

Team

A. Orlati, S. Righini, L. Monti, M. Fiorentini, F. Vitello - **INAF-IRA**
 M. Buttu, G. Carboni, A. Fara, C. Migoni, S. Poppi - **INAF-OAC**
 M. Landoni - **INAF-OA-BRERA**



A COMMON STRATEGY

The **DISCOS** (*Development of Italian Single-dish COntrol Software*) project has been engaged for years in the design and development of software for the operation of INAF radio telescopes (Medicina, Noto and SRT). Our software is produced with the aim of being common to the three antennas, thus optimizing resources and ensuring reliability, maturity and durability over time.

The complexity of coping with three heterogeneous instruments was handled by designing a software development infrastructure with a wide monolithic codebase (libraries, components and generic interfaces), which is completely shared among the three product lines while specific differences are managed via configuration files or via polymorphism only when strictly necessary.

The common infrastructure created by DISCOS allows, also, to optimize the resources used for maintenance, operations and development and also improves the accessibility of the facilities as it guarantees common user interfaces, tools, procedures, data formats, documentation and training.



The Italian radiotelescope sites.

SOFTWARE ARCHITECTURE

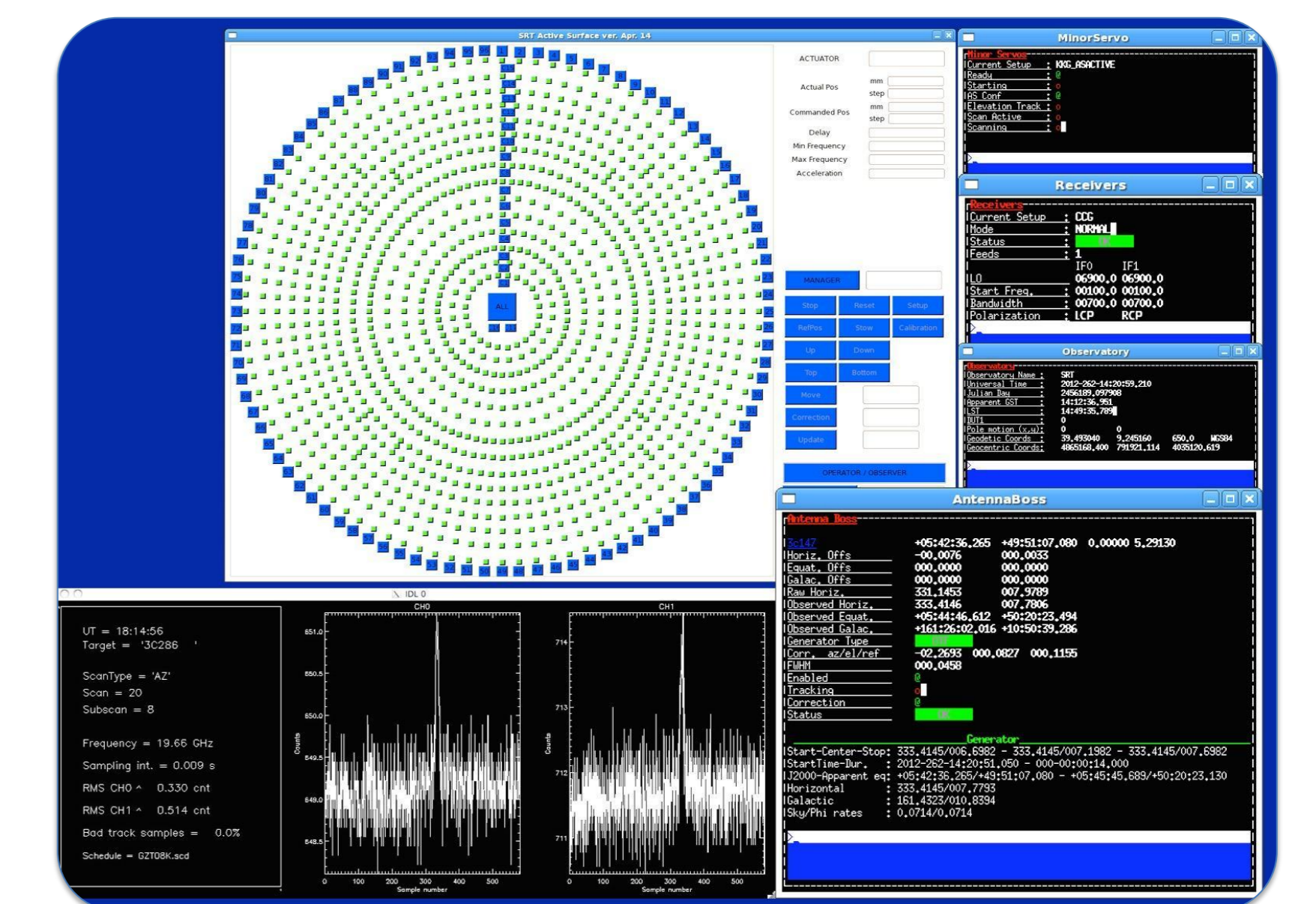
The architecture is composed by several packages or subsystems. Packages are sub-sets of the whole system: they group all the elements (components, configuration schemas, codes and interfaces) that logically or functionally stay together. Packages are themselves hierarchically grouped into two layers: the User and the Telescope Layer. The *User Layer* runs on top of the *Telescope Layer* and hides its complexity to the upper one.

Telescope layer

It contains the packages that directly deal with the telescope hardware (Antenna, Receivers, back-ends, servo systems, etc.). These subsystems work independently from each other.

User layer

It is essentially composed by a *Management* package, the user interfaces and other tools used for science operations. The Management subsystem is responsible for schedule execution, instrument setup, supervision, remote access, data file output, log handling, etc.



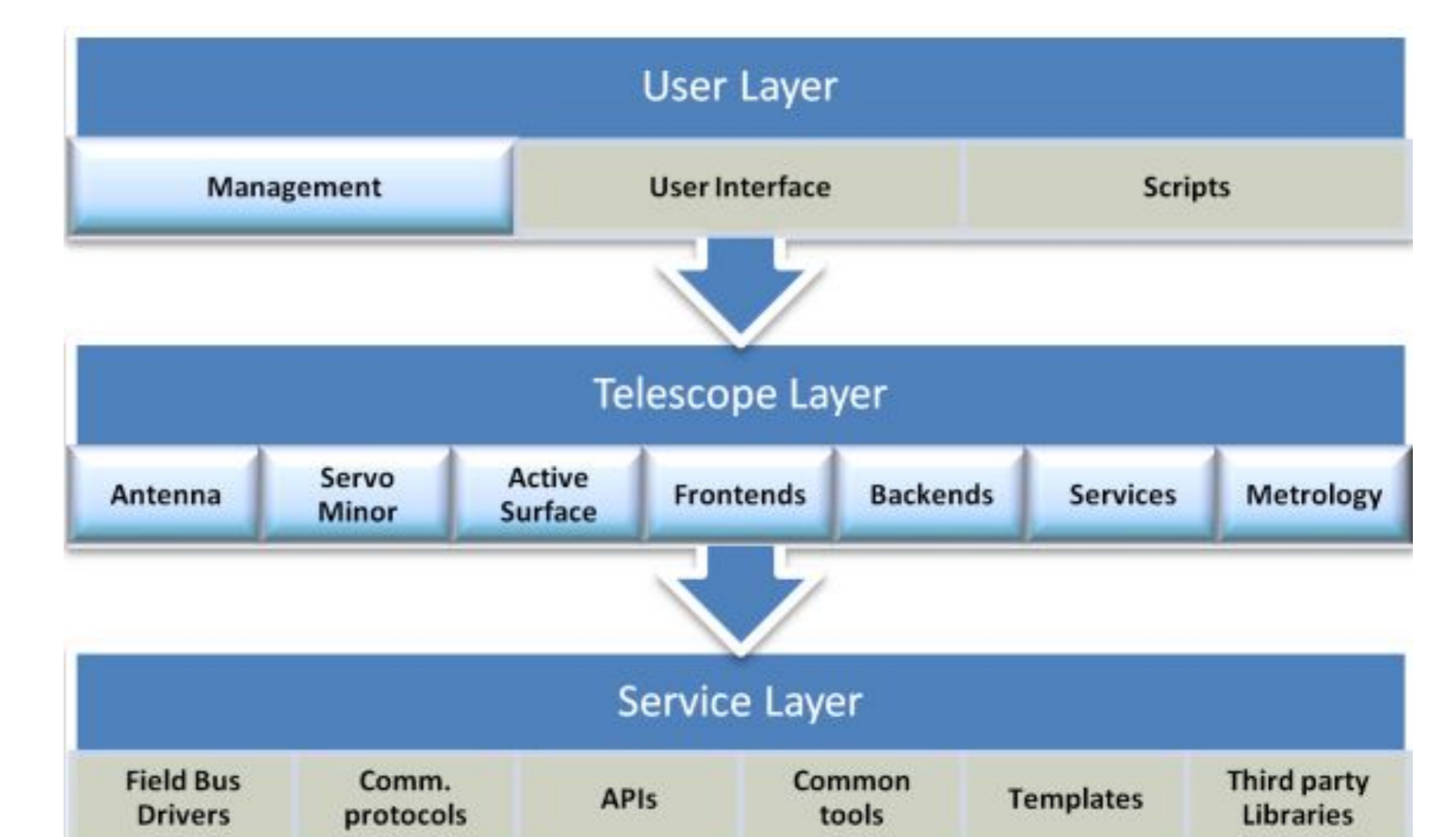
Screenshot of the DISCOS GUI.

TECHNOLOGIES

Development and maintenance is a *continuous process* which is carried over trying to follow the state-of-the-art software methodologies (Test Driven Development, Continuous Integration on simulated environments and automatic provisioning) so that high-quality software is obtained.

The key to a successful **Continuous Integration** strategy adoption is the automation of repetitive tasks and the standardization of procedures among environments and among developers. Once the standardization of the environments was obtained, we automated the build process for each of our product lines.

Continuous integration is an effective practice as long as the development process is integrated with tests, ensuring that every stage fulfills its requirements. Drawn from our experience, we have developed new guidelines in the adoption of an approach called **Behavior Driven Development** (BDD). It has a different goal as compared to **Test Driven Development** (TDD): it aims at testing the software behavior, not its implementation, yet it must be used together with TDD and unit testing strategies.



Architecture layers. The DISCOS project documentation can be found following the QR code.

