CSP.LMC Prototype Proposed Design

E.Giani, C.Baffa

LMC harmonization through Telescopes
Step2: LMC Peer Review
Meeting 1

Madrid, 11-13 April 2016

Main Topics

- General introduction.
- CSP.LMC architecture and functionalities.
- CSP.LMC Prototype structure.
- CSP.LMC Prototype Tango Devices and Tango Classes.
- Prototype Tango Attributes and Commands.
- Prototype monitoring strategy.
- Logging and Alarms.

Reference framework

The reference framework is established by:

- SKA1_MID Telescope Interface Control Document CSP to TM (EICD)
- Interface Control Document LMC to CSP Sub-elements (IICD)
- SKA CSP Local Monitor and Control Sub-element Detailed Design Description
- LMC Interface Guidelines Document (LIG)
- Tango Interface Guidelines

The Tango Control System Manual version 8.1 and 9.1

Horizontal Prototype & boundary conditions

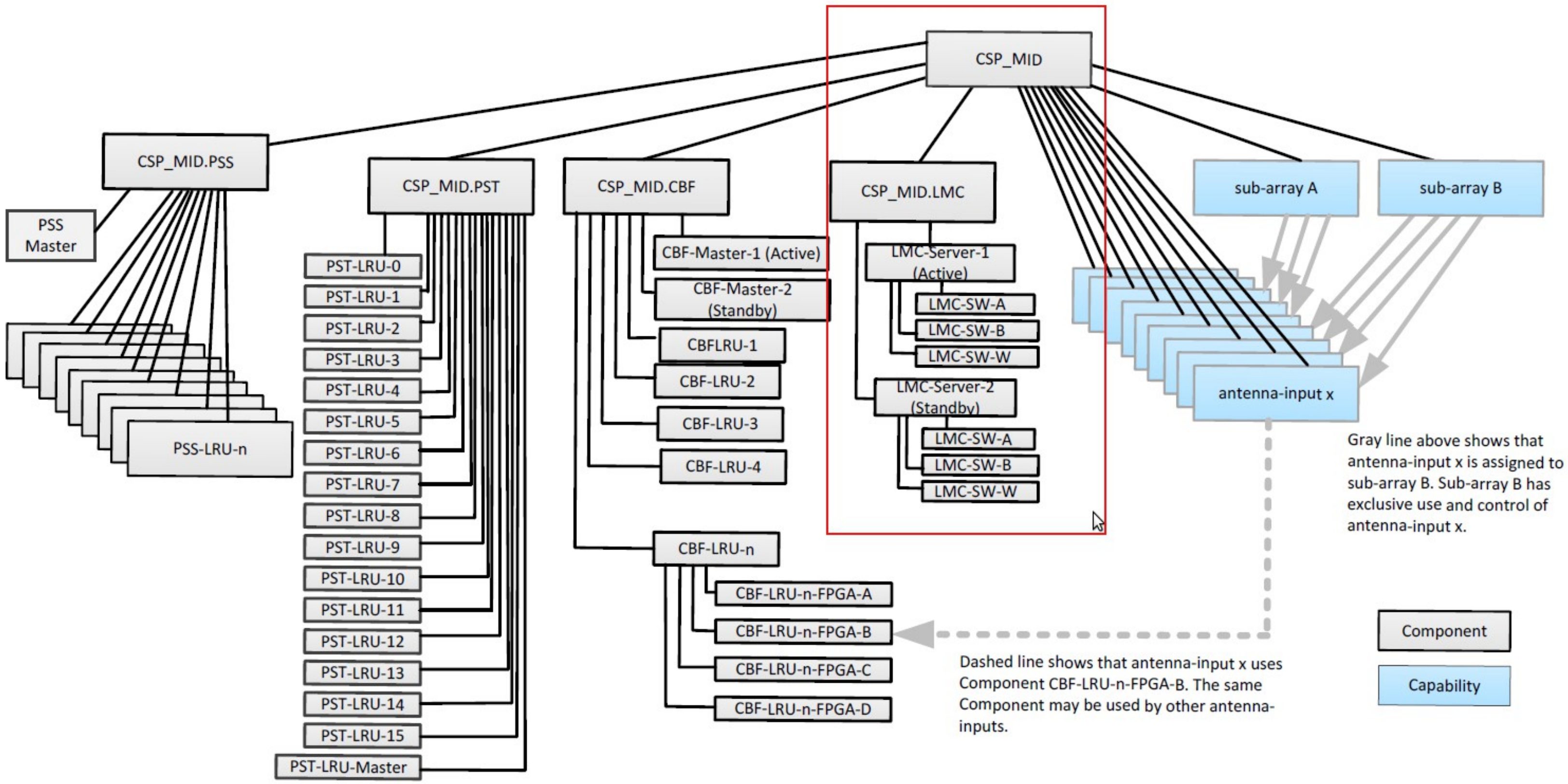
- A prototype with a nearly complete interface towards TM.
- Prototype development performed in the MID mental framework but the functionalities are in common with LOW.

- TM has an unique point of access to CSP.LMC during normal operations.
- All CSP SubElements have an unique point of access to CSP.LMC during normal operations.
- **Use of Tango as control framework.**

Why to develop a prototype?

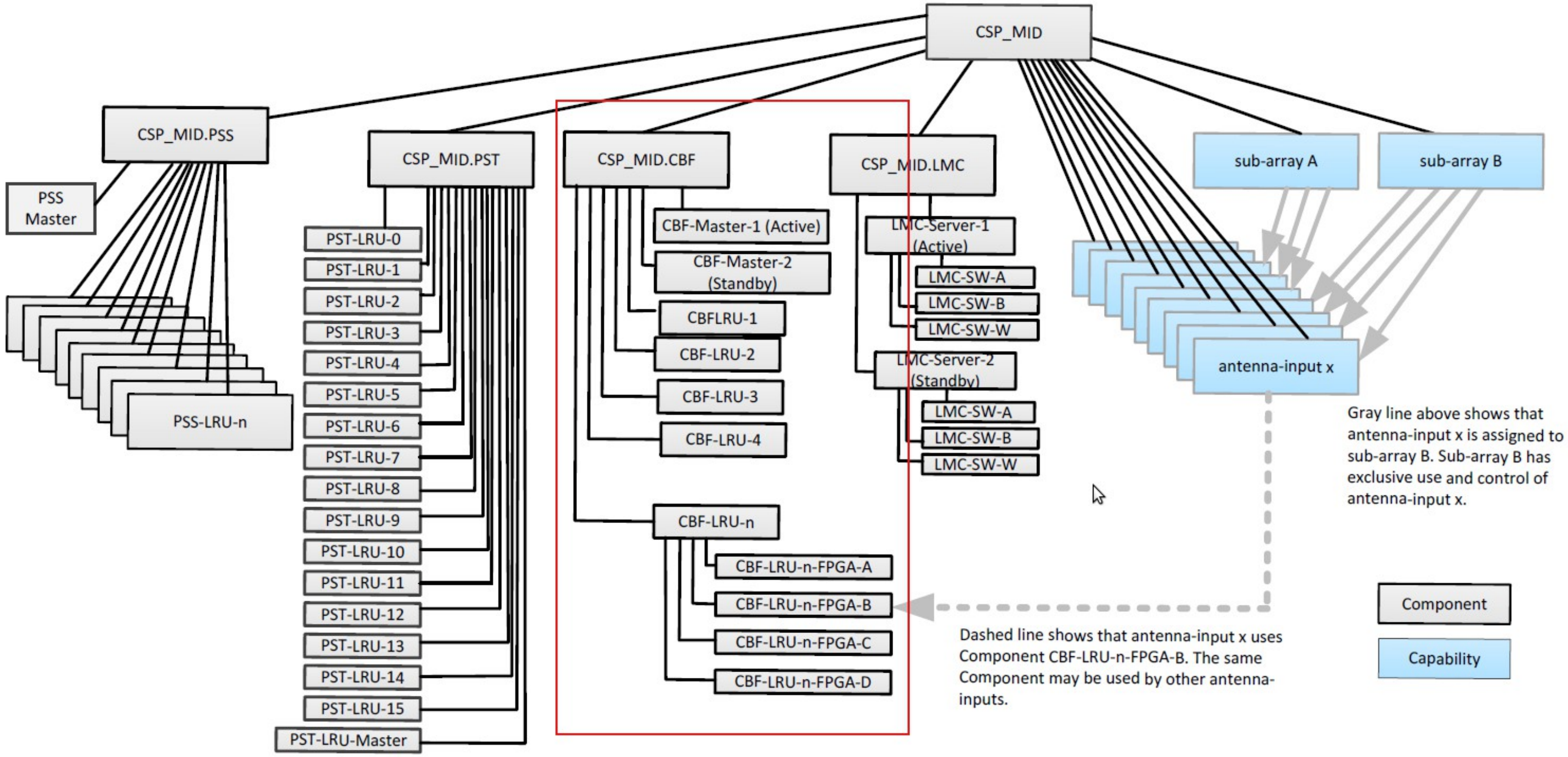
- Test the Tango ability and find the best approaches to implement the main CSP.LMC functionalities.
- Reduce the risks of the requirements.
- Verify the compliance with the requirements of timings in critical operations.
- Test, if possible, a small subset of design alternatives.

CSP M&C Hierarchical Structure



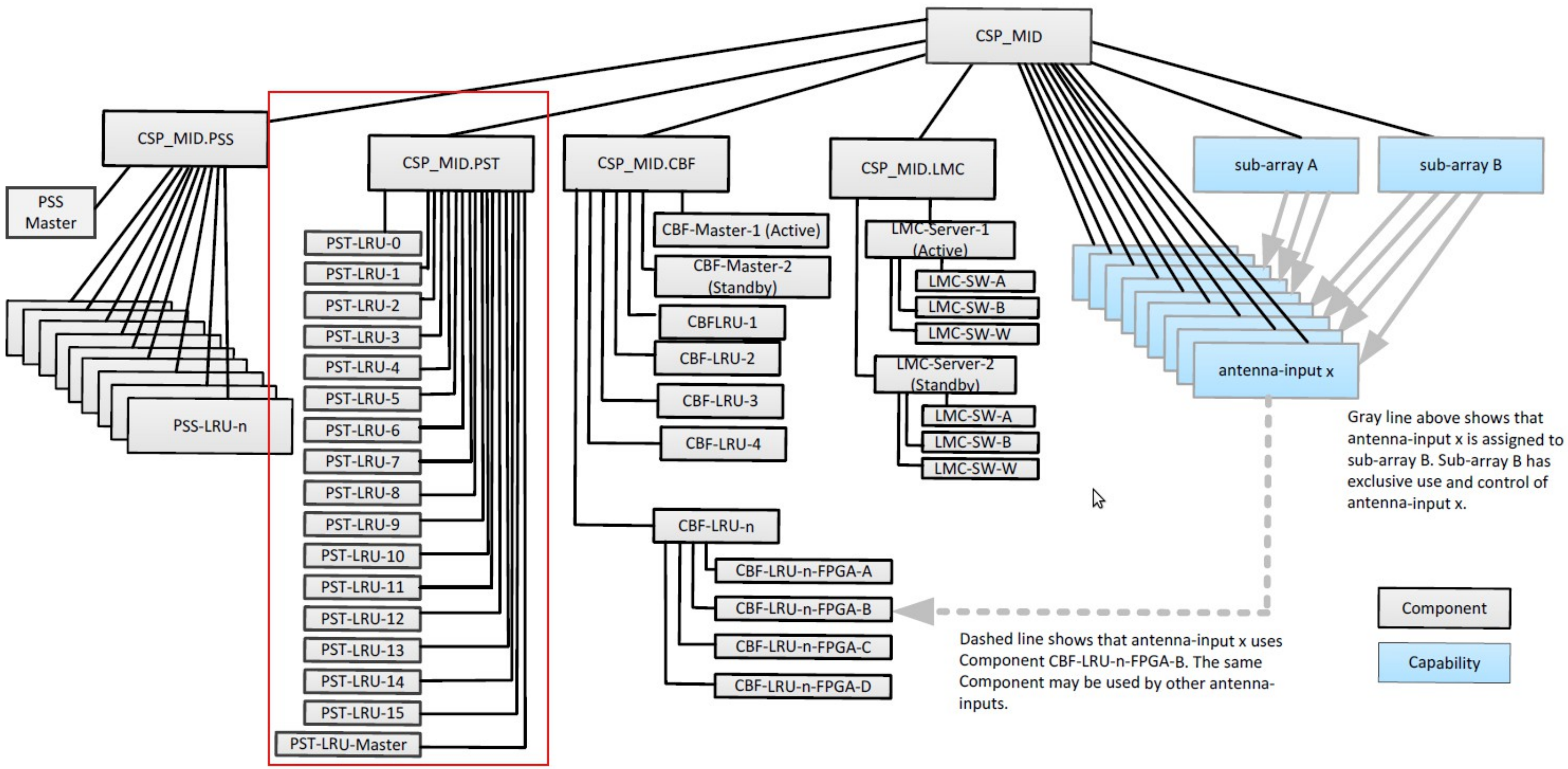
From: *S.Vrcjc SKA ICD SKA Document*

CSP M&C Hierarchical Structure



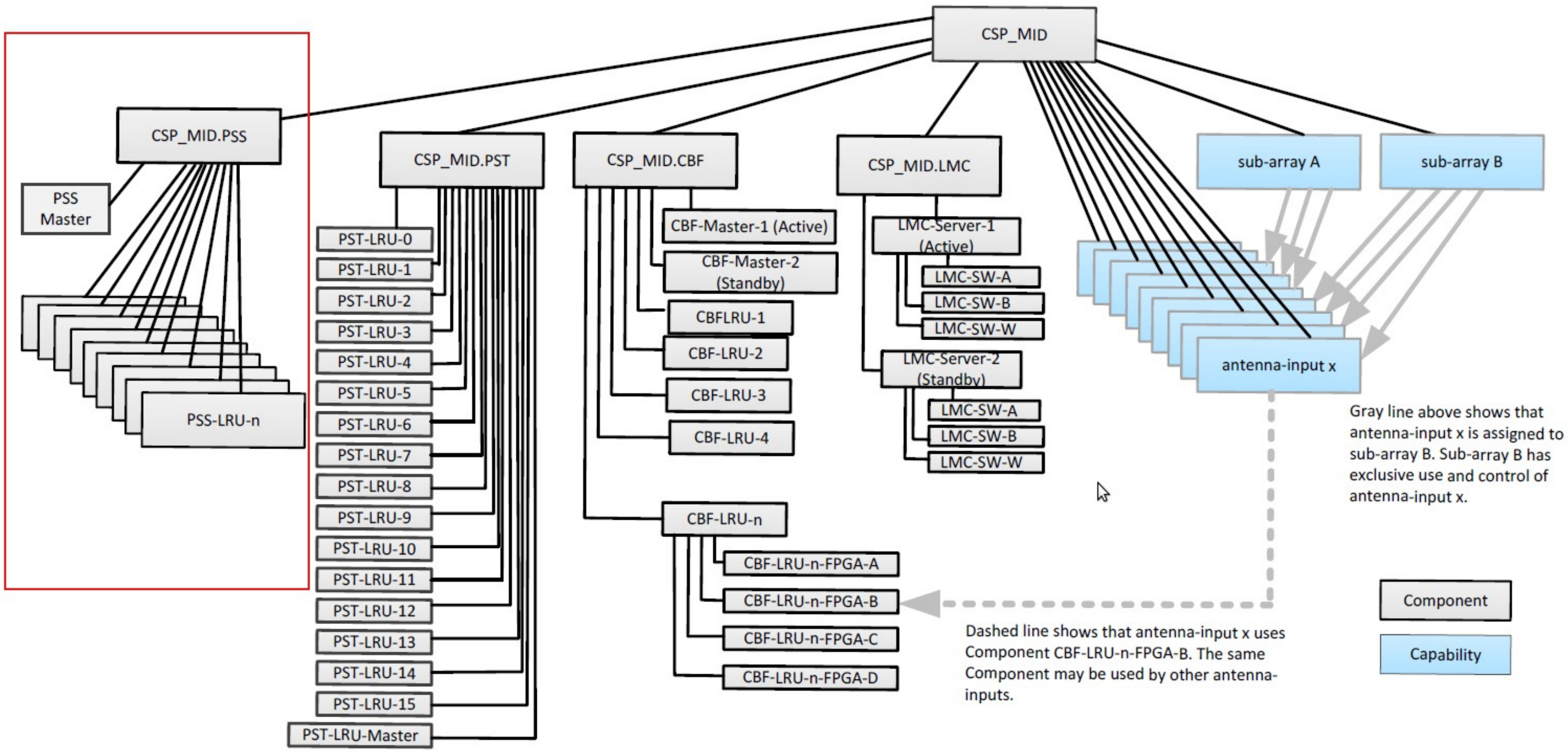
From: *S.Vrcjc SKA ICD SKA Document*

CSP M&C Hierarchical Structure



From: *S.Vrcjc SKA ICD SKA Document*

CSP M&C Hierarchical Structure



From: *S.Vrcjc SKA ICD SKA Document*

The Prototype M & C Functions

The Prototype will implement some **CSP** M&C functionalities. In particular, it will:

- Implement the interface with TM and SubElements.
- Maintain and control the overall CSP status.
- Receive, execute TM commands and generate replies.
- Perform mapping of TM commands to command for individual CSP SubElements.
- Handle timed commands.
- On the behalf of TM configure *SubArrays* and allocate the *Capabilities* to them.
- Collect and forward to TM the alarms, events and other messages generated by CSP SubElements.
- Maintain a log of all the activities.

The prototype structure (1)

The modeling of an equipment is the first step to implement a Tango Device.

The Prototype structure is modeled on the CSP architecture:

- Each M&C entity is implemented as a TDS (Tango Device Server) running one or more TDs (Tango Devices)

↓

- ✓ One TDS for CSP Element
- ✓ One TDS for each SubElement: CBF, PSS and PST
- ✓ Each TDS runs on a separate PC (Master Node)

- The Prototype will implement as TDs all its M&C functionalities (but the logging).
- The Prototype will use the TLS (Tango System Logging) for logging.

The prototype structure (2)

Tango Device (CSP Master Node)	Functionality
CSP.LMC	Monitor & control the CSP
CSP.SYS	Monitor & Control of the Master Node
Scheduler	Handle the command queue for timed command received from TM
Alarm Handler	Handle of alarms generated by the whole CSP
SubArray	Maintain status and configuration of SubArrays
Capability	Maintain status and configuration of Capabilities

The prototype structure (3)

Tango Device (PSS Master Node)	Functionality
PSS.LMC	Monitor & control the PSS
PSS.SYS	Monitor & Control of the Master Node
Scheduler	Handle the command queue for timed command received from CSP.LMC
Alarm Handler	Handling of alarms generated by PSS

These devices may run in a single TDS (as a multiclass device) or in separate TDSs. From the point of view of the Prototype implementation, this means little changes. It is a matter of logical grouping.

Taxonomy of the prototype classes (1)

The functionalities of a *Tango Device* are described by a *Tango Class*. The CSP.LMC Prototype defines four different families of *Tango Classes*:

1. **TopCsp Class**: implement interface forl M&C functionalities CSP Entity.
2. **Capability Class**: implement interface for CSP.LMC capabilities.
3. **Alarm Class**: implement interface for alarm handling.
4. **Scheduler Class**: implement interface for delayed commands.

The last two classes were borrowed from external Tango Projects.

Inside the prototype Tango Classes

The interface of a Tango Class is defined by the Tango attributes and Tango commands.

The Prototype Tango Classes are organized into a classification hierarchy: from more general classes (abstract) to specialized ones (concrete).

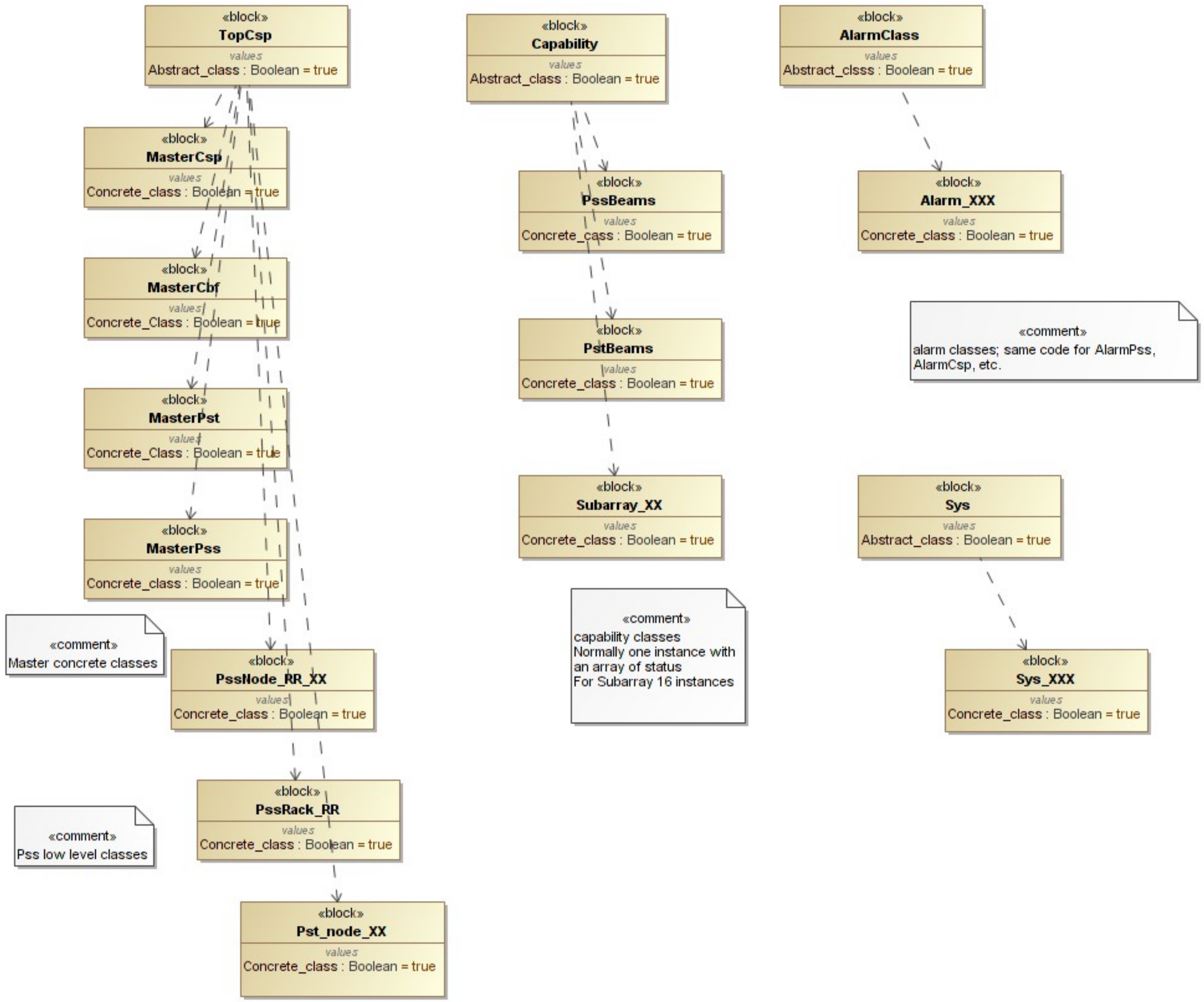
From the *EICD*

- Attributes and methods common to all elements, sub-elements and capabilities → generated few *abstract* classes.

From the *IICD*

- Attributes and methods specific to each SubElement and Capabilities → generated all the concrete Tango Classes.

Taxonomy of the prototype classes (2)



EICD / IICD Parameters

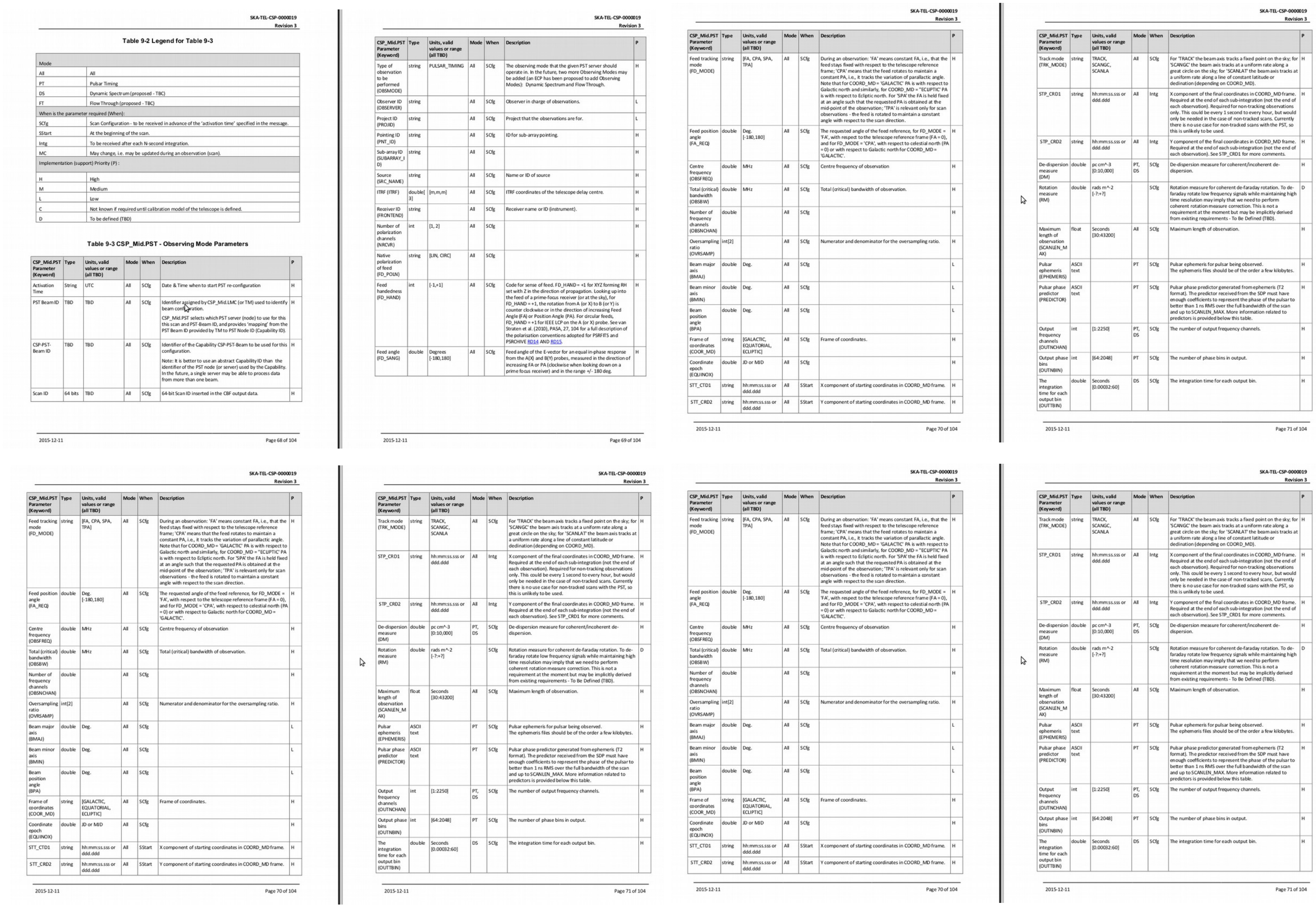
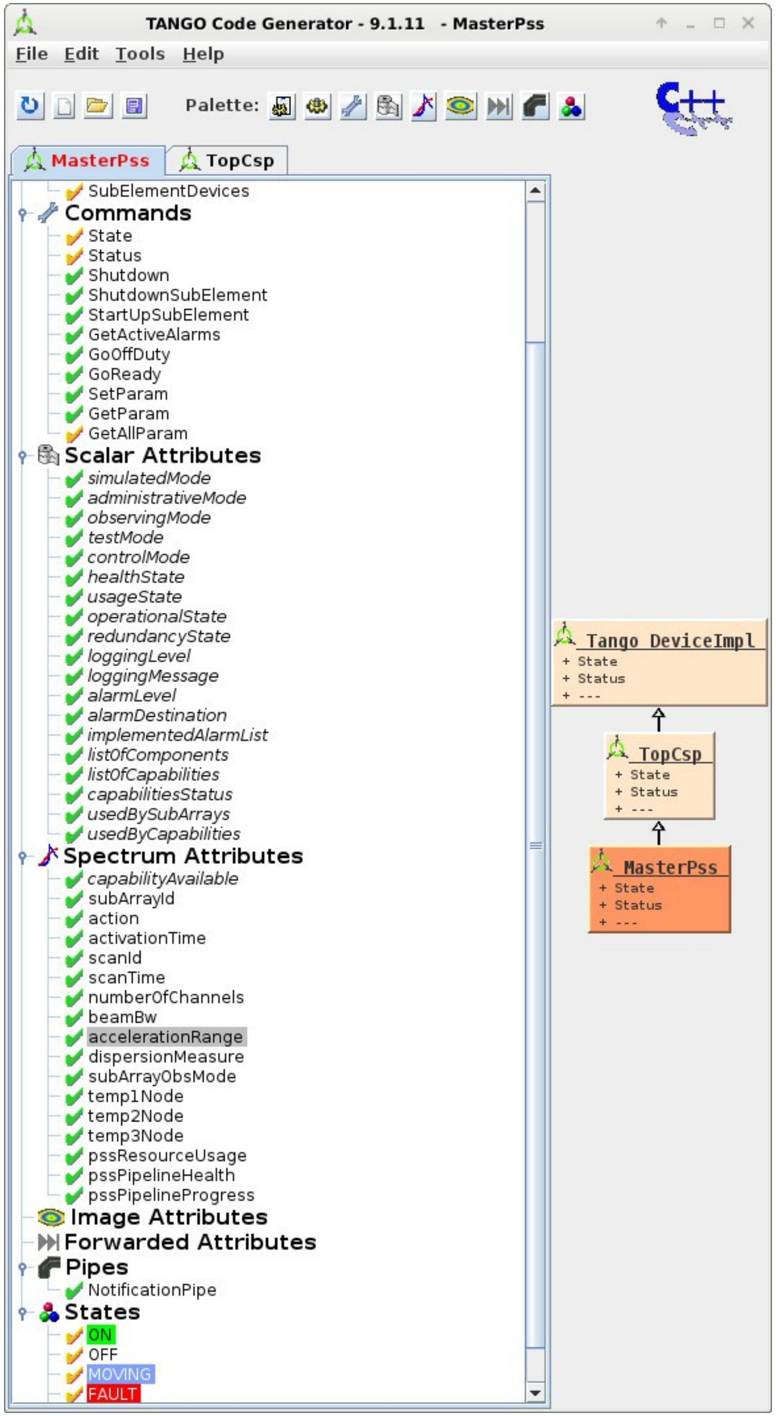

SKA-TEL-CSP-0000019
Revision 3

Table 9-2 Legend for Table 9-3

Table 9-3 CSP_Mid.PST - Observing Mode Parameters

Parameters

- Modes and States attributes defined by the SCM (SKA Control Model)
  - Configuration specific attributes for each entity.
    - Specified on SubArray base.
    - Specified on Capability base (PssBeams, PstBeams).
  - Engineering specific attributes for each entity.
  - Monitoring specific attributes for each entity.



Commands (1)

There is no definition of a common set of control commands to be implemented by all TM elements.

From LIG and Tango Guidelines:

- A preliminary list: power-down, power-up, upgrade software…

From ICDs:

- Two general commands common to CSP_Mid enities:
  - GetParam: to get CSP elements configuration attributes and status
  - SetParam: to specify a single message to configure one or more parameters. It can specify also an *action* as R/W parameters (es: Observing Mode).

Basic assumptions

- TM sends coherent and complete commands to CSP.LMC.
  - CSP.LMC performs syntactic and minimum safety checks, not extensive one.
- TM sends detailed configurations for scan programming (EICD and IICD).
- TM can send compounded settings for parameters and compounded commands.

Example:

- Setting the observing mode for PSS.
- Creation of a sub-array and the allocation of receptors and beams.

Commands (2)

Issues:

- A massive scan reconfiguration can require to specify a high number of parameters.
- Tango accepts/returns only one argument.
- Parameters configuration pass through several layers of hierarchy.

Questions:

How can do it in Tango?

Prototype Proposal:

- I/O arguments for commands are string in Json format. (as in Tango LMC Guidelines)
- Can this solution be considered a Tango anti-pattern?

Capability Strategy

- Capabilities as a different view to the real hardware, more like a mental organization tool.
- Most of processing intelligence inside the CSP.LMC
  - Centralized control.
  - Capabilities as information and configuration container
  - Capabilities can be updated and interrogated by CSP.LMC.

This approach separates hardware handling from logical entities handling, as per Tango approach.

Monitoring Points (1)

From ICD:

- CSP Monitor Points (MPs) are parameters that are periodically. monitored.
- CSP report status of MPs on request, periodically on on changes.
- The CSP (entities) shall implement Health State as MPs.

From Tango Guidelines:

- Each MP is implemented as a Tango Attribute.
- A preliminary list of MPs.

At the moment, there is no uniform monitoring policy.

Monitoring Points (2)

Need to understand what is really useful at higher level, which attributes need to be subscribed and which only read on TM requests.

- Health State: how is it?
- Operational State: what is it doing?
- Usage State: is busy?
- Progress Status: at what point?
- .....
- Detailed status of all components? No thanks!

To be Tango compliant
→ build a hierarchy

CSP SubElement	Number of monitor points
PST	~ 300
PSS	Up to 30.000
CBF	Up to 3.000.000

Monitoring Points Strategy

Analyze Pulsar Search case:

- 750 Nodes - ~ 60 racks - 1500 beams.
- 10 monitoring points for each node.
- 10 monitoring point for each rack of computers.
- ~16 monitoring points for each beam.

Single PSS Node	PSS.LMC	CSP.LMC
sensor1	sensor1[750]	sensor1_max sensor1_min sensor1_mean
sensor2	sensor2[750]	sensor2_max sensor2_min sensor2_mean

Logging (1)

From ICD & LIG:

- CSP.LMC should be required to log to a Central Log file and keep a limited number of latest logs (10).
- CSP Central Log file with Alarms and Events with at least 10.000 most recent records.
- Each CSP Component that has a capacity, maintains own log file.
- Log file rotation when capacity is reached
- Access to TM for copy, search, configuring level, destination.
- Remote logging.
- Use of a standard format, content and logging level as defined in LIG

Logging (2)

What does Tango offer?

Tango includes a logging service (TLS)

- Different targets: console, file, device (Log Consumer -LC)
- Several output layouts (log4tango library)
- Logging levels.
- A graphical interface (LogViewer) based on log4j package, associated to a LC device.
- Log file rotation on dimension basis. Saved only the last one.

Issues:

- Rotation of log files: more than one save.
- All logging targets share the same log level.

Logging (3)

Generally is useful a more verbose output for the log file.

Alternatives:

- Two different files with different logging level (see LIG):
  - Central Log less verbose (> INFO).
  - Local one more verbose (DEBUG).
- Only the Central Log and a Tango LC device which:
  - Acts as a device target for the SubElement devices.
  - Filters out logging messages with DEBUG level.

Prototype proposal:

- The Prototype will log on device and file.
- Implement a log level attribute to configure the logging level.
- Implement a LC Device to filter out the DEBUG level messages.

Alarms handling (1)

No alarm policy defined.

From the LIG & ICD:

- Classification of alarms.
- List Element LMC responsibilities: generation, suppression, filtering, clear, list, add, update.
- Define levels of alarms and messages format.
- Alarm level configurable by TM.
- Define a preliminary list of a Common alarms.
- Logs all CSP Alarms and Events in the Central Log file.

From Tango LIG:

- Use pipe to signal alarms.

Alarms handling (2)

From CSP Requirements:

- Alarm Reporting Latency: CSP.LMC shall send an alarm message within 3 sec.
- Alarm Forwarding Latency: CSP.LMC shall forward alarm message to TM within 1 sec.

Alarm management is critical for the performance and maintenance of any complex system, such as the SKA components.

Alarm handling not possible with only pipes.

Alarms handling (3)

What do we need?

Perhaps an Alarm System that generally provides:

- Generation of basic and key alarms
- Definition of actions to take and of escalation procedures.
- Definition of a system of alarm acceptance.
- Advanced alarm handling techniques:
  - Suppression:
    - Alarm shelving, alarm flood, state based alarms.
  - Alarm reduction.
- Tools for analysis and statistics of alarms log.

Alarms handling (4)

What does Tango offer?

- Generation of basic alarms. No Alarm engine in Tango core.

Tango Community has developed two Alarm Handler devices:

- Alarm System by *Elettra*
- PyAlarm by *Alba*

Prototype Proposal:

- Implementation of the Alarm System.
- Use of a hierarchical structure with CSP.LMC Alarm Handler as central supervisor.

Question:

- Should alarm messages log into the standard log file?
- Is alarm logs policy on file rotation the same of ordinary log file?

Alarms handling (5)

What does the Tango Alarm System offer?

- A read attribute: arrays of alarms strings present in the alarm table.
  - Alarm string defined by several fields are partially equivalent to the LIG ones.
- Alarms log and alarms configuration stored in a MySQL DB.
- A set of commands to ack, list alarms, load/remove.
- Alarm formula, time threshold, actions.

- It can be appropriately configured to support handling of:
  - Nuisance alarms, state-base alarms, escalation procedures.
- No support for Alarm shelving and reduction.

Comments and Suggestions?

Thank you!

Special thanks:
Marina Vela Nuñez for the presentation review
Luca B. for the presentation layout