Dish LMC: a prototype control system for the SKA1-Mid telescopes

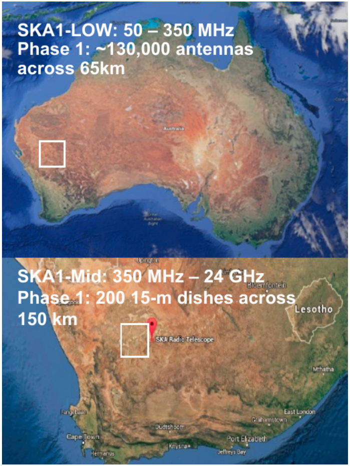
S. Riggi – INAF-OACT

INAF ICT Workshop, 10-14 September 2018 - Catania

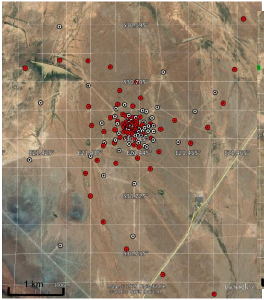
The Square Kilometer Array (SKA)

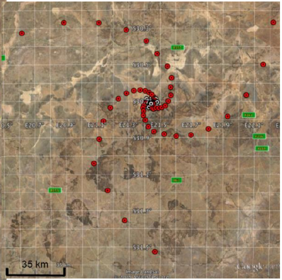
SKA: the largest and most sensitive radio observatory ever built. . .

- **2 antenna arrays** (SKA-Mid, SKA-Low) to be built in South Africa and Western Australia
- **2 construction phases foreseen**
  - *Phase 1* (2018-2024): SKA1-Mid + Meerkat + SKA1-Low (cost cap: 675 M€)
  - *Phase 2* (mid 2020s): expansion of both arrays in Southern Africa and Australia
- **Pre-construction phase ongoing**
  - lead by 8 Consortia (**Dish**, TM, CSP, LFAA, SDP, SaDT, INFRA, AIV) coordinated by the SKA Organization (SKAO)
  - 10 member countries (more to join), >100 research institutions and companies across 20 countries
  - Consortia busy to complete the *Critical Design Reviews* (CDRs)
- Moving to an **intergovernmental organisation (IGO)** for long-term governance and funding



SKA1-Mid Dish array





- 133 15-m diameter dishes + 64 13.5-m Meerkat dishes
- Spread over ∼150 km in the South Africa's Karoo region
- 5 frequency bands foreseen: 0.35-13.8 GHz
  - Band 1, 2, 5 (5a, 5b) prioritized in Phase 1
  - Band 5 installed in 67 dishes only
- Expected performance boost ∼5-10 wrt existing arrays
  - Sensitivity: $\sim$0.75 μJy hr$^{1/2}$ (continuum), $\sim$66μJy hr$^{1/2}$ (spectral-line)
  - Resolution(@1.67 GHz): ∼0.25"
  - Survey speed: $\sim 1.19\times10^6$ deg^2 m^4 K^2
- Extension to ∼2500 antennas in SKA 2

SKA Dish design organization & status

Design roadmap (T0: 4 Nov 2013)

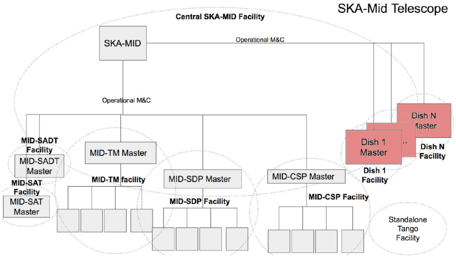
- May 2014: *Concept Design Review* (CoDR)
- Nov-Dec 2014: *Preliminary Design Review* (PDR)
- May-Dec 2016: *Detailed Design Review* (DDR)
- Feb 2018: *First SKA dish prototype assembled in China (SKA-P)*
- Mar-June 2018: *Second SKA dish prototype assembled in South Africa (SKA-MPI)*
- May-Dec 2018: *Critical Design Review* (CDR)
- Early 2019: *Dish Qualification & Integration on SKA-MPI*
- 2019-2020: *Construction bridging + Early Production Array (EPA)* (TBD)

Dish Consortium organized in 4 major work packages or sub-elements

- **LMC** (Local Monitoring and Control): INAF (OACT + OATS), SAM, EIE (Italy)
- **SPF** (Single Pixel Feeds): Onsala Space Observatory (Sweden), EMSS (SA), Oxford University/STFC (UK)
- **SPFRx** (Receivers): NRC (Canada) + Bordeaux University (France)
- **DS** (Dish Structure): CETC54 (China) + MTM (Germany) + SAM (Italy)

The SKA Control System (CS)

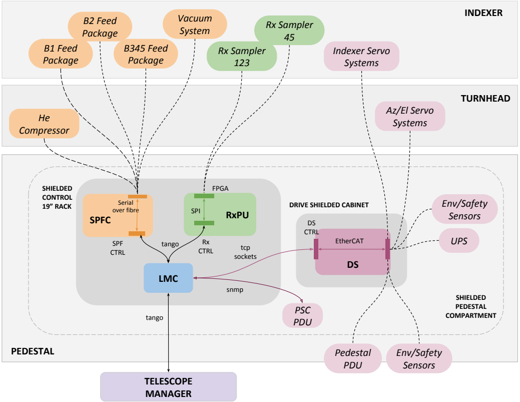
- **SKA CS: a technological and organizational challenge**
  - large & eterogenous system ($>10^6$ moni points), scalability & future maintenability
  - group geographical dispersion & previous background
- **Harmonization process main outcomes** (see *SKA Control Guidelines*)
  - *Tango* selected over other frameworks (EPICS, ACS, Meerkat CAM) (Mar 2015)
  - Standardized architectural components (Element master, logging, alarm handler, archiver)
  - Standardized *SKA Control Model* (SCM) (operating states/modes, . . .)
  - Standardized control & configuration patterns
  - Standardized programming languages & tools
  - Some areas still to be covered (e.g. deployment, configuration, testing strategies, etc)



- **SKA CS organized in hierarchical Element facilities** (=Tango domains)
  - Each SKA Element has a Local Monitoring and Control (LMC) sub-element
  - LMC providing dish master control and rolled-up monitoring to Telescope Manager (TM)

The SKA Dish Control System

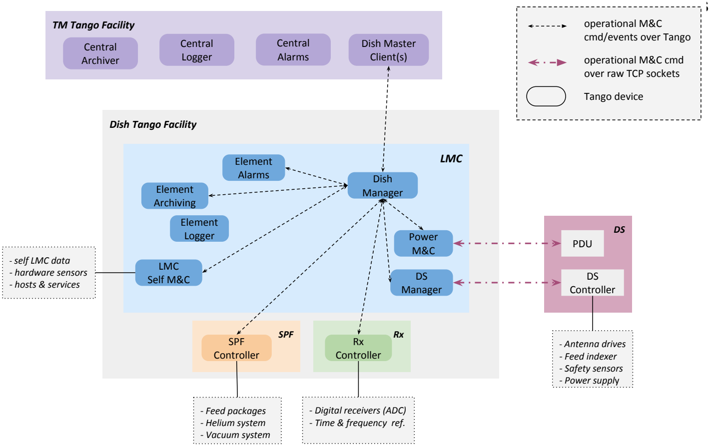
- Dish equipment distributed in 3 zones (*pedestal*, *yoke*, *indexer*)
- **SPF**: M&C of feed packages (B1/B2/B345), Helium/Vacuum systems
  - 6 Tango Device Servers (TDS) provided (one per sub-system + main controller) to LMC
  - Low-level interface: serial over fibre
- **SPFRx**: M&C of receiver digiter system (B1,. . . ,5a, 5b)
  - 1 TDS provided to LMC
  - Low-level interface: SPI



- **DS**: low-level M&C of antenna servo/safety systems and power distribution
  - No TDS provided, interface via custom protocol (tcp sockets)
  - Low-level interface: *Beckoff EtherCAT*
- **LMC**: high-level Dish M&C (logging/archiving/. . .) towards TM, PDU control
  - 10 TDS provided
- Relatively small system from the M&C view
  - Moni points: <1000
  - M&C data rate flowing centrally: $\sim$200 kbps
  - Fastest M&C rate: 100 ms (ACU loop is faster)
  - Moderate number of TDS: <20

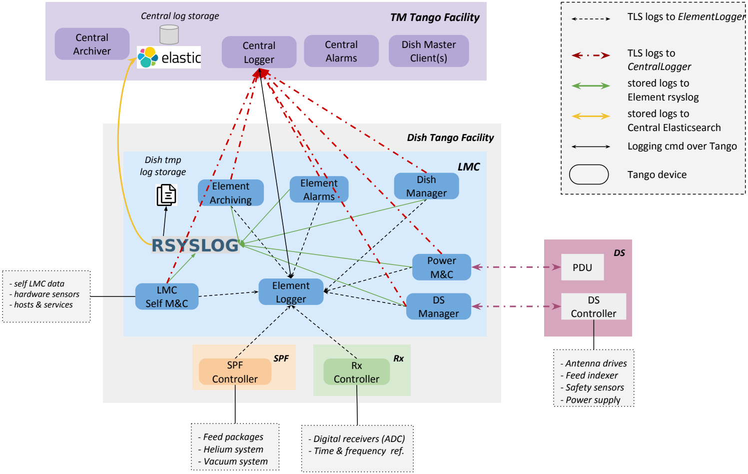
Dish LMC Design

- **10 Tango device servers (TDS) in LMC**
  - 1 Dish master TDS providing high-level control and rolled-up monitoring status to TM
  - 1 logger, 4 archivers, 1 alarm handler TDSs
  - 1 DS manager TDS, providing Tango interface to DS subsystems
  - 1 PDU manager TDS, providing PDU pedestal control
  - 1 LMC monitor TDS, providing interface to Nagios and self monitoring parameters



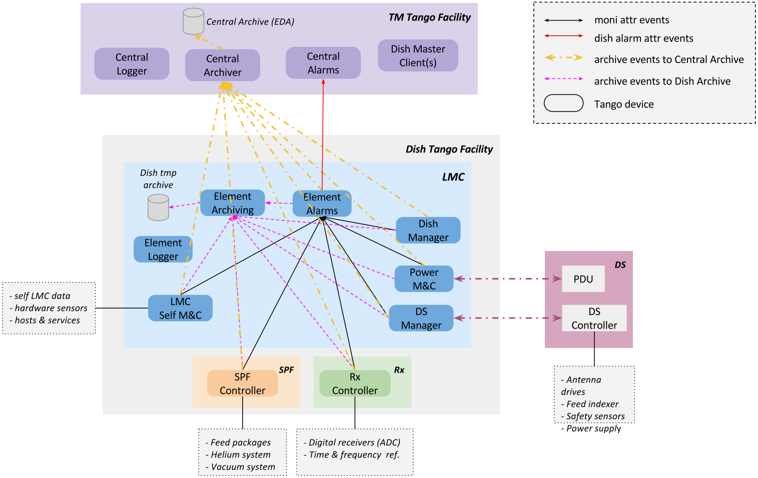
Dish LMC - Logging

- **Logging data flow: 3 logging targets per Tango DS**
  - *Central logger* (Tango *LogConsumer*) for cross-facility log viewing @ default WARN level
  - *Element logger* (Tango *LogConsumer*) for viewing purposes @ default INFO level
  - *rsyslog server* for tmp log storage (+ forwarding to central *Elasticsearch* for permanent storage & analysis) @ default INFO level

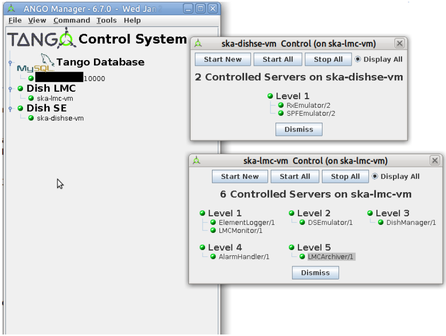


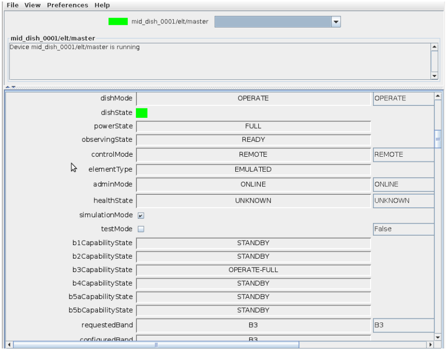
Dish LMC - Alarm & Archiving

- **Alarm and archiving data flow**
  - Alarms interpreted on behalf of dish sub-elements using alarm rules (compliant to *IEC 62682 - Alarm Management Lifecycle* standard)
  - Dish data archive is temporary (12 h at least)



Dish LMC software prototype



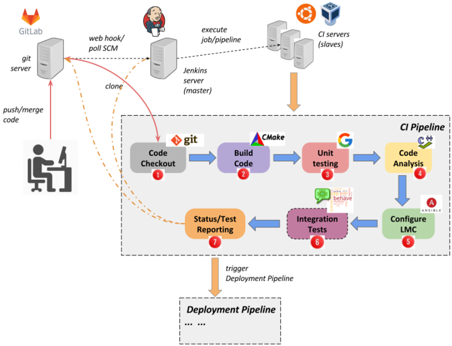


- **LMC prototype implementation started after DDR Phase**
  - Main scope: architecture validation, dish integration
  - Adopted technologies
    - ✓ Developed in C++/python, Tango 9
    - ✓ Addon libs/tools: Nagios 4, Tango Addons (hdb++/yat/alarm), boost/log4cxx/pugixml, . . .
    - ✓ Build system: cmake
    - ✓ Version control: git
    - ✓ Unit/Integration testing: Google Test, pytango/nose
    - ✓ CM + CI: Ansible + Jenkins
    - ✓ OS: Ubuntu 14.04/16.04, CentOS 7

Dish LMC - Tango addons

- **Some SKA/Dish required functionalities missing in Tango framework**
  - Pre-configured SKA loggers, including syslog target
  - No attribute transition rules or attribute-based command state machine (only for DevState)
    - ✓ Support SKA/Dish Control Model (mostly based on DevEnum attribute)
  - Formula attributes (not in Tango Core, community device available)
    - ✓ Support rolled-up dish monitoring
  - Control task sequencer not available
    - ✓ Support dish control operations and event handling
- **Developed additional features in a base Tango device (*LMCDevice*)**
  1. *Standardized logging*: syslog target, custom logging macros
  2. *Attribute transition & command state-machine rules* (loaded at device startup)
  3. *Dynamic attribute generation from XML config files* (based on yat4tango lib)
     - ✓ follow ICD continuous changes, ease of integration with CM tools (Ansible), supporting command & attribute rules
  4. *Formula attributes*: based on C++ Exprtk lib
  5. *Device proxy utilities*: proxy, event and handlers registration, proxy monitoring
     - ✓ Registration: manually (e.g. via macros), from properties, automatically for formula vars
  6. *Task sequence management*
     - ✓ tasks (=device commands) pushed in a queue and processed in a dedicated thread according to priority, start/timeout time, dependency on other tasks in the sequence
     - ✓ status of long running tasks determined from dedicated progress attributes
- **Other SKA Elements implemented similar or additional functionalities**
  → *LMC Commons*

Dish LMC - Development process



- **Simple CI pipeline set up for pre-construction**
  - *Jenkins* to manage configuration and deployment pipelines
  - Pipeline jobs triggered manually/on commits
  - LMC development strategy being standardized at SKA level for construction
- **Development status**
  - 6/7 LMC servers implemented
  - *DSManager* server implementation ongoing (DS-LMC ICD delayed)
- **LMC configuration approach**
  - Initial server configuration automated with *Ansible* playbooks (XML template config files)
  - LMC server configuration stored in TangoDB
  - Current configuration can be changed at runtime using Tango UI (*Jive*) or *pytango* API
- **LMC deployment approach**
  - LMC deployed on a set of virtual machines
  - Expected to move to *Docker*/Singularity containers for construction (TBD)

Dish LMC - Testing process

- **Testing process**
  - $>$100 test procedures designed for CDR
  - Software interface simulators (SPF, Rx) developed for testing purposes (based on ICD)
  - Non-automated testing done using Tango builtin UI tools
  - Automating qualification testing
    - ✓ Test procedure scripts implemented in python using pytango API (a script per test case)
    - ✓ Test framework provided by LMC Eurosoft Srl partner
- **Testing status**
  - First qualification testing performed in both manual and automated way (May-Aug 2018)
  - $\sim$40% automated testing coverage (wrt to manual tests)
  - Overall testing and integration ongoing
- **Potential improvements and developments**
  - *Behavior Driver Development (BDD)* approach explored
  - Qualification tests defined in *Gherkin* textual language (masking details to stakeholders)
  - Many frameworks available: python behave + nose considered for prototyping
  - Integration with *Jenkins* tested

```
@stow_dish
Feature: Stow SKA dish
  For dish qualification I want to test dish stowing operation
  under different dish mode cases

  Background: Control system is started
    Given TangoDB is up and running
    And LMC is up and running

  Scenario Outline: Stow dish in STANDBY-LP
    Given I am an "<LMC>" client (TM or operator)
    And Dish is in STANDBY-LP mode
    When I request dish LMC to stow
    Then I should be acknowledged that stowing was initiated by LMC
    And I should be notified within 30 s that dish is in STOW mode

  Examples: Device name
    |           LMC            |
    | mid_dish_0001/elt/master |
    | dish_master              |
```

Features Statistics

The following graphs show passing and failing statistics for features

Features

	Steps						Scenarios			Features	
Feature	Passed	Failed	Skipped	Pending	Undefined	Total	Passed	Failed	Total	Duration	Status
Stow SKA dish	0	2	14	0	0	16	0	2	2	000ms	Failed
1	**0**	**2**	**14**	**0**	**0**	**16**	**0**	**2**	**2**	**000ms**	
	0.00%	**12.50%**	**87.50%**	**0.00%**	**0.00%**		**0.00%**	**100.00%**			**0.00%**

Summary

- **SKA Dish is completing the CDR phase**
- **SKA Dish LMC prototype implemented and under test**
  - Experience with Tango & interaction with community very positive
  - Tango extensions developed to fullfil SKA & dish requirements, pull requests made for Tango v10
  - Interface with Dish Structure missing to complete the prototype
  - Focusing on DS integration, testing and documentation for the next months
- **Short-term roadmap**
  - CDR review (Sep 2018)
  - post-CDR activities (Oct-Dec 2018)
  - LMC integration on the field (mid 2019)
- **Towards SKA construction (construction bridging)**
  - setup and integration in the SAFE framework for the development of SKA software
  - development of SKA common libraries and additional SKA standards (LMC Commons)
- **SKA construction is TBD!**
  - Waiting for IGO and INAF decisions. . .