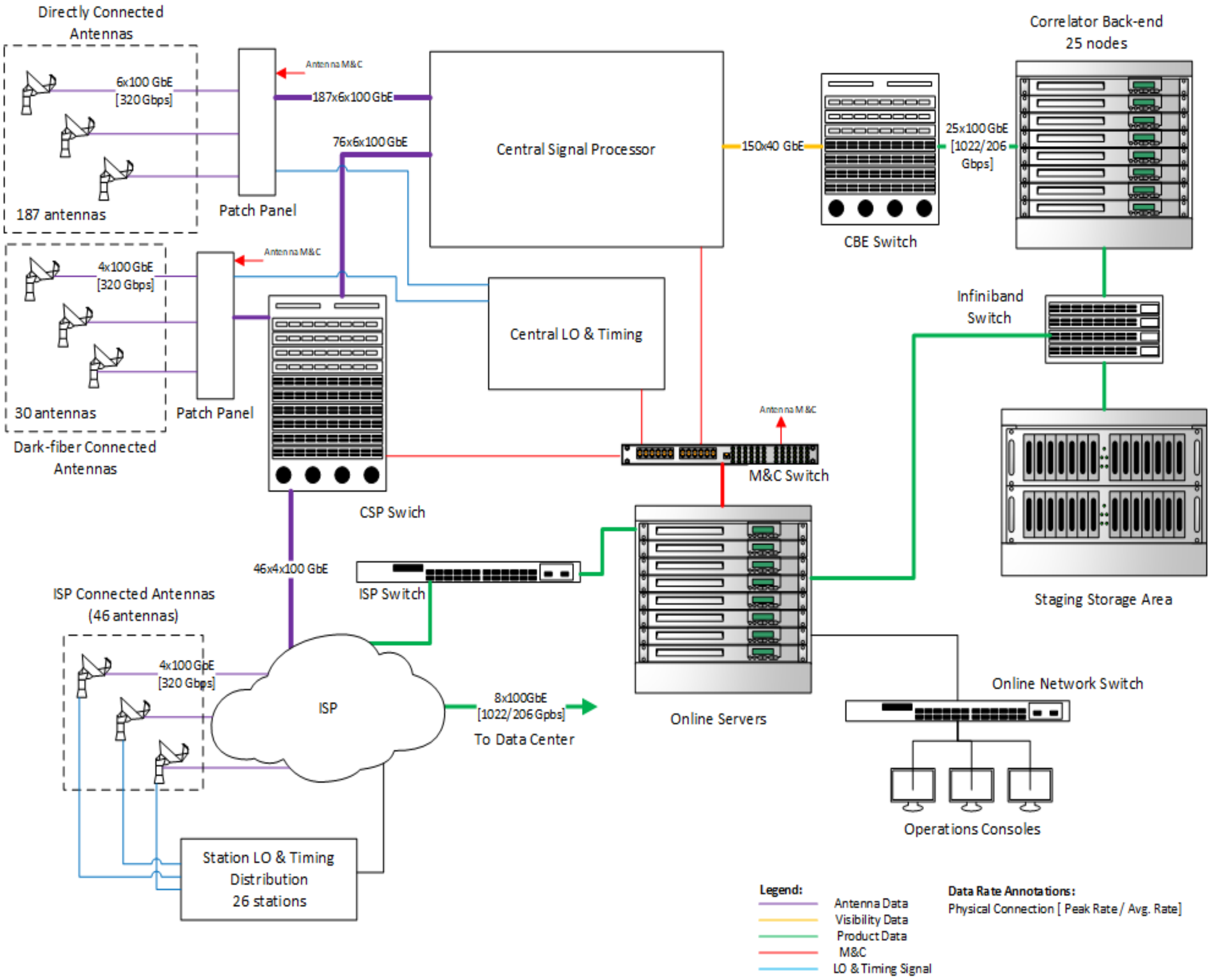

Figure 4 - Online subsystem physical architecture.

The Online Servers rack includes 13 servers, defined in Table 2. It may be possible to decrease the number of physical machines by virtualization.

Name	Quantity	Purpose
General Services Server	1	General purpose. Build and deployment, troubleshooting, etc.
CSP Supervisory Computer	1	Real time computer, controls the CSP.
LO and Timing Supervisory Computer	1	Real time computer, controls the LO reference and timing systems.
Calibration Support Server	1	Executes the Calibration module software.
Observation Execution Server	1	Executes the Observation Executor. It may require real-time capabilities.
Observation Support Server	1	Executes other components belonging to the Observation module, e.g., the Delay Calculation component, Metadata Capturer, etc.
Data Transmission Server	1	Executes software that reads the raw data and transmit it to the Science Operations and Data Center.
Operations Console	3	Operator consoles.
Scheduling Database Server	1	Executes a DBMS system containing the scheduling blocks.
Engineering Database Server	1	Executes a DBMS system for the Engineering Database.
Configuration Database Server	1	Executes a DBMS system for the Configuration Database.
Monitoring Server	1	Executes software from the Monitoring module.

Table 2 - Online servers.

6 Offline Subsystem

The Offline Subsystem is responsible for all the telescope functions that occur after the observation raw data has been stored in the Science Archive. These functions include the generation of derived data products (images, catalogs), support for quality assurance activities, and interfaces for searching, visualizing, and retrieving raw data and derived products.

Table 3 shows the telescope expected data rates. These figures are large enough to make it difficult for end users to have in-house computational resources to calibrate and image the raw data locally. The ngVLA system will be integrated with the Science Ready Data Products (SRDP) project, a general NRAO initiative originated with the goal of creating the necessary infrastructure and interfaces for the generation and distribution of science-ready calibrated datasets, images, and catalogs, for current and future NRAO telescopes. The ngVLA Offline subsystem will be integrated into the SRDP architecture, extending it to supply additional computational and storage resources, and data analysis pipelines. The size of the expected datasets makes necessary the development of large-scale parallelization algorithms for calibration and imaging.

This section discusses only aspects of the SRDP project that affect the ngVLA architecture. See [AD05] for additional information about SRDP.

Science Case	Use Fraction	Vis Per Hour	Data Rate	Storage Rate
KSG1 Driving Cont Band 6 eg Taurus disk	8%	73.19 GVis	0.081 GB/s	0.21 PB/Month
KSG1 Driving Cont Band 4 eg Taurus disk	4%	216.28 GVis	0.240 GB/s	0.63 PB/Month
KSG2 Driving Line Band 5 eg Sgr B2(N)	4%	97241.83 GVis	108.046 GB/s	284.14 PB/Month
KSG2 Driving Line Band 4 eg Sgr B2(N)	1%	72129.85 GVis	80.144 GB/s	210.76 PB/Month
KSG2 Driving Line Band 3 eg Sgr B2(N)	1%	119342.01 GVis	132.602 GB/s	348.72 PB/Month
KSG3 Driving Line Band 5 eg COSMOS	4%	5985.35 GVis	6.650 GB/s	17.49 PB/Month
KSG3 Driving Line Band 4 eg COSMOS	1%	2996.82 GVis	3.330 GB/s	8.76 PB/Month
KSG3 Driving Line Band 3 eg COSMOS	1%	3030.45 GVis	3.367 GB/s	8.85 PB/Month
KSG3 Driving Line Band 6 eg Spiderweb galaxy	2%	11.16 GVis	0.012 GB/s	0.03 PB/Month
KSG3 Driving Line Band 5 eg Spiderweb galaxy	1%	11.16 GVis	0.012 GB/s	0.03 PB/Month
KSG3 Driving Line Band 4 eg Spiderweb galaxy	1%	5.58 GVis	0.006 GB/s	0.02 PB/Month
KSG3 Driving Line Band 6 eg Virgo Cluster	7%	3232.05 GVis	3.591 GB/s	9.44 PB/Month
KSG3 Driving Line Band 1 eg M81 Group	10%	149.48 GVis	0.166 GB/s	0.44 PB/Month
KSG3 Driving Line Band 1 eg M81 Group	12%	4.66 GVis	0.005 GB/s	0.01 PB/Month
KSG5 Driving Cont Band 1 OTF Find LIGO event	7%	7347.53 GVis	8.164 GB/s	21.47 PB/Month
KSG5 Driving Cont Band 4 OTF Find LISA event	7%	1090.82 GVis	1.212 GB/s	3.19 PB/Month
KSG5+4 Driving Cont Band 2 OTF Find BHs+PossiblePulsars	8%	2034.17 GVis	2.260 GB/s	5.94 PB/Month
KSG5 Driving Cont Band 3 Gw170817@200Mpc	23%	4.18 GVis	0.005 GB/s	0.01 PB/Month
	Average:	6714.55 GVis	7.461 GB/s	19.62 PB/Month

Table 3 - Expected data rates, from science use cases. It is assumed that full polarization is required, and visibilities are stored in half precision (2 bytes per number, 4 bytes per visibility).

The Offline Subsystem is divided into the following modules:

- **Science Archive:** The Science Archive is the final repository for all data products. It consists of a metadata database, a file storage system, and services constructed around them. This system will be provided by the SRDP project, although it will be extended to accommodate additional ngVLA requirements. The ngVLA project will require the addition of storage nodes and changes in the metadata database.
- **Observatory Interfaces:** This module consists of Web applications that provide retrieval interfaces and services constructed around the data stored in the Science Archive. This is also provided by the SRDP project, although extensions will be required. This includes visualization and processing interfaces, in order to examine large images without having to download them first. It also includes Virtual Observatory compatible interfaces.
- **Data Processing Management:** This module includes several components required to manage and integrate the Data Processing Pipelines with the Science Archive. These infrastructural components will also be provided by the SRDP project. They include components to copy the data from the Archive file storage system to processing space, manage caches, implement workflows, track pipeline executions, etc.
- **Processing Resources:** This module encapsulates high performance storage and processing hardware and associated management software. These resources include a local cluster in the Data Processing Center and external resources such as the Open Science Grid, Amazon Web Services, and XSEDE. Although the exact distribution of local and remote resources is still TBD, the system will probably operate in a hybrid mode where local, dedicated, high-utilization resources handle the reduction of raw data to generate standard data products up to certain capacity, while external resources are used to process jobs that overflow this level.
- **Data Processing Pipelines:** This module includes pipelines for calibration, imaging, and catalog generation. The infrastructure will be based on the CASA package and the ALMA/EVLA Pipeline. The ngVLA project will require changes in algorithms and extensive improvements on parallelization and performance.
- **Quality Assurance Interfaces:** This module includes interfaces to assess the quality of the data products generated by the Data Processing Pipelines, and support quality assurance operations. These will be provided by the SRDP project, although they may require extensions for ngVLA.
- **Data Analysis Tools:** These are tools necessary to analyze ngVLA datasets and images, which will be provided as a package for users to download and install in their machines. This will be based on CASA, although extensions and modifications may be necessary to fulfill ngVLA requirements.

The following sections describe the subsystem functions.

6.1 Subsystem Function: Raw Data Calibration

The raw data acquired for Proposals that adhere to the ngVLA standard observing modes will be automatically calibrated. Each execution of a Scheduling Block is defined as an Execution Block. Execution Blocks are calibrated independently. A set of Execution Blocks that are processed together by the imaging pipeline is known as a Program. Calibration is triggered by the Data Processing Management module. If the raw data is still the temporary storage (where it is being left after data transmission), the calibration pipeline is executed immediately; if not, it is retrieved from the Science Archive first.

The calibration pipeline is executed in the processing cluster, removing instrument-dependent effects from the data, producing a set of calibration tables and a calibrated dataset along with QA reports and other auxiliary products. These are analyzed by QA staff by means of the Quality Assurance Interfaces, and if accepted the data products are ingested into the Archive. The calibrated dataset is not stored into the Archive, but the calibration tables are. When necessary, the calibration tables can be applied on the raw data to create a *restored* calibrated dataset.

If the quality of the dataset is deemed insufficient, it can be re-processed (typically after additional flagging) or re-observed. A problem tracking system, part of the Observatory Interfaces module, allows one to communicate and follow-up on problems with the PI.

For visibility data, the pipeline performs calibrations for phase, amplitude, band-pass, flux, polarization, and other standard calibrations. Pulsar data is usually calibrated for flux and polarization.

Currently, the algorithms used to solve the calibration tables are not parallel. Depending on timing results, it may be necessary to investigate parallelization possibilities. In lower frequencies, self-calibration may be needed in order to achieve high dynamic range. This ties calibration performance requirements with the imaging.

6.2 Subsystem Function: Imaging Product Generation

Once all the datasets that belong to the same Imaging Program have been successfully calibrated and accepted by QA, the Data Processing Management module triggers the imaging pipeline. In order to avoid unnecessary copies, the recently calibrated dataset is kept in a cache, from where it is read by the imaging processes.

The imaging process runs the CLEAN algorithm. This algorithm iteratively executes gridding, degridding, FFT and deconvolution operations in a *major cycle*, with the embedded deconvolution operation being iterative itself, executing several *minor cycles* during each major cycle. The gridding operation includes also a convolution operation of the corrected visibility data with a *support function or kernel*, which incorporates corrections for wide field distortions and other effects.

Given the scale of the ngVLA datasets, extensive parallelization and performance improvements on the data reduction software will be needed. This will require algorithm research during the Design & Development phase of the project.

The production of imaging data products follows a workflow similar to calibration. Once the pipeline has finished, the images are examined for quality assurance, and if accepted, they are ingested into the Archive.

6.3 Subsystem Function: Source Catalog Product Generation

Images that have been successfully accepted by QA and ingested into the Archive can be used as input of processes that generate catalogs. These processes scan the images to find sources and extract several properties from them. These tables go through the same quality assurance processes as other data products and are ingested into the Archive.

6.4 Subsystem Function: Project Tracking

This function refers to the provision of interfaces for PIs to track the status of their accepted proposals. This includes the capability of tracking the status of scheduled observations and quality assurance processes for the raw and derived data products.

6.5 Subsystem Function: Data Product Discovery

This function encompass the capability of interacting with the Archive to search its contents, visualize results, save queries or results, and select data products for later use. These capabilities will be provided by the SRDP project.

6.6 Subsystem Function: Data Product Retrieval

This function refers to the capability of retrieving the metadata and binary files of a data product. Several delivery mechanisms will be provided: a password protected URL, a download manager, staging into NRAO processing cluster storage, or external services such as AWS. These capabilities will be provided by the SRDP project.

6.7 Subsystem Function: Data Product On-Demand Post-Processing

Once one or more data products have been selected using the Archive interfaces, they can be used as inputs in data processing pipelines. The system allows one to customize the execution parameters. This capability will be provided by the SRDP project.

6.8 Subsystem Function: Data Product Analysis

The CASA package currently provides tools to enable analysis of resulting science products. The ngVLA project will extend this package with ngVLA specific capabilities.

6.9 Subsystem Function: Security

This function refers to the implementation of strategies and countermeasures to deal with internal and external security threats.

6.10 Interfaces and Data Flows

Figure 5 (next page) shows the offline flow of data from the Array Operations Center to Observatory Interfaces and external processing centers.

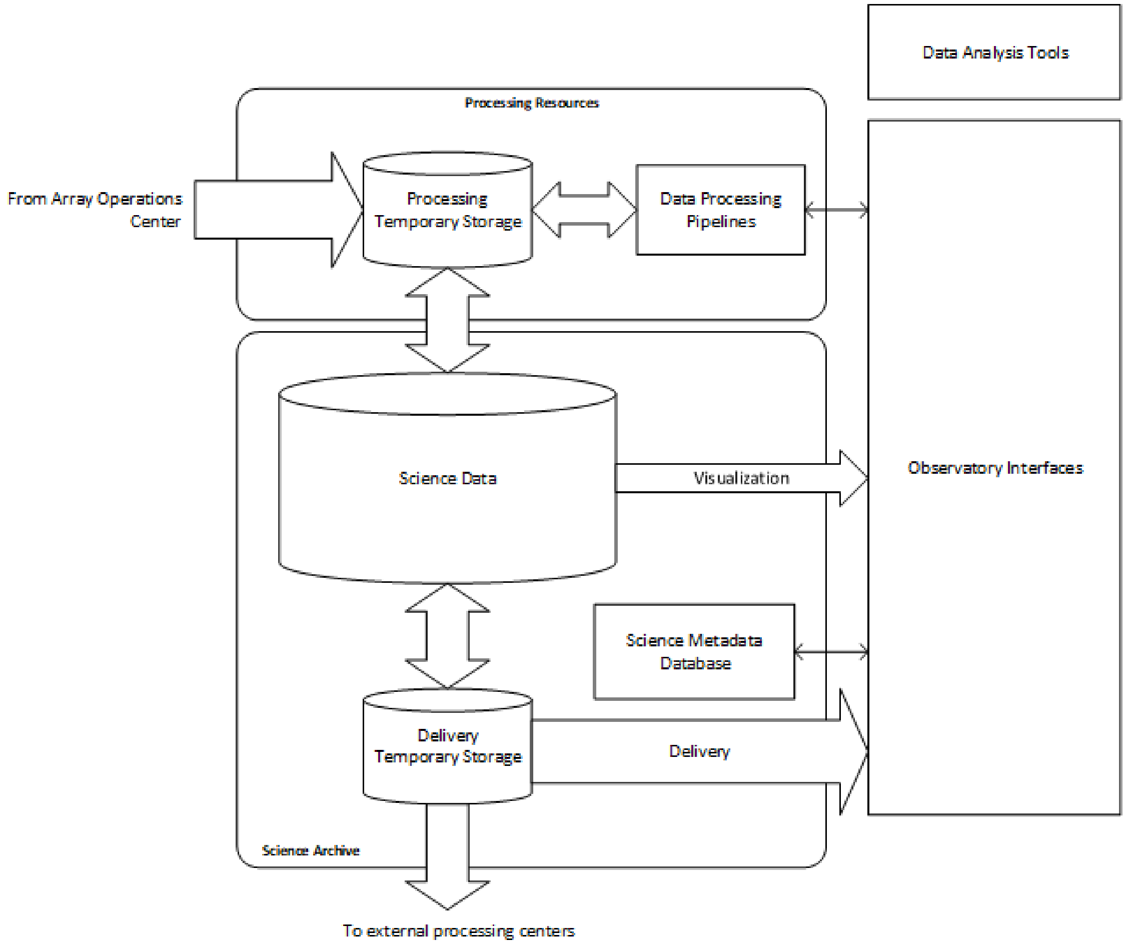

Figure 5 - Offline data flows. Data streams are shown as wide arrows, while single line arrows denote control and query paths.

6.10.1 Science Archive Interfaces

- **Archive Metadata Database Interface:** This interface allows one to submit queries and retrieve results from the Archive Metadata Interface.
- **Archive Storage System Interface:** This interface provides access to the Archive Storage System. It allows one to search and list files, retrieve file metadata, download files, upload files and delete files.
- **User Database Interface:** This interface permits retrieval of information about user accounts.
- **Calibrator Database Interface:** This interface provides access to the Calibrator Database. Lists of calibration sources are maintained in this database, along with their properties and time series.

6.10.2 Observatory Interfaces

- **Archive Web UI:** This Web application allows users to search the Archive contents.
- **Workspace UI:** This Web application lets users access their *workspace*, where they save references to data products and queries, submit post-processing requests, and manage their execution.
- **Helpdesk Interface:** This interface provides access to NRAO Helpdesk, which allows users to submit and follow up problems related with observations and post-processing jobs.
- **Virtual Observatory Interface:** This interface provides VO-compatible services.

6.10.3 Data Processing Management Interfaces

- **Data Processing Management UI:** This interface permits management of automated data processing. It uses the Product and Job interfaces.
- **Product Interface:** This REST interface permits management of product information. It provides methods to create new products, retrieve product information, update the product status, manage dependent products, etc.
- **Job Interface:** This REST interface permits management of product-generation jobs.

6.10.4 Processing Resource Interfaces

- **Processing Resource Management Interface:** This interface permits management of processing resources. It provides a common interface to heterogeneous processing resources (local clusters, external resources such as Amazon Web Services, etc.).

6.10.5 Data Processing Pipeline Interfaces

- **Pipeline Processing Request Interface:** This interface is based on an XML file (PPR.xml) that is parsed by the pipeline. The file contains data elements such as the project information, the PI name, the Measurement Set, etc., and processing instructions about the pipeline tasks that will be executed, along with their parameters.
- **Pipeline Scripting Interface:** The Pipeline Scripting Interface is an API based on the pipeline tasks, which are constructed using CASA tasking infrastructure.
- **Weblog Interface:** The Weblog is one of the products generated by the Pipeline. It consists of a set of generated HTML pages and plots describing several aspects of the generated data products and associated processing.

6.11 Quality Assurance Interfaces

- **Quality Assurance UI:** This Web application allows QA personnel to access job execution information, Weblogs, etc. The application permits introducing comments and performing actions to follow up QA decisions (accept data products, reject data products, submit for re-processing, submit Helpdesk issues, etc.). This UI uses the Quality Assurance Interface.
- **Quality Assurance Interface:** This REST interface permits management of QA information.

6.12 Data Analysis Package Interfaces

- **Plot MS UI:** This is a plotting application for the Measurement Set format.
- **Image Viewer UI:** This application allows viewing and manipulating astronomical images.
- **Data Analysis Task Interface:** This permits defining parameters and executing data analysis tasks.

6.13 Data Streams

- **Archive/Processing Data Stream:** This is the data stream between the temporary storage resources in the Processing Resources and the Archive Storage System. Both local and external processing resources need to be considered.
- **Delivery Data Stream:** This is the data stream between the Archive Storage System and the user, when downloading data products. In practice, the data product will be staged in a delivery storage area, from where it will be streamed to the end user. The staged data product can be the result of

delivery processing tasks such averaging, mosaicking, image cutout, restoration of calibrated MS from calibration tables, etc.

- **Visualization Data Stream:** NRAO is working in an advanced visualization system that will be provided as a browser application, reducing the need to retrieve full image files for viewing. This will originate a data stream between the Archive and the user browser.

6.14 Physical Architecture

Figure 6 shows the Offline Subsystem physical architecture.

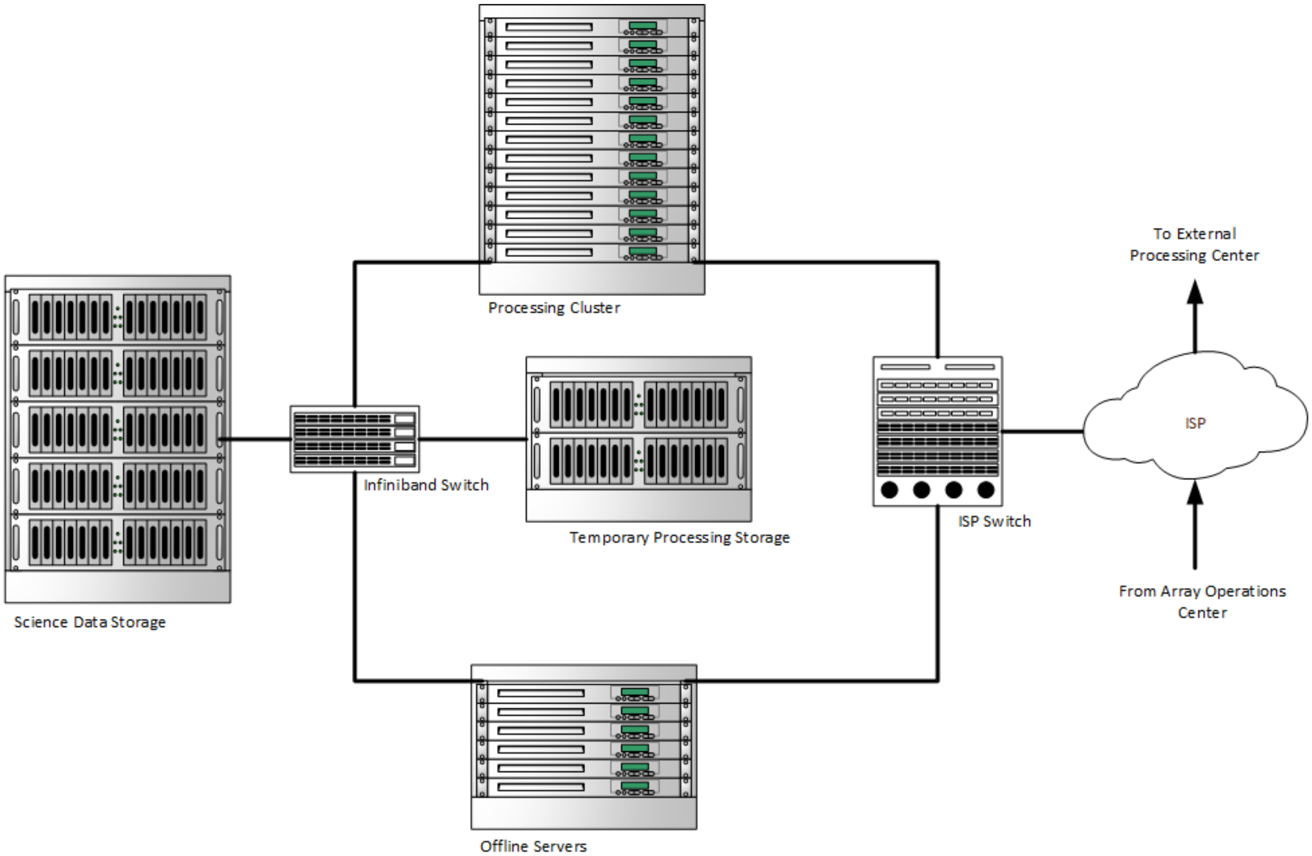

Figure 6 - Offline subsystem physical architecture.

The Offline Servers rack includes nine machines, described in Table 4.

Server Name	Quantity	Description
Archive Services Server	1	Executes services from Science Archive module.
Observatory Interfaces Server	2	Deploys Observatory Interfaces services.
Quality Assurance Support Server	1	Executes QA support software.
General Services Server	1	Supports software maintenance and development.
Metadata Database Server	1	Runs the Metadata Database.
Processing Management Server	1	Executes software from Processing Management module.
Proposal Management Database Server	1	Runs Proposal Management Database.
Proposal Management Server	1	Executes Proposal Management software.

Table 4 - Offline subsystem servers.

6.14.1 Data Reduction Performance Requirements

The main requirement for data processing is the ability of the system to maintain the throughput while reducing (calibrating and imaging) the raw data stream from observations. It is not possible to accumulate observations waiting for processing resources, beyond normal variations. The system is sized to sustain the average throughput, while latency is considered a “floating” variable. This section documents preliminary analyses and measurements for estimating the required system performance.

The computing load for one gridding cycle is

$$CL_{gridding} = \frac{\text{Visibilities}}{\text{second}} \cdot \frac{\text{FLOPs}}{\text{Visibility}}$$

The number of FLOPs per visibility depends on the size of the convolution kernel used during gridding, which varies depending on the algorithm used (W-projection, A-projection, etc.)

It is assumed that the total number of operations involved in reducing a dataset is proportional to $CL_{gridding}$:

$$CL = k \cdot CL_{gridding}$$

The factor k includes the repetition gridding /degridding in each major cycle, the deconvolution operations performed in each minor cycle, the effect of multi-term and multi-scale algorithms, and the operations performed in calibration tasks.

We define the observed performance of a single core (or more generally a “processor,” as the system is likely to include accelerator hardware) as:

$$CP^{(obs,sc)} = \frac{\text{FLOPs}}{\text{Execution time for a single core}}$$

measured by executing test runs with simulated datasets and applying scaling laws.

Similarly, the observed performance on a parallel system is

$$CP^{(obs,par)} = \frac{\text{FLOPs}}{\text{Exec. time for parallel system}}$$

$$= CP^{(obs,sc)} \cdot \frac{\text{Execution time for single core}}{\text{Execution time for parallel system}}$$

$$= CP^{(obs,sc)} \cdot Sp(N_c)$$

The speedup $Sp(N_c)$ can be modeled from Amdahl's Law, and measured by observing how the execution time scales as the number of cores increases.

The observed core performance differs from the nominal core performance by the core efficiency ε_c factor:

$$CP^{(obs,sc)} = \varepsilon_c CP^{(sc)}$$

The nominal performance is the peak figure reported in the processor specifications. Achieving it would require keeping the processor busy only doing floating point operations, using SIMD instructions.

Matching the required computing load with the parallel system throughput:

$$CL = \varepsilon_c CP^{(sc)} Sp(N_c)$$

Multiplying and dividing by the number of core N_c, and defining the parallelization efficiency as $\varepsilon_p = Sp(N_c)/N_c$, the ratio of the real speedup and the ideal speedup, we obtain

$$CL = CP^{(sc)} N_c \varepsilon_c \varepsilon_p$$

The single-core performance times the number of cores is the quantity that is usually reported as the "system performance" of an HPC system. Naming this quantity $SP = CP^{(sc)} N_c$, the required system performance can be expressed as

$$SP = \frac{kCL_{gridding}}{\varepsilon_c \varepsilon_p}$$

The parameters used on this estimation are shown in Table 5, the algorithmic requirements for each science case can be found in Table 6), and Table 7 (next page) shows the estimated system performance for each science case.

Parameter	Value	Notes
Base Convolution Kernel Size	7	Prolate spheroidal. Linear size, i.e. the kernel matrix dimensions are 7x7.
Primary Beam Kernel Size	8	Linear size. When A-proj is used, this kernel is convolved with the base kernel to produce the final convolution kernel.
W-projection Kernel Size	100	Linear size. When W-proj is used, this kernel is convolved with the base kernel to produce the final convolution kernel.
Ionospheric Corr. Kernel Size	8	Linear size. When this alg. is used, this kernel is convolved with the base kernel to produce the final convolution kernel.
Gridding to Full Expansion Factor	25.0	Base case for estimating k.
Multi-Scale Factor	1.2	Multiplies k when multi-scale is used.
Multi-Term Factor	3.0	Multiplies k when multi-term is used.

Parameter	Value	Notes
Base FLOPs/Visibility	3900	Preliminary measurement for base case. It is scaled with the combined kernel size when other algorithms are used.
Core Efficiency	0.1	ε_c. From preliminary measurements.
Parallelization Efficiency	0.9	ε_p. Educated guess. This parameter needs to be measured (there is no currently software available that allows this measurement).
Arithmetic Intensity for Gridding	170 FLOPs/byte	Preliminary measurements for gridding.

Table 5 - Parameters used for estimating the required computing performance for reducing ngVLA observations.

Science Case	w-proj	PB correction	Ionospheric Correction	Multi-scale	Multi-term
KSG1 Driving Cont. Band 6 e.g. Taurus disk	No	No	No	Yes	Yes
KSG1 Driving Cont. Band 4 e.g. Taurus disk	No	No	No	Yes	No
KSG2 Driving Line Band 5 e.g. Sgr B2(N)	No	Yes	No	Yes	No
KSG2 Driving Line Band 4 e.g. Sgr B2(N)	No	Yes	No	Yes	No
KSG2 Driving Line Band 3 e.g. Sgr B2(N)	No	Yes	No	Yes	No
KSG3 Driving Line Band 5 e.g. COSMOS	No	Yes	No	Yes	No
KSG3 Driving Line Band 4 e.g. COSMOS	No	Yes	No	Yes	No
KSG3 Driving Line Band 3 e.g. COSMOS	No	Yes	No	Yes	No
KSG3 Driving Line Band 6 e.g. Spiderweb galaxy	No	Yes	No	Yes	No
KSG3 Driving Line Band 5 e.g. Spiderweb galaxy	No	Yes	No	Yes	No
KSG3 Driving Line Band 4 e.g. Spiderweb galaxy	No	Yes	No	Yes	No
KSG3 Driving Line Band 6 e.g. Virgo Cluster	No	Yes	No	Yes	No
KSG3 Driving Line Band 1 e.g. M81 Group	Yes	Yes	Yes	Yes	No
KSG3 Driving Line Band 1 e.g. M81 Group	No	Yes	Yes	Yes	No
KSG5 Driving Cont. Band 1 OTF Find LIGO event	Yes	Yes	Yes	Yes	Yes
KSG5 Driving Cont. Band 4 OTF Find LISA event	No	Yes	No	Yes	Yes
KSG5+4 Driving Cont. Band 2 OTF Find BHs + Possible Pulsars	Yes	Yes	No	Yes	Yes
KSG5 Driving Cont. Band 3 Gw170817@200Mpc	No	No	No	Yes	Yes

Table 6 - Algorithmic requirements for each science case.

Science Case	Use Fraction	Required System Perf.
KSG1 Driving Cont. Band 6 e.g. Taurus disk	8%	0.079 PFLOPs/s
KSG1 Driving Cont. Band 4 e.g. Taurus disk	4%	0.078 PFLOPs/s
KSG2 Driving Line Band 5 e.g. Sgr. B2(N)	4%	412.782 PFLOPs/s
KSG2 Driving Line Band 4 e.g. Sgr. B2(N)	1%	306.184 PFLOPs/s
KSG2 Driving Line Band 3 e.g. Sgr. B2(N)	1%	506.595 PFLOPs/s
KSG3 Driving Line Band 5 e.g. COSMOS	4%	25.407 PFLOPs/s
KSG3 Driving Line Band 4 e.g. COSMOS	1%	12.721 PFLOPs/s
KSG3 Driving Line Band 3 e.g. COSMOS	1%	12.864 PFLOPs/s
KSG3 Driving Line Band 6 e.g. Spiderweb galaxy	2%	0.047 PFLOPs/s
KSG3 Driving Line Band 5 e.g. Spiderweb galaxy	1%	0.047 PFLOPs/s
KSG3 Driving Line Band 4 e.g. Spiderweb galaxy	1%	0.024 PFLOPs/s
KSG3 Driving Line Band 6 e.g. Virgo Cluster	7%	13.720 PFLOPs/s
KSG3 Driving Line Band 1 e.g. M81 Group	10%	24.130 PFLOPs/s
KSG3 Driving Line Band 1 e.g. M81 Group	12%	0.055 PFLOPs/s
KSG5 Driving Cont. Band 1 OTF Find LIGO event	7%	3558.202 PFLOPs/s
KSG5 Driving Cont. Band 4 OTF Find LISA event	7%	13.891 PFLOPs/s
KSG5+4 Driving Cont. Band 2 OTF Find BHs + Possible Pulsars	8%	783.613 PFLOPs/s
KSG5 Driving Cont. Band 3 Gw170817@200Mpc	23%	0.005 PFLOPs/s
	Average:	322.907 PFLOPs/s

Table 7 - Required system performance for ngVLA data reduction.

7 Maintenance and Support Subsystem

The Maintenance and Support Subsystem is composed of the following modules:

- **Computerized Maintenance Management System (CMMS):** This COTS software package maintains a database on observatory maintenance operations. This system provides several functions aimed to effectively organize maintenance operations.
- **CMMS Integration Module:** This integrates the Engineering Database with the CMMS system.
- **Issue Tracking System:** This COTS software maintains lists of issues and helps organize activities needed to resolve them. This may be provided by issue tracking system already in use by NRAO.
- **Integrated Support Module:** This module provides a centralized interface for support personnel to gather troubleshooting information, such as logs, alarms, and monitoring data.

7.1 Subsystem Function: Preventive Maintenance Analysis

This function refers to the analysis of monitoring data and alarms to detect conditions or trends that can indicate the need to service equipment before actual problems occur. This function is allocated in the CMMS Integration Module.

7.2 Subsystem Function: Maintenance Schedule Management

This function refers to the management of the maintenance schedule. It is allocated in the CMMS system.

7.3 Subsystem Function: Inventory Management

This function refers to the management of the spare parts inventory and is allocated in the CMMS system.

7.4 Subsystem Function: Troubleshooting Support

This function provides support for troubleshooting operations. It is allocated to the Integrated Support Module. This module allows users to search, visualize and correlate data from the Engineering Database.

7.5 Subsystem Function: Security

This function refers to the implementation of strategies and countermeasures to deal with internal and external security threats.

7.6 Interfaces and Dataflows

- **CMMS Interface:** This interface allows Operations staff to access the CMMS functionality.
- **Engineering Support Interface:** This interface, provided by the Integrated Support Module, allows Engineering support staff to access data from the Engineering Database.

7.7 Physical Architecture

Table 8 lists the servers included in this subsystem.

Name	Quantity	Description
Maintenance and Support Server	1	Executes custom software developed in this subsystem.
CMMS Server	1	Executes the CMMS system.
Maintenance Issue Tracking Server	1	Executes the Maintenance Issue Tracking system.

Table 8 - Maintenance and support subsystem servers.

8 Development Support Subsystem

This subsystem includes software modules that support software development activities. These generally include system simulators, concurrent versioning systems, continuous integration systems, testing infrastructure, build and deployment infrastructure, and quality assurance software packages.

8.1 Physical Architecture

Table 9 lists the servers included in this subsystem.

Name	Quantity	Description
Continuous Integration Server	1	Executes software to manage the continuous integration process (automatic build and test system).
Software Configuration Management Server	1	Executes a software configuration management system.
Software Development Services Server	2	Execute services to support software development (issue tracking and planning, software reviews, documentation, simulators, test databases).
Software Test Server	4	Used by developers to test several versions of the system.

Table 9 - Development support servers.

9 Appendix

9.1 Abbreviations and Acronyms

Acronym	Description
ACU	Antenna Control Unit
ADC	Analog to Digital Converter
API	Application Programming Interface
AWS	Amazon Web Services
CASA	Common Astronomy Software Applications
CBF	Correlator Beamformer
CBE	Correlator Back End
CDP	Central Data Processor
CMMS	Computerized Maintenance Management System
COTS	Commercial Off-The Shelf
CSP	Central Signal Processor
DBE	Digital Back End
DBMS	Database Management System
DTS	Data Transmission System
FE	Front End System
FSP	Frequency Slice Processor
GPIO	General Purpose Input Output
IF	Intermediate Frequency
IPT	Integrated Product Team
IRD	Integrated Downconverter/Digitizer Module
IQ	In-Phase and Quadrature
ISP	Internet Service Provider
KSG	Key Science Goal
LO	Local Oscillator
LRU	Line Replaceable Unit
M&C	Monitor and Control
MPI	Message Passing Interface
MS	Measurement Set
ngVLA	Next Generation VLA
NTP	Network Time Protocol
OPT	Observation Preparation Tool
OTFM	On The Fly Mosaic
PBT	Proposal Builder Tool
PE	Pulsar Engine
PHT	Proposal Handling Tool
PPS	Pulse Per Second
PST	Proposal Submission Tool
QA	Quality Assurance
REST	Representational State Transfer
RFI	Radio Frequency Interference
SDM	Science Data Model

Acronym	Description
SKA	Square Kilometer Array
SPI	Serial Peripheral Interface
SRDP	Science-Ready Data Products
TAC	Telescope Allocation Committee
UI	User Interface
UT	Universal Time
VCC	Very Coarse Channelizer
VLA	Jansky Very Large Array
VLBI	Very Long Baseline Interferometry
VO	Virtual Observatory
WVR	Water Vapor Radiometer