Next Generation GMRT M&C System : A SKA Telescope Manager Prototype

Presenter – Jitendra Kodilkar

Australia-India Research & Development in Radio Astronomy Meeting (ARDRA)
Nov 15,2019, Pune, India

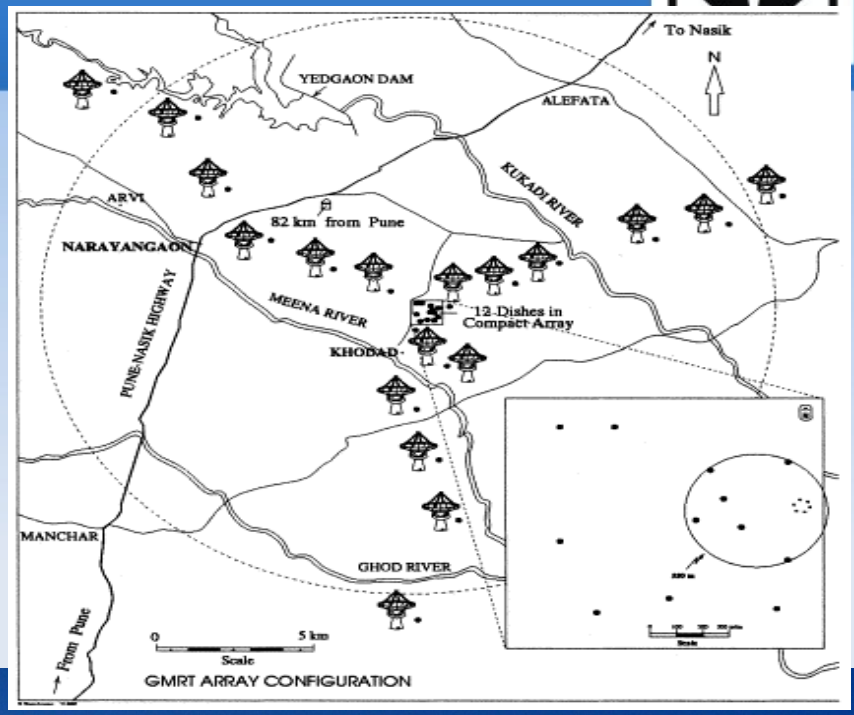
Operation Group, GMRT-NCRA members

Jitendra Kodilkar, Rajsingh Uprade, Charu Kanade,
Deepak Bhong, Sachin Sherkar, Santaji Katore, Nilesh Raskar,
S. Nayak, and Prof. Y. Gupta
+
Vikas Kumthekar, and Tata Consultancy Services Ltd., Pune Team

uGMRT (Giant Metrewave Radio Telescope)

- Located at Khodad (Lat 19.1◦ N , Long 74.05◦ E)
- Total 30 Parabolic Dishes each of 45 diameter
- 14 Antennas in Central Sq. area of ~ 1 x 1 km
  Remaining spread over ~30 km diameter
  Shortest baseline ~ 100 meter
- Observing Frequency bands are 150 to 1500 MHz with instantenous bandwidh of 100-400 MHz.
- Support Interferometry and Beam forming modes

Observations are scheduled twice in a year each of ~5.5 month duration.
More than factor of 2 oversubscribed, ~50% users are from outside India.



Outline of the talk

- Introduction : Tango Based GMRT Control (TGC) System
- TGC Software Architecture
- Software Features
- Software Development : Process and Timeline
- Compatibility with SKA-TM
- Lessons Learned from the exploratory Prototype

GMRT Upgrade : M&C System Objectives

- **Modernization of Existing M&C platform using latest tools & techonology**

- **End-to-end Radio Telescope Software solution, starting from auto execution of scheduled observing sessions to meta data generation (Observation log, M&C system log, and flag-file for flagging astronomical bad-data) .**

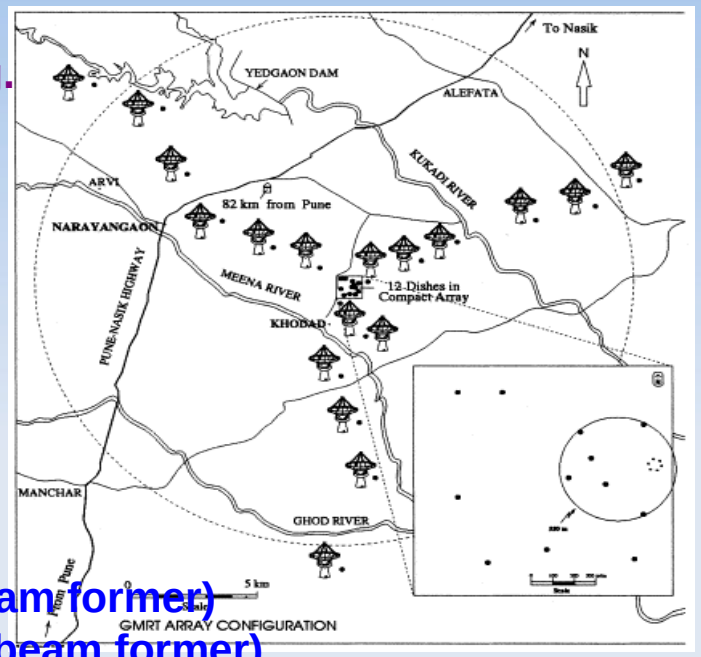
- **As a part of Indian participation in SKA, the new M&C system development aims to mitigate SKA-TM risks by :**

  - **Exploring M&C system control framework, architecture design, and technology choices which are aligned with the SKA-TM (Tango Prototype report Oct 2015)**

  - **Produce Functional components that can be reused or evolved into SKA-TM products.**

Introduction : TGC System

- Monitor and controlling of 30 distributed antennas , and signal processing Backends located at the Central building.



30 GMRT antenna comprises 30 x 6 = 180 subsystems in total :

- Two servomechanism systems –
  for antenna rotation, and feed rotation (FPS).
- RF Receiver chain, and signal conditioning systems –
  RF, IF, LO, OFS

At the GMRT Central Electronic Building ~ 46 Subsystems :

- 30 GMRT Analog Backend control Units, one per antenna
- GMRT Software Backend – 6 Subsystems (1 correlator + 2 beam former)
- GMRT Wideband Backend – 10 Subsystems (1 correlator + 4 beam former)

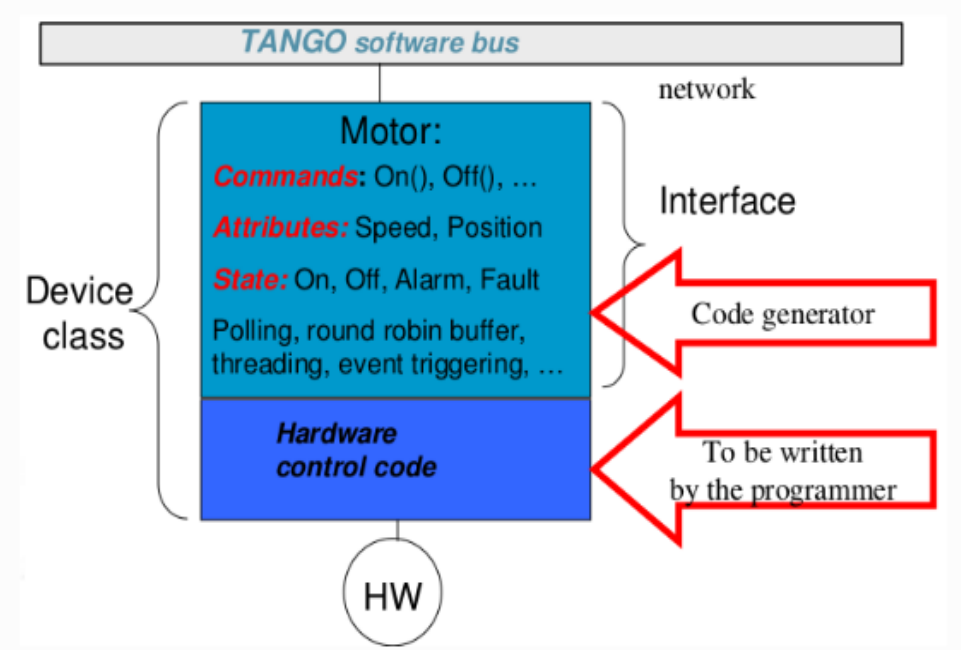
- Thus, ~226 GMRT sub-systems are being monitor and Control by the TGC system in real time.

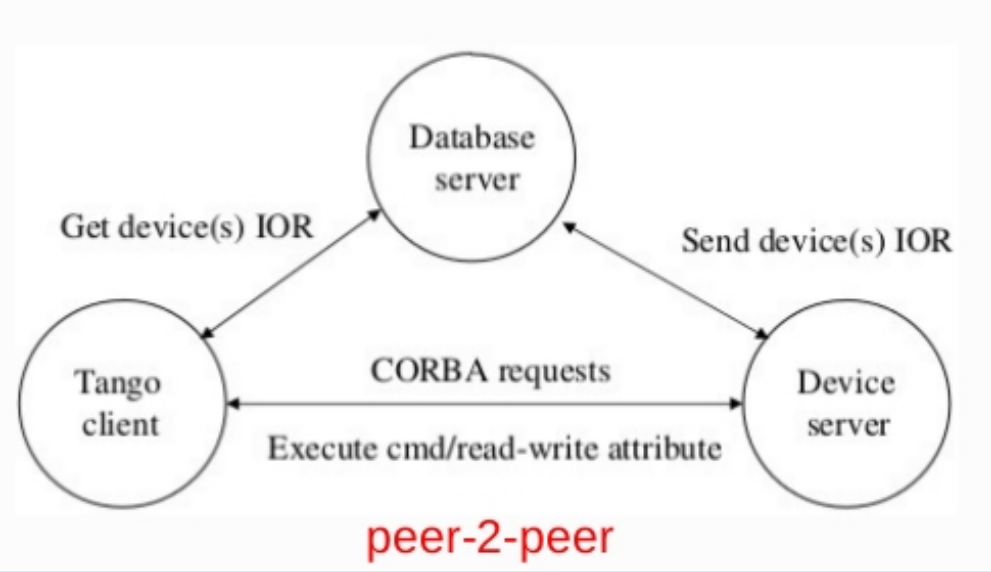
Link	Device Interface	Control Cards	Central and Local M&C machines	OS and Softwares	User Interface
Ethernet 1 Gbps	(i) Ethernet TCP/IP – 100 Mbps/ 1 Gbps (ii) RS 485, RS 422	(i) MCM – Rabbit 4300 (ii) Servo PC104 (iii) 80c535 legacy MCM cards.	(i) Dell precision Tower 5810 – 16 core, 64 GB Ram (Application & Data server) (ii) Miltech Quadcore-i3 processor	Ubuntu 16.04 / 18.04 LTS, TANGO frame work, C++, Java and Python	GUI - Taurus PyQt4, PyTango CLI – python Scripting Environment

Tango (Taco Next Generation Object) (Overview)

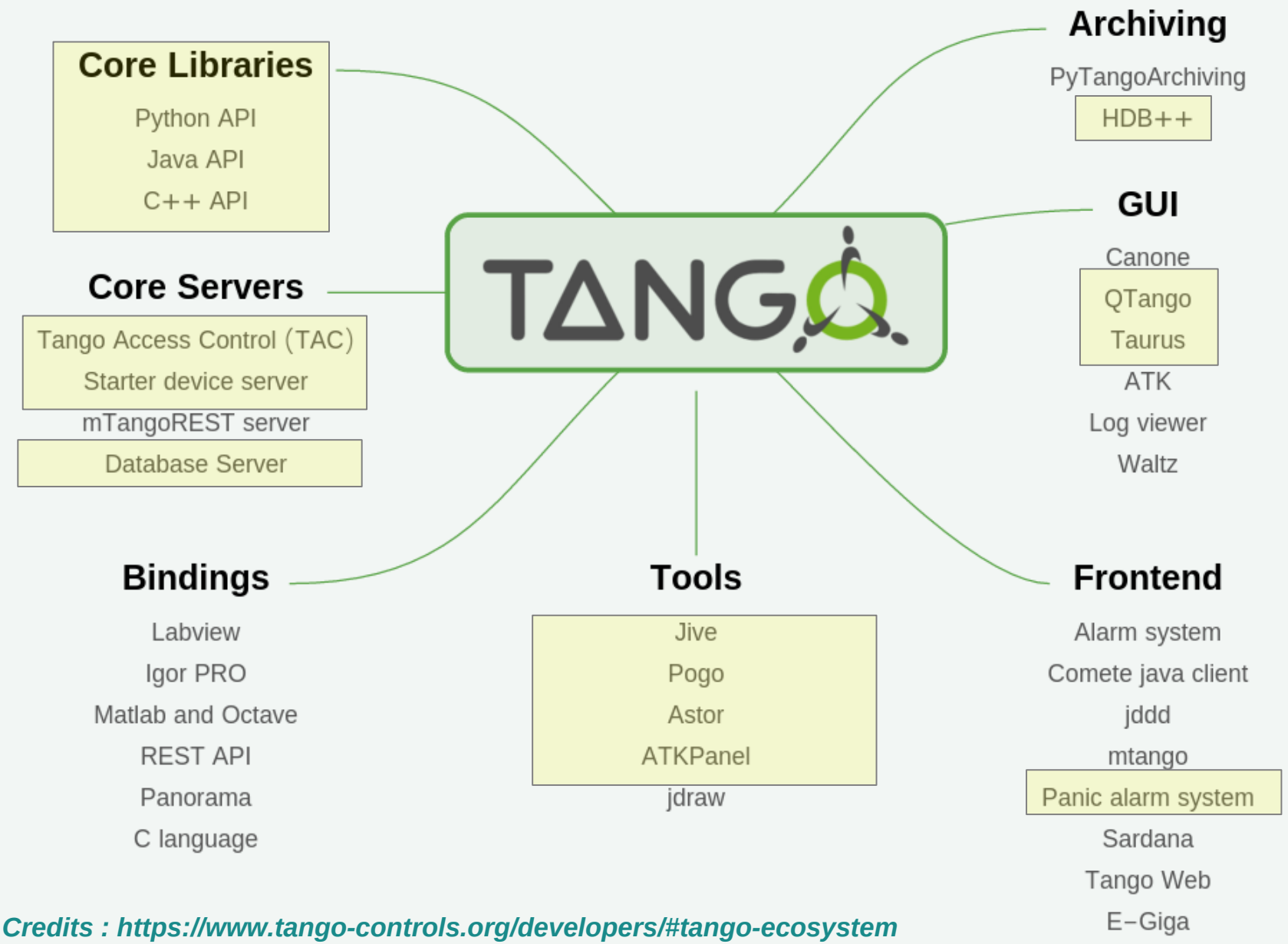
Tango is a toolkit for building Object oriented Control System, based on CORBA (developed at the ESRF). In TANGO, all control objects are treated as devices.

The Device model in TANGO implements service oriented approach by providing interface for commands/methods, attributes, data fields and properties.
Number of Tango-Device class instances are running in Tango Device Server container. All the relevant information is stored in the MySQL database by creating a TANGO database schema.
Tango Device is identified and accessed by a Unique name across the network – *e.g.*
tango://c01:10000//LMC/C01/SERVO/<current, voltage> ,
tango://cmsserver:10000/MNC/CMC/AGN1



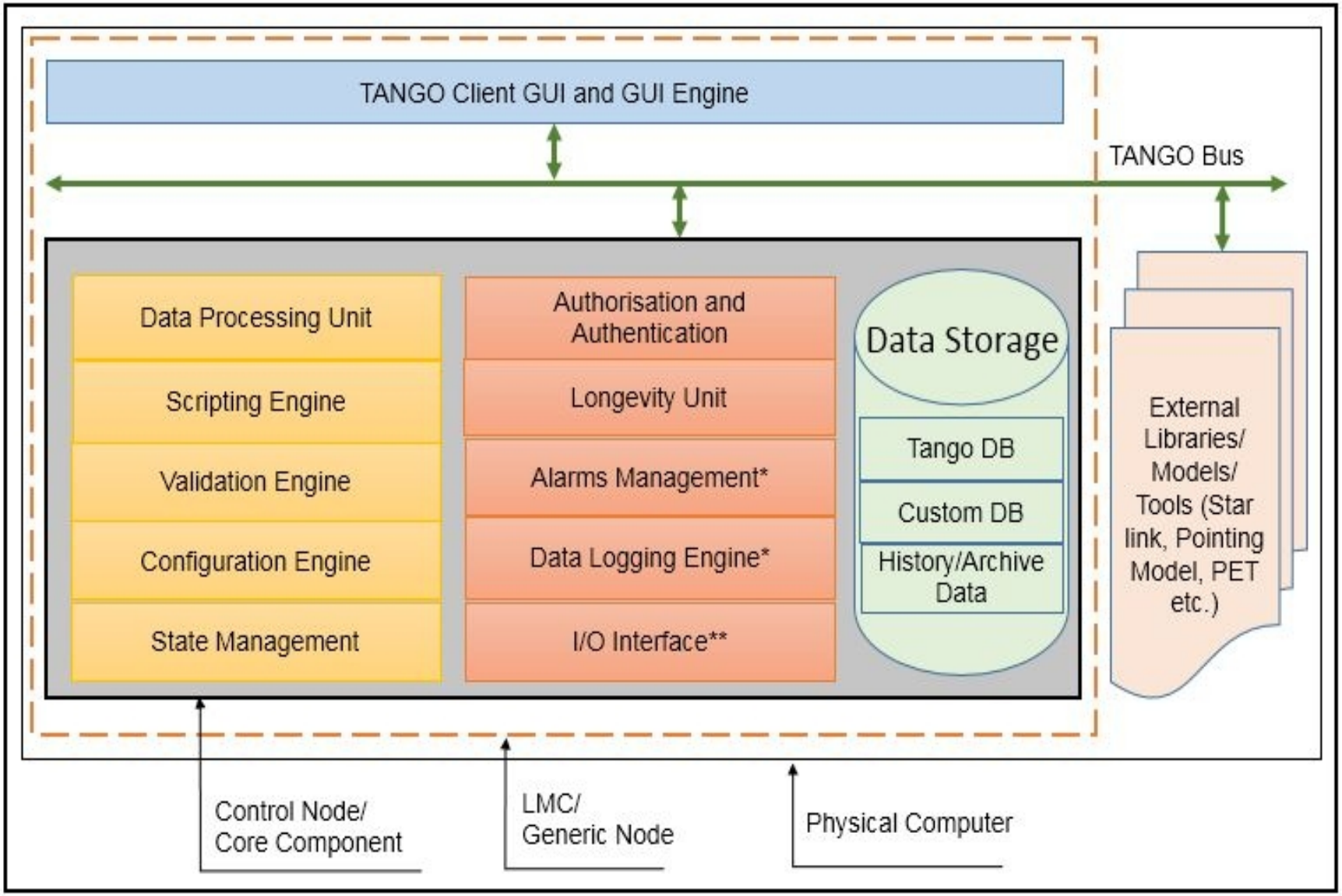


Tango (Taco Next Generation Object) (Overview – continued..)



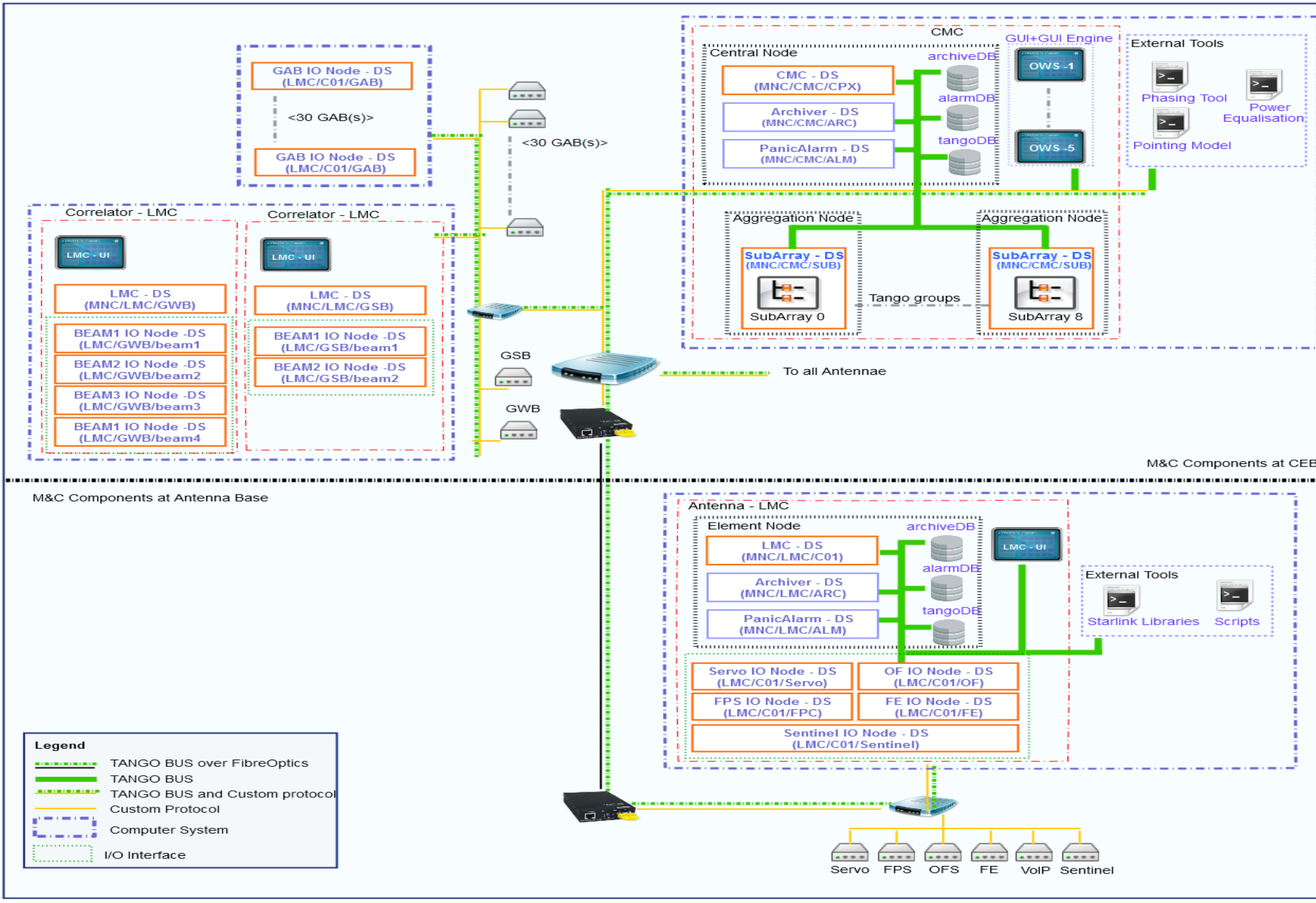
Credits : https://www.tango-controls.org/developers/#tango-ecosystem

A Generic Control-Node Architecture



Credits : https://www.tango-controls.org/developers/#tango-ecosystem

Hierarchical Composition of Control Nodes



Tango based M&C Systemimplementation

Tango Node	Tango Commands
IO (Input/Output) Subsystem C++ Device Server	init, shutdown, reset, validate state, script-launch,_pause,_resume >systemcommand
LMC (Local M&C) Antnna Level Java Device Server	Init, shutdown, suspend, validate state, script- launch, _pause, _resume. > SystemCmd, set_System_state Set maintenance, operational-Automatic/manual, SetsystemState, AuthnCmd, Logout, switchusercommand
CMC/AGN Central M&C Aggeregation Java Main Control-Nodes	Init, shutdown, suspend, validate state, script- launch, _pause, _resume. > SystemCmd, set_System_state Set maintenance, operational-Automatic/manual, SetsystemState, AuthnCmd, Logout, switchusercommand Allocate_Antenna, Deallocate_Antenna , create_subarray, destroy_subarray

- As per Tango Guidelines – Generic Service Oriented interface. Reusable node for adding future antennas and systems (where knwoledge of M&C implementation is optional).

- Hence, command-data, argument-data, and states are externilize through custom-database

Tango based M&C System implementation (Continued..)

Tango Host	Number of Tango-Devices/ Servers per Machine	Devices	Attributes
Local M&C Machin	23	LMCs , FECB, FPS, OPFSNT, Servo, GAB, HDB++ (ES, CM), SNAP (Arch, Managr, Extr), PyAlarm, Scripting, and Starter	~450 to 500 (polling 1-3 sec)
Central Application server	40	AGN(0-6), CPX, HDB++ (ES, CM), SNAP (Arch, managr, extract), Scripting , Starter, Pyalarm	~1200 (Polling 1-3 sec)

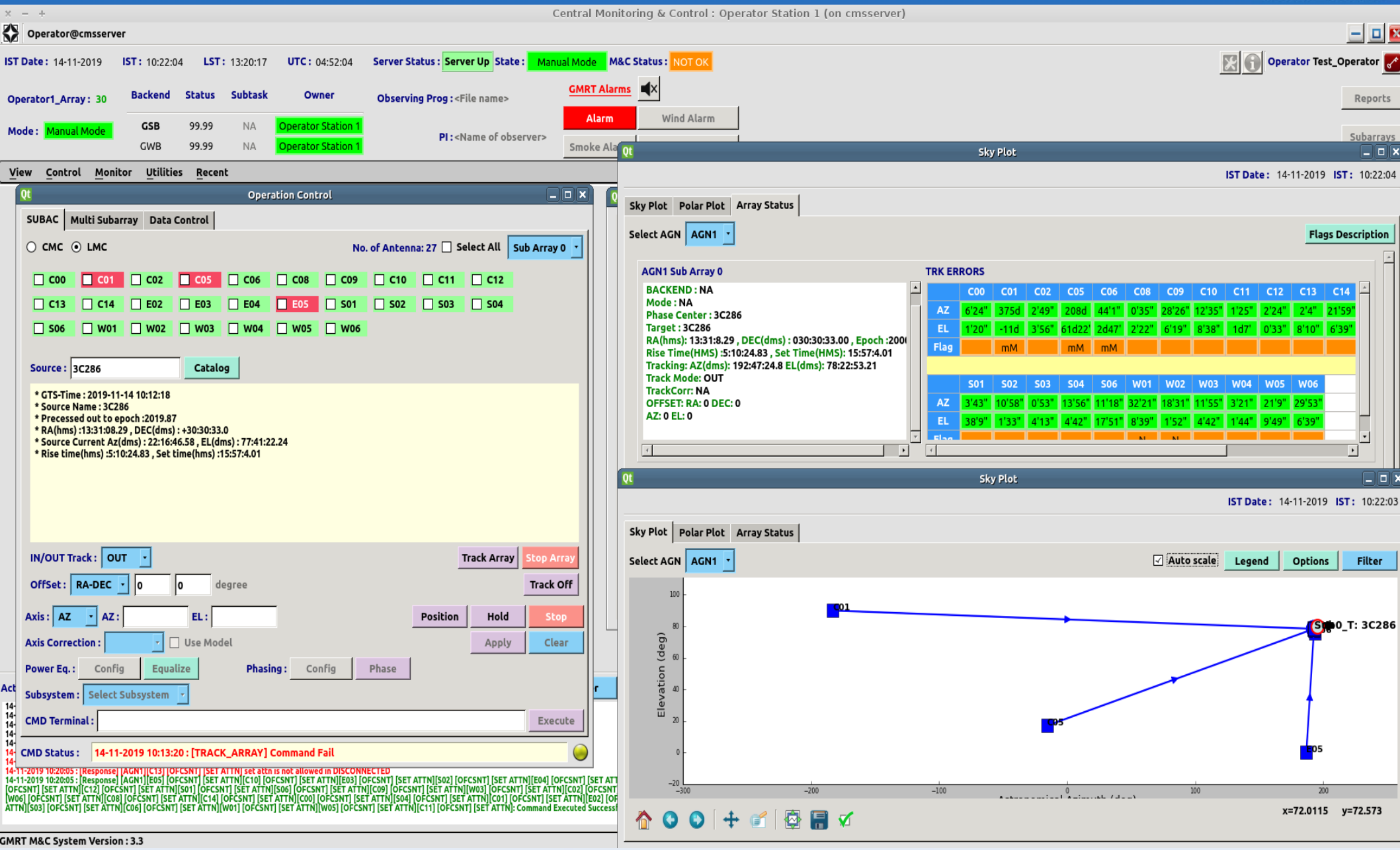
M&C Comprises, at present at 33 LMCs

- Total DS = LMC DS ~759 + CMS DS 40 = 799
- Total Attributes : ~ 17000

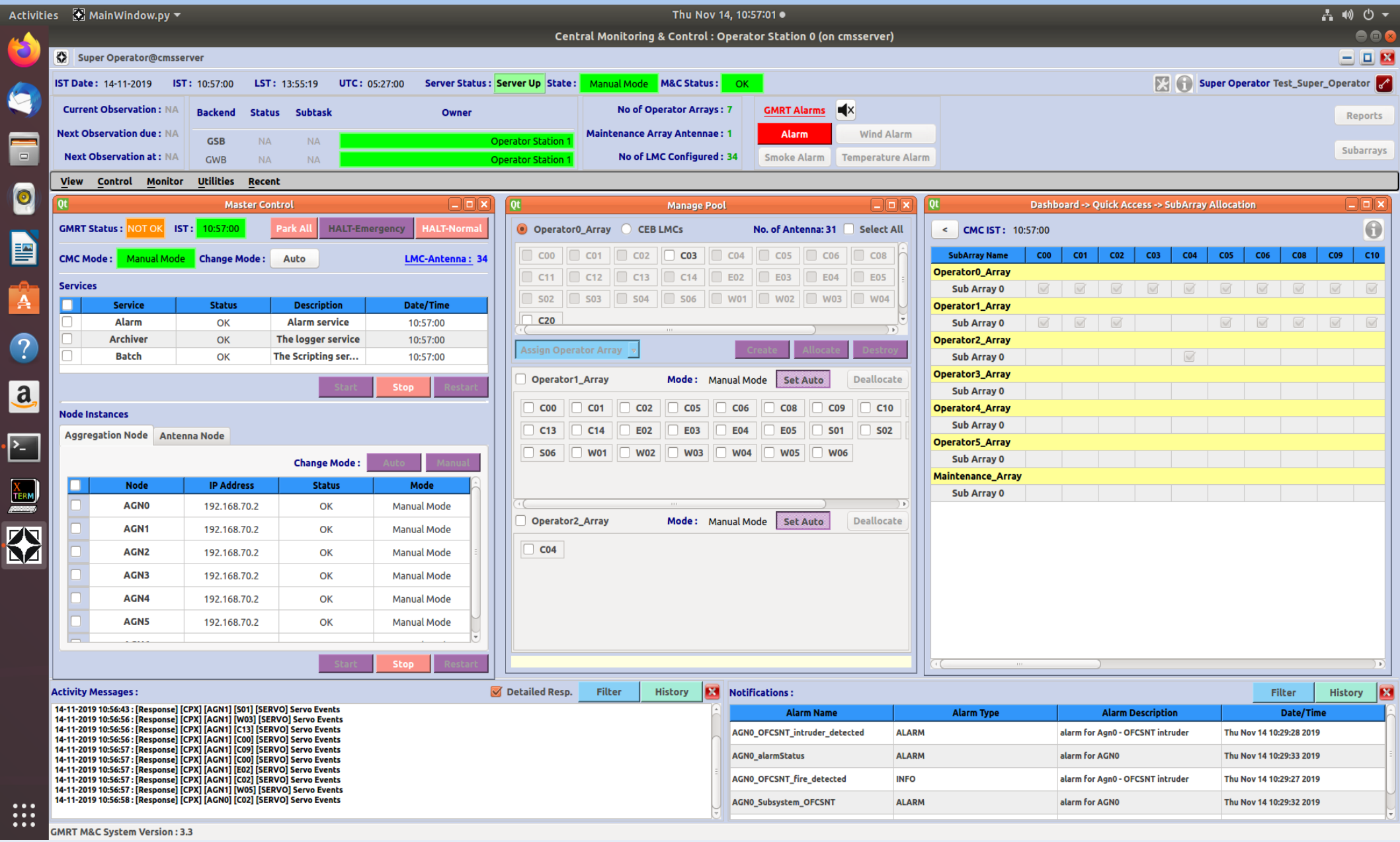
TGC M&C System Features

Phase I : A Core M&C system with GMRT antennas	Phase II : Bussiness Logic and auxilliary applications
Monitor and Control of Antenna base systems using the thick client interfaces from the Central Building	User Authentication and Role based authorisation – context based UI
Subarray Creation, Deletion and resource allocation	Multiple Sub-Arrays, group Commands
Python script environment	Antenna Tracking, pointing, and Source Catalog Management and associated utilities.
Alarm Management – PANIC tool Alarm monitoring, warning, alarm handling (acknowledgement, enable/disable/auto-action/manual action etc.)	Data backend Interface (CSP) – GMRT software backend and GMRT wide backend
Data archival HDB++	RF Signal Power equalization and Phasing
Implementation of Tango based architecture with configurable node	Automated execution of observation – Trigger Task
Risk driven scenarios – electrical power fails, system restart at I/O, LMC , AGN and CMC node level	Rich Python environment along with the M&C functionality library to conduct observation/Eng. experiment
Configuration driven GUI/Taurus development	Observation Report, Fault sheet Report, Data rendering tools (Plots, statistics associated with it).

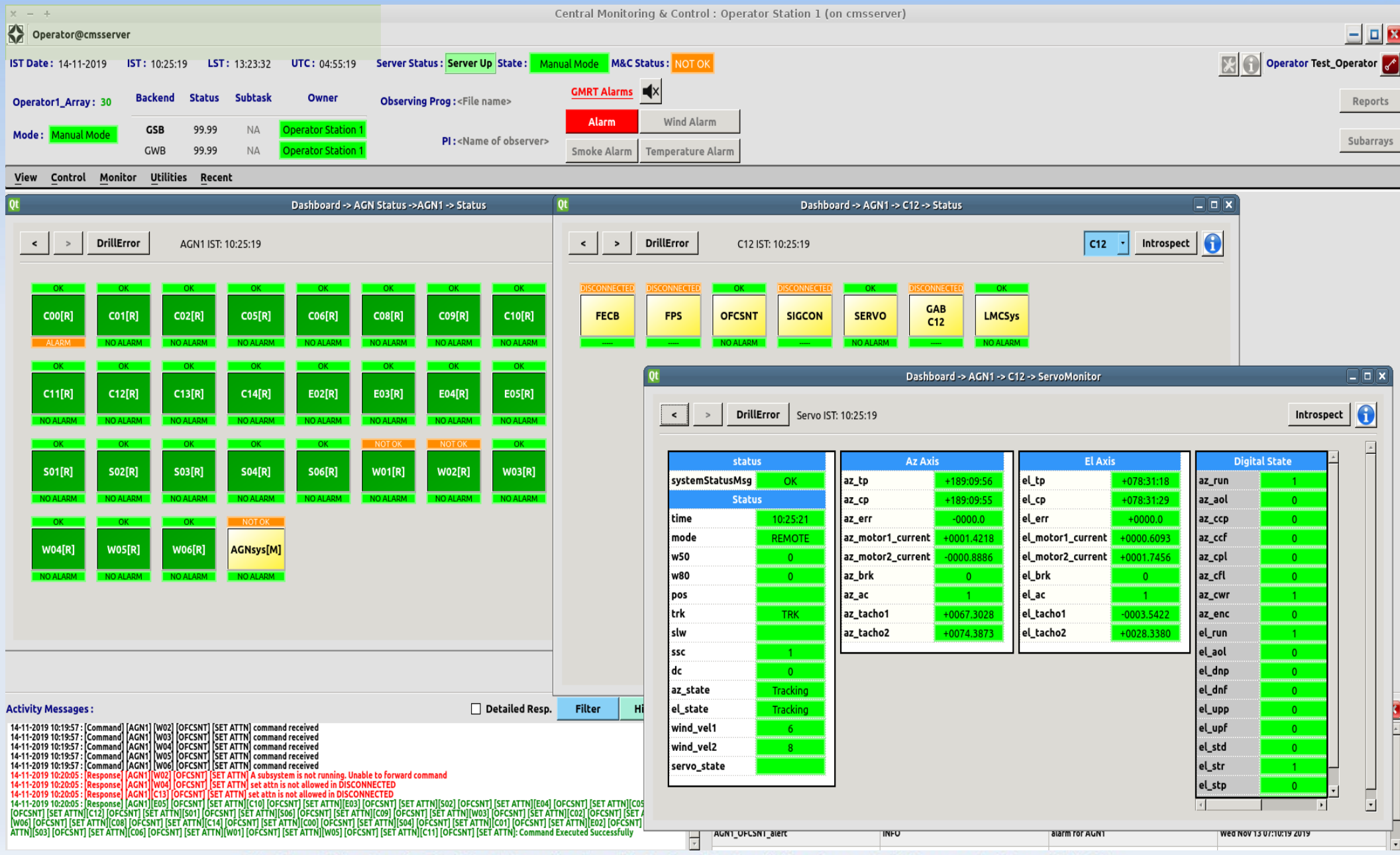
TGC Features – Sub-Array Tracking



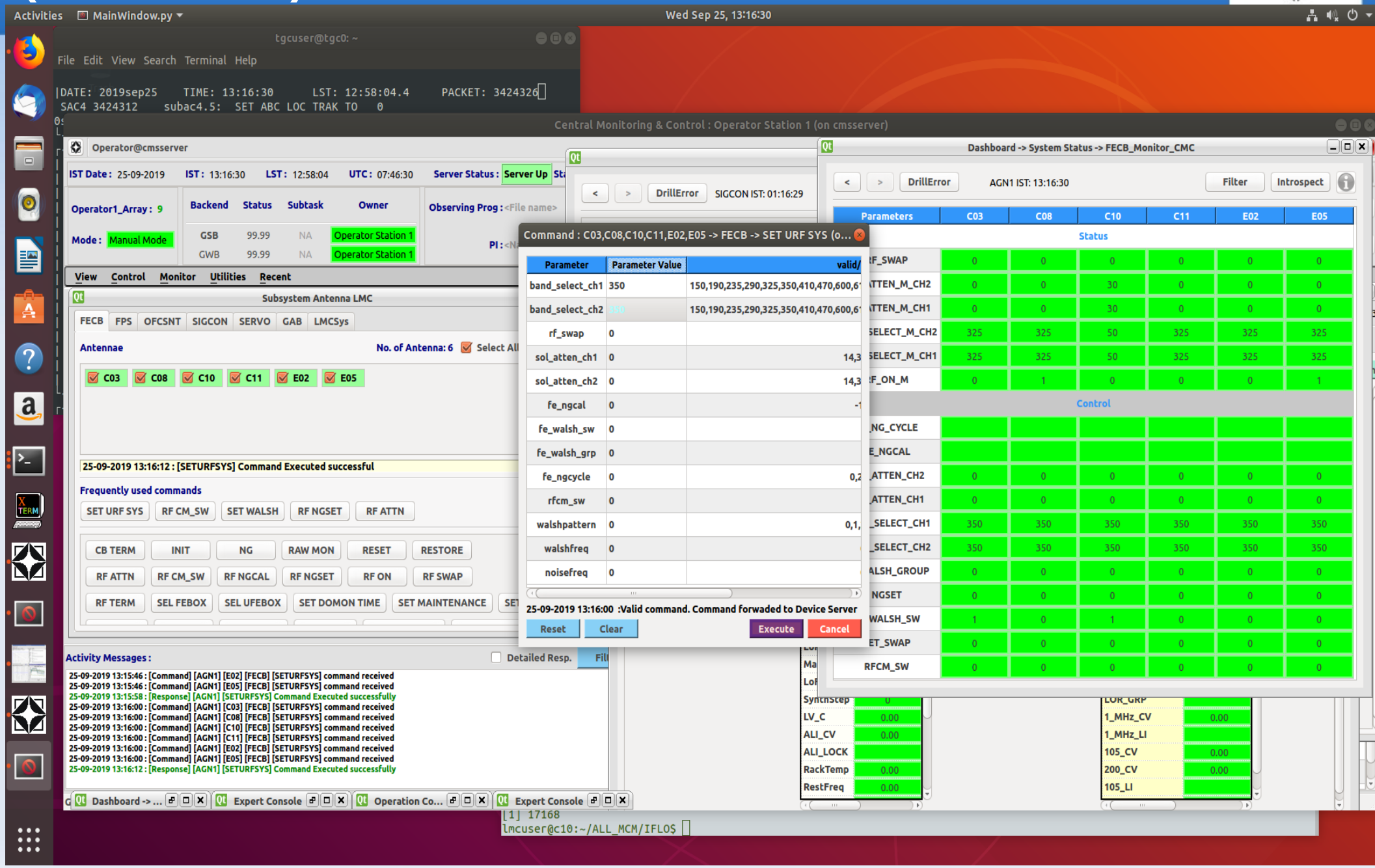
TGC Features : Fault Isolation/Tolerence, Array Allocation



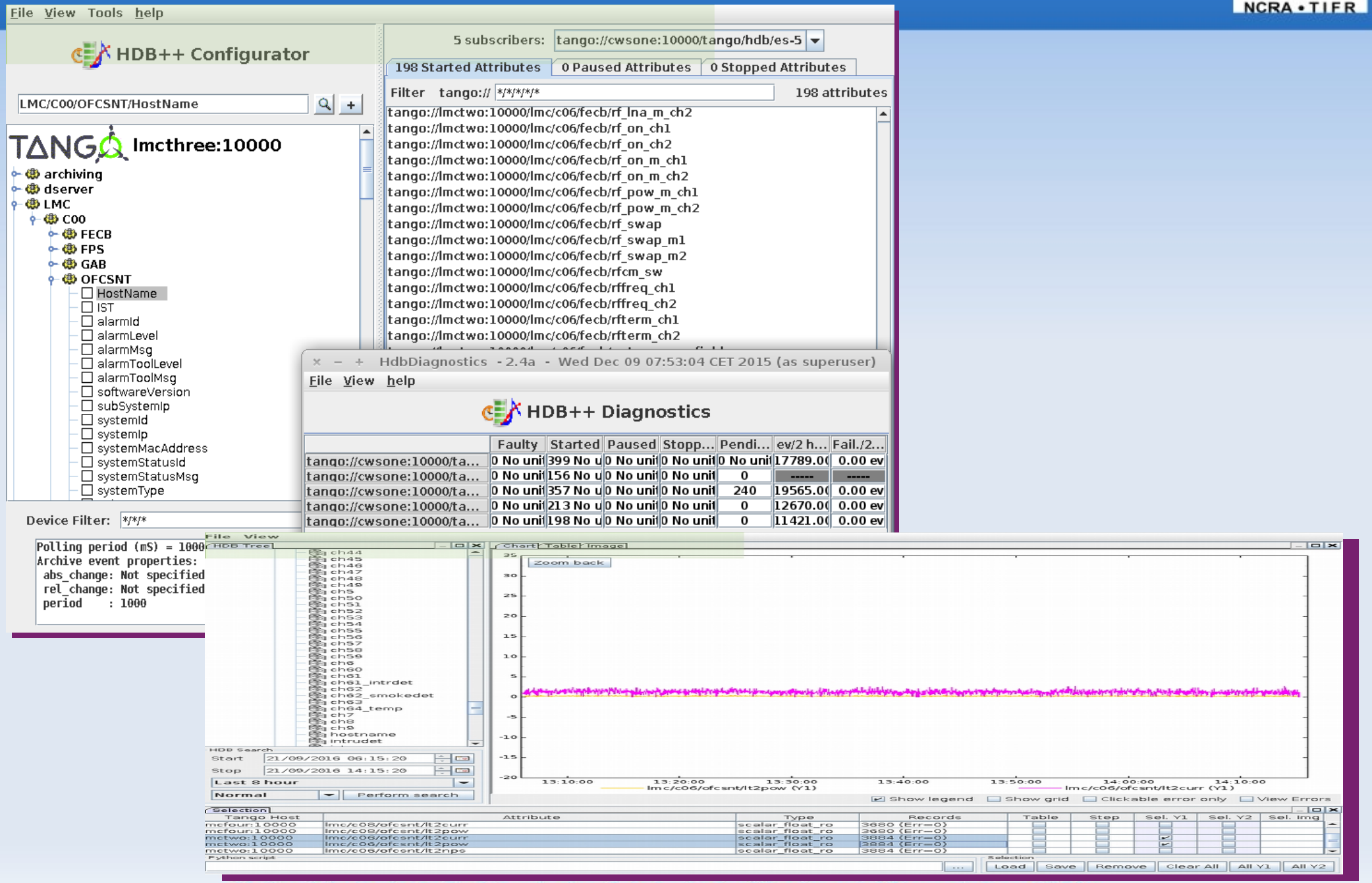
TGC Features : Drill-Down Monitoring (Screen shot)



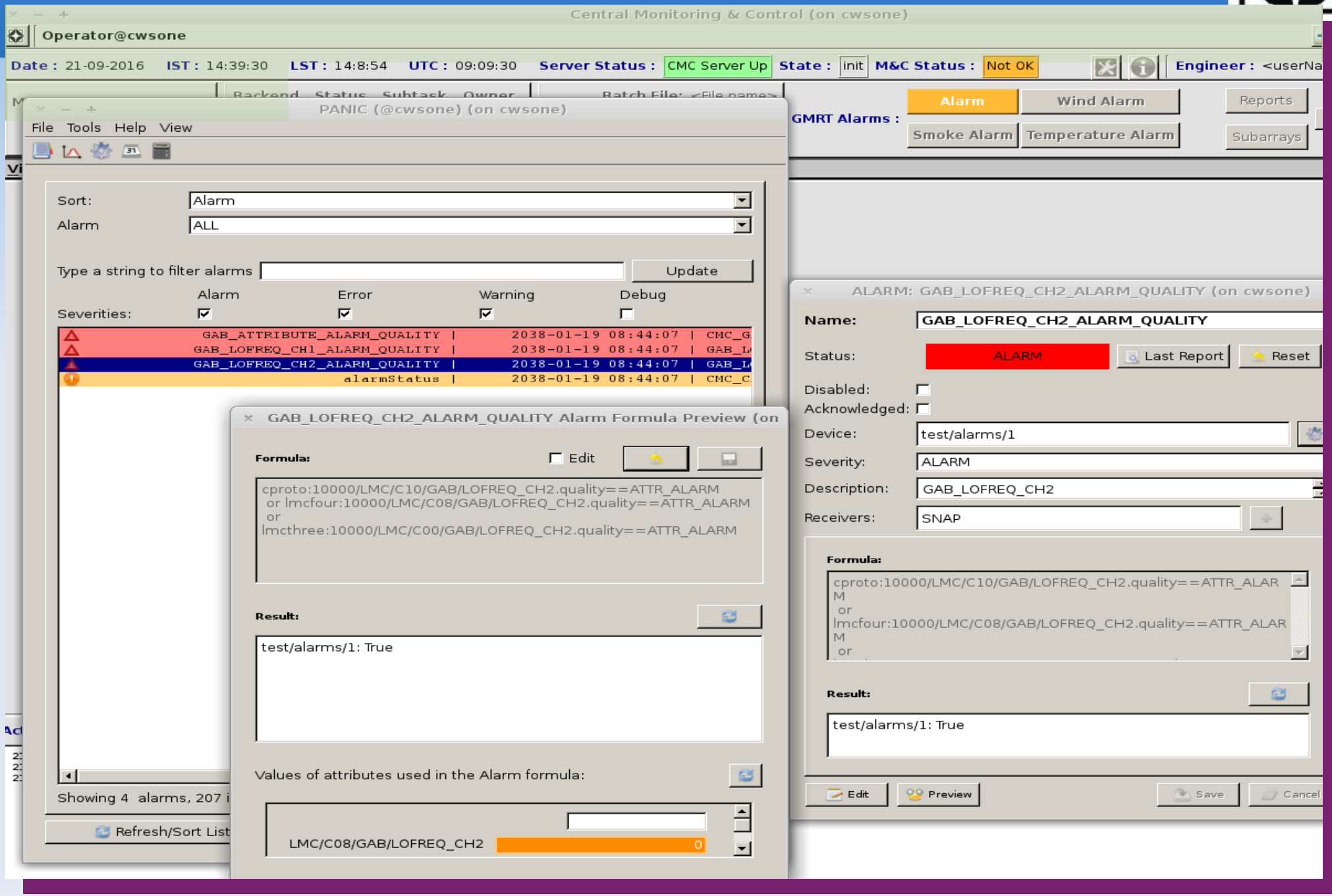
TGC Features : Request-Response & System monitoring (screen shot)



TGC Features : Archiveing - Hdb++ Config and View



TGC Features : Panic Alarm Tool (Screen shots)



Software Development : Process and Timeline

DEFINITION PHASE of the project :

(1) User Requirement Specifications (ver 2.0, March 2013) : Use cases identified and documented based on over 15 years of telescope operation experience.
(2) Software Requirement Specifications (ver 1.0, Aug 2013)
(3) Software Architecture document (ver 1.1, Dec 2014) and consistent GUI design (Ver 1.1, Nov 2014) across the element.
(4) Technology Evaluation : EPICS (ver 0.6, May 2014), TANGO (Ver 1.3, Oct 2015)

DEVELOPMENT PHASE : Planned Deliveries are

Phase-1 : A core M&C system with essential functionality to control GMRT antennas

Phase-2 : Astronomical Calibration and Auxiliary applications to conduct astronomical observing sessions (Completed in April 2019)

Iterative Waterfall Model :
Deliveries are sub-phased again for short-duration based on Requirement Prioritization

Scheduled routine discussion : Early feedback on incremental Development process like detail design of sub-phase, validation and integration with the main software.

GMRT team contributes 25 % of domain based software development and testing : Such as Java/Python wrappers to the NOVAS Positional Astronomy Library, DataServers and Back-end Controller, Phasing scripts, and conversion format Libraries.

Software Development

TGC Components	Open-Source s/w used		
	IO	LMC	CMC
Control-Server	Tango 9.2.2, OmniORB 4.2.1, ZeroMQ 4.0.7	Jtango 9.5.0, JacORB	
Logging	log4cpp	Slf4j – log4j 2.6.1	
GUI	Not Applicable	Taurus GUI framework – 3.7.0, PyQt4 (4.11.4) , PyTango (8.1.8)	
Scripting	Python 2.7, PyTango 8.1.8 (Along with Boost library for C++ interface)		
A & A	Kerberos V5 implementation		
Positional Astronomy	Novas-3.1.1.3 (Noval Observatory Vector Astronomy Software)		
Archiving	Hisorical Database HDB++ : hdb++-cm 1.1.0, hdb++es 1.0.1		
Alarm	Panic Alarm 6.5.1, Fandgo 14.3.0		
Longitivity	Astor / Starter – Tango Manger 6.7, Jive 6.9. POGO 9.2.4		
Database : Mysql 5.7,	Code Generation : Pogo 9.2.4,	Deployment : Ansible 2.3.2 , bash-scripts	

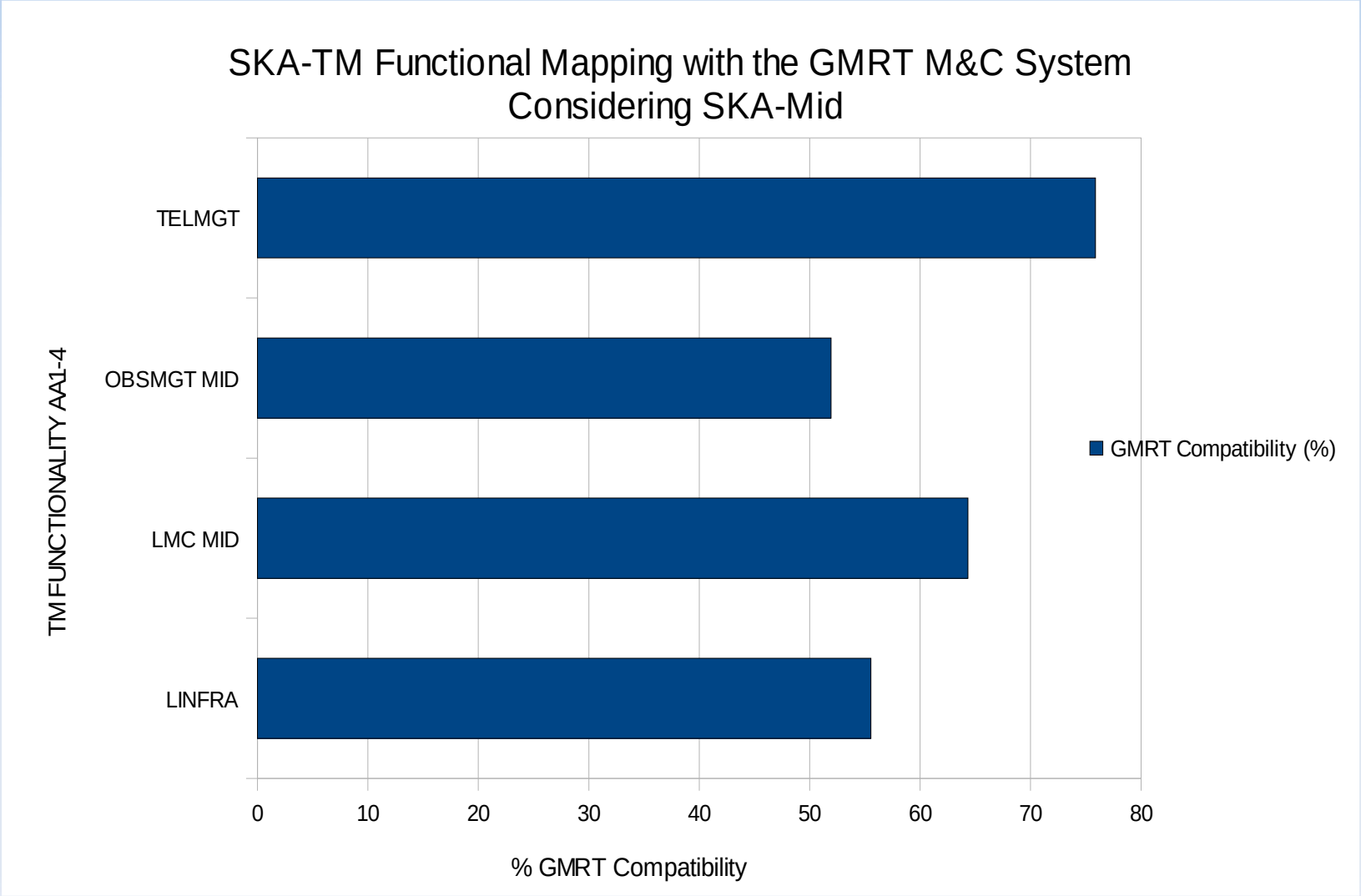
GMRT Soft Architecture : Alignment with SKA-TM

- **Hiearchical Control**
- **Support Multiple Tango facility**
- **Modular and Configurable Data Driven Architecture**
- **Local and Centralized Data Archiving**
- **Alarm Detection and Handling : Local and Centralized**

SKA1 Control and Monitoring Hierarchy	GMRT Control and Monitoring Hierarchy
Central Node	Central Monitor & Control (CMC) node
Subarray Nodes (1 to 16) - Dynamic grouping of leaf-nodes based on observation	OPER-1 to OPER-5 - Operation Terminal for conducting observing sessions or engineering experiments using sub-arrays.
Aggregation Nodes - Static grouping of leaf-nodes for engineering Operations like emergency control and debugging.	Super-Operator Terminal Static Sub-array containing all 30 antennas for the manual intervention or emergency control of LMCs.
Leaf-Node	Local Monitoring Control (LMC) at the Antenna base
Element LMC for controlling of Dish, Single Pixel Feed, LFAA elements etc.	I/O Nodes to interface with Telescope Subsystems

SKA-TM Functional Mapping : Compatibility with the GMRT M&C System

TM Functional mapping to Requirements and TM Product List



Lessons Learned From the Exploratory Prototype

- The TGC Control-Node requirement, concpets and design has been useful, and adapted while developing the TELMGT LMC base class prototype.
- GMRT M&C System implementation can be a reference points for feasibility study of the SKA-TM design, feedback on usage of the TANGO framework, improving effort estimates and time planning for the construction of TM.
- The GMRT Control Requirements directed towards exploring the PANIC Alarm, HDB++ Archiving required for remote and distributed hosts running across different LAN/Network.
- The project has led to the development of skilled manpower in Indian industry for telescope control and radio astronomy applications (last 4-5 years).

➢ Higher level Parent Tango-Device hangs if Child Tango-device-server is down suddenly (Due to netowork problem or power-down), Aggregation node and GUI hangs becase number of attributes can not unsubscribe at a time.

➢ Performance issues are hiting because each command-response cycle goes through A&A, command validation etc. Events are missing when mysqld consume more than 100 % cpu time continuosly.

➢ Expert reviewer shall be planned while the development of product.

✂ Schedule slippage due lack of resources, simultaneous project/shared work-load.

Thanks…

Authors : J. Kodilkar [1], V. Kumthekar [2], P. Patwari [3], R. Uprade[1], C. Kanade[1], S. Katore[1], D. Bhong[1], S. Nayak[1], Y. Gupta[1], K. Thigale[2], M. Phule[2], B. Amin[2], A. Pritesh[2], K. Khatri[2], G. Rajas[2], B. Sarojini[2], B. Chitalia[2], A. Banerjee[3], S. Chaudhuri[3], N. Swaminathan[3]

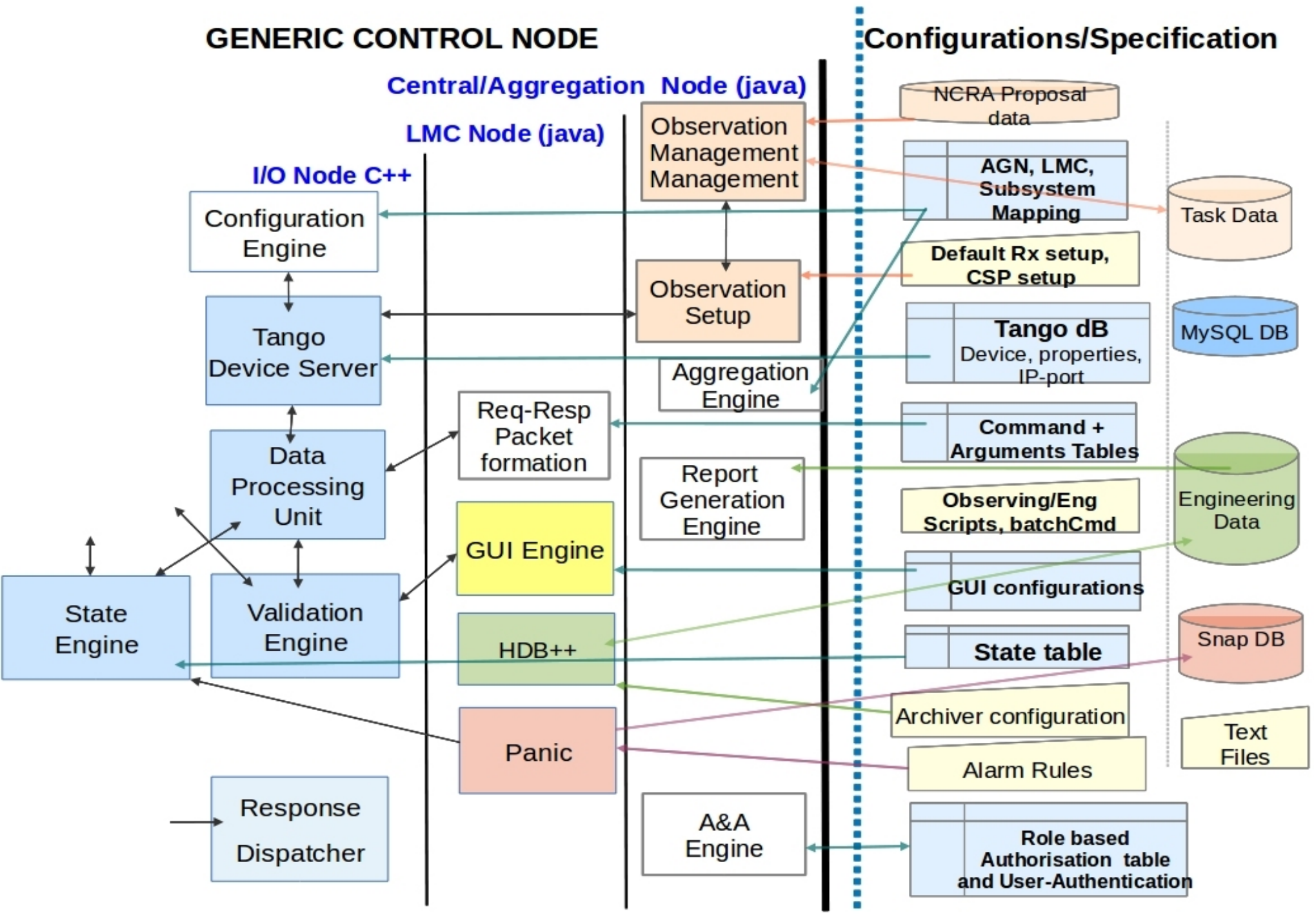
Affilations : 1 National Centre for Radio Astrophysics - Tata Institute of Fundamental Research, Pune, India
2 Tata Consultancy Services, Pune, India.
3 Tatat Researach Development & Design Centre - TCS, Pune, India

References : [1] Development of Next Generation Monitor & Control System for Radio Telescopes, 2013 IOP Conf. Ser.: Mater. Sci. Eng. 44 012026
[2] GMRT M&C System Software Architecture Description Document, December 2014
[3] Report On Tango Control System Framework Evaluation, Version 1.3, October, 2015 [4] Tango Controls, http://www.tangocontrols.org

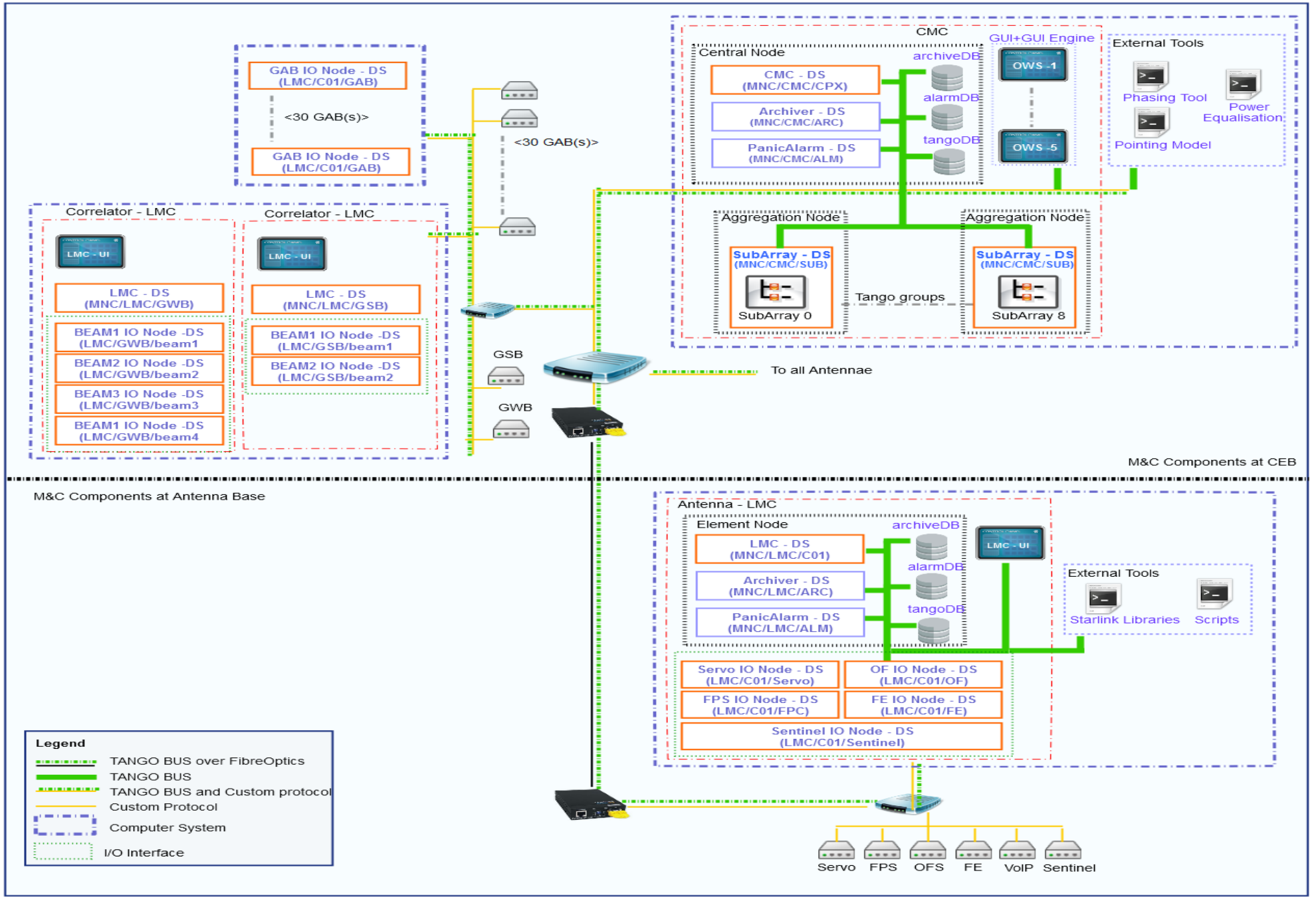
Demonstration

- Engine based Data Driven architecture
- Database Schema
- Aggregation Node – Request Response
- Scripting
- Alarm : Panic Tool
- Data Archiving – Retreiving : HDB++
- Scalability and Configurability

Engine Based , Data Driven Architecture



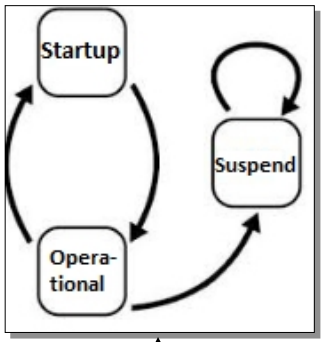
Implementation : Tango Based M&C System Architecture



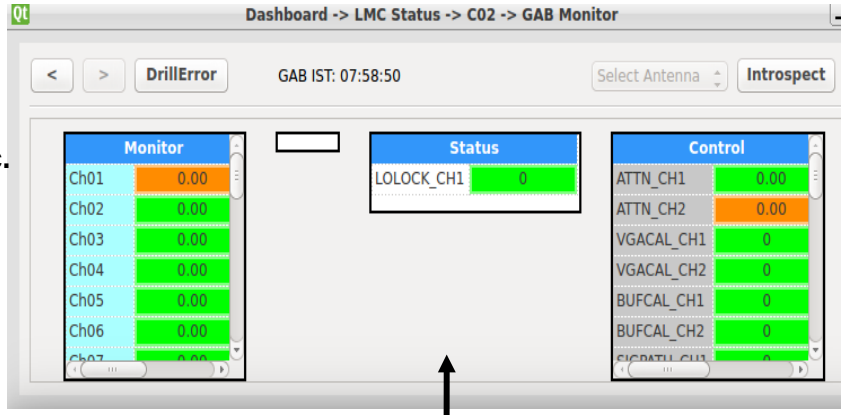
Data Driven flow Req-Response, State and Event/Alarm

State transition executed as per the specifications in DB

- Allowable state transitions
- Actions to be performed upon state transition
- Control commands allowed in specific state etc.

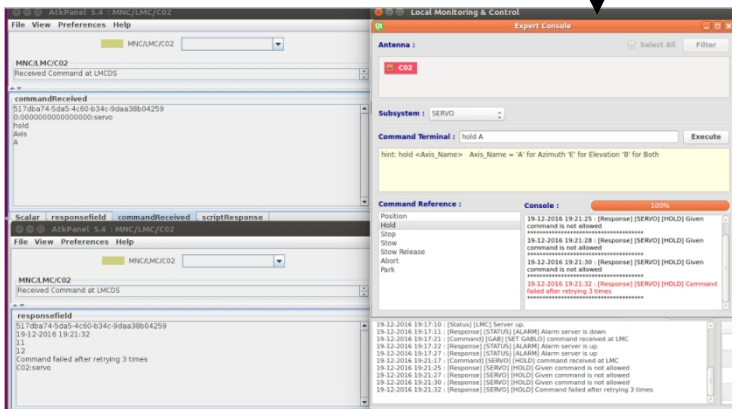


- Grouping of parameters
- Option to display/ Hide
- Background Colours etc.



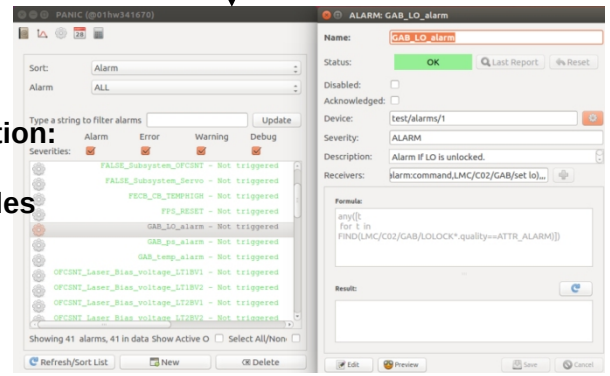
stateORcmdId	init	operationalManual	operationalAutomatic	maintenance	suspend	shutdown
initialisation	0	1	0	0	1	0
operationalManual	0	0	1	1	1	parkAll.py
operationalAutomatic	0	1	0	0	1	0
maintenance	0	1	0	0	1	0
suspend	0	1	1	1	0	1
shutdown	0	1	0	0	1	0
84	0	1	0	0	0	0

contain 1	table	tableDispla	titleColor	displayAtNan
1	1	Monitor	3296FA	tableMonitor
1	2	Sensor	3296FA	tableSensors
3	1	Status	3296FA	tableStatus
4	1	Control	3296FA	tableControl



Command packet configuration:

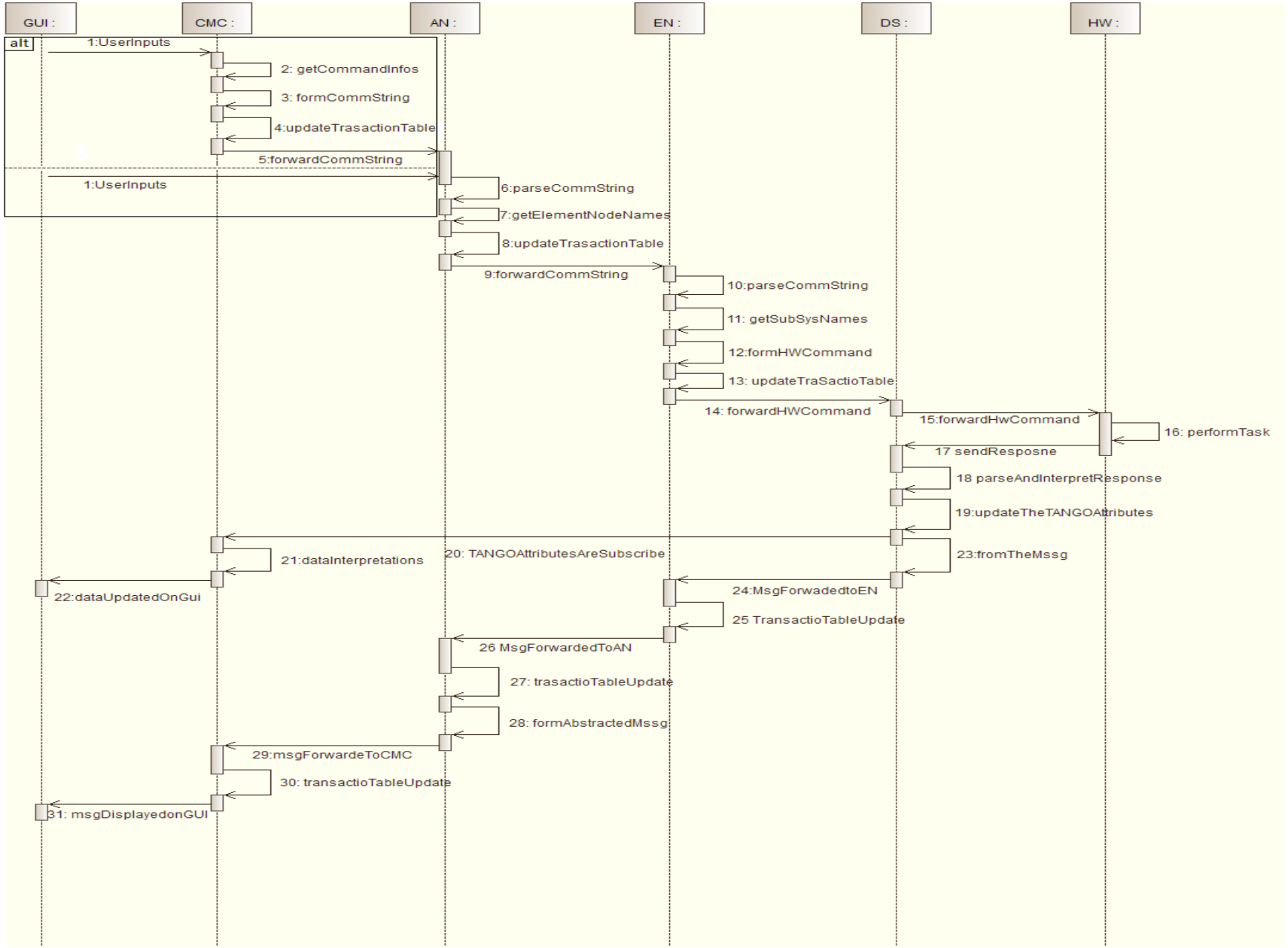
- Origin of command
- Destination
- Command Priority
- Response (Intermediate
- Final /Success, Fail etc.)
- Retry
- Timeout



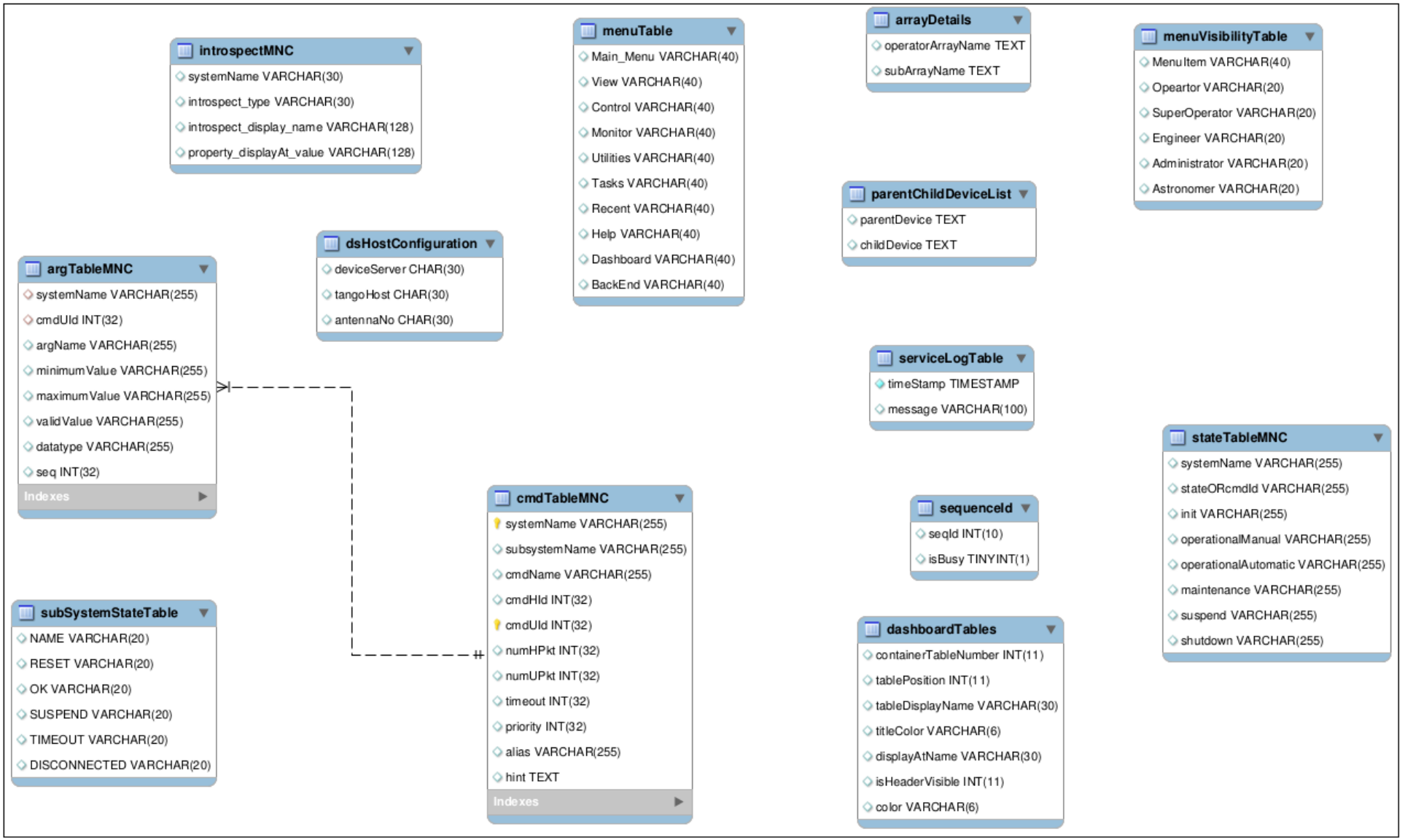
Alarms configuration:

- Alarm evaluation rules
- Hi/ Lo Values
- Automatic actions

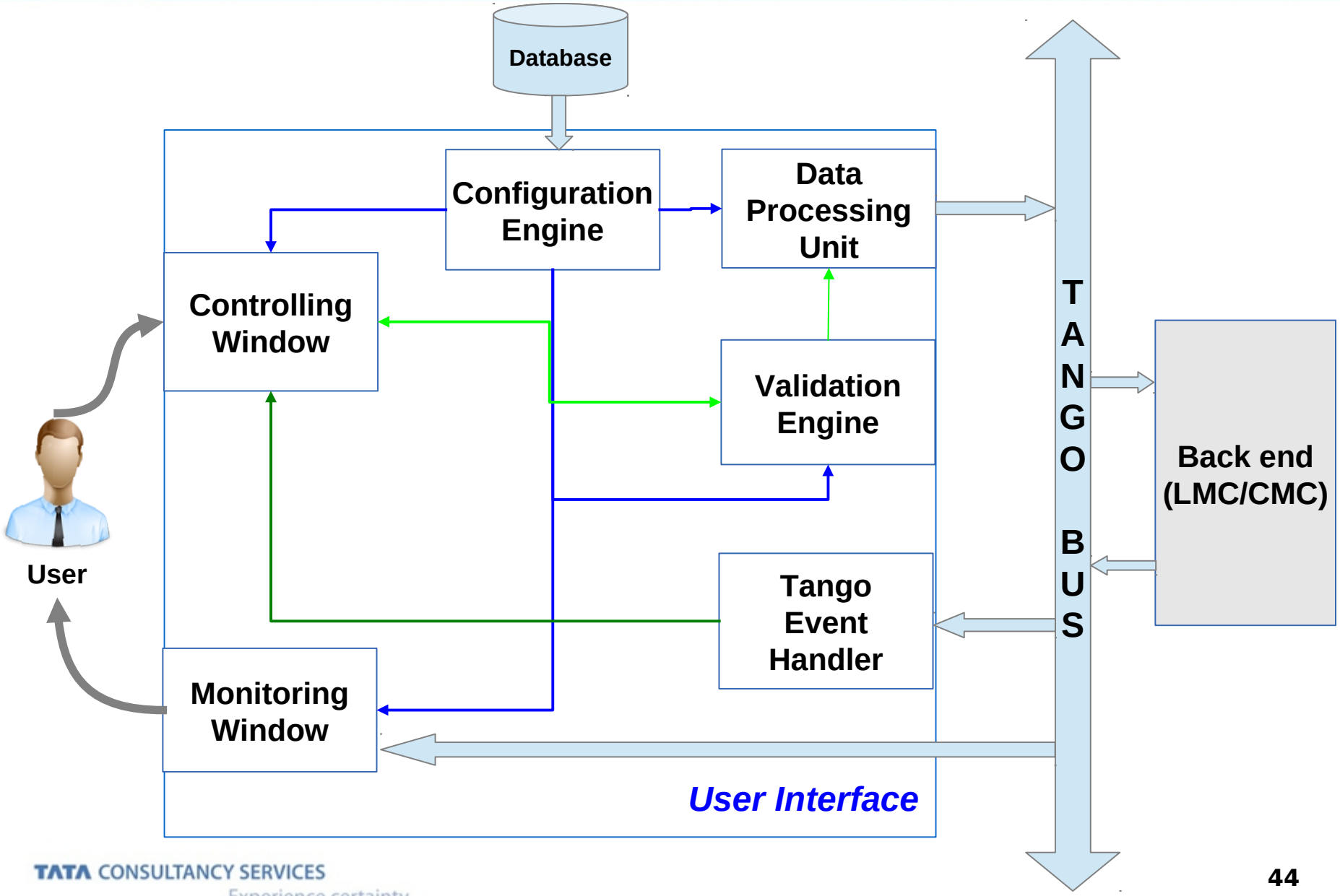
Typical* End to End Command Response Flow



High Level View of Database and Tables



GUI Block Diagram



GUI Module Overview

- **Configuration Engine**
  - **dbConfig.py**
  - **errorMsgs.py**
  - **menuReadFromDb.py**
  - **mySqlInterface.py**
- **Controlling Window:**
  - **commonCommandEnvironment**
  - **connectIoDeviceServer**
  - **expertConsole**
  - **subsystem**
- **Basic Layout**
  - **mainWindow.py**
  - **titleBar**
  - **R1,R2**
  - **openNewMdi.py**
  - **B1,B2**
  - **GmrtVersion,icon**
- **Validation Engine + Data Processing Unit**
- **validateCmdMNC.py**
- **Tango Event Handler :**
  - **cmdRespChangeEvent.py**
- **Monitoring Window**
  - **dashboardControl**
  - **introspect**
  - **quickAccess**
  - **systemStatus**
- **Major Features**
  - **activityMessageFilter**
  - **selectionFilter**
  - **systemFilter**
  - **managePool**
  - **masterControl**
  - **observationProgram**
  - **systemVariables**
  - **tuneReceiver**

Alarms and Archiver

Back end – Node (CMC)

Scripting Engine

- **Executes given script**
- **Ability to execute batch script**
- **Executes script for any command**
- **Executes script for any state**

Command & response mechanism

- **Send command to multiple LMCs at a time & gives individual & aggregated response**
- **Start & stop services like alarm, archiver**

- **Subarray functionality**
  - **create & destroy subarray, allocate & deallocate operator to particular subarray**

  Validation Engine
  - **Validates incoming command**

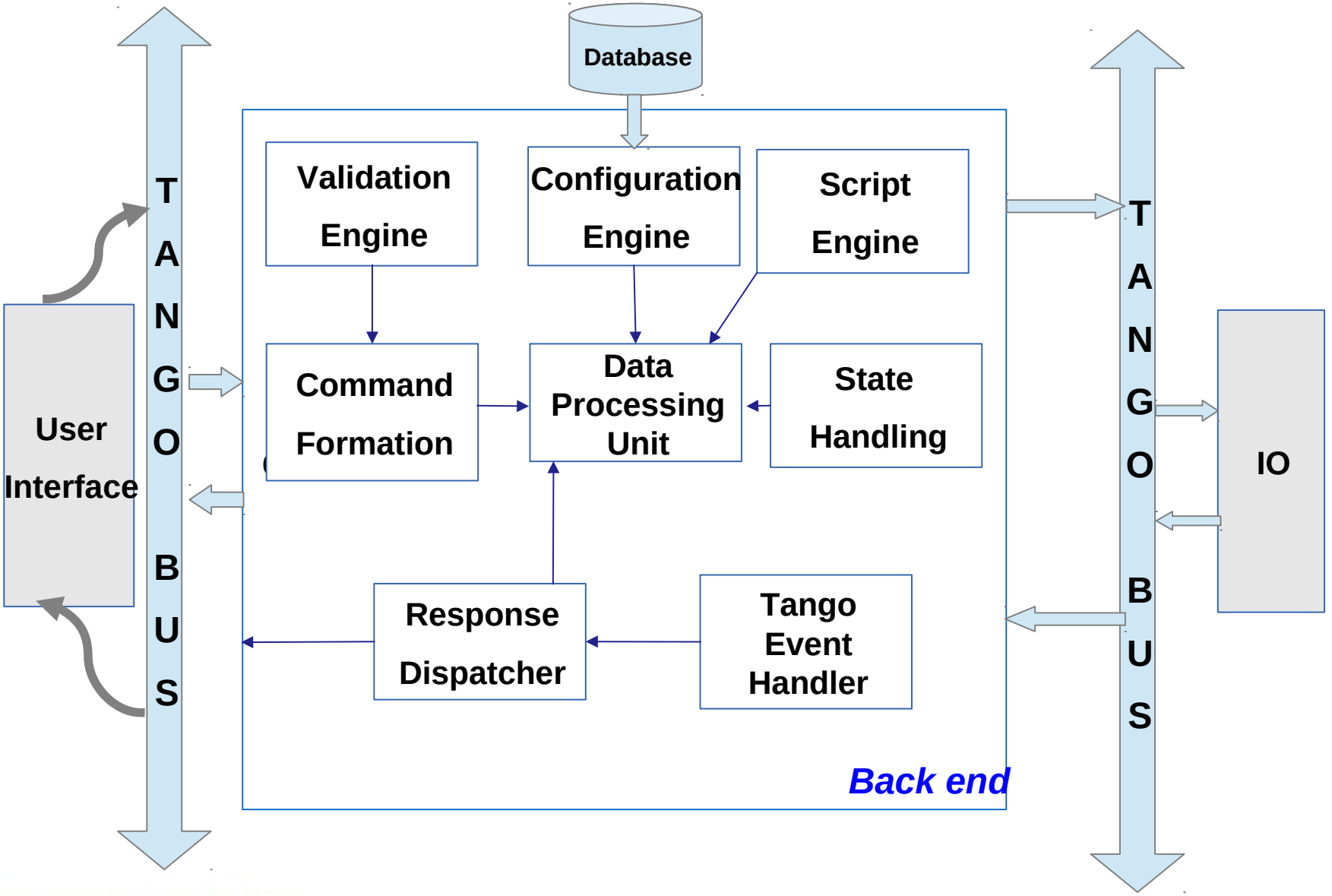
  State Machine
  - **Checks whether state transition is allowed/not**
  - **Checks whether command is allowed in current state**

  Longevity Unit
  - **Lounge the subsystems, archiver and alarm**

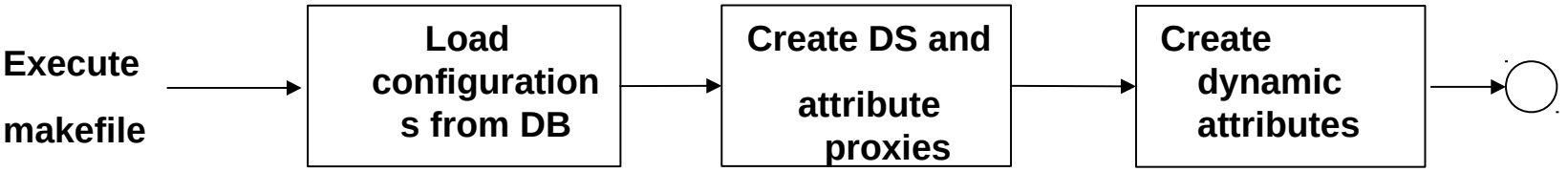
  Monitoring Unit
  - **Maintains longed DS health information (alive/dead)**
  - **Gets aggregated system status & alarm status of subsystems**

- **Configuration Engine**
  - **Interacts with database to get the required information**
  - **Gets information about number of LMCs available**
  - **Gets information about number of archiver & alarm available**

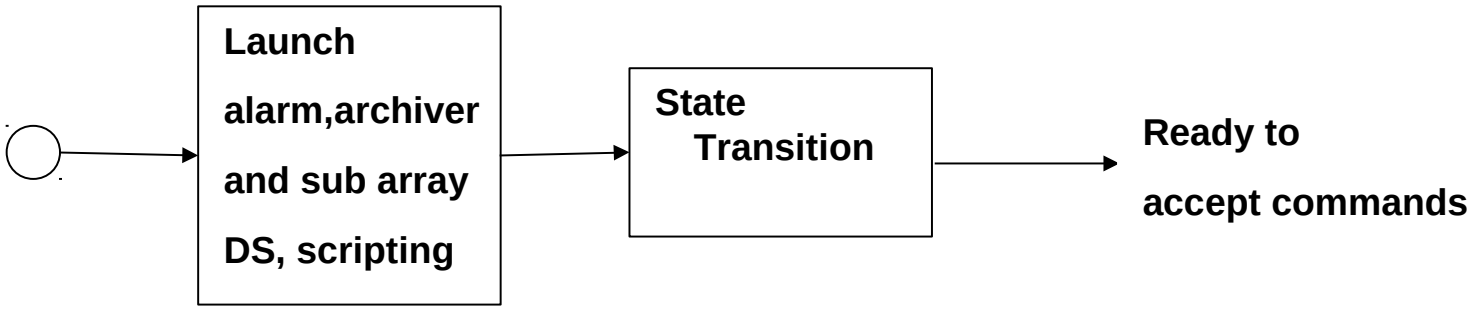
Back end System Overview



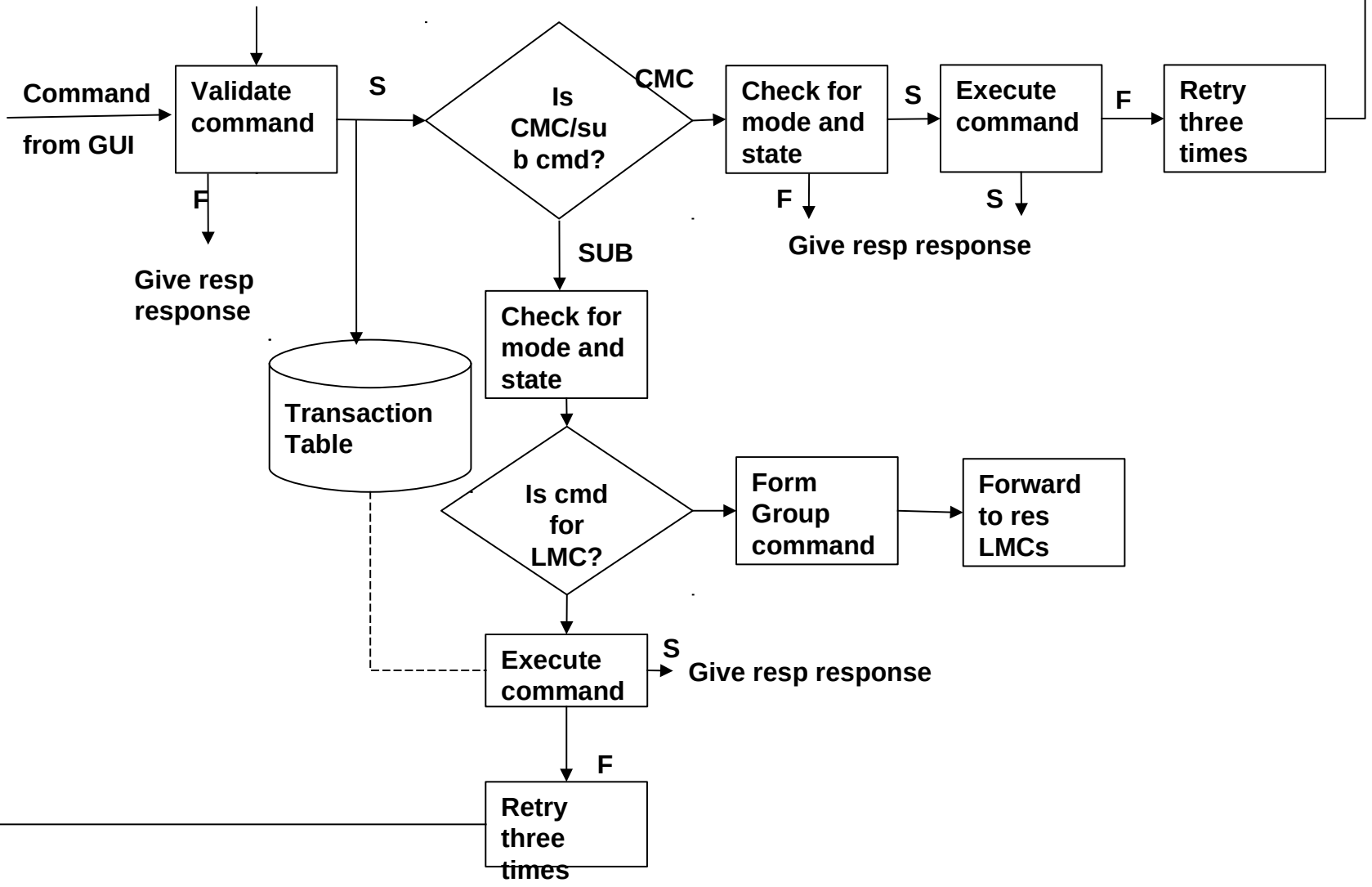
Back end - System Start Up



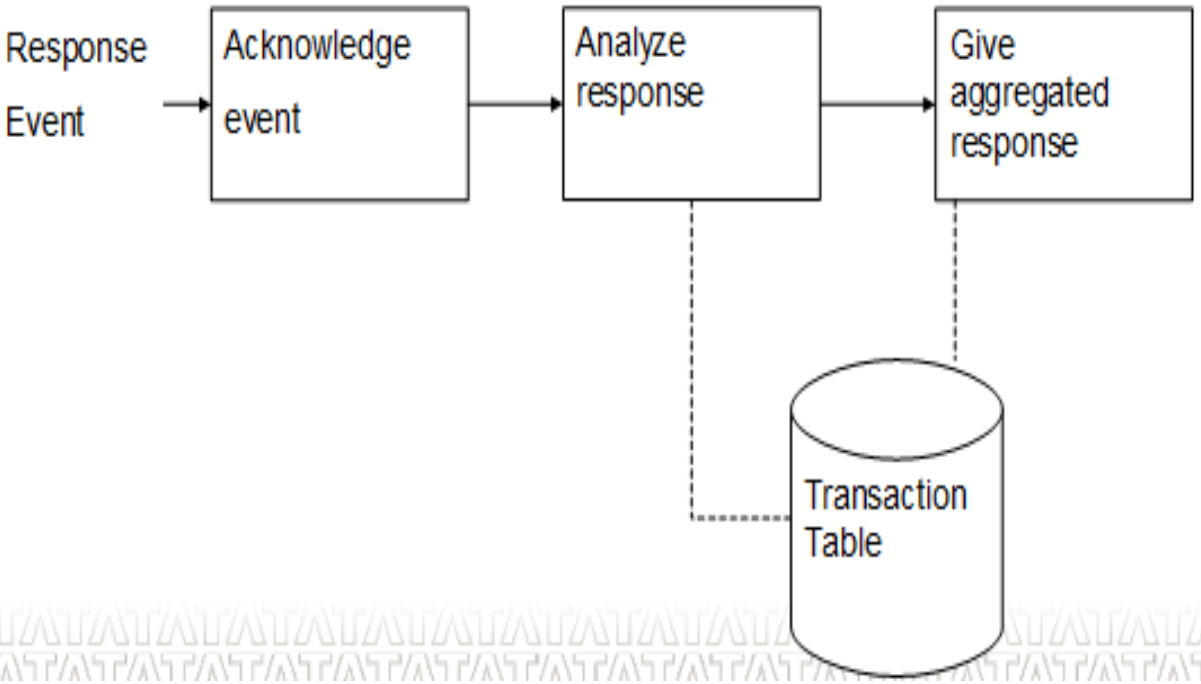
- Status,alarm attributes
For IO DS , quick access parameters(temp,wind).
- Introspect parameters(CPU usage)



Back end – Command Flow



Back end- Response Dispatcher



IO Module

Key functionality of IO module :

- Communication with GMRT subsystems
- Command-response mechanism
- Handling of hardware specific commands
- Data processing of hardware responses
- State mechanism

Engines used by IO module :

- Configuration Engine
- Validation Engine
- State Engine
- Scripting Engine

IO Block Diagram