
ASTRON
Netherlands Institute for Radio Astronomy
SKAO

LOFAR2.0: Station Control Upgrade

Thomas Juerges (SKAO, ASTRON),
Jan David Mol, Thijs Snijder (ASTRON)

ICALEPCS 2021

ASTRON
Netherlands Institute for Radio Astronomy

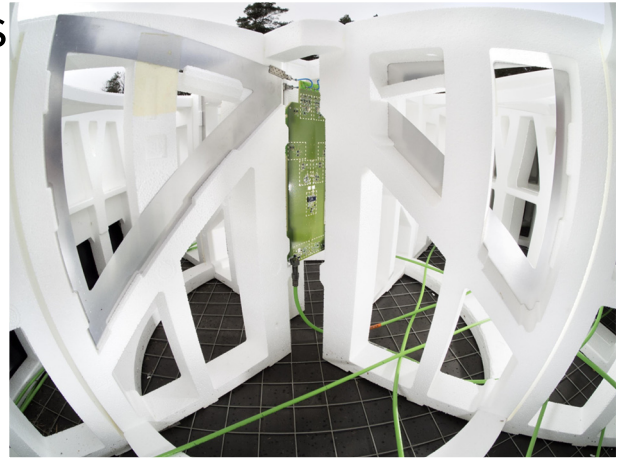
LOFAR

- 10 – 270 MHz
- 38 Dutch stations + 14 Intl
  - EU and non-EU now…
- Central Processing
  - 225 – 390 Gbit/s input
  - GPU & CPU clusters (NL)
- Archives
  - ~20PB
  - 3 locations (NL, DE, PL)

LOFAR

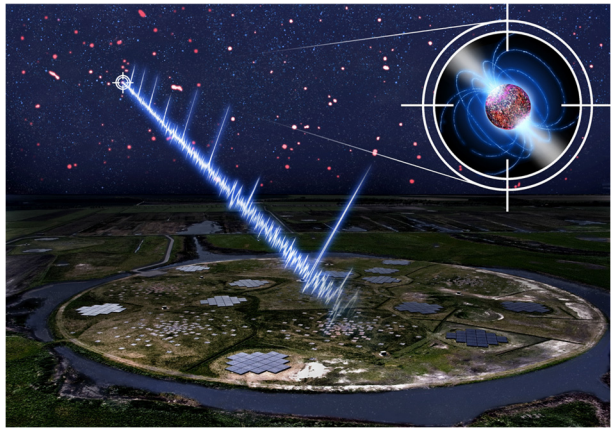
LOFAR Station

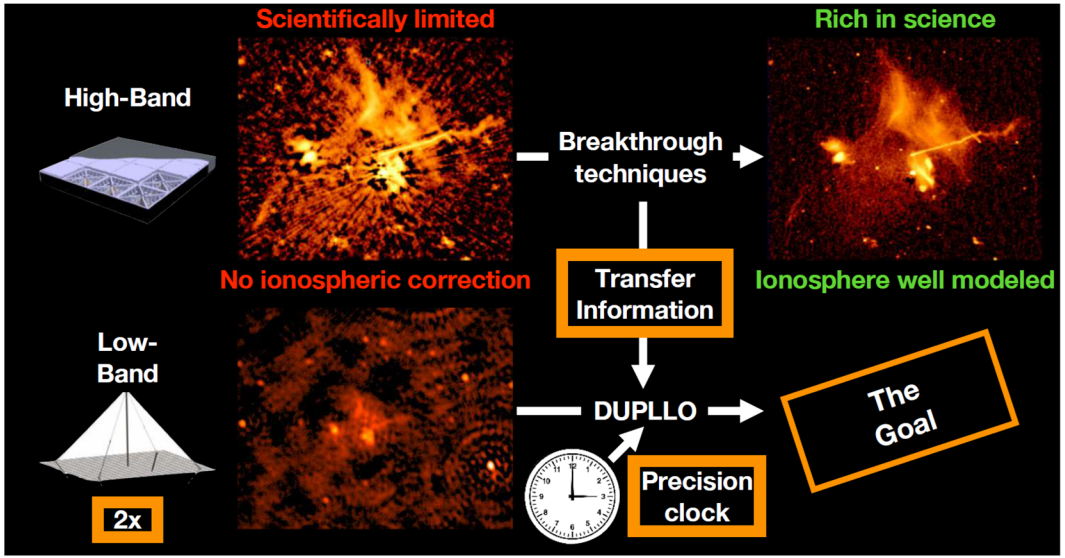
- 96 low-band antennas
- 48-96 high-band tiles
- Digital processor



LOFAR 2.0

- Upgrade to new generation
  - New hardware → New Software
  - OPC-UA hardware protocol interface
  - *Tango Controls as new monitoring & control system*
- Stations first
  - All inputs simultaneously
  - 2-3x increase
- Deployment: 2022 - 2023





LOFAR

Station Internals

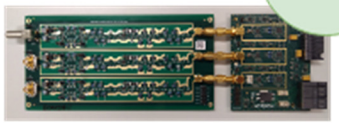
Cabinet

Monitoring & Control

Central M&C

Antennas

Filters & ADC



Digital Processing

Central Processing

Timing Distribution

Central Clock

New Software & Interface Design

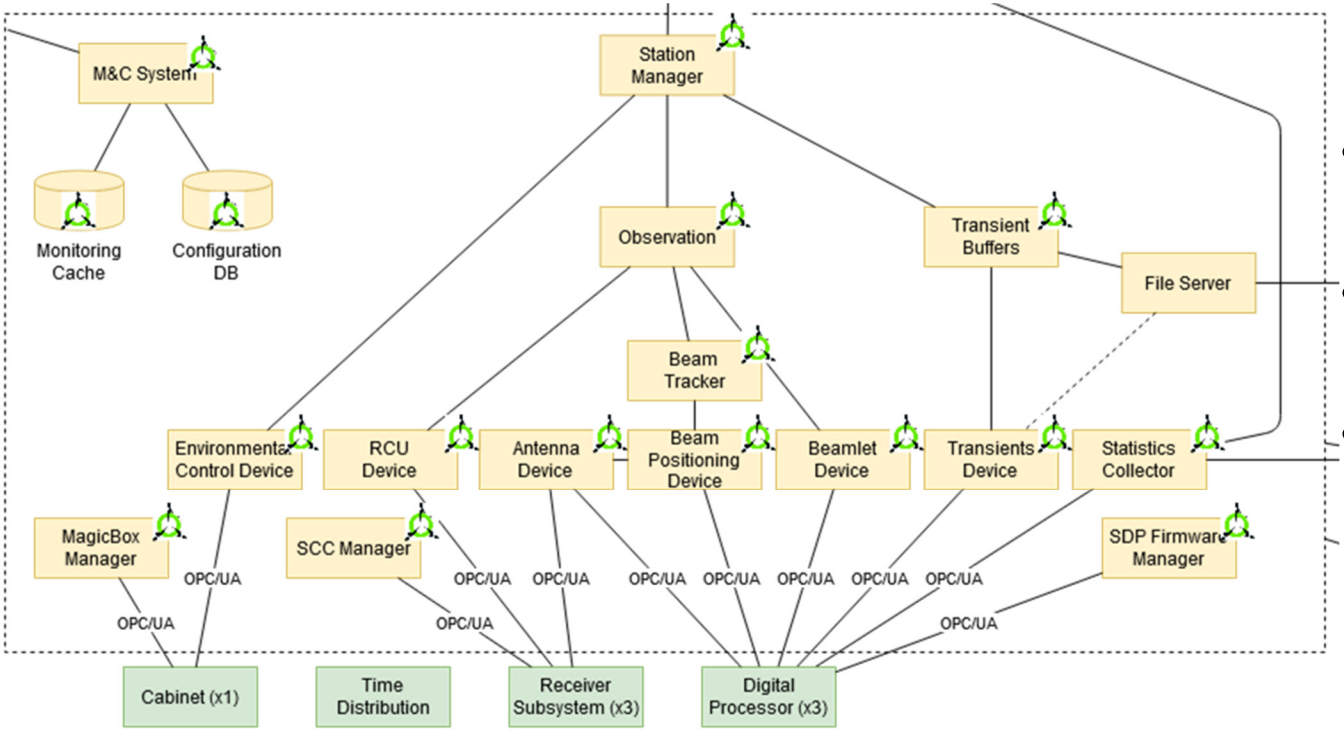
Property	LOFAR1	LOFAR2
Control	Custom	Tango Controls
Monitoring	Custom + WinCC/OA	Tango Controls
Software interfaces	Custom	Tango Controls
Hardware interfaces	Custom (raw Ethernet)	OPC/UA + TCP/IP
Software language	C++98	Python 3

- Easier to operate
- Easier to maintain
- Easier to debug
- Easier to integrate
- Easier to extend

➠ Lower operational cost

➠ Improvements make it quicker into production

Software Architecture



- Model functionality
- Software devices
- Open interfaces

Tango Controls helps a lot!

Object-oriented Monitor and Control framework for the 21st century

- Modelling hardware functionality in software is simple (Object-oriented design)
- Dependencies within the hardware is easy to model (hierarchical devices)
- Python 3
  - Fully supported (many, many good and mature libraries available)
  - CI/CD integration simple (Gitlab is your friend)
- Docker
  - Deployment is host independent ➤ Less headaches
  - External frameworks are configured out of the box

Jupyter: Easy access to Tango Controls Devices

jupyter Untitled1 (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted StationControl

Run Code

In [7]:
```
for device in devices:
    try:
        print(f"{device}: {device.state()}")
    except (ConnectionFailed, CommunicationFailed):
        print(f"{device}: UNREACHABLE")
```

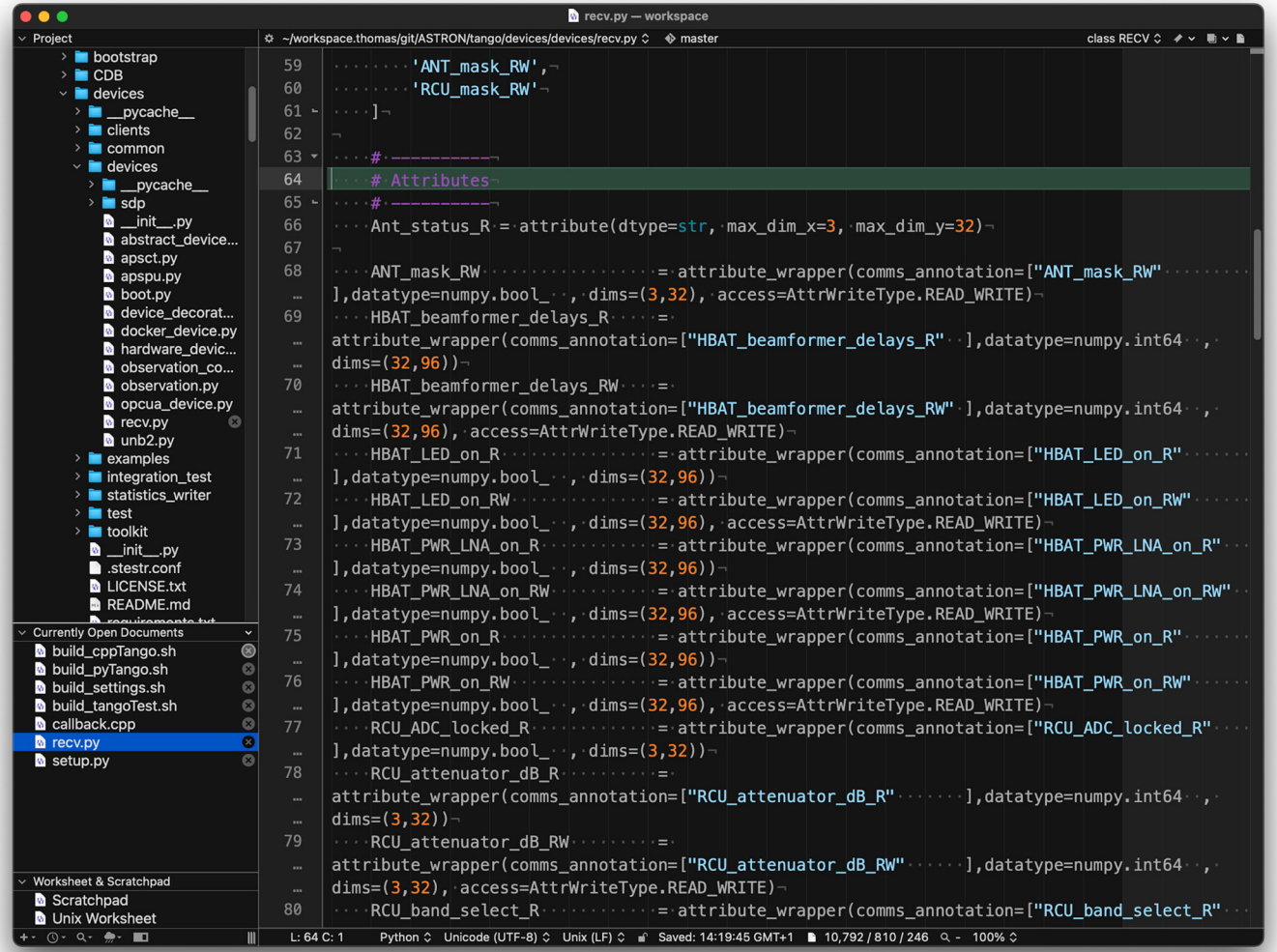
```
RECV(lts/recv/1): OFF
SDP(lts/sdp/1): ON
SST(lts/sst/1): OFF
XST(lts/xst/1): OFF
UNB2(lts/unb2/1): OFF
```

In [9]:
```
sdp.FPGA_global_node_index_R
```

Out[9]:
```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15],
      dtype=uint32)
```

In [ ]:

Grafana: passive UIs made easy



Easy to maintain source code

Success stories

- Jupyter notebooks provide full access to all Tango Devices
- Archival of Tango Controls attributes is simple
- Grafana provides "human understandable" monitoring GUIs
- HW engineers can easily access "their" HW and "see" in real time what it is doing. - Look, no LabView! :-)
- OPC-UA servers in simulation mode allow for Station Control software development independently of HW development cadence

Collaborations

- INAF:
  - Front-end (RCU2) co-development
  - 1 FTE software development

- SKAO:
  - Informal contacts for expertise & advice
  - Docker images!


Thank you for your time!
Tango Controls Lead: Thomas Jürges (thomas.juerges@skao.int)
Software Architect: Jan David Mol (mol@astron.nl)
Software Engineer: Thijs Snijder (snijder@astron.nl)
ASTRON is looking for Tango Controls programmers! See http://tiny.cc/help_lofar_tango
ASTRON
Netherlands Institute for Radio Astronomy
2011 © AEROPHOTO EELDE