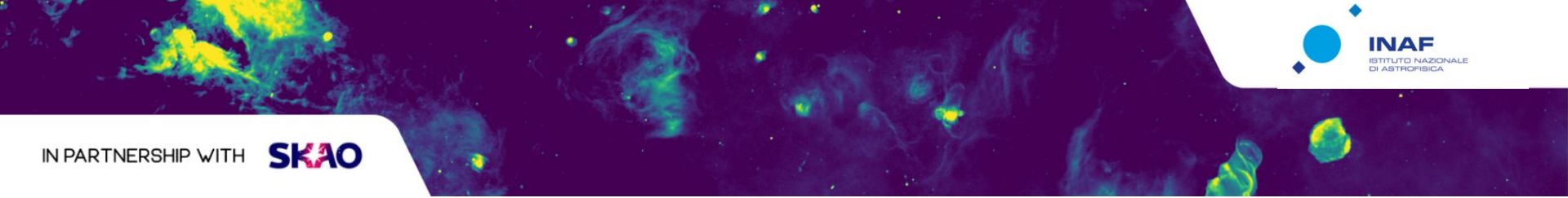


SKA Observatory Management and Control Software: the INAF contribution

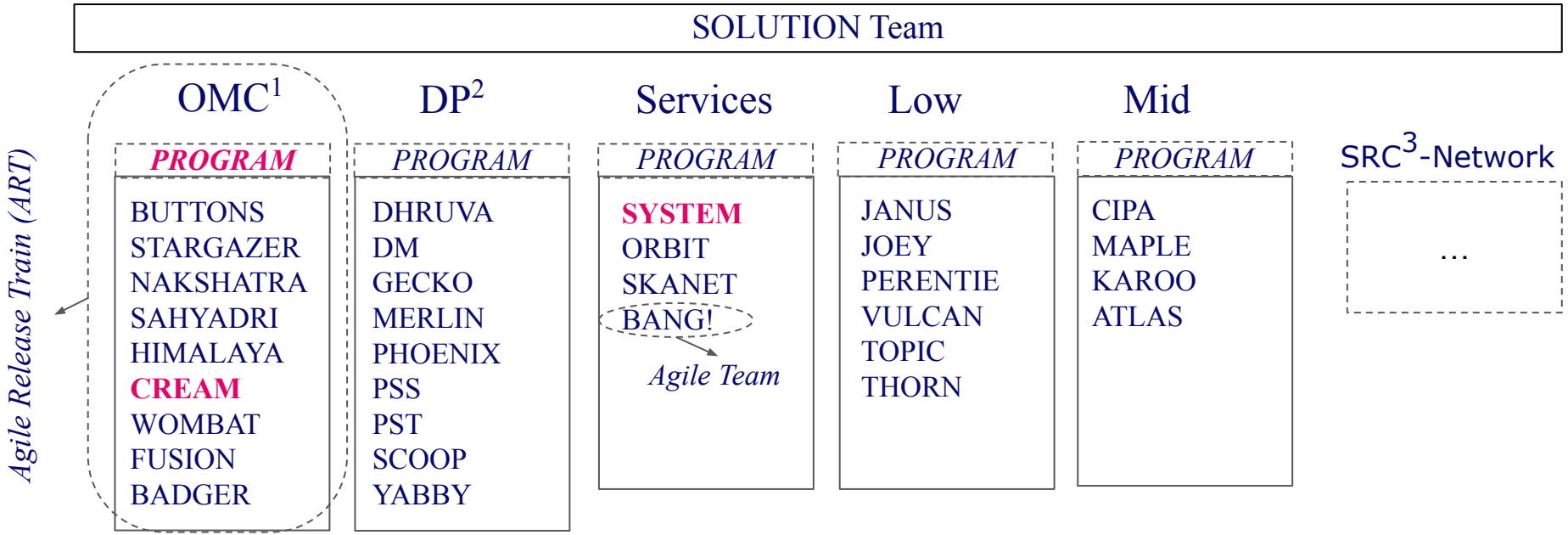
Gianluca Marotta, Valentina Alberti, Carlo Baffa, Matteo Canzari, Matteo Di Carlo, Stefano Di Frischia, Elisabetta Giani, Teresa Pulvirenti and Mauro Dolci

The SKA software engineering group

The SKA software engineering group

- more than 200 people involved from 15 different countries
- organized with Scaled Agile Framework (SAFe)
- work is iteratively planned every 3 months (Program Increment - PI)

SAFe® PROVIDED BY SCALED AGILE



[1] *Observatory Management and Control* [2] *Data Processing* [2] *SKA Regional Centers*

The INAF Group

Program Team

OATs	Valentina Alberti	UI/UX[1] Expert (OMC Program)
OAA	Carlo Baffa	CREAM Product Owner
OAA	Elisabetta Giani	CREAM Developer
OAA	Gianluca Marotta	CREAM Developer
OAAb	Stefano Di Frischia	CREAM Developer
OAAb	Matteo Canzari	CREAM Developer
OAAb	Matteo Di Carlo	SYSTEM Developer
OAAb	Mauro Dolci	Scientific Responsible
IRA	Teresa Pulvirenti	Contract Management

CSP[2] Local Monitoring and Control

TARANTA TANGO ON WEB

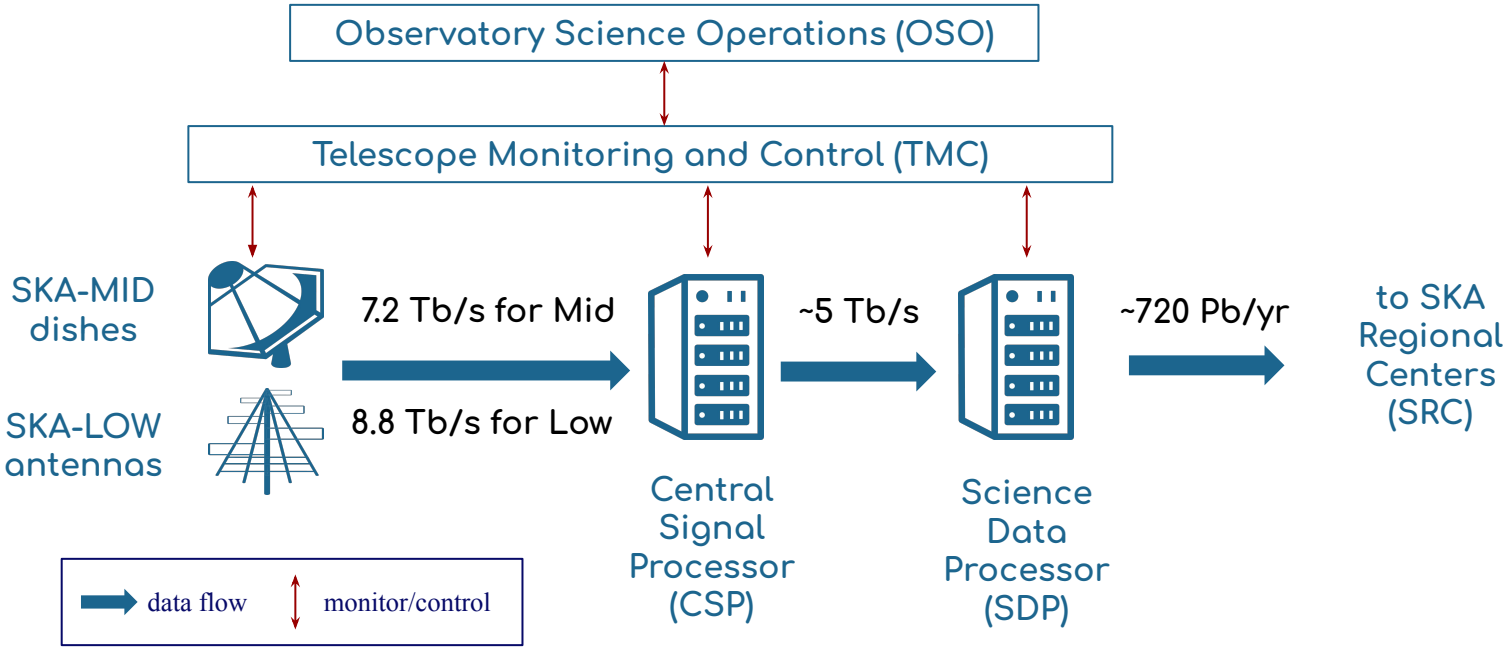
System Infrastructure

N.B. CREAM and SYSTEM are also composed by people coming from other countries and companies/institutions

[1] *User Interfaces/User eXperience* [2] *Central Signal Processor*

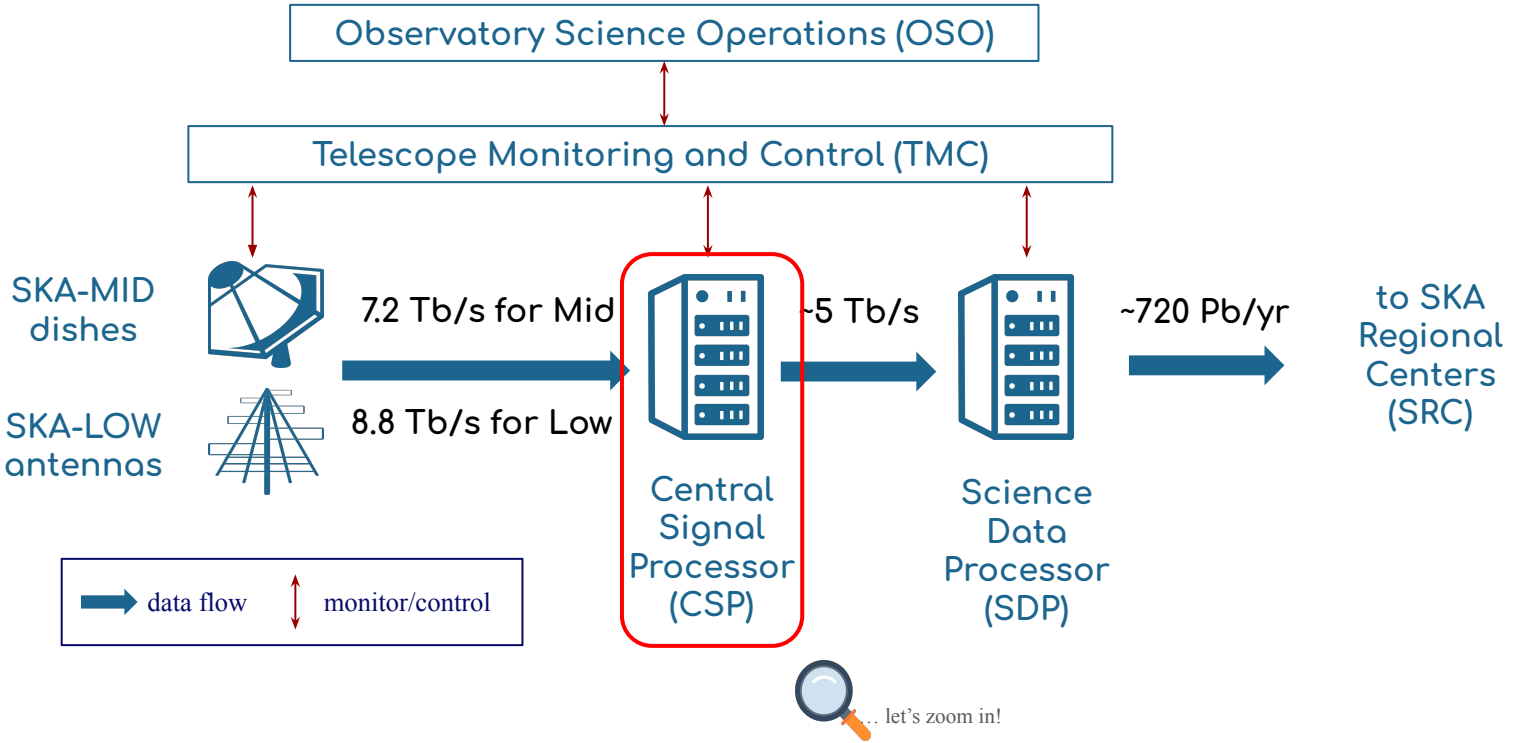
The SKA Data Flow and Control System

… a very simplified view

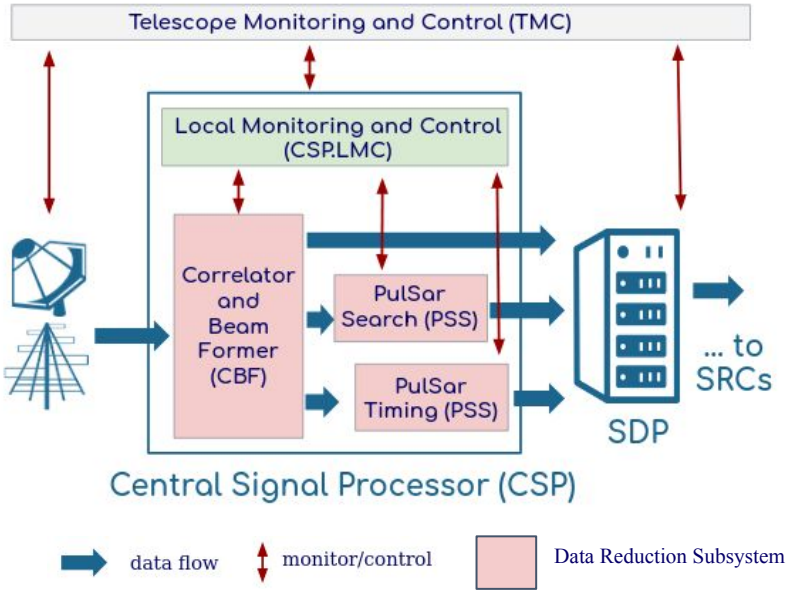


The SKA Data Flow and Control System

… a very simplified view



The Central Signal Processor

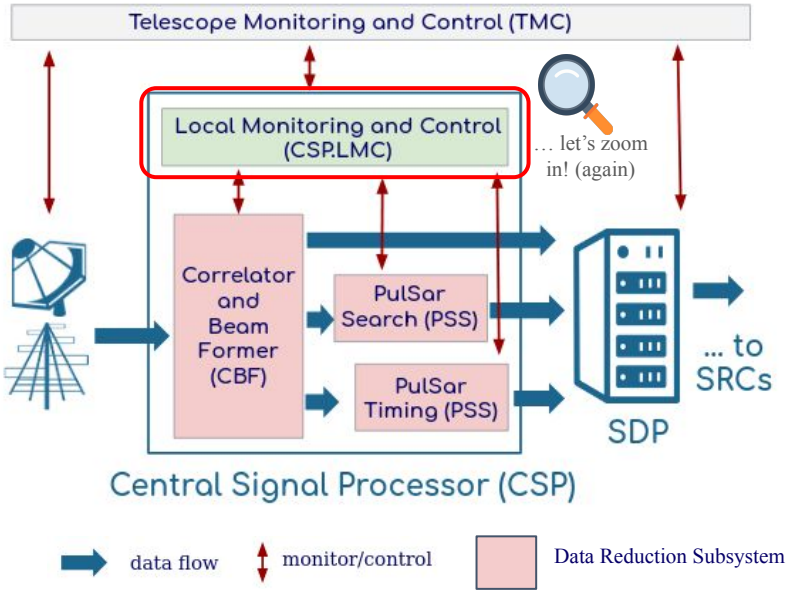


CSP is composed of three main subsystems:

- the **Correlator and Beam Former (CBF),** to creates the visibilities and the data beams;
- the **Pulsar Search (PSS)**, to perform an all-sky pulsar search survey;
- the **Pulsar Timing (PST),** to measures the frequency of the pulsar candidates

CSP.LMC provides the *interface* to TMC *without exposing CSP internal complexity.*

The Central Signal Processor



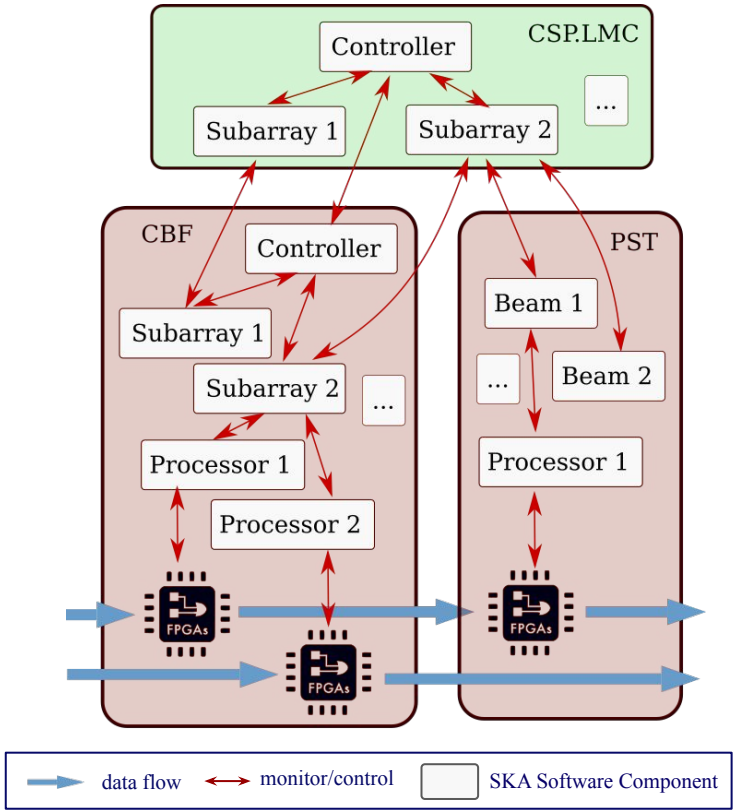
CSP is composed of three main subsystems:

- the **Correlator and Beam Former (CBF),** to creates the visibilities and the data beams;
- the **Pulsar Search (PSS)**, to perform an all-sky pulsar search survey;
- the **Pulsar Timing (PST),** to measures the frequency of the pulsar candidates

CSP.LMC provides the *interface* to TMC *without exposing CSP internal complexity.*

The CSP Local Monitoring and Control (CSP.LMC)

… a very simplified view



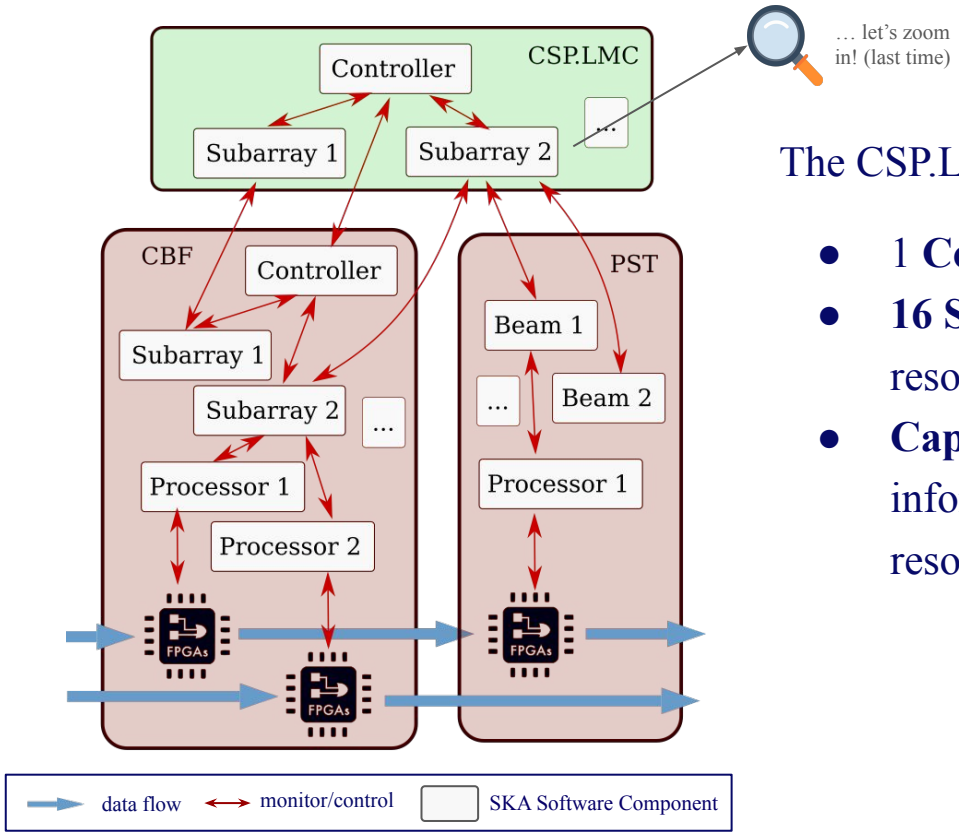
The CSP.LMC is composed by:

- 1 **Controller,** i.e. the primary point of access for CSP
- **16 Subarray**, representing subsets of the telescope resources that can be used for one observation
- **Capability** devices, apt to monitor and report to TMC information and statistical data about specific CSP resources (e.g. CBF processors, PST Beams)

CSP **L**ocal **M**onitoring and **C**ontrol

The CSP Local Monitoring and Control (CSP.LMC)

… a very simplified view



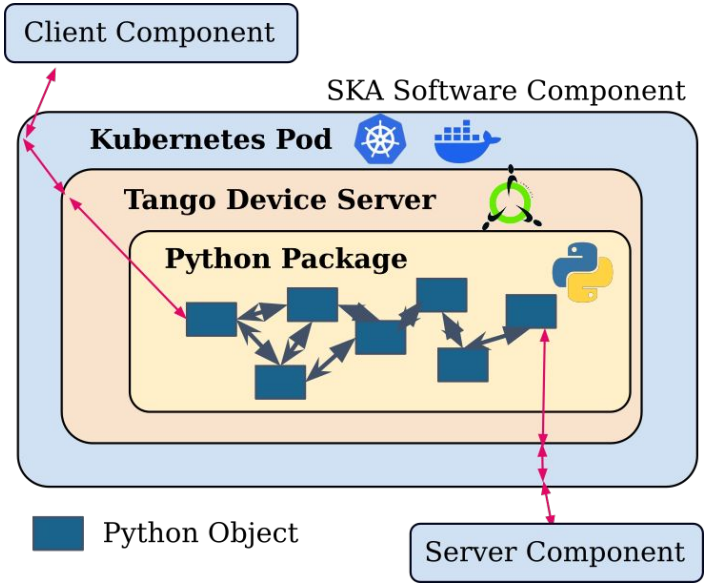
The CSP.LMC is composed by:

- 1 **Controller,** i.e. the primary point of access for CSP
- **16 Subarray**, representing subsets of the CSP resources that can be used for one observation
- **Capability** devices, apt to monitor and report to TMC information and statistical data about specific CSP resources (e.g. CBF processors, PST Beams)

CSP **L**ocal **M**onitoring and **C**ontrol

The SKA Software Component

Software is **executed by** a **Tango Device Server** process, that runs in a **Docker container** that is **orchestrated** in a cluster **by Kubernetes**



TANGO

"a free open source device-oriented controls toolkit for controlling any kind of hardware or software and building SCADA[1] systems"[2].

docker

"is an open platform for developing, shipping, and running applications [...] enables you to separate your applications from your infrastructure[3]"

kubernetes

"is a portable, extensible, open source platform for managing containerized workloads and services …[4]"

* for simplicity only one server/client is reported

[1] Supervisory Control And Data Acquisition [2] https://www.tango-controls.org/ [3] https://docs.docker.com [4] https://kubernetes.io/docs

The contribution to TANGO Control System

The Tango Community has **more than 45 institutional** and **more than 15 industrial partners** from all over the World

INAF is member of the TANGO steering committee and active actor in the community

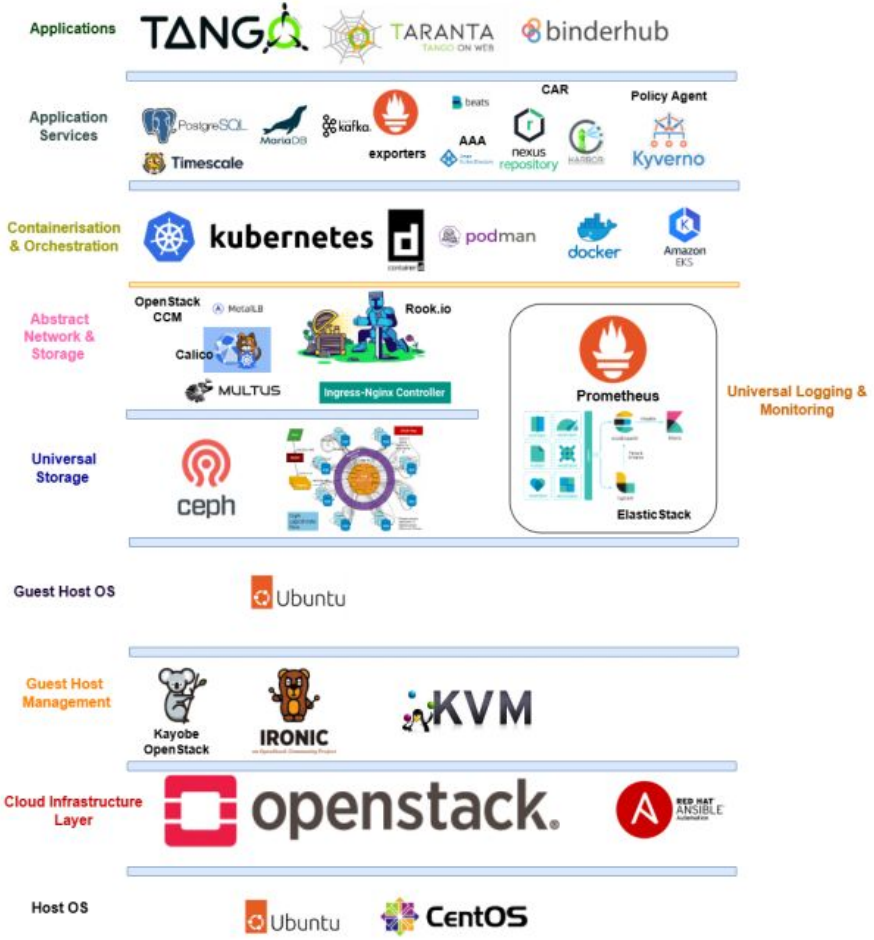
TANGO collaboration meeting - hosted by INAF OAAb - May 2025

The System Infrastructure

SKA Control Software is meant to be deployed with a large use of **cloud-native technologies**

The infrastructure is developed and maintained by System Team

Tools for *deploying*, *testing* and *monitoring* are provided to developers



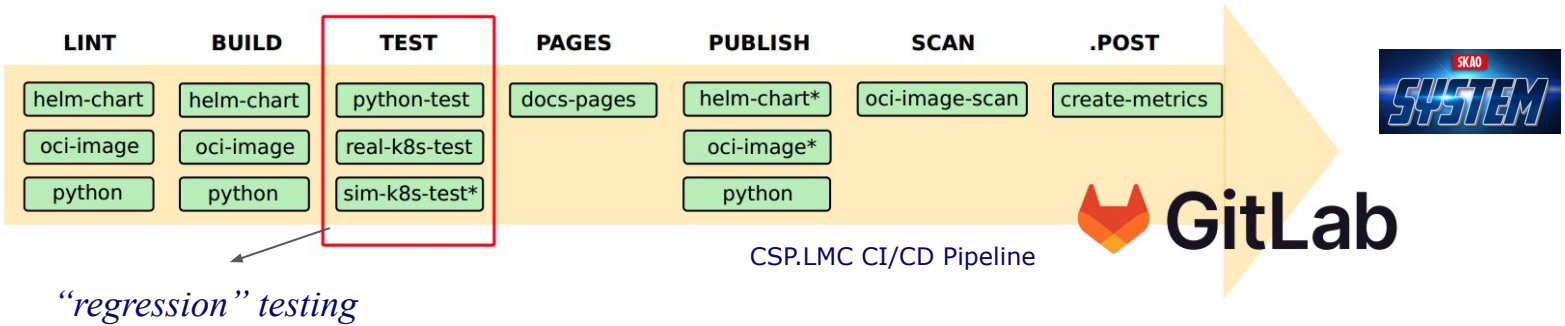
M.Di Carlo et al, "Monitoring the SKA Infrastructure for CICD", *Proceedings of the 19th ICALEPCS* (2023)
M.Di Carlo et al, "SKA Tango Operator", *Proceedings of the 19th ICALEPCS* (2023)

Continuous Integration/Continuous Delivery/Deployment

Continuous Integration/Continuous Delivery/Deployment (CI/CD) refers to development practices:

- single source repository for each component;
- automated build;
- ˆautomated testing;
- ˆ**every commit** should build on an integration machine (with tango/kubernetes)

CI/CD practices are ensured by the use of **Gitlab pipelines,** based on System Team templates.



Testing the SKA Software

"[Testing is] the process [...] to determine that they [software products] **satisfy specified requirements**, to demonstrate that **they are fit for purpose** and to **detect bugs**."[1]

SKAO Software Testing Policy and Strategy - https://developer.skao.int/

- Individual teams are responsible for software component quality and testing strategy
- Verification Tests based on requirements are done by AIV teams
- A *Testing Community of Practice* gather developers from different teams to share knowledge and practices

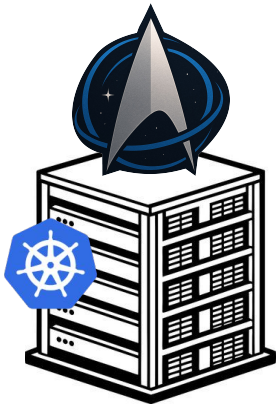
Testing can represent a **considerable amount of time** of developlemnt (about 50% in CSP.LMC)

CREAM Team plays a pioneering role in SKA software testing, tackling *"test flakiness"* and *"unhappy-path testing"*

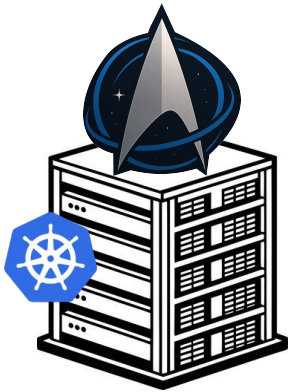
HW Infrastructure for SKA Software @ INAF

Finanziato dall'Unione europea
NextGenerationEU

Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA



@INAF-OAA



@INAF-OAAb

Federation Clusters

- made available free of charge
- provide kubernetes
- hosted and managed by INAF-OAA and INAF-OAAb members
- hosts Cream Team VMs for development
- hosts CSP.LMC CI/CD pipelines

Taranta: a web UI for Tango

TARANTA
TANGO ON WEB

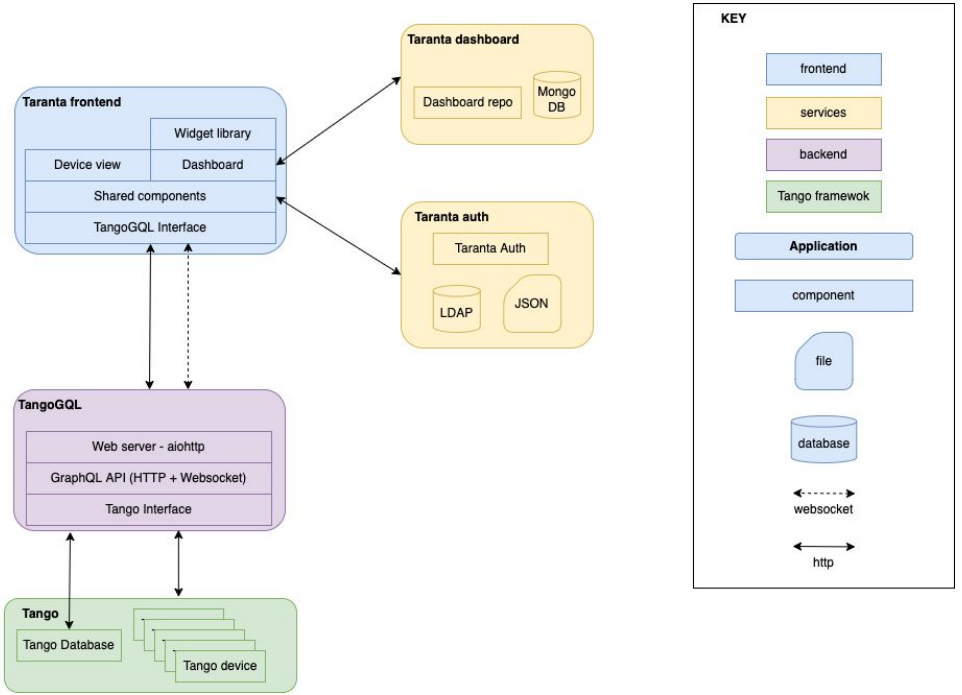
https://taranta.readthedocs.io

Taranta offers a **no-code**, **web-based** approach for creating dashboards that integrate multiple *Tango devices*.

Taranta is developed by CREAM team in collaboration with MAX IV institute

SKAO CREAM TEAM

MAX IV

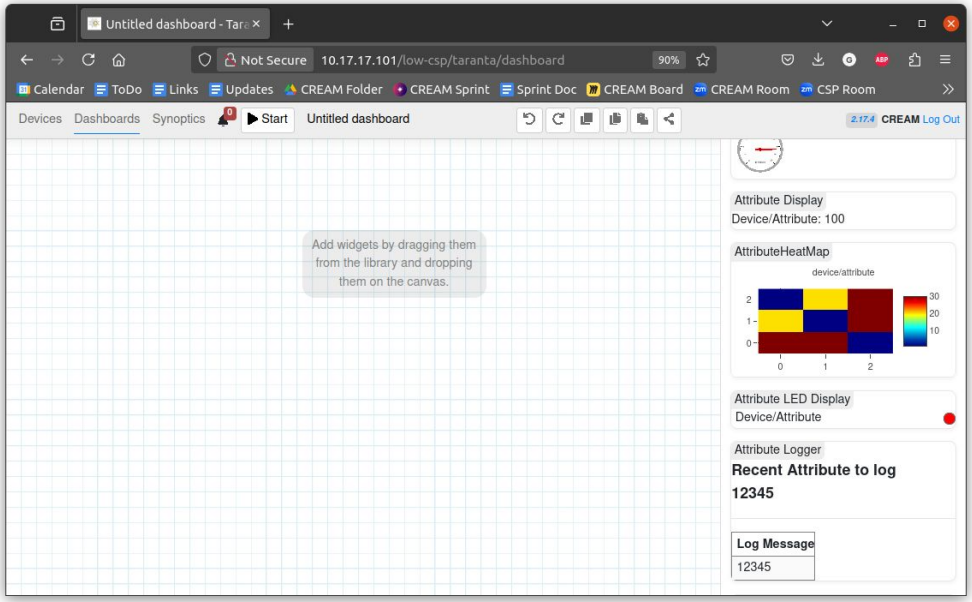


M. Canzari et al., "How Taranta provides tools to build user interfaces for TANGO devices in the SKA integration environment without writing a line of code" *Software and Cyberinfrastructure for Astronomy VII*; 121890W (2022)

Taranta: a web UI for Tango

TARANTA
TANGO ON WEB

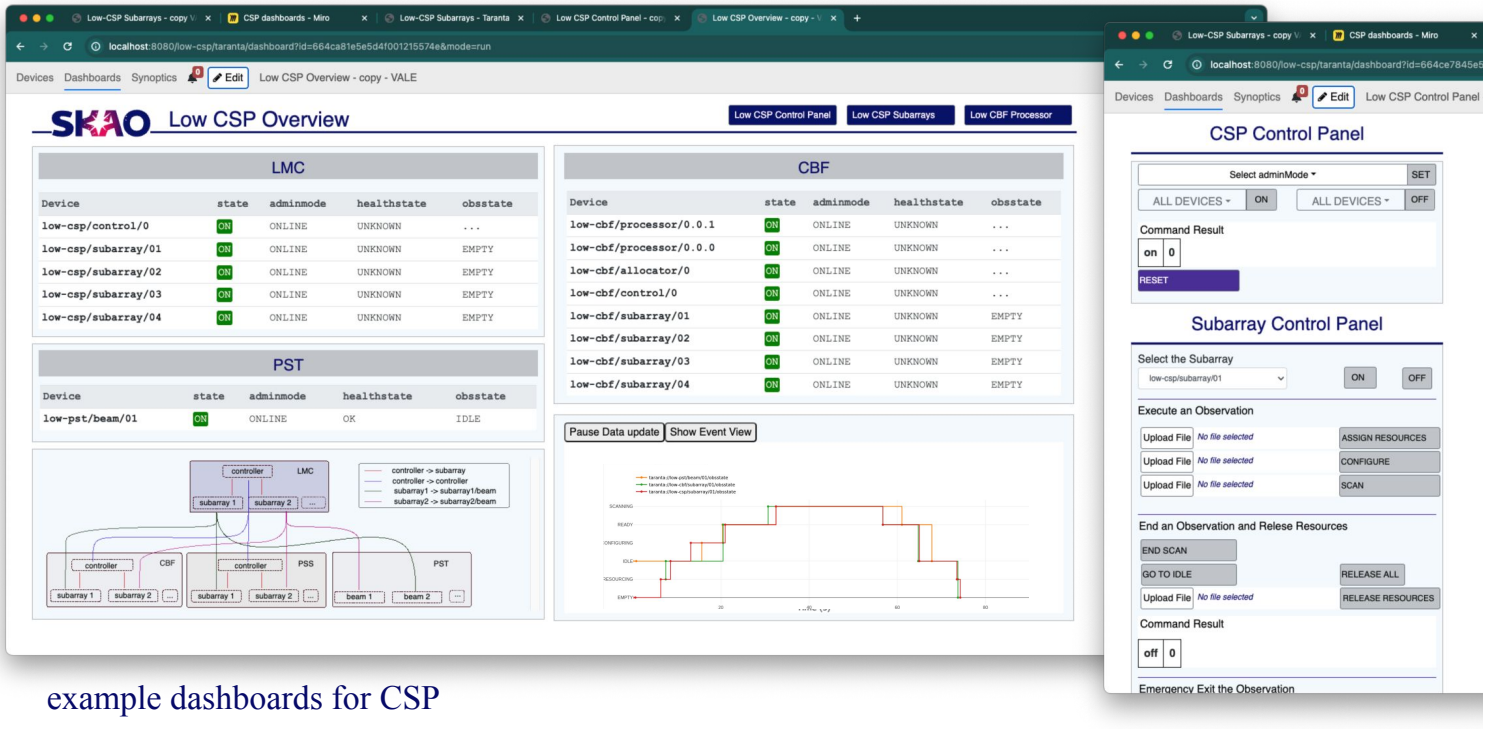
Taranta is largely used by teams for creating **engineering dashboards**



- simple to use (drag and drop)
- new functionalities based on user feedback
- possibility to create vectorial and synoptic dashboards with Inkscape

Taranta: a web UI for Tango

TARANTA
TANGO ON WEB

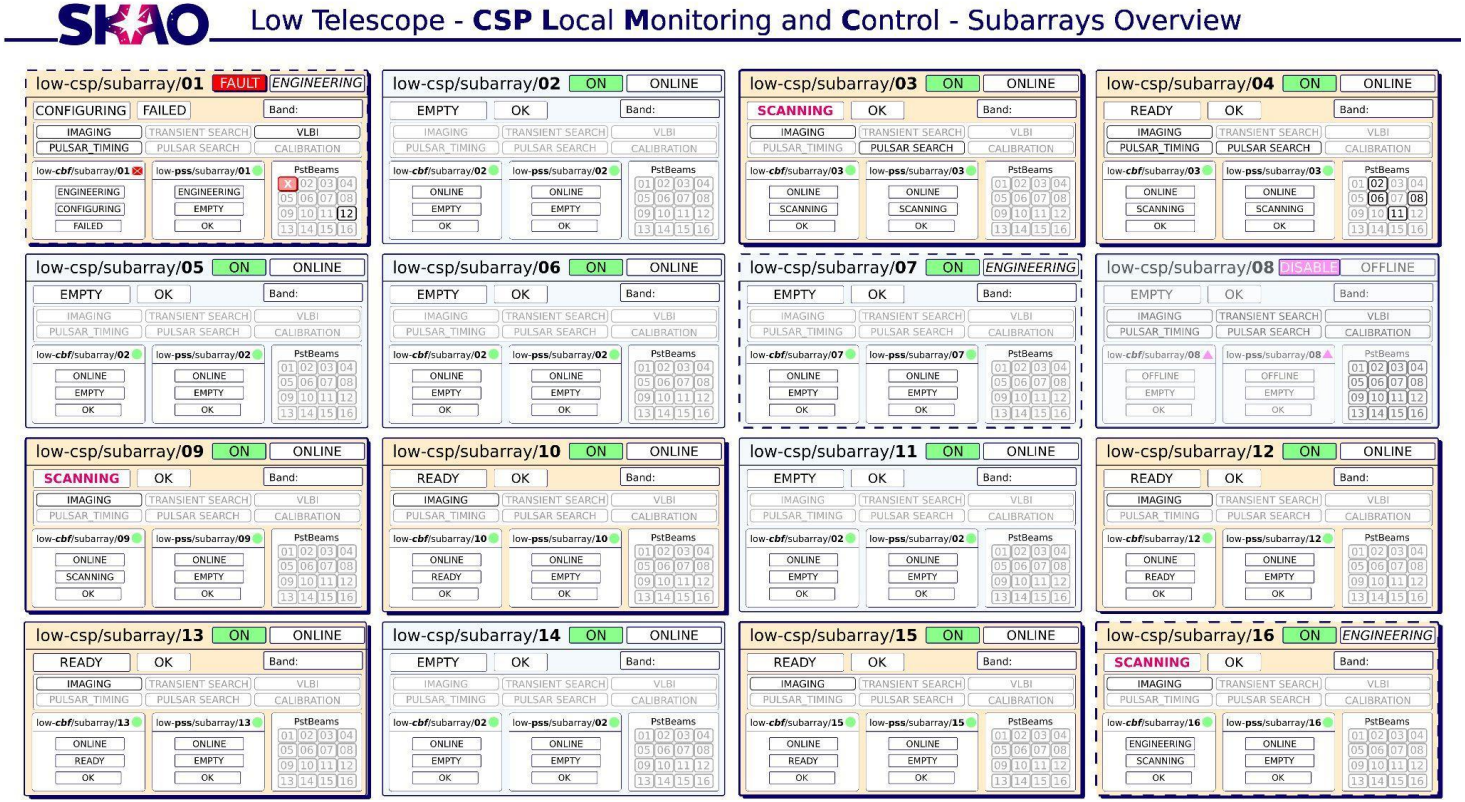


example dashboards for CSP

Taranta: the “new” synoptic view

INKSCAPE

TARANTA
TANGO ON WEB

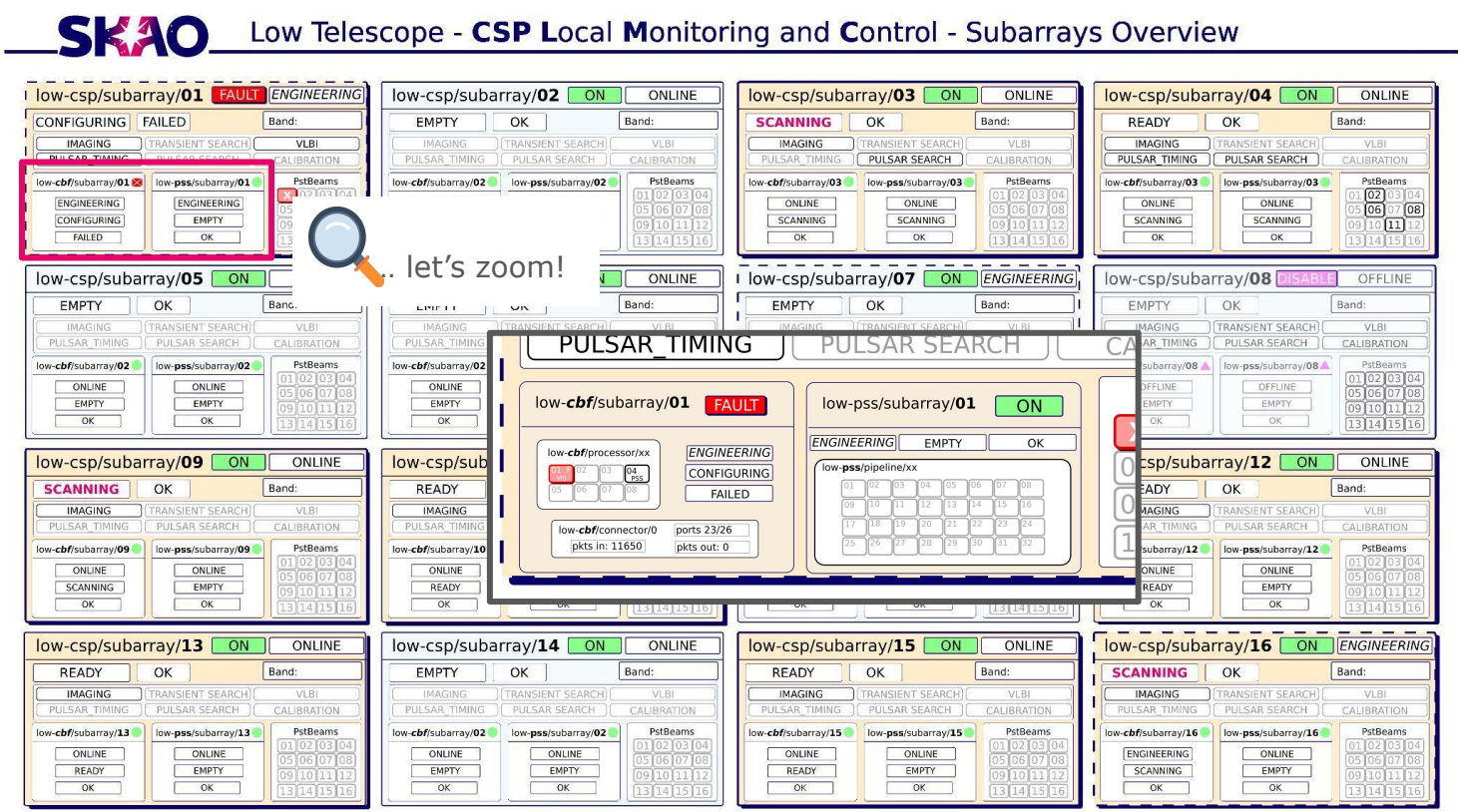


a prototype synoptic for CSP

Taranta: the “new” synoptic view

INKSCAPE

TARANTA
TANGO ON WEB

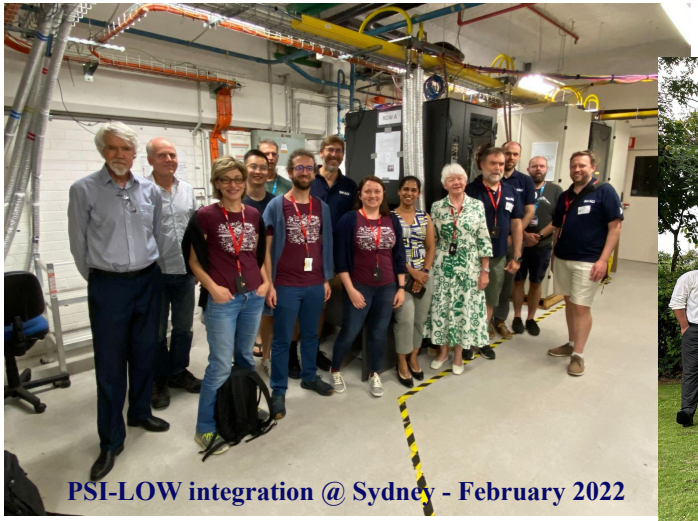


a prototype synoptic for CSP

Collaborating over the world

OSO TMC CSP SUBSYSTEMS AIV

teams that interact with CREAM on a regular basis


PSI-LOW integration @ Sydney - February 2022


TMC Workshop @ Pune - May 2024


UI Workshop @ Australia - March 2025

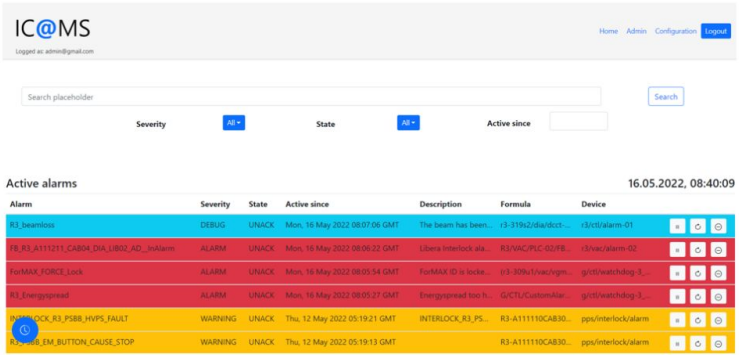
…furthermore

- CSP.LMC developers are also in charge of PSS.LMC[1]

detection of first known Pulsar using CSP.LMC and PSS.LMC developed by CREAM TEAM

NOTE: PSS.LMC will control >1000 pulsar engines in the future

- Taranta developers will take over the SKA Alarm handling UI (IC@MS)



[1] *PulSar Search - Local Monitoring and Control*

INAF
ISTITUTO NAZIONALE
DI ASTROFISICA

SKAO

Program Team

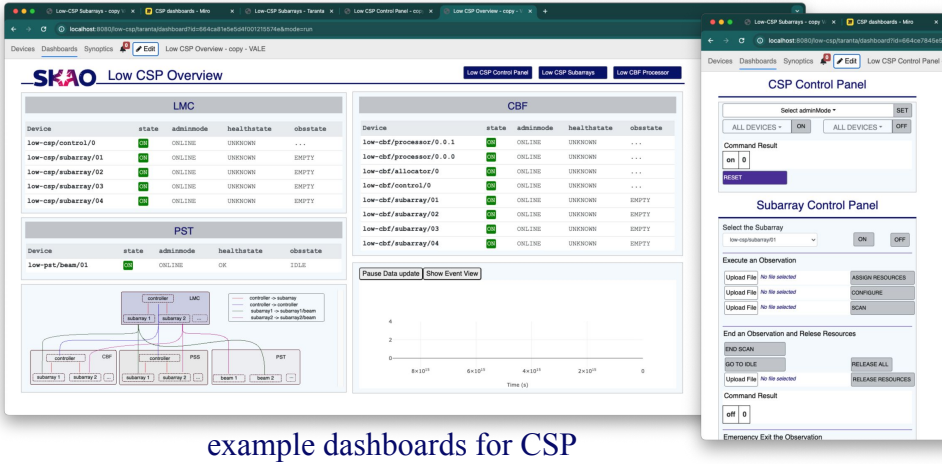
Thank you for your attention!

Taranta: a web UI for Tango

TARANTA
TANGO ON WEB

Taranta offers a **no-code**, **web-based** approach for creating dashboards that integrate multiple Tango devices.

SKAO MAX IV



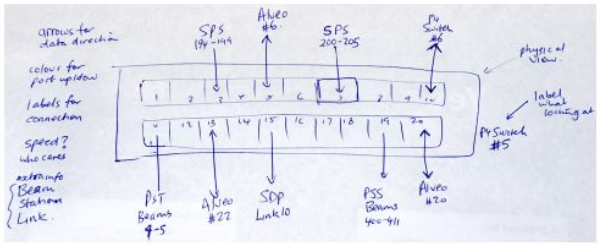
example dashboards for CSP

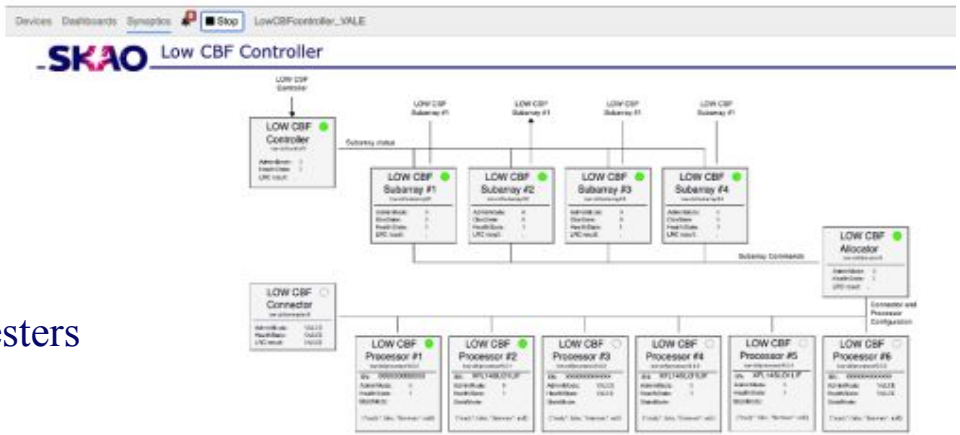
INKSCAPE

The new **synoptic view** enables the integration of **svg files** that reacts to the changes of the attributes of the TANGO Devices

Creating engineering UIs: lean UX

The creation of User Interfaces for LOW-CSP has been experimented as a cross team collaboration using a **LeanUX approach**

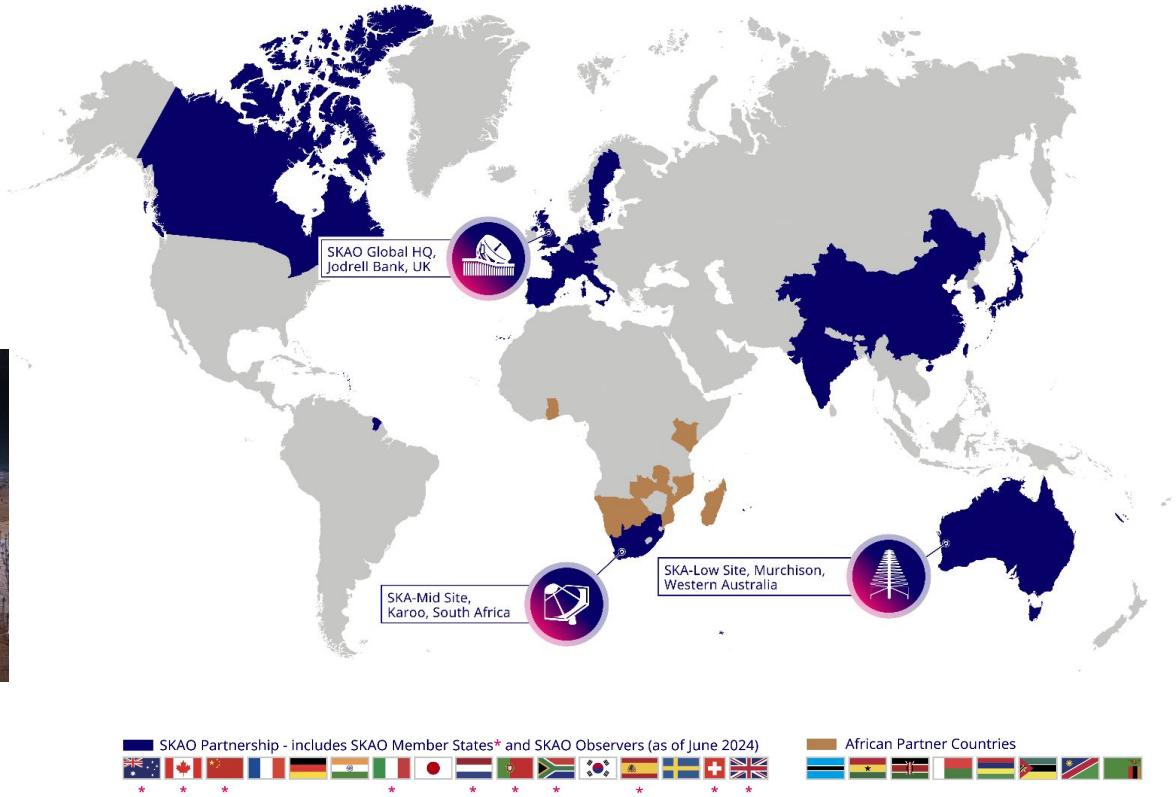




Taranta Users are both developer and testers

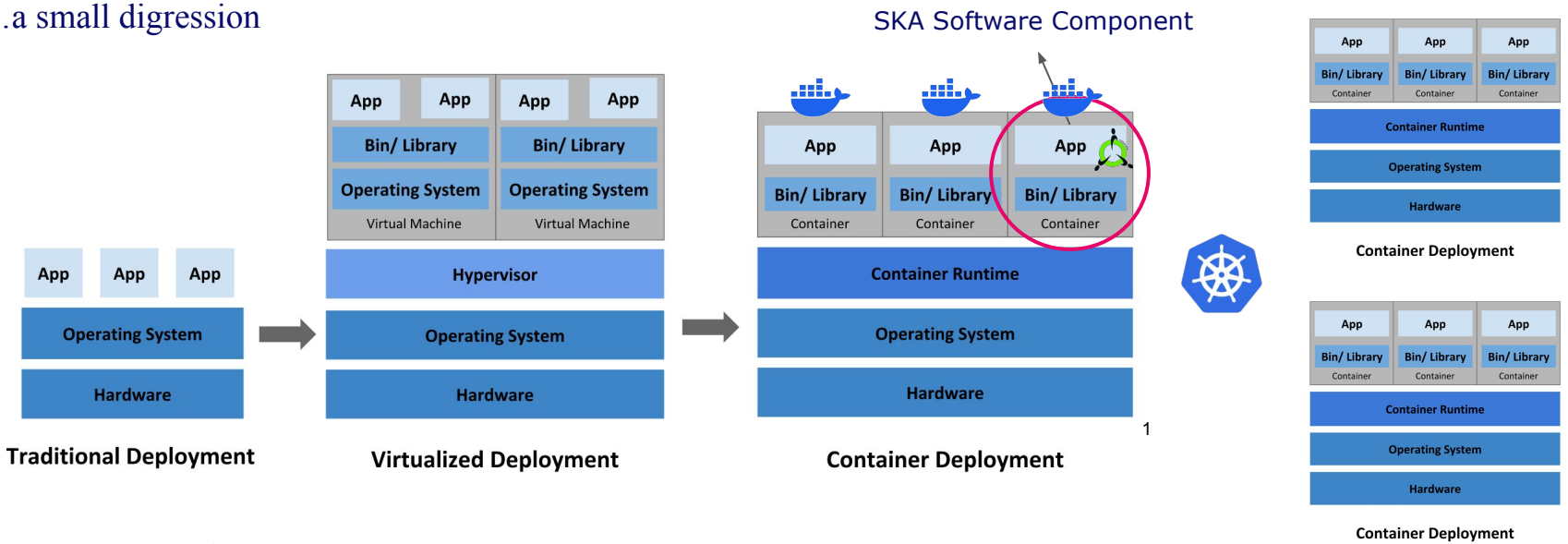
The SKA Project

The Square Kilometer Array (SKA) is an international effort to construct the **two** *world's biggest radio telescopes*.



Docker and Kubernetes

…a small digression

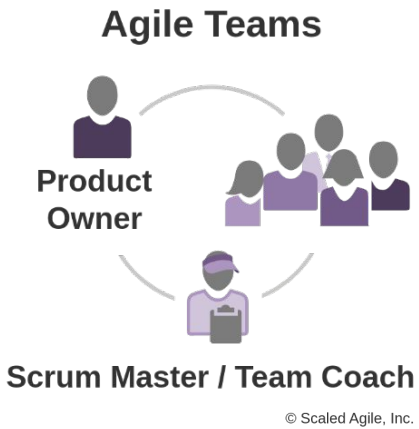


Kubernetes:

- establishes the network between containers,
- allows shared storage,
- manage secrets,
- distributes workloads in cluster
- is self-healing,
- ensure scalability

[1] https://kubernetes.io/docs

The Agile Team



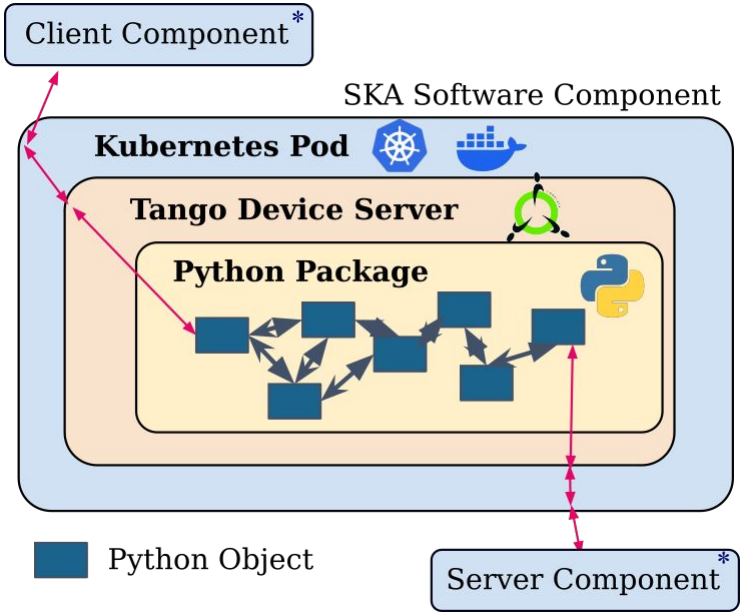
- each Agile Team is responsible for one or more SW Product;
- code is developed iteratively;
- Team plans the work within the ART[1] every 3 months (*Program Iteration - PI*);
- Team revise its PI plan every 2 weeks (*sprint*);
- the *Scrum Master* "helps implement and maintain Agile practices, [...] optimizes and improves team performance"[2]
- the *Product Owner* "contributes to the Vision and roadmap [...] and prioritize the team's work"[2].

SAFe® PROVIDED BY SCALED AGILE

more about it: **tomorrow**'s training session on "Agile Framework" conducted by Valentina Alberti and Matteo Di Carlo (**h14:00**)

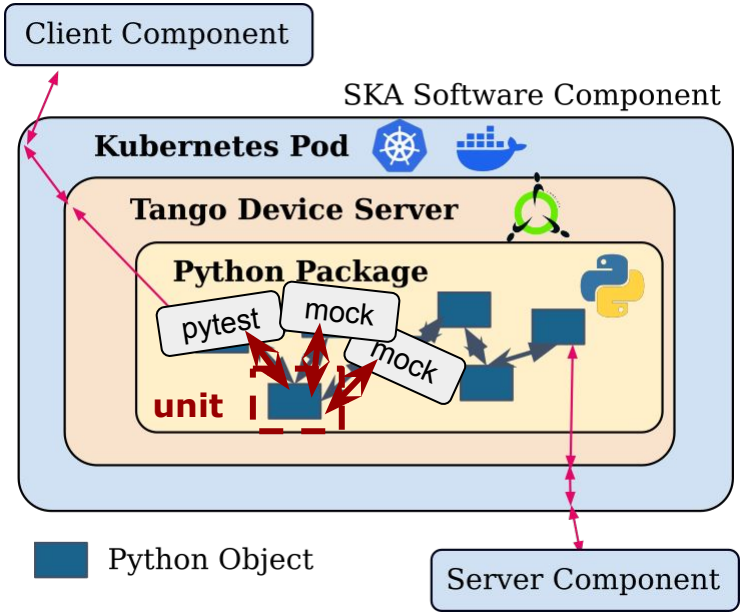
[1] *Agile Release Train* [2] *https://scaledagileframework.com/agile-teams/*

Testing strategy of CSP.LMC



* for simplicity only one server/client is reported

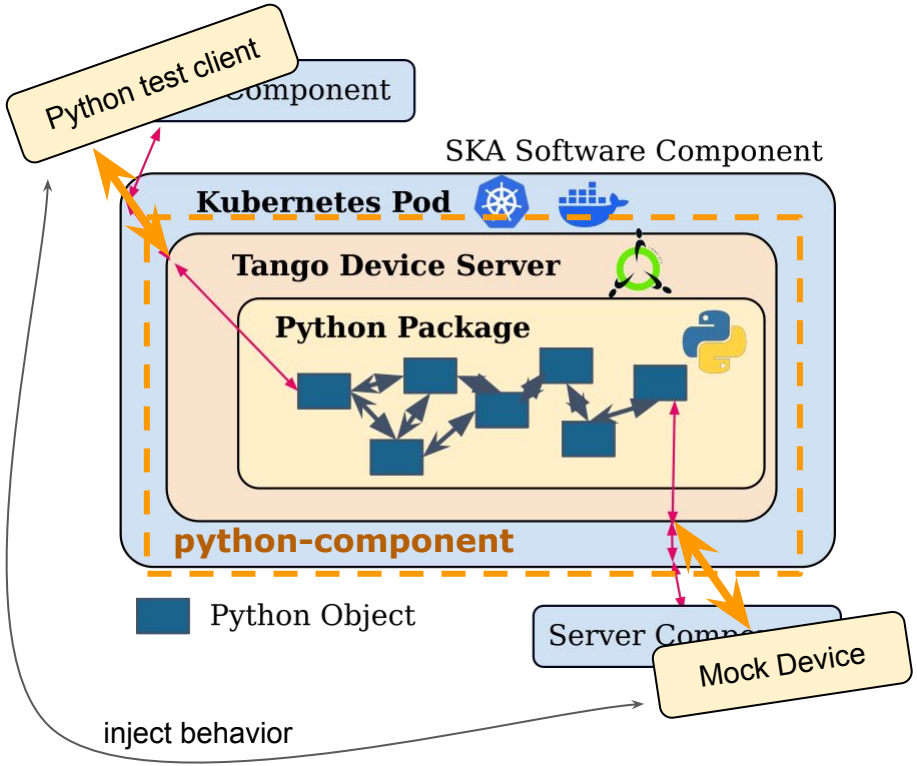
Testing strategy of CSP.LMC



Tests are performed with a *multi-level strategy:*

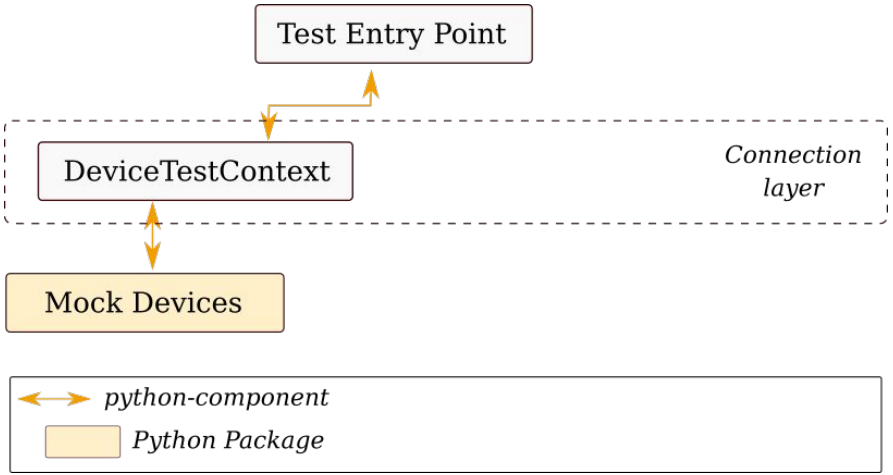
- *unit* tests

Testing strategy of CSP.LMC

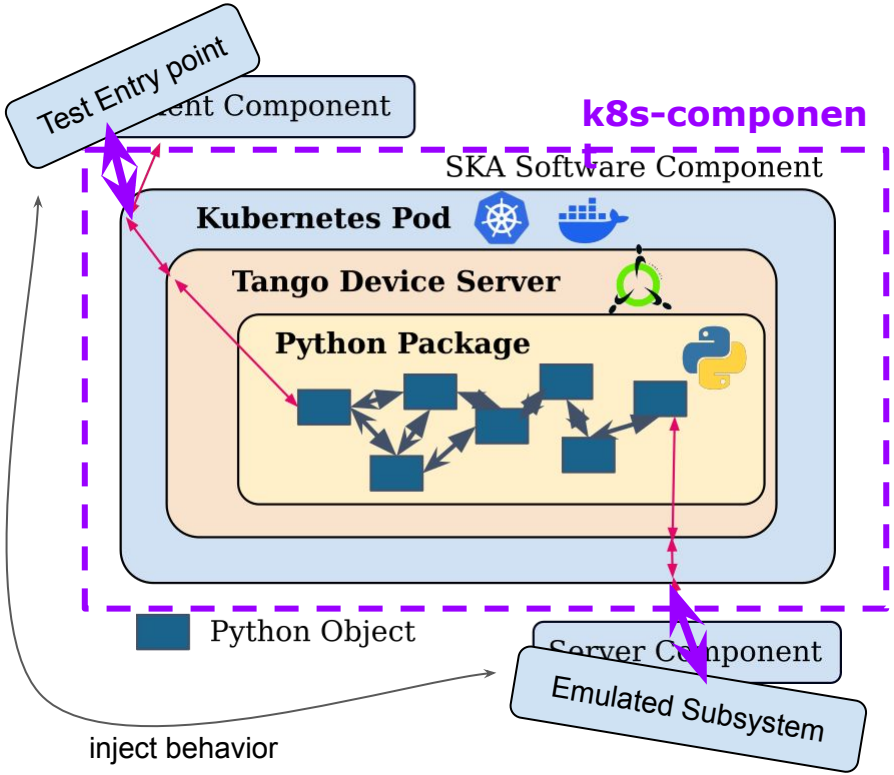


Tests are performed with a *multi-level strategy:*

- *unit* tests
- *python-component* tests

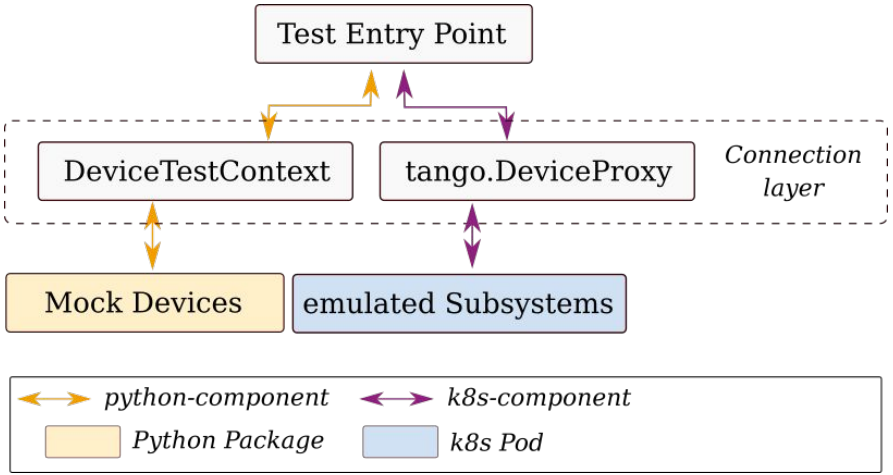


Testing strategy of CSP.LMC

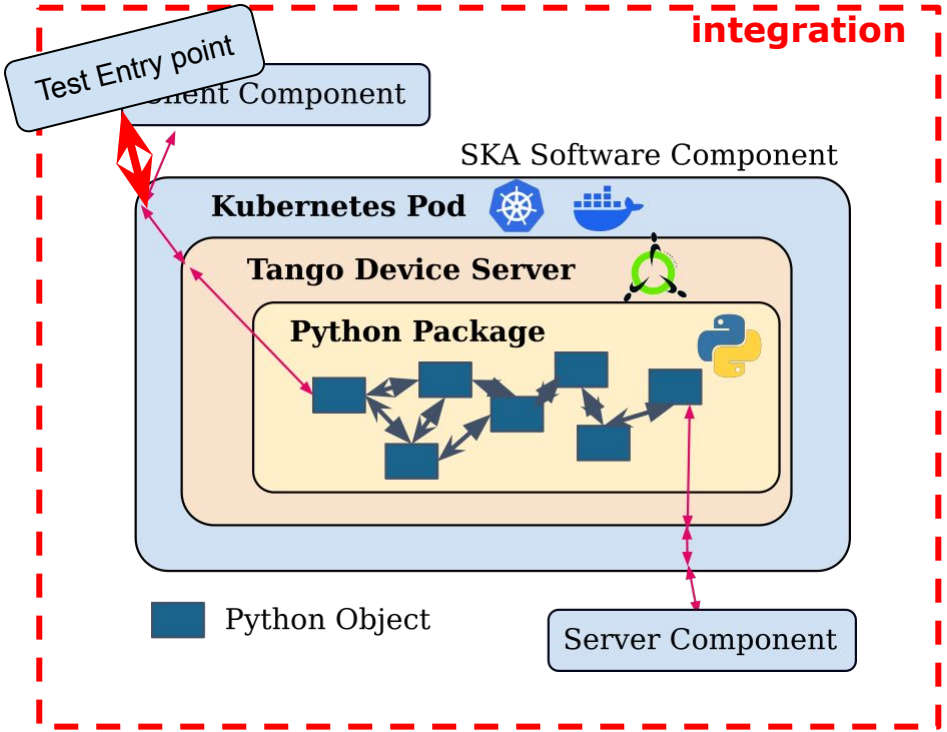


Tests are performed with a *multi-level strategy:*

- *unit* tests
- *python-component* tests
- *k8s-component* tests

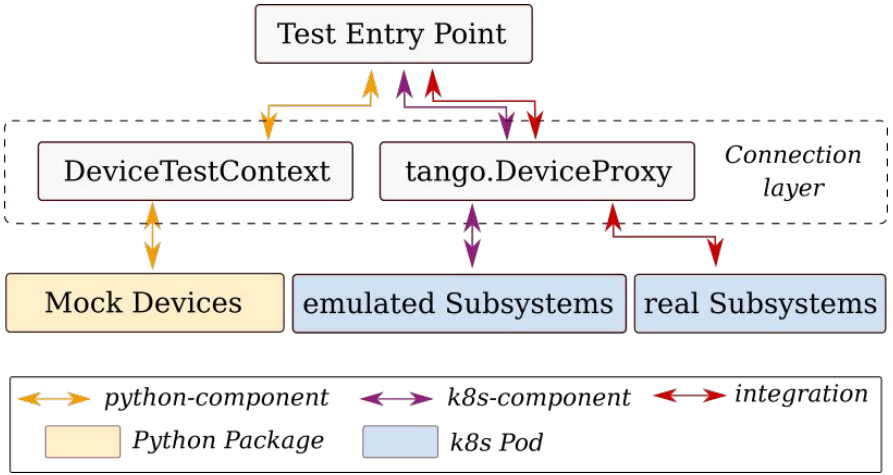


Testing strategy of CSP.LMC



Tests are performed with a *multi-level strategy:*

- *unit* tests
- *python-component* tests
- *k8s-component* tests
- *integration* tests



BDD testing

"The purpose of testing is to **increase confidence** for **stakeholders** through **evidence**"[1]

Integration and *component* tests follow the *Behaviour Driven Development (BDD)* approach. They are written in the *Gherkin* language.

"step"

Passed	Failed	
✓	✓	**Given:** All subsystems are fresh initialized
✓	✓	**When:** On Command is issued on CspController
✓	✓	**Then:** CbfController longRunningCommandStatus is (0, COMPLETED)
✓	✗	**And:** CbfController state is ON
✓		**And:** CbfSubarray0 state is ON

Each "step" is translated to a specific Python function and can be utilised in different tests

Gherkin files can be used as **living documentation**

"regression" testing

A single failing test let the CI/CD pipeline to fail and prevent any MR to be effective!

[1]Dan North "We need to talk about testing"- dannorth.net

“Test flakiness” and data mining

Tests **randomly fail** during CI/CD pipeline execution → *“test flakiness”*

test name	fail rate	num of execution	most failed step - mfs	mfs frequency
cspcontroller healthstate is unknown 1	0.9333	105	CspController HealthState is UNKNOWN	1.0
assignresources rejected on subarray01 without pst beams	0.6667	6	All subsystems are fresh initialized without PST beams	1.0
csp controller reports simulationmode	0.6667	111	CspController SimulationMode is FALSE	1.0
obsstate subscription on subarray01	0.0631	111	All subsystems are fresh initialized	1.0
state subscription on controller and subarray01	0.0541	111	CspController state is subscribed for archiving	0.5
configure rejected on ready subarray01 with pst beams	0.0472	106	All subsystems are fresh initialized with PST beams	1.0
configure rejected on idle subarray01 with pst beams	0.036	111	All subsystems are fresh initialized with PST beams	1.0
assignresources rejected on subarray01 with pst beams	0.036	111	All subsystems are fresh initialized with PST beams	1.0
all commands on subarray01 with pst beam	0.036	111	All subsystems are fresh initialized with PST beams	1.0
csp controller reports healthstate	0.027	111	All subsystems are fresh initialized	1.0

an example of test statistics: the 10 most failing tests

data mining on test result helps in providing:

- better metrics on test quality;
- hints on “deeply hidden” bugs

❓ *Predictive maintenance* also for software?

About Tango Controls

"Tango Controls is an object oriented, *distributed control system framework*."

- it is built around the concept of **devices,** that run into device servers
  - each device has *state machine*, *commands* , *pipes* and *attributes*
- each Tango system has a centralised **database** that:
  - stores configuration data to start up device servers;
  - acts as a name server storing the dynamic network addresses;
- uses CORBA (syncronous) and ZMQ (asyncronous) to communicate between device server and clients;
- kernel written in C++;
- can be programmed in C++, **Python** or Java.