Present, past and future of the development of the control software for the Italian radio telescopes

Sergio Poppi
on behalf of the DISCOS team

DISCOS Team

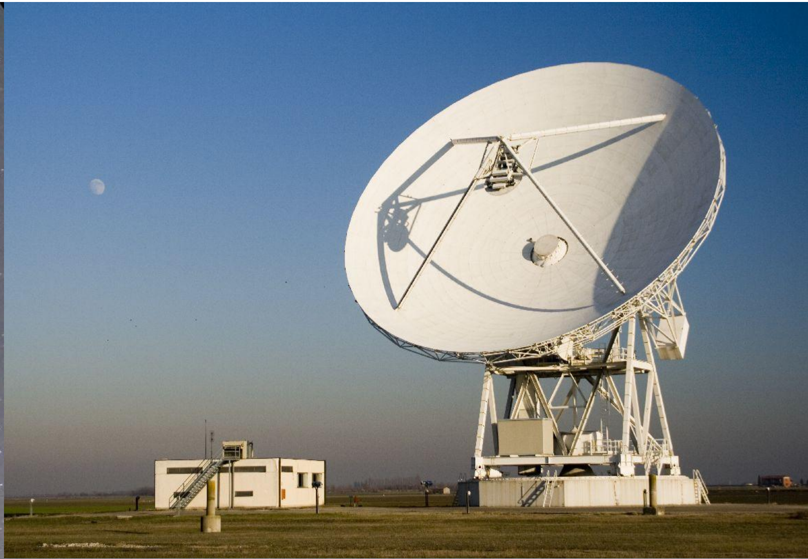
- **Andrea Orlati (INAF - IRA): team leader - project manager - core developer**
- **Carlo Migoni (INAF - Cagliari):** core developer, VLBI integration.
- **Marco Buttu (INAF - Cagliari):** core developer, test driven development.
- **Giuseppe Carboni (INAF - Cagliari):** simulators, continuous integration.
- **Simona Righini (INAF - IRA):** astronomical advisor, observations, documentation.
- **Antonietta Fara (INAF - Cagliari):** system administrator.
- **Sergio Poppi (INAF - Cagliari):** core developer, astronomical advisor, observations

\+ Many contributions over the time
M.Bartolini, F.Palagi, G.Maccaferri, M.Murgia,
F.Schilliro', M. De Biaggi

Project History

- 2004 Development SRT Control Software - NURAGHE started
  - team: Giuseppe Maccaferri, Andrea Orlati, Francesco Palagi, Carlo Migoni, Matteo Murgia, Francesco Schillirò (GAI SOFTWARE - SRT)
  - Goal:
    - Provide the Sardinia Radio Telescope of control software with enhanced performances.
    - Build a common infrastructure for the three radio telescopes.
- 2007 ESCS Enhanced Single-dish Control System (Medicina and Noto)
  - team: GAI SOFTWARE + Simona Righini, Rashmi Verma, P.Libardi
- 2015 DISCOS: unifies the three development lines.
- 2018 INAF UTG-II endorsement


Sergio Poppi©



Telescopes Configurations

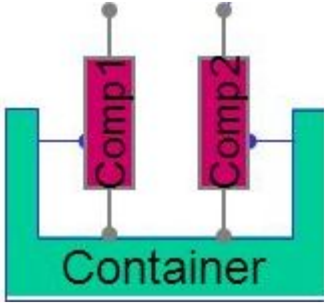
	SRT	Medicina	Noto
Main mirror	64 m	32 m	32 m
Optical configuration	Gregorian	Cassegrain	Cassegrain
Mount	Altazimuthal, fully steerable 12 motors + cable wrap	Altazimuthal, fully steerable 4 motors	Altazimuthal, fully steerable 4 motors
Antenna Control Unit (main servo system)	Beckhoff PLC ethernet vendor protocol	VxWorks based PC ethernet vendor protocol	VxWorks based PC ethernet vendor protocol
Primary Focus	three degrees of freedom INAF defined protocol	three degrees of freedom INAF defines protocol	
Secondary Focus	six degrees of freedom ethernet INAF protocol	five degrees of freedom ethernet INAF protocol	five degrees of freedom RS232 vendor protocol
Active Surface	1008 aluminium panels 1116 actuators rs485/ethernet vendor protocol	not available	240 aluminium panels 244 actuators rs232 vendor protocol

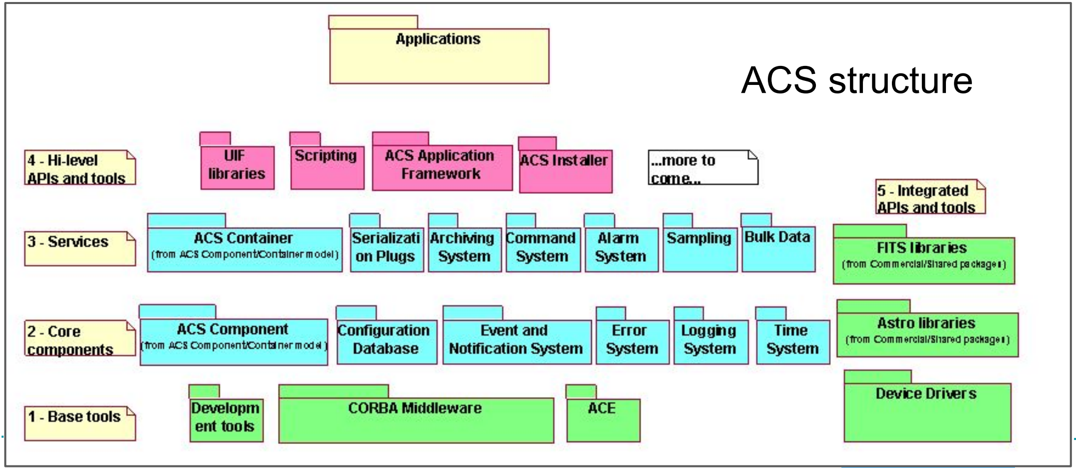
Telescopes Configurations

	SRT	Medicina	Noto
Main mirror	64 m	32 m	32 m
Receivers *	0.305-0.410 1.3-1.8 5.7-7.7 18.0-26.0, 7 feeds GPIB and ethernet INAF protocol	1.35-1.45 1.595-1.715 2.2-2.36 4.30-5.80 5.90-7.10 8.18-8.98 18.0-26.0, 2 feeds GPIB and ethernet and RS232 various protocols	0.317-0.320 1.40-1.72 2.20-2.36 4.70-5.05 8.18-8.58 22.18-22.46 39.0-43.3 GPIB and RS232 various protocols
Backends*	TotalPower (continuum) 0.1-2.1, 1-1000 ms, 14 inputs XARCOS (spectro-polarimetry) 0.0005-0.125, 10 s, 2048 bins, 14 inputs SARDARA(spectro-polarimetry) 1500MHz , 10-1000 ms,16384 bins, up to 14 inputs DFB3(pulsar) 1.024, 1-4000 ms, 8192 bins, 4 inputs DBBC	TotalPower (continuum) 0.1-2.1, 1-1000 ms, 4 inputs XARCOS (spectro-polarimetry) 0.0005-0.125, 10 s, 2048 bins, 14 inputs	TotalPower (continuum) 0.1-2.1, 1 ms, 4 inputs DBBC

DISCOS framework

- Applications based on ALMA **Common** Software
  - Distributed objects architecture
  - ACS component as the basic unit which performs tasks
  - Components expose interfaces to other components.

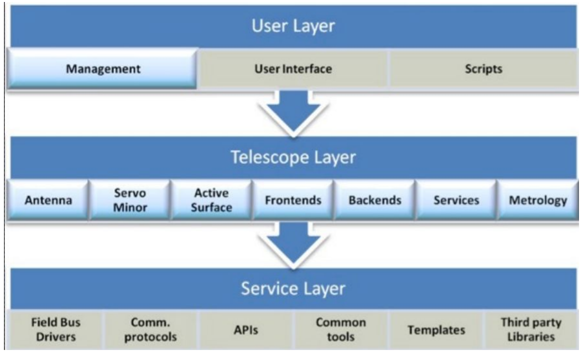


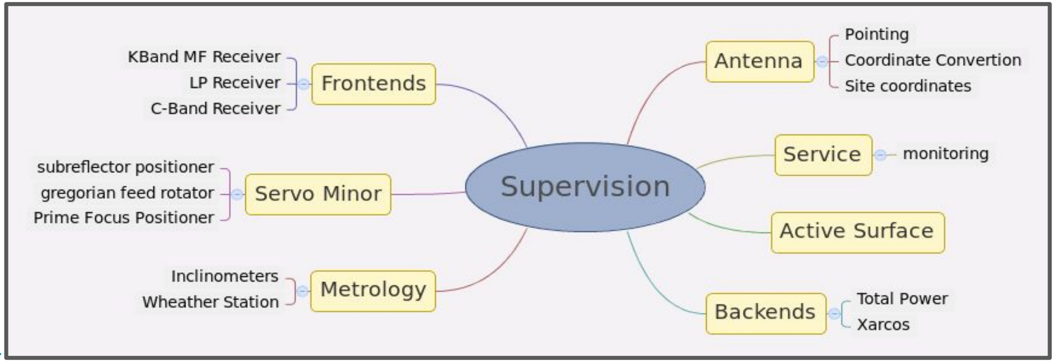


ACS structure

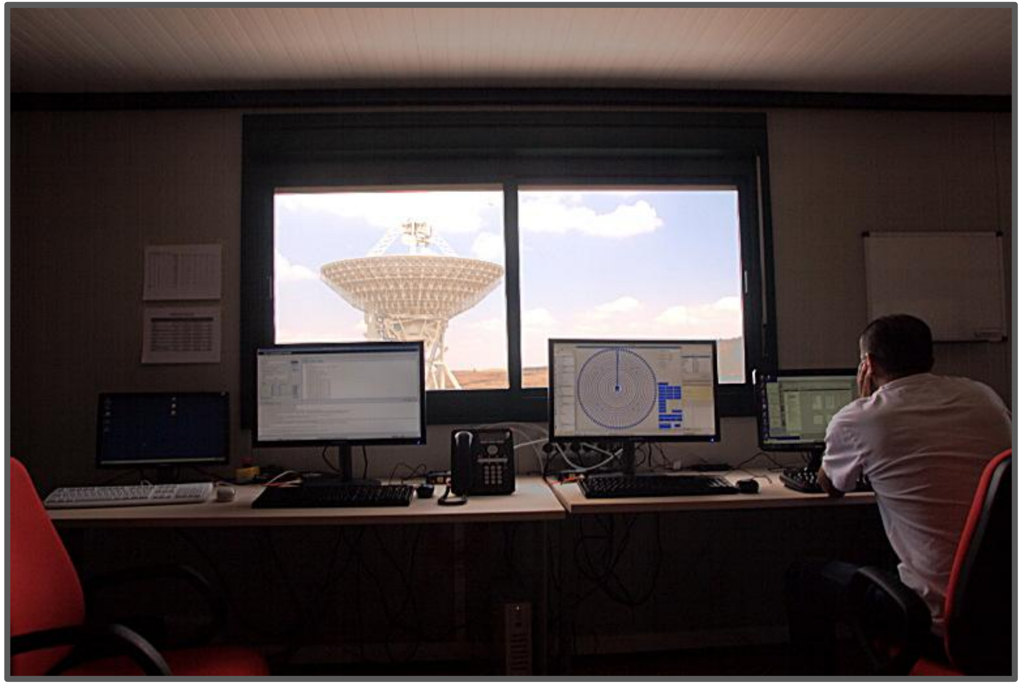
DISCOS Design

- **Common** interfaces design for the three telescopes
- Components organised in subsystems
- Each subsystem has a “boss” component, which has in charge the communications inward and outward the subsystem



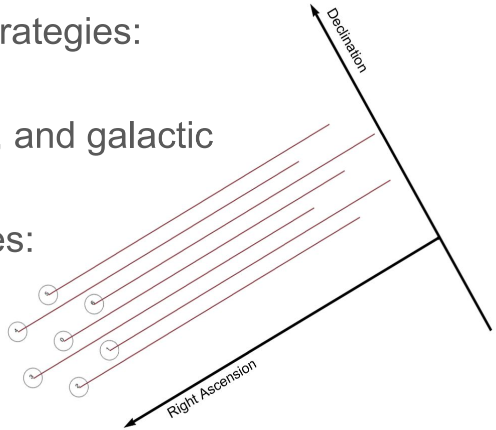


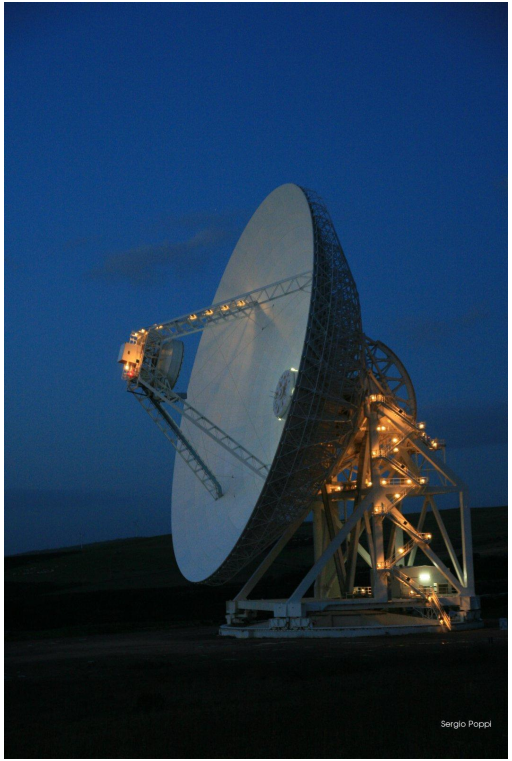
What the control software must do?



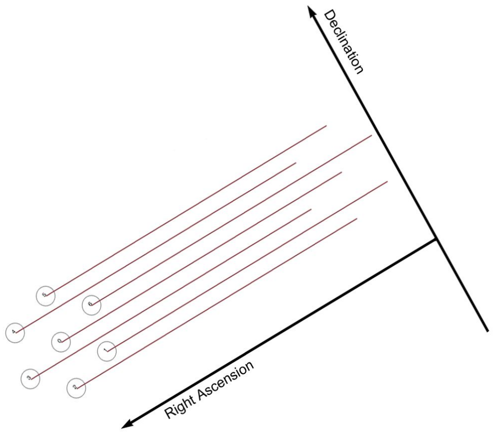
It drives the telescope...

- The antenna control unit needs time tagged azimuth and elevation coordinates
  - The antenna subsystem computes t-tagged coordinates to perform scanning strategies:
    - Sidereal tracking
    - OTF in equatorial, horizontal, and galactic coordinate system
    - Spectroscopy mode strategies:
      - Position switching
      - Nodding
      - Raster mapping





Observing modes

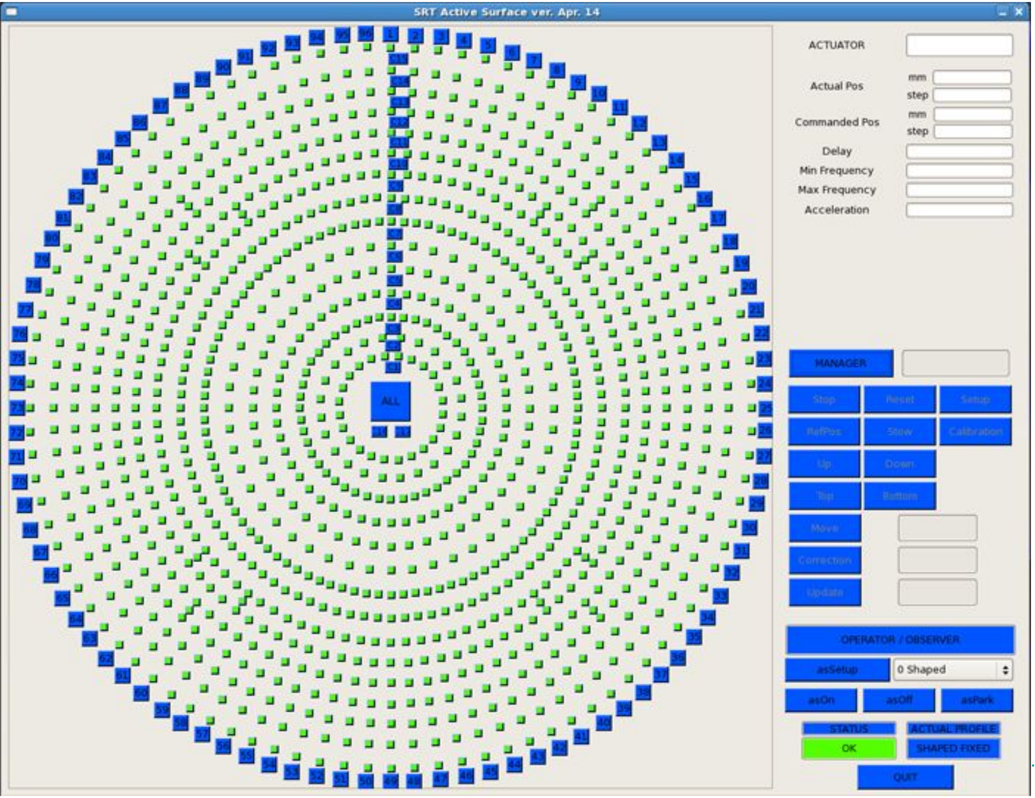


- Observing modes are:
  - Sidereal tracking (pulsars and VLBI)
  - On-the-fly Mapping or scanning (continuum, spectropolarimetry)
  - Spectroscopy mode strategies:
    - Position switching
    - Nodding
    - Raster mapping
- Italian radio telescopes are versatile instruments for different scientific cases

It set the receivers in the right focal position...

- The minor servo subsystem:
- Set the chosen receiver into it focal position
- Drives the subreflector tracking the best focus as the elevation changes

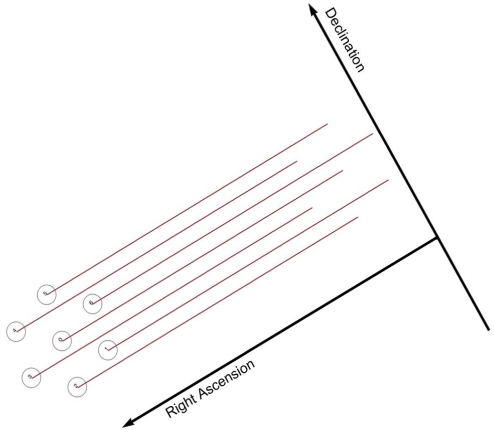
It allows the primary mirror to be in “good shape”

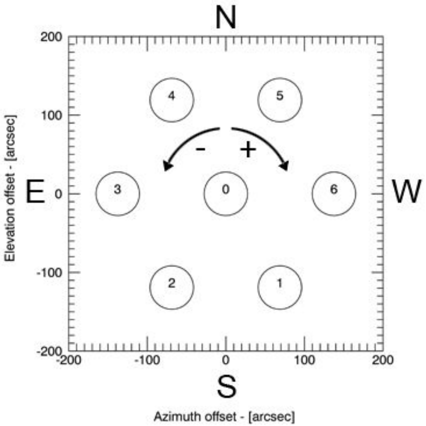


- The AS subsys set each of the 1116 actuators to its elongation for commanded elevation

It derotates multifeeds

- The Multifeed K-band receiver hosts an hardware derotator to follow the parallactic angle
- If needed, it rotates the receiver following the paralactic angle





Allows data acquisition with housekeeping infos!!!

- Data ara acquired by digital backends (total power, ROACH2 based BE) at high sampling rate (down to 10 ms)
- Housekeeping ancillary infos are mandatory. For each sample there are
  - Data
  - Timestamp
  - Pointing coordinates
  - Weather parameters
  - Goodness of the sample (all the subsystem where working correctly?)

What if the sample rate is one sample every 10 ms?

Discos must continuously

1. **Check that the telescope, at time t, is at the right position**
2. **Check each device, at time t , is at the right position**
3. **Save the acquired data each 10 ms, together with a status flag (1 if all the check are ok) together with the all the other HK parameters**

- Synchronism is critical

Another Critical point: data rate!

- Totalpower BE produces   MAX 64 KB/s
- SARDARA ROACH2 BE (16384) produces 128 MB/s (7-beam mode 900 MB/s)

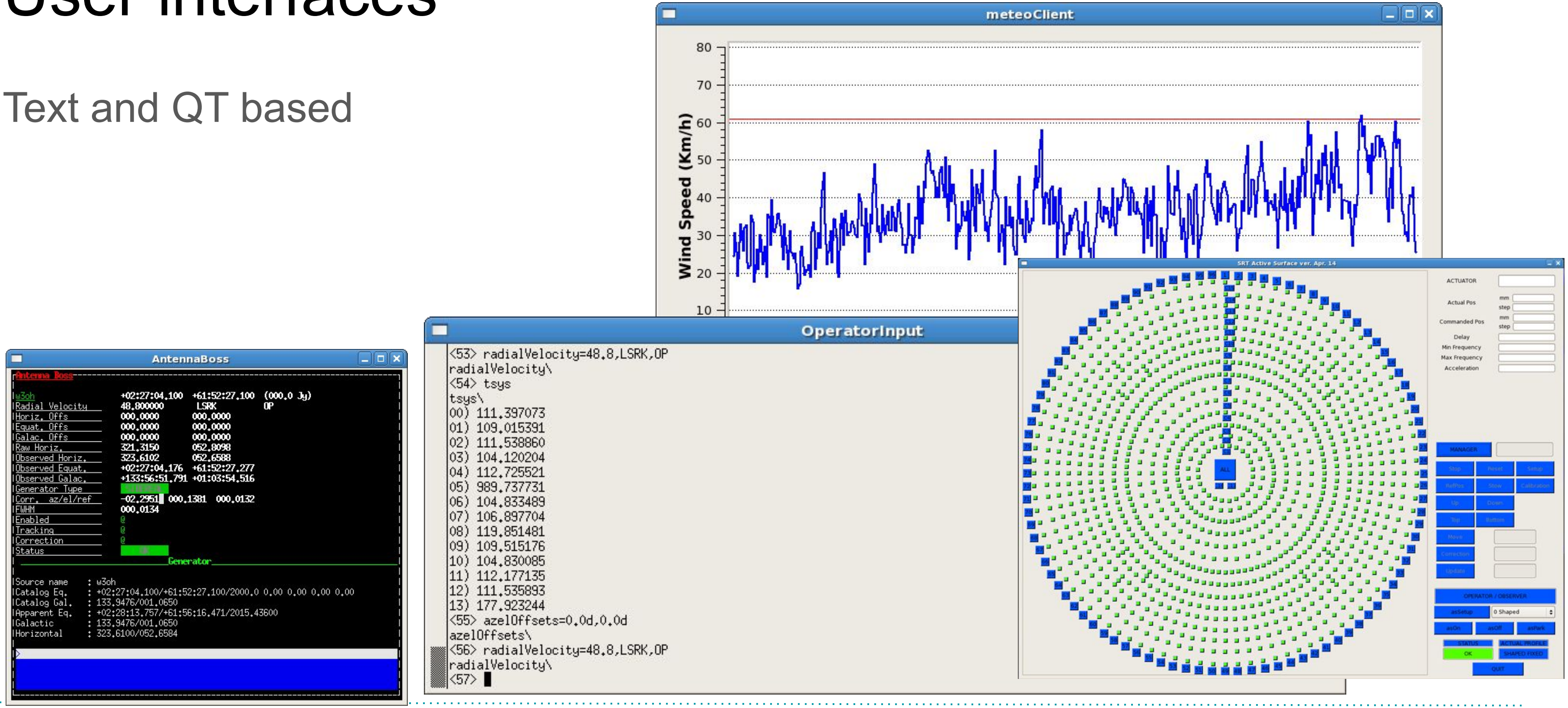
The data rate is close to the disk rate limit

Optimizing disk and filesystem is mandatory!

Are we facing BIG DATA problems?

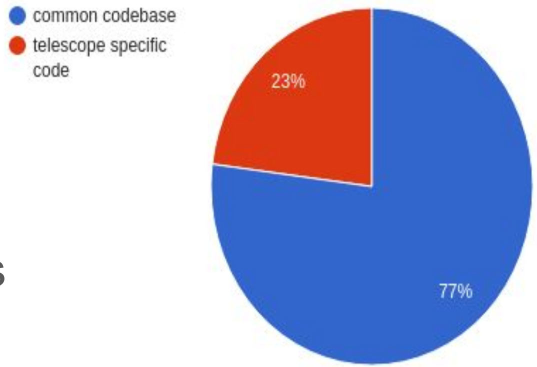
User interfaces

Text and QT based



DISCOS in a nutshell

- A common monolithic codebase (77%):
  - management (scheduling, observing modes)
  - subsystem bosses
  - user interfaces
  - libraries
- Specific code coping differences among telescopes



Lesson Learned - How big is Discos

Totals grouped by language (dominant language first):

cpp:	72.59%
xml:	16.26%
ansic:	5.84%
python:	5.03%
sh:	0.25%
fortran:	0.03%
perl:	0.00%

Total Physical Source Lines of Code (SLOC) = 533782

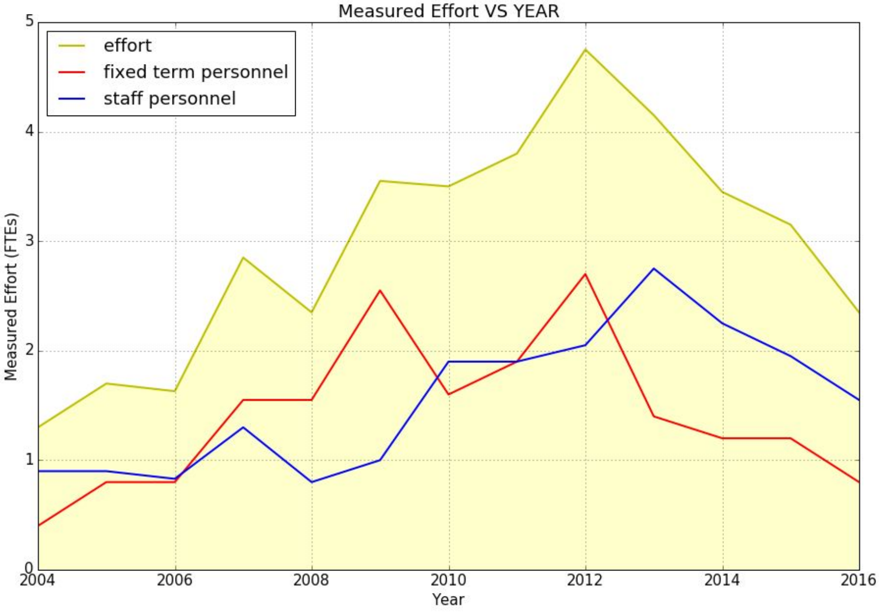
generated using David A. Wheeler's 'SLOCCount'.

statistics by www.openhub.net

Astropy Codebase Statistics	
First commit date	25 July 2011
Number of distinct contributors	191
Number of commits	15,475
Lines of Python code	200,295
Lines of Python code (core package)	125,737
Lines of C code	13,957
Lines of shell code	1,065
Total lines of code	215,317
Development effort estimate (person-years)	56.33
Schedule estimate	2.48 years with 22.72 developers
Estimated cost	$8,449,937

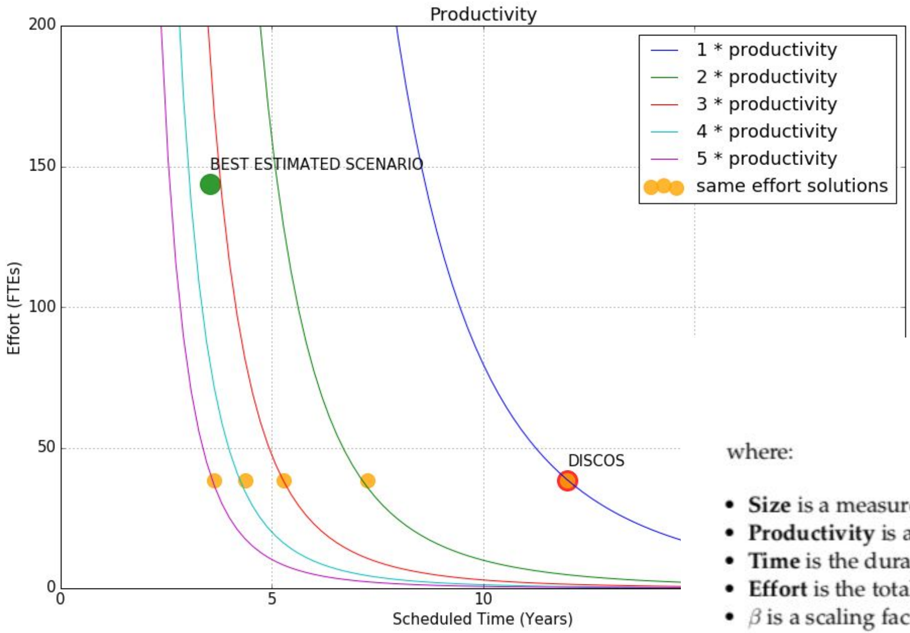
Lesson Learned - Costs analysis (2016)

Project	FTE	% of Project Costs
Alma	450	5.7
VLT	422	7.1
ASKAP	37	4.3
SRT	39	2.8



Total costs: around 1600 k€

Lesson Learned - Cost in-effective planning



$$Size = Productivity * Time^{\frac{4}{3}} * \left(\frac{Effort}{\beta}\right)^{\frac{1}{3}}$$

where:

- **Size** is a measure of the final size of the project, here we use LOC counts
- **Productivity** is a productivity factor of the organization
- **Time** is the duration of the project schedule in Years
- **Effort** is the total effort of the organization in person-years
- β is a scaling factor for the effort depending on the project size

Lesson Learned - Design and Requirements

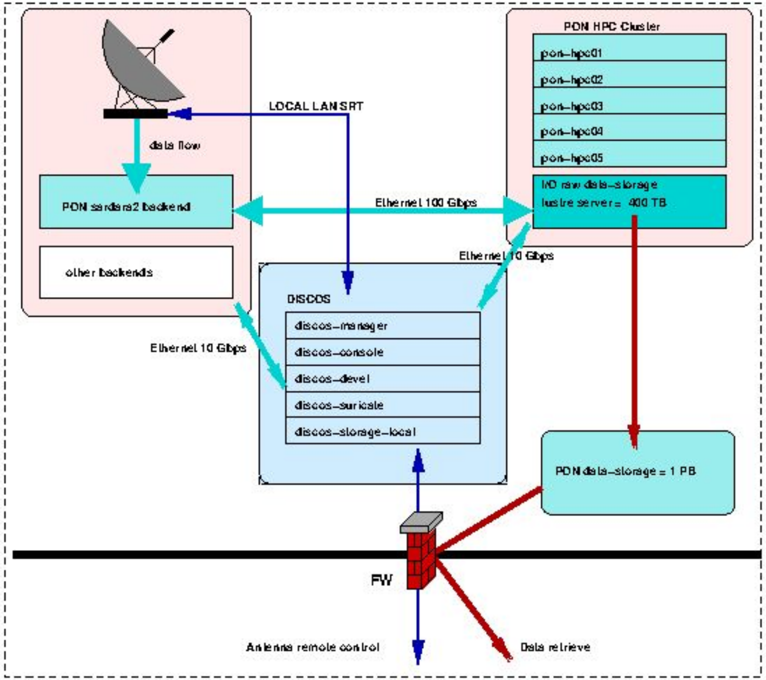
1. Speed up technology surveys in order to anticipate the start of the development.
2. The requirements from your stakeholders (i.e. astronomers and scientists) not well defined
3. Underestimate the technology progress during the years, see Big Data and data rates.
4. Code with poor test coverage
5. Delay the design and the development of your GUI to a later stage
6. Relaying on a framework is certainly a great benefit especially during development but, on the other hand, you lose control on a fundamental part of you application and you are anchored to technologies which tends to grow old rapidly

Lesson Learned - Drawbacks

Automatic machines provisioning

Ansible

- Network setup
- shared file system setup
- Users creation
- dependencies installation (e.g. ACS)
- observing tools installation
- DISCOS installation

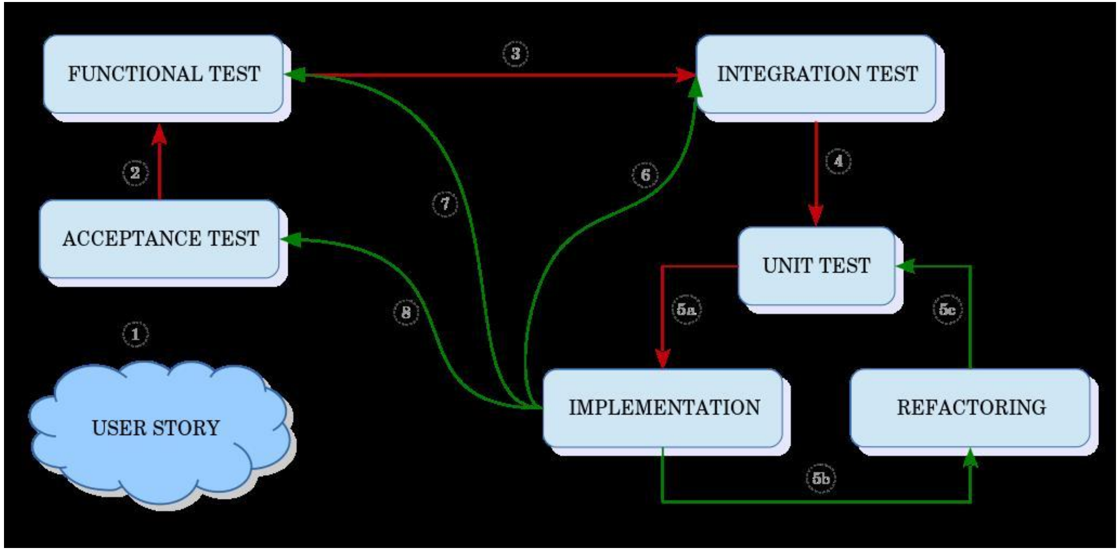


ANSIBLE

HashiCorp Vagrant

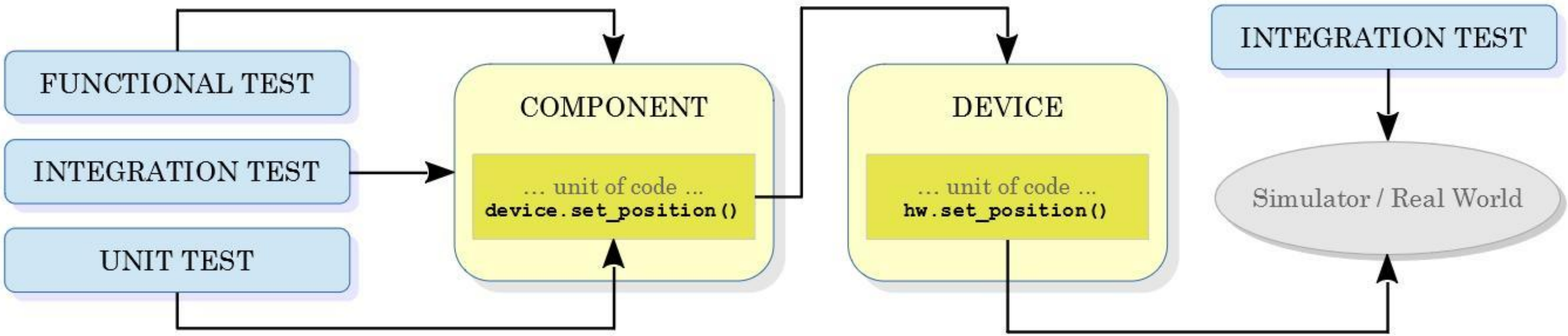
Test driven development

- Development of new functionality will be test driven



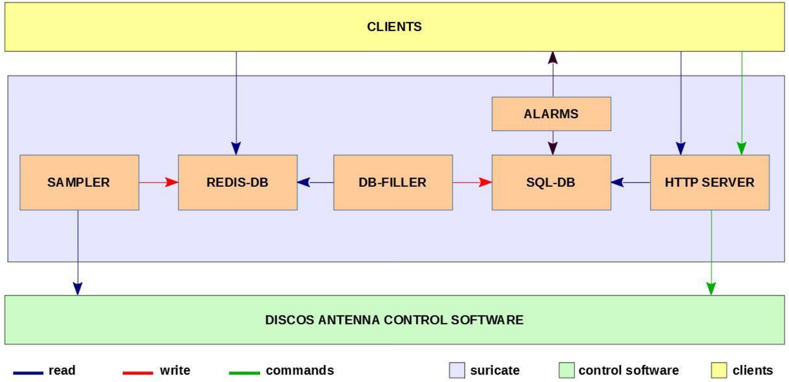
Testing with Simulators

- Simulators provide a means to perform both high and low level offline tests
- Same tests should yield the same results with both simulators and hardware
- We can verify that the real hardware APIs behave as expected



Suricate

- Handle communication inward and outward the CS
- Clients do not depend on the framework.



Future/present

- Italian radiotelescopes will be upgraded within PON Research and Innovation 2014-2020 project “Potenziamento del Sardinia Radio Telescope per lo studio dell’Universo alle alte frequenze radio”
- New receivers will be installed:
  - cryogenic receiver in W Band for SRT (75-116 GHz)
  - cryogenic receiver in Q Band for SRT (33-50 GHz)
  - Millimetre camera for SRT (80-116 GHz)
  - triple-Band receiving system for the three Italian radio telescopes (18-26 ; 35-50; 85-116 GHz)
- Metrology: shape and structure deformations measure to have closed loop control of mirrors
- New backends/detectors

Questions?

See:

https://discos.readthedocs.io/en/latest/

https://github.com/discos

https://www.ict.inaf.it/gitlab/INAF/DISCOS