



throughout the years

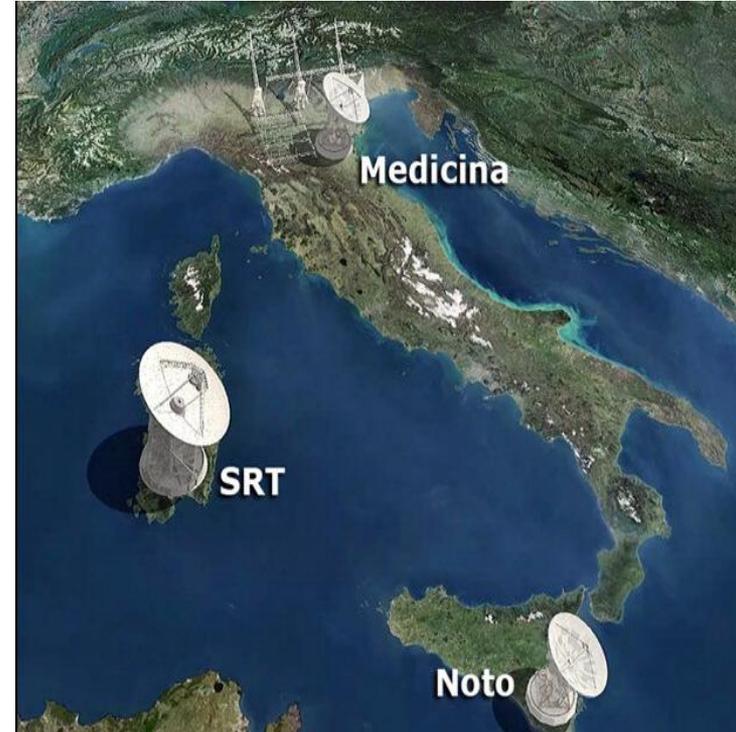
Evolution and modernization of the Italian single-dish radio telescopes control software

Giuseppe Carboni – INAF – Osservatorio Astronomico di Cagliari

USC-C General Assembly – 9 - 13 March 2026 – Trieste

DISCOS

- *Development of Italian Single-dish **C**ontrol System*
- Control software for the INAF single-dish radio telescopes
- Distributed architecture based on *Alma Common Software (ACS)*
- Controls and coordinates all the telescope subsystems
- Supports single-dish and VLBI observations



Origins of DISCOS

- 2007: *ESCS (Enhanced Single-dish Control System)* developed for the Medicina and Noto radio telescopes
- Early 2010s: Development of *Nuraghe*, the control software for the Sardinia Radio Telescope
- *Nuraghe* reused a significant portion of the existing *ESCS* core
- Mid 2010s: Convergence of the two projects into a unified system

DISCOS Architecture

- Control system organized as distributed software components
- Components run inside ACS containers
- Components communicate through CORBA-based middleware
- Each component controls a specific telescope subsystem
- Separation between control logic and hardware interfaces

Applications Places MANAGEMENT/CalibrationTool Mon 08:30

SRT Active Surface

MANAGEMENT/CalibrationTool

Project Observer Filename

Ongoing Subscan

Completed Subscan

Source: w3oh
 Flux:
 Device Id: 0
 Scan Id: 0
 SubScan Id: 0
 HPBW (arcmin): 0.000
 Amplitude (K): 0.000
 PeakOffset (deg): 0.000
 Slope: 0.000
 Tsys (K): 0.000
 ScanAxis - Pointing: MNG_NO_AXIS
 Recording: OFF

Sky Offsets
 Azimuth (Deg): 0.000
 Elevation (Deg): 0.000

Close

asSetup 0 Shaped
 asOn asOff asPark
 STATUS CURRENT PROFILE
 OK SHAPED
 QUIT

04:20:00 05:18:20 07:06:40

Air Temp 23.0

289.0 deg

SRT Active Surface meteoClient MANAGEMENT/CalibrationTool

Challenges in the evolution of DISCOS

- Hardware dependency
- Testing and simulation
- Reproducible environments
- Deployment automation
- CI/CD workflows
- Legacy technologies
- ACS/CORBA coupling
- Modern interfaces
- System resilience

Challenge #1: Hardware Dependency

Problem:

- Testing the control software required access to real telescope subsystems
- Hardware availability limited by observing schedules
- Difficult debugging and experimentation
- Slow development and testing cycles

Challenge #1: Hardware Dependency

Problem:

- Testing the control software required access to real telescope subsystems
- Hardware availability limited by observing schedules
- Difficult debugging and experimentation
- Slow development and testing cycles

Solution:

- Python based simulators of telescope subsystems
- Run DISCOS without telescope hardware
- Decouples software development from telescope availability
- Enables Test-Driven Development (TDD) and automated testing workflows

Challenge #2: Reproducible Environments

Problem:

- Difficult to reproduce the full telescope software environment
- Large number of services and dependencies
- Manual configuration of servers and development machines
- Inconsistent environments across installations

Challenge #2: Reproducible Environments

Problem:

- Difficult to reproduce the full telescope software environment
- Large number of services and dependencies
- Manual configuration of servers and development machines
- Inconsistent environments across installations

Solution:

- Automated environment provisioning
- Infrastructure configuration managed through Ansible
- Unified setup for telescope machines and development environments
- Rapid provisioning of virtual machines for testing and development

Challenge #3: Testing the Deployment Infrastructure

Problem:

- Deployment relies on automated Ansible procedures
- Changes in roles or dependencies may break the setup
- Validation previously required manual provisioning of test VMs
- Manual testing made the feedback loop slow and error-prone

Challenge #3: Testing the Deployment Infrastructure

Problem:

- Deployment relies on automated Ansible procedures
- Changes in roles or dependencies may break the setup
- Validation previously required manual provisioning of test VMs
- Manual testing made the feedback loop slow and error-prone

Solution:

- Continuous Integration pipeline using GitHub Actions
- Automatic provisioning of test environments
- Automated validation of the deployment workflow
- Ongoing transition from virtual machines to containers

Challenge #4: Legacy Middleware Coupling

Problem:

- DISCOS communication built on ACS/CORBA middleware
- Interactions with components requires a full ACS runtime environment
- Heavy dependencies for external clients and integrations
- Limited interoperability with modern software ecosystems

Challenge #4: Legacy Middleware Coupling

Problem:

- DISCOS communication built on ACS/CORBA middleware
- Interactions with components requires a full ACS runtime environment
- Heavy dependencies for external clients and integrations
- Limited interoperability with modern software ecosystems

Solution (ongoing work):

- Introduction of a new ZeroMQ-based communication layer
- ACS runtime no longer required for external clients
- Enables flexible access to telemetry and system state
- Allows development of internal and third-party monitoring and control tools

Challenge #5: System Resilience

Problem:

- Many DISCOS components mapped 1:1 to hardware devices
- Components could not start if the corresponding hardware was unavailable
- Hardware failures could prevent parts of the control system from running
- Limited ability to expose hardware status and availability information

Challenge #5: System Resilience

Problem:

- Many DISCOS components mapped 1:1 to hardware devices
- Components could not start if the corresponding hardware was unavailable
- Hardware failures could prevent parts of the control system from running
- Limited ability to expose hardware status and availability information

Solution (ongoing work):

- Components designed to start independently from hardware availability
- Graceful handling of missing or unreachable devices
- Explicit reporting of hardware status and connection state
- Increased robustness of the distributed control system

CURRENT TARGET

w3oh

Catalog Right Ascension	02:27:04.100
Catalog Declination	61:52:27.100
Apparent Azimuth	32.233764
Apparent Elevation	29.898870
Apparent Right Ascension	02:29:02.000
Apparent Declination	61:59:42.575
Apparent Galactic Longitude	133:56:51.356
Apparent Galactic Latitude	01:03:54.159
Julian Epoch	2026.18708631
Generator Type	SIDEREAL

OK
TRACKING

SCHEDULER

Project Code	Maintenance
Schedule Name	--
Scan / SubScan	0 / 0
Backend	TotalPower
Recorder	FitsZilla
Device	0
Rest Frequency	0

OK
TRACKING

SITE INFO

Observatory Name	SRT
Universal Time	2026-03-10T07:59:55.046Z
Modified Julian Day	61109.333276
Apparent Greenwich Sidereal Time	19:11:58.651
Local Sidereal Time	19:48:57.451
DUT1	0
Polar Motion	0 0
Geodetic Coordinates	39:29:34.000 09:14:42.000 700 WGS84
Geocentric Coordinates	4865226.972 791916.701 4035129.949

ANTENNA POINTING

COORDINATE	OBSERVED	OFFSETS	RAW	POINTING CORRECTION	REFRACTION CORRECTION
Azimuth	32.233623	0.000000	389.944097	-2.289526	
Elevation	29.898862	0.000000	30.085637	0.153880	0.032895
Right Ascension	02:27:04.127	0.000000			
Declination	61:52:27.257	0.000000			
Galactic Longitude	133:56:51.479	0.000000			
Galactic Latitude	01:03:54.375	0.000000			

FWHM: 0.012389

MINOR SERVOS

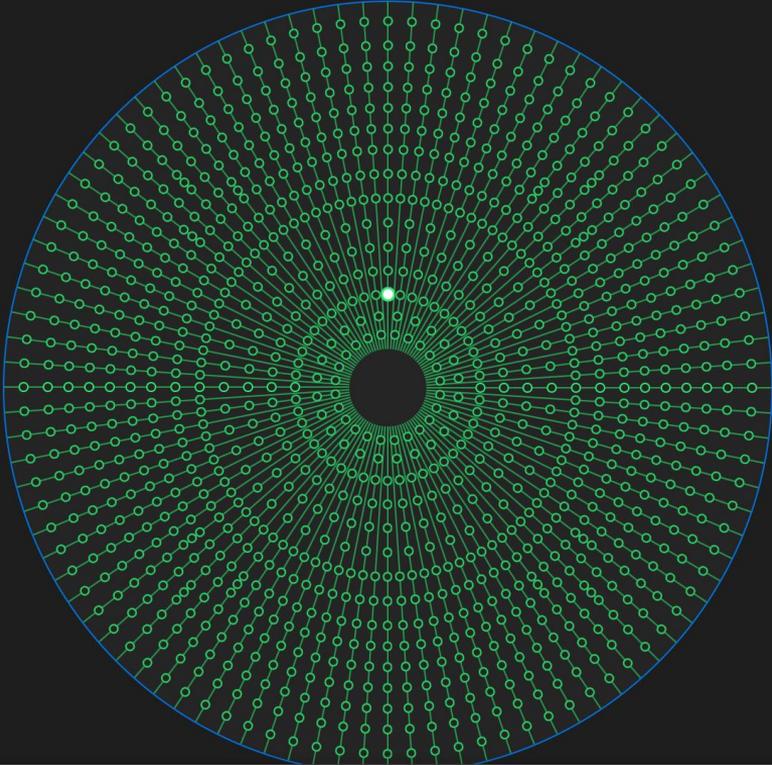
Current Configuration	KKG
Motion Info	TRACKING

OK
TRACKING

ISSUED	RECEIVED	STATUS	COMMAND	OUTPUT

> Type command...
SEND

ACTIVE SURFACE



OK

Loop Table File act_rev02.txt

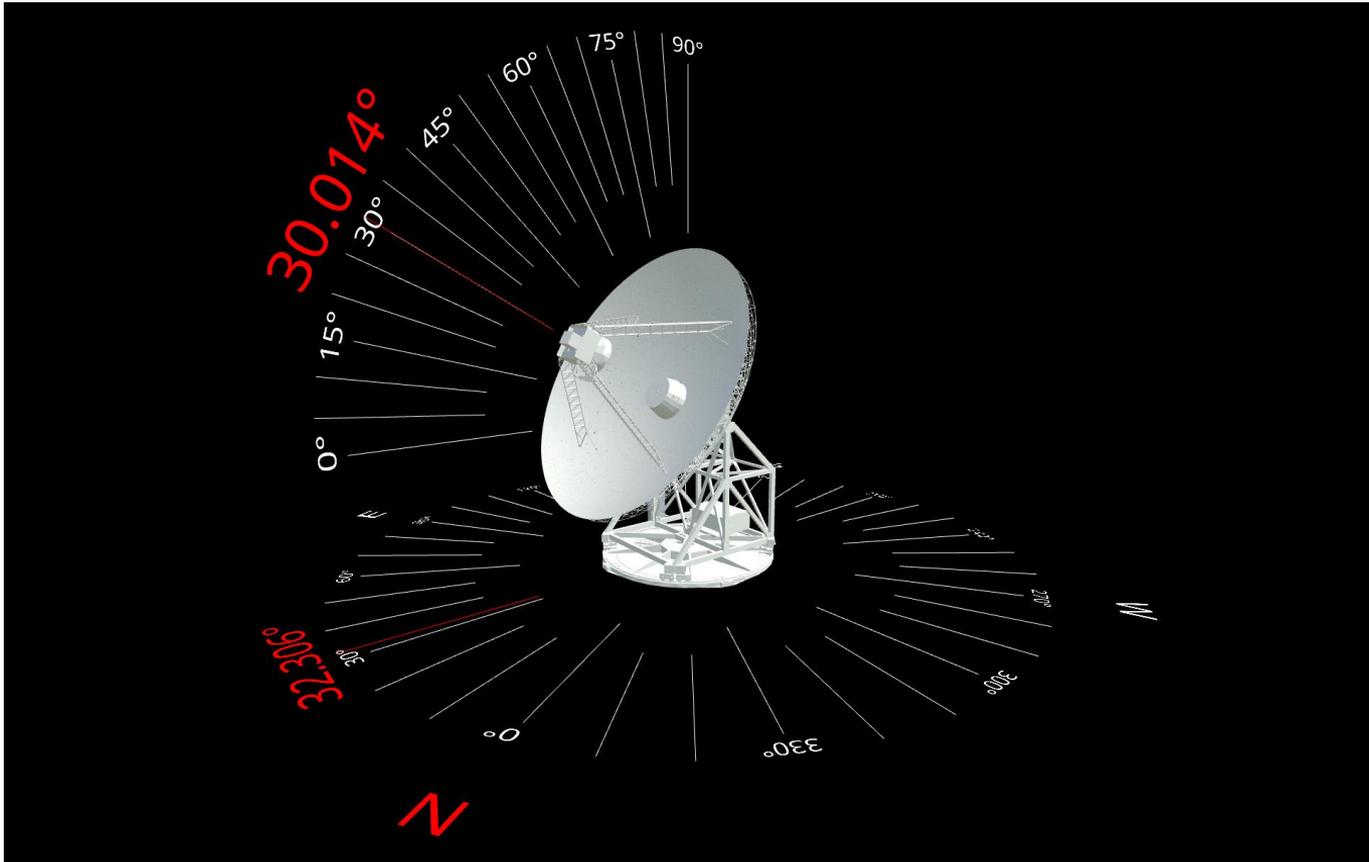
SECTOR01/LAN01/USD03

AVAILABLE	ENABLED
CALIBRATED	RUNNING

USD Type USD60xxx

Software Version	1.3
Commanded Position	2169
Current Position	2169
Minimum Frequency	10
Maximum Frequency	50
Acceleration Factor	50
Delay	2560

> Type command... SEND



USC-C General Assembly – 9 - 13 March 2026 – Trieste

Source code and resources

DISCOS organization on GitHub
github.com/discos

DISCOS Simulators
github.com/discos/simulators

DISCOS Deployment
github.com/discos/deployment

DISCOS Client / telemetry tools
github.com/discos/discosclient

giuseppe.carboni@inaf.it

Thank you for the attention