The Square Kilometer Array (SKA): the world’s largest and most sensitive radio observatory

- 2 radio antenna arrays (Fig. 1)
  - SKA1-Mid: South Africa
  - SKA1-Low: Western Australia

- 2 construction phases foreseen
  - Phase 1 (2018-2024): SKA1-Mid + SKA1-Low arrays
  - Phase 2 (mid 2020s): expansions of both arrays in Southern Africa and Australia

- SKA pre-construction phase 1 ongoing
  - Lead by 8 major consortia coordinated by the SKA Organization
  - 10 member countries, >100 research institutions and companies across 20 countries
  - Consortia completing their critical design

SKA DISH OVERVIEW

Dish instrumentation (Fig. 3) mounted at 3 major locations (indexer, pedestal, yoke) and designed by:

- Dish Structure (DS): antenna structure and optics, feed indexer, servo systems, power distribution and safety systems.
- Single-Pixel Feed (SPF): feed packages (OMTs, LNAs), feed helium cooling and vacuum system and relative controllers.
- Receiver (SPRxs): RF digitizer system and relative controllers.

DISH LMC ROLE IN SKA CS

- SKA Control System (CS) design (Fig. 4)
  - based on Tango framework
  - organized in hierarchical Element facilities (~Tango domains)
  - Harmonized across Elements with common guidelines & patterns

- Each Dish has a Local Monitoring and Control (LMC) system
  - LMC provides dish master control and rolled-up monitoring to Telescope Manager (TM)
  - TM remotely coordinates the telescope scientific and M&C operations

- Dish LMC “size” from a M&C view
  - Monitoring points: <2000
  - M&C data flow to TM: ~200 kbps
  - Fastest M&C rate: 100 ms
  - Number of Tango Device Servers (TDS): <20

DISH LMC DESIGN AND PROTOTYPE

- Dish LMC high-level architecture designed (Fig. 5)
  - 1 master TDS for high-level Dish M&C
  - 1 TDS controller per Dish sub-element
  - 1 logger/archive/alarm handler TDS per Dish instance
  - Additional TDS for LMC self M&C, power control

- Prototype implementation started
  - Aim: design validation + dish qualification
  - Technologies selected (Table 1)
  - Tango extensions implemented

Table 1: List of Dish LMC prototype employed technologies

<table>
<thead>
<tr>
<th>Type</th>
<th>Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program language(s)</td>
<td>C++ (control system), python (configuration &amp; testing)</td>
</tr>
<tr>
<td>M&amp;C framework</td>
<td>Tango 9</td>
</tr>
<tr>
<td>Tango tools</td>
<td>PyTango, TangoBuilder, ELIXIR Alarm System, RRM++ Archer, Tango GUI</td>
</tr>
<tr>
<td>Libraries &amp; tools</td>
<td>boost, pyOpenPip, pyOpenPip, Nagios 4</td>
</tr>
<tr>
<td>Build system</td>
<td>cmake</td>
</tr>
<tr>
<td>Version control</td>
<td>git (GitHub)</td>
</tr>
<tr>
<td>Testing</td>
<td>Google Test, pyElxTest, nose</td>
</tr>
<tr>
<td>Configuration management</td>
<td>Ansible</td>
</tr>
<tr>
<td>Continuous integration</td>
<td>Ansible</td>
</tr>
<tr>
<td>Documentation</td>
<td>ELixir + SphinxBridge</td>
</tr>
<tr>
<td>Virtualization &amp; OS</td>
<td>VirtualBox, Ubuntu 14.04</td>
</tr>
</tbody>
</table>

Contact: s.riggi@oact.inaf.it

(*) Full author list available in the accompanying paper