



ASTRON

Netherlands Institute for Radio Astronomy

SKAO

LOFAR2.0: Station Control Upgrade

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

ICALEPCS 2021



LOFAR



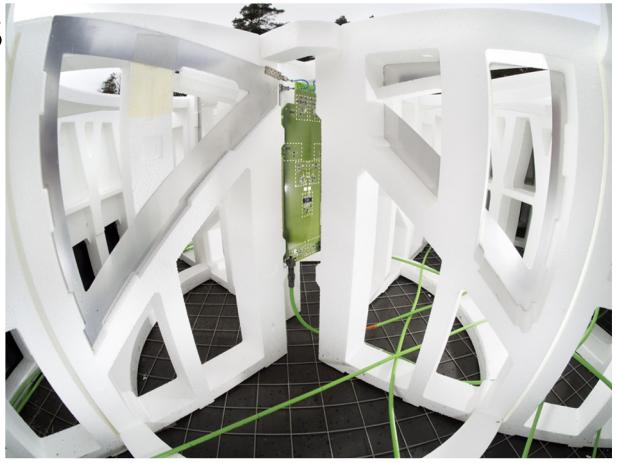
- 10 – 270 MHz
- 38 Dutch stations + 14 Intl
 - EU and non-EU now...
- Central Processing
 - 225 – 390 Gbit/s input
 - GPU & CPU clusters (NL)
- Archives
 - ~20PB
 - 3 locations (NL, DE, PL)



LOFAR

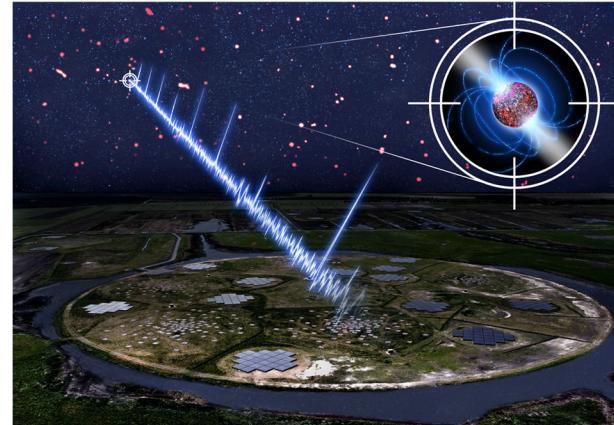
LOFAR Station

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

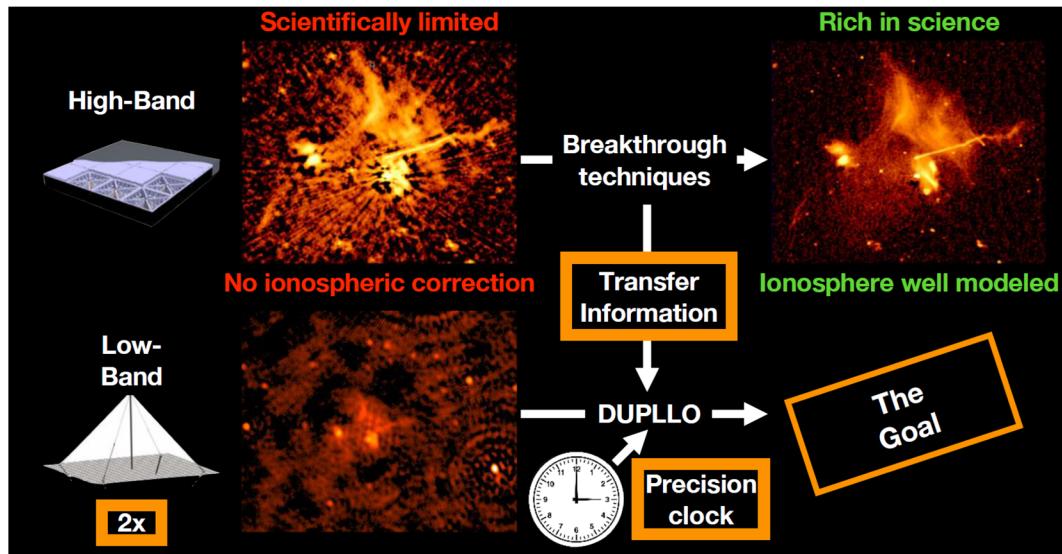


LOFAR 2.0

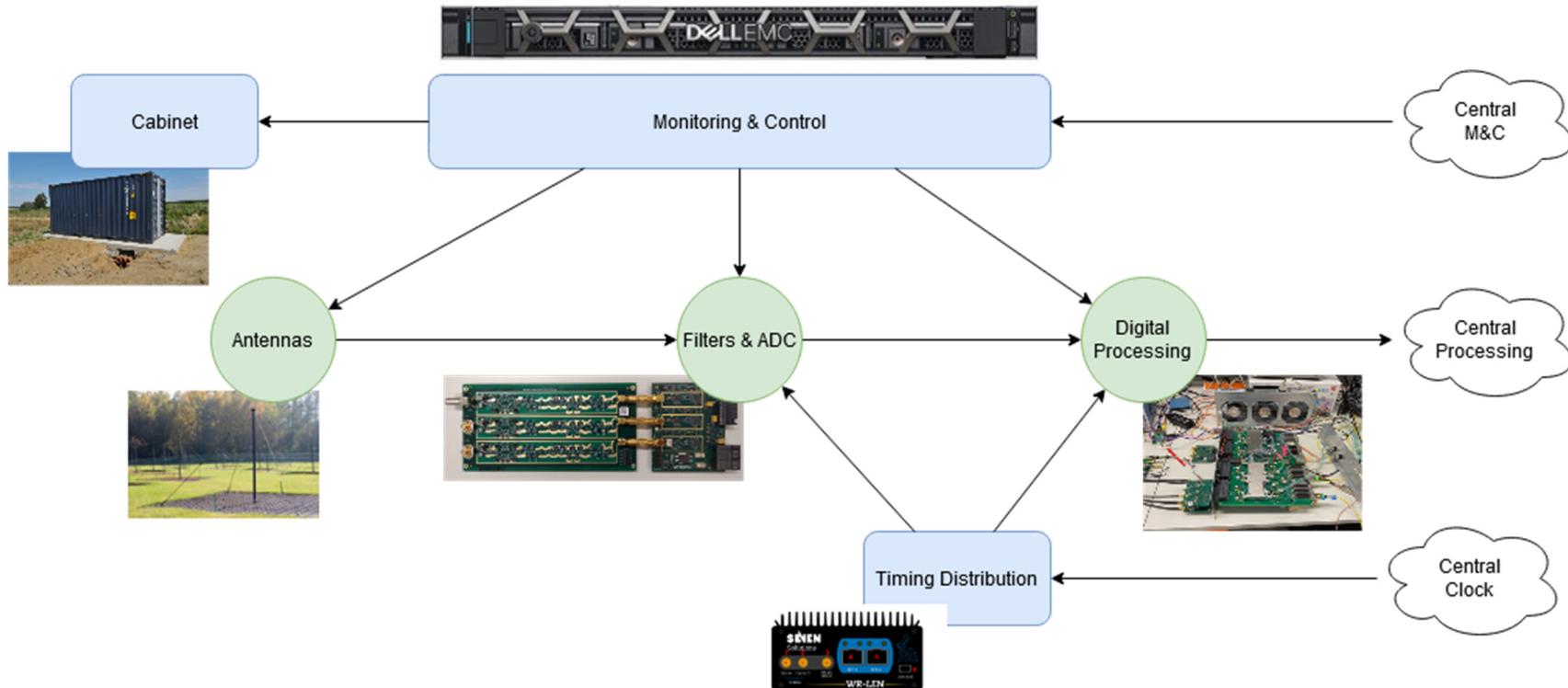
- Upgrade to new generation
 - New hardware → New Software
 - OPC-UA hardware protocol interface
 - *Tango Controls as new monitoring & control system*
- Stations first
 - All inputs simultaneously
 - 2-3x increase
- Deployment: 2022 - 2023



LOFAR



Station Internals



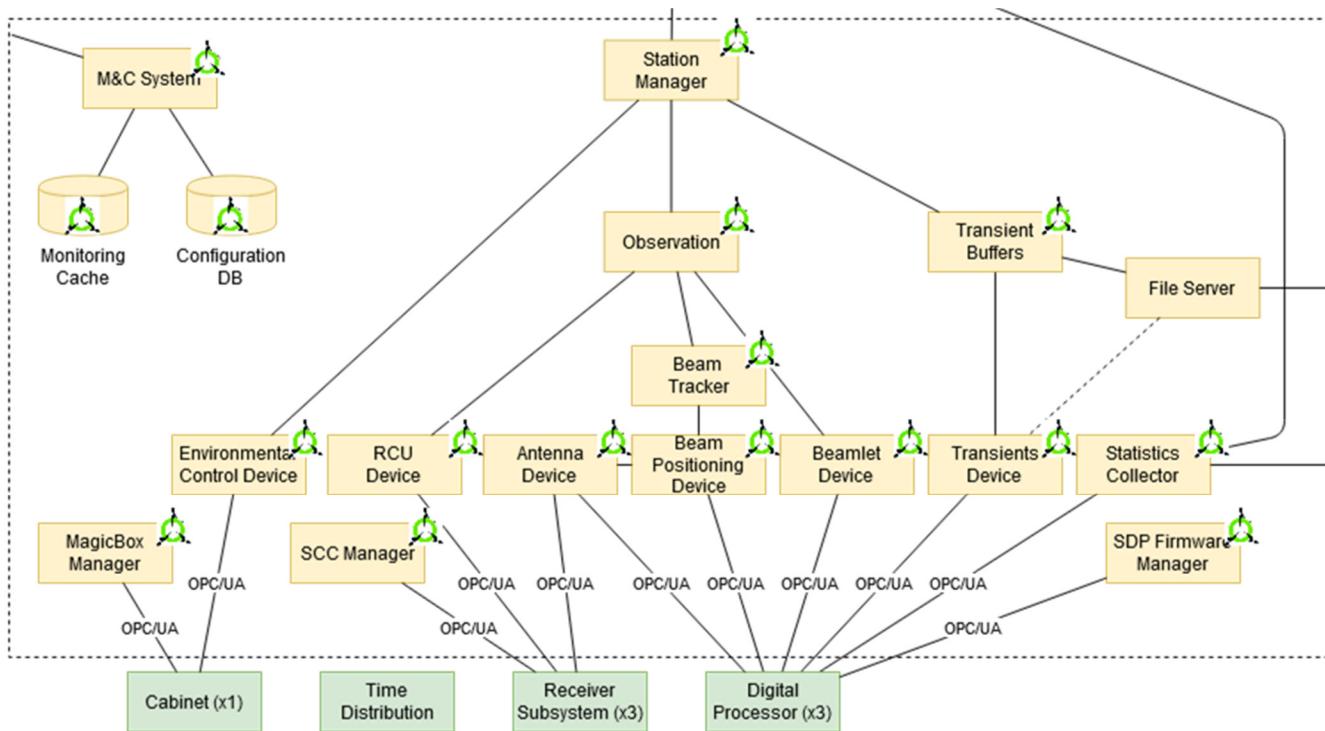
New Software & Interface Design

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 O

In [7]:

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

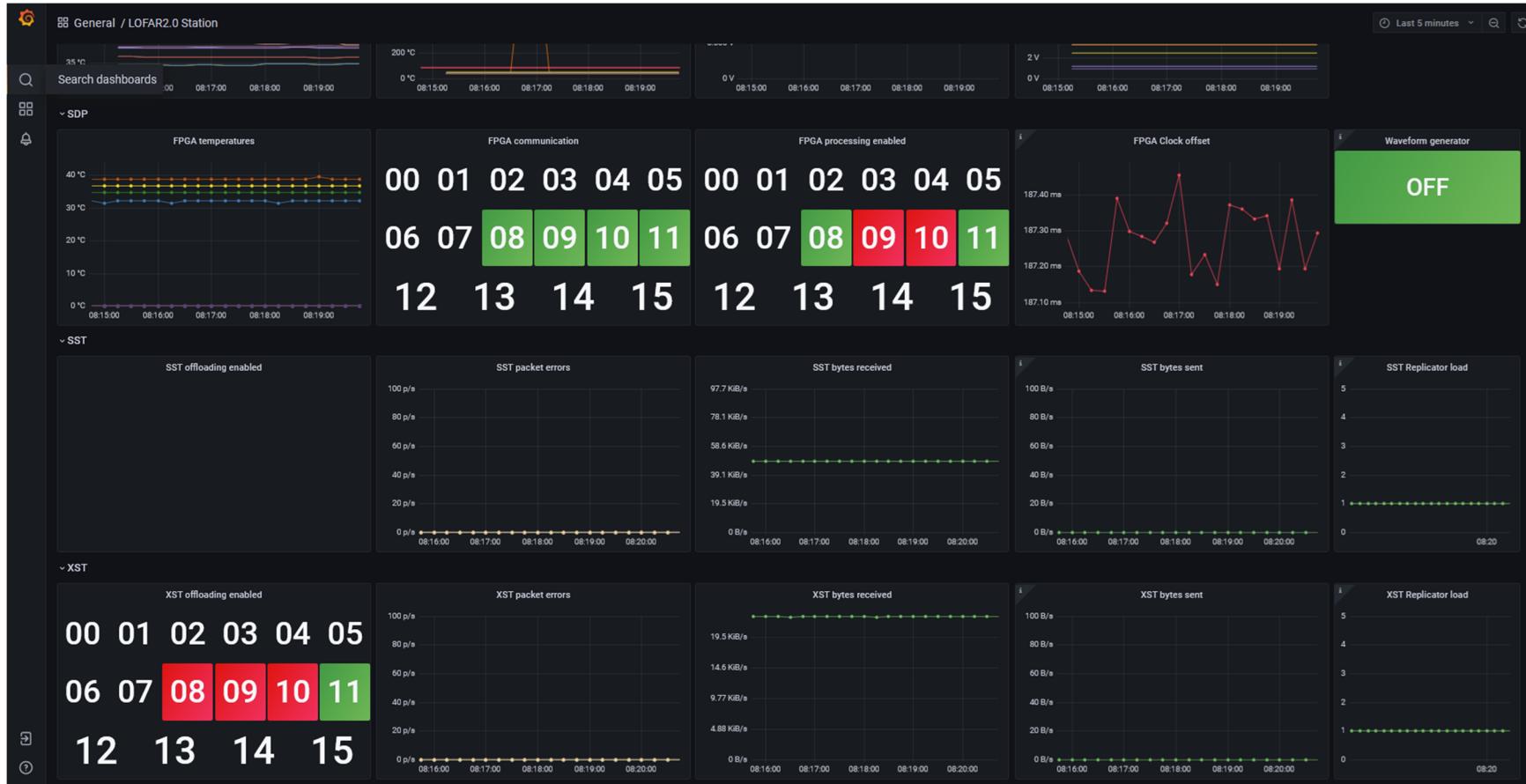
RECV(lts/recv/1): OFF
SDP(lts/sdp/1): ON
SST(lts/sst/1): OFF
XST(lts/xst/1): OFF
UNB2(lts/unb2/1): OFF

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



recv.py — workspace

Project

- > bootstrap
- > CDB
- > devices
 - > __pycache__
 - > clients
 - > common
 - > devices
 - > __pycache__
 - > sdp
 - __init__.py
 - abstract_device...
 - apsct.py
 - apspu.py
 - boot.py
 - device_decorat...
 - docker_device.py
 - hardware_devic...
 - observation_co...
 - observation.py
 - opcua_device.py
 - recv.py
 - unb2.py
 - > examples
 - > integration_test
 - > statistics_writer
 - > test
 - > toolkit
 - __init__.py
 - .sestr.conf
 - LICENSE.txt
 - README.md
- > requirements.txt

Currently Open Documents

- build_cppTango.sh
- build_pyTango.sh
- build_settings.sh
- build_tangoTest.sh
- callback.cpp
- recv.py
- setup.py

Worksheet & Scratchpad

- Scratchpad
- Unix Worksheet

59 'ANT_mask_RW',
60 'RCU_mask_RW'
61]
62
63 # -----
64 # Attributes
65 # -----
66 Ant_status_R = attribute(dtype=str, max_dim_x=3, max_dim_y=32)
67
68 ANT_mask_RW = attribute(annotation=["ANT_mask_Rw"],
69 datatype=numpy.bool_, dims=(3,32), access=AttrWriteType.READ_WRITE)
70 HBAT_beamformer_delays_R =
71 attribute(annotation=["HBAT_beamformer_delays_R"],
72 datatype=numpy.int64, dims=(32,96))
73 HBAT_beamformer_delays_RW =
74 attribute(annotation=["HBAT_beamformer_delays_RW"],
75 datatype=numpy.int64, dims=(32,96), access=AttrWriteType.READ_WRITE)
76 HBAT_LED_on_R =
77 attribute(annotation=["HBAT_LED_on_R"],
78 datatype=numpy.bool_, dims=(32,96))
79 HBAT_LED_on_RW =
80 attribute(annotation=["HBAT_LED_on_RW"],
81 datatype=numpy.bool_, dims=(32,96), access=AttrWriteType.READ_WRITE)
82 HBAT_PWR_LNA_on_R =
83 attribute(annotation=["HBAT_PWR_LNA_on_R"],
84 datatype=numpy.bool_, dims=(32,96))
85 HBAT_PWR_LNA_on_RW =
86 attribute(annotation=["HBAT_PWR_LNA_on_RW"],
87 datatype=numpy.bool_, dims=(32,96), access=AttrWriteType.READ_WRITE)
88 HBAT_PWR_on_R =
89 attribute(annotation=["HBAT_PWR_on_R"],
90 datatype=numpy.bool_, dims=(32,96))
91 HBAT_PWR_on_RW =
92 attribute(annotation=["HBAT_PWR_on_RW"],
93 datatype=numpy.bool_, dims=(32,96), access=AttrWriteType.READ_WRITE)
94 RCU_ADC_locked_R =
95 attribute(annotation=["RCU_ADC_locked_R"],
96 datatype=numpy.bool_, dims=(3,32))
97 RCU_attenuator_dB_R =
98 attribute(annotation=["RCU_attenuator_dB_R"],
99 datatype=numpy.int64, dims=(3,32))
100 RCU_attenuator_dB_RW =
101 attribute(annotation=["RCU_attenuator_dB_RW"],
102 datatype=numpy.int64, dims=(3,32), access=AttrWriteType.READ_WRITE)
103 RCU_band_select_R =
104 attribute(annotation=["RCU_band_select_R"],

L: 64 C: 1 Python Unicode (UTF-8) Unix (LF) Saved: 14:19:45 GMT+1 10,792 / 810 / 246 100%

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!



An aerial photograph of the Low Frequency Array (LOFAR) telescope. The array consists of numerous white rectangular panels arranged in a grid pattern across a green field. These panels are part of larger, more complex antenna structures. The field is surrounded by a network of small, winding blue waterways and patches of brown and green agricultural land. The overall pattern of the array is roughly circular.

Thank you for your time!

Tango Controls Lead: Thomas Jürges (thomas.juerges@skao.int)
Software Architect: Jan David Mol ([mol@astron.nl](mailto/mol@astron.nl))
Software Engineer: Thijs Snijder (snijder@astron.nl)

ASTRON is looking for Tango Controls programmers! See http://tiny.cc/help_lofar_tango



Netherlands Institute for Radio Astronomy

2011 © AEROPHOTO EELDE