



ASTRON

Netherlands Institute for Radio Astronomy

LOFAR 2.0 - Hardware, Software & Pytango

High level overview

Corne Lukken (Dantali0n / PD3SU)

Contents

- LOFAR 2.0
- Station Hardware
- Dataflow & Software components
- Station Software
- Tango



July 3, 2023

4

ASTRON

Netherlands Institute for Radio Astronomy

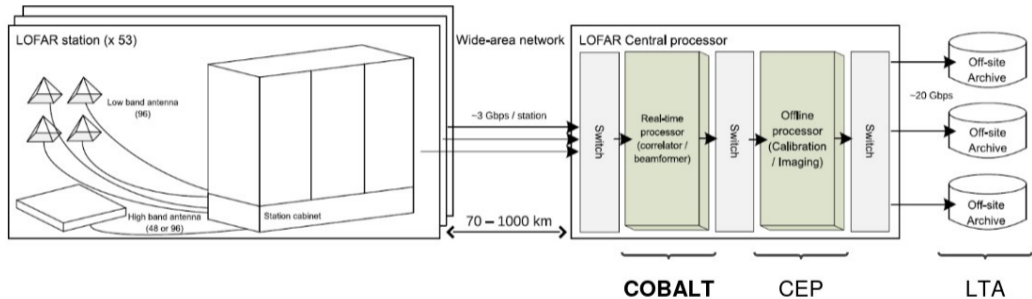


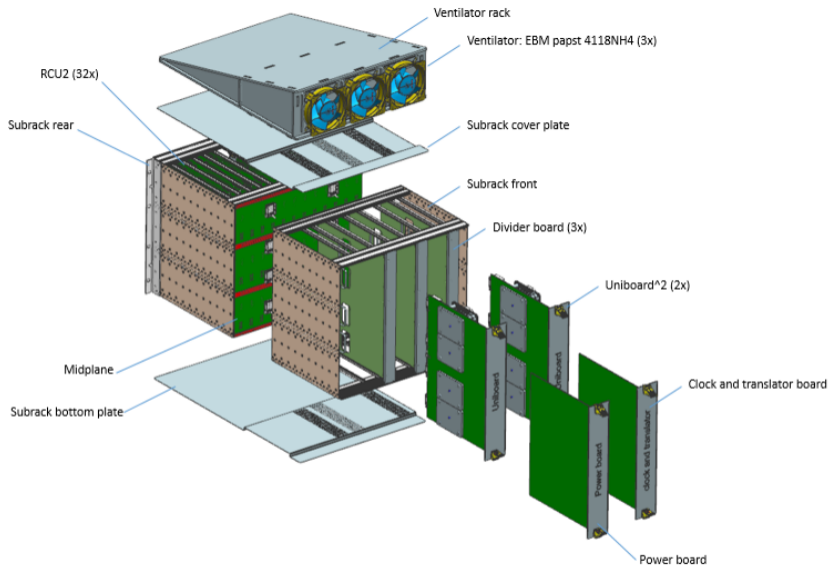
July 3, 2023

5

ASTRON

Netherlands Institute for Radio Astronomy



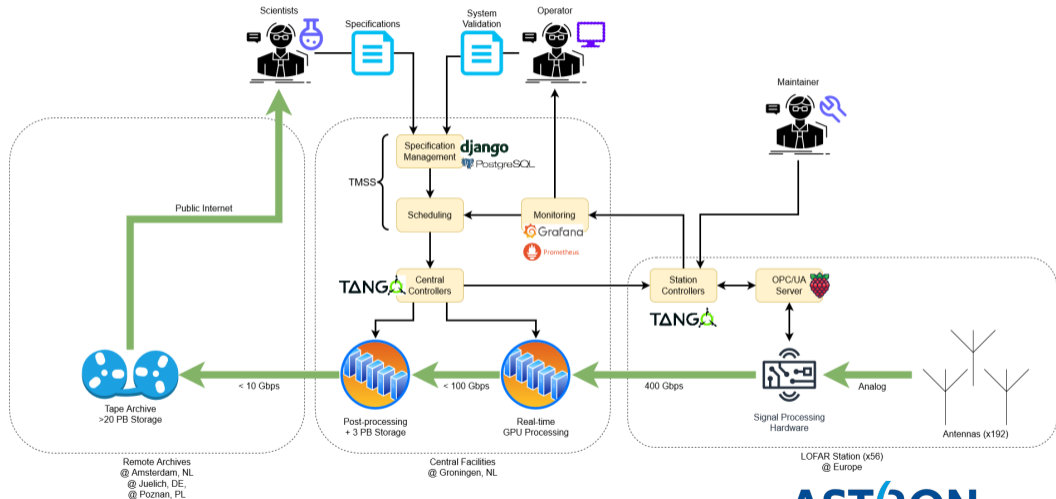


July 3, 2023

7

ASTRON

Netherlands Institute for Radio Astronomy



July 3, 2023

8

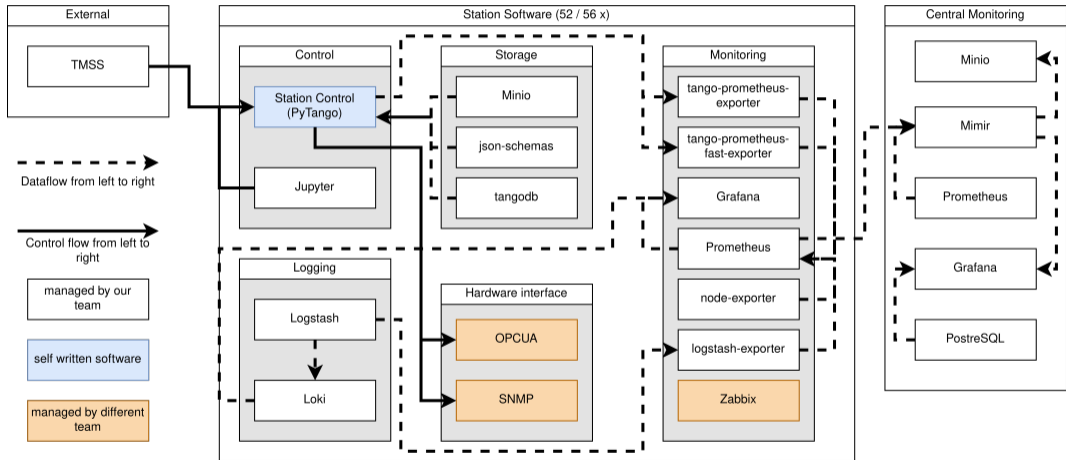
ASTRON

Netherlands Institute for Radio Astronomy

What is LOFAR 2.0

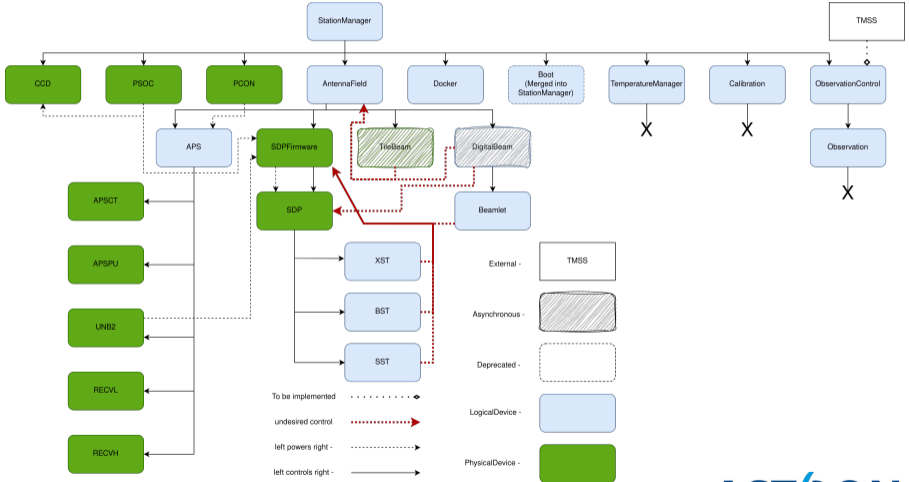
- Monitoring with Grafana
- Observations combine HBA & LBA antenna
- Observations multiple beams (Mega Mode)
- Replaced station SCADA with Pytango
- Upgraded hardware (RCU2, Uniboard2)
- Upgraded timing (White Rabbit)

Station Software Components



The tango prometheus exporter is developed at [SKA](#)

PyTango device hierarchy



Challenges

- Long term archiving of monitoring data & visualizations
 - <https://gitlab.com/ska-telescope/TANGO-grafana/-/tree/master/exporter>
- Hardware translation, attribute read / write function boilerplate
- Local & international stations (hardware differences)
- Use of DeviceTestContext, code coverage and mocking

AttributeWrapper

git.astron.nl/lofar2.0/attributewrapper

- Generate attribute read / write functions at runtime
- Custom clients for underlying hardware
 - OPCUA - <https://tinyurl.com/opcuaclient>
 - SNMP - <https://tinyurl.com/snmpclient>
 - Dimensionality conversions
 - Transparent reconnections
- Codebase could use refactoring / documentation / examples

AttributeWrapper

```
UNB2TR_I2C_bus_QSFP_error_R = AttributeWrapper(  
    comms_annotation=["UNB2TR_I2C_bus_QSFP_error_R"],  
    datatype=numpy.int64,  
    dims=(N_unb, N_qsfp),  
)
```

```
async def read_function(self):  
    """  
    Read_R function  
    """  
  
    value = await self.node.get_value()  
  
    try:  
        def fix_string(s):  
            return s.encode("latin-1", errors="replace").decode("latin-1")  
  
        if type(value) == list and len(value) > 0 and type(value[0]) == str:  
            value = [fix_string(v) for v in value]  
        elif type(value) == str:  
            value = fix_string(value)  
  
        if self.dim_y + self.dim_x == 1:  
            # scalar  
            return value  
        elif self.dim_y != 0:  
            # 2D array  
            value = numpy.array(  
                numpy.split(numpy.array(value), indices_or_sections=self.dim_y)  
            )  
        else:  
            # 1D array  
            value = numpy.array(value)  
  
    return value
```

```
async def _connect_opcua(self):  
    # connect  
    await self.opcua_connection.start()  
  
    # map an access helper class  
    for i in self.attr_list():  
        try:  
            if not i.comms_id or i.comms_id == OPCUAConnection:  
                await i.async_set_comm_client(self, self.opcua_connection)
```

Code coverage demo

Impediments

- Device server configuration management
- Understanding and identifying PyTango features
- Update code for solved issues
 - Lack of unit tests for device servers
 - Almost no use of event subscriptions
- Integration with real hardware

Resources

- Repositories, it's FOSS! (Apache 2)
 - git.astron.nl/lofar2.0/tango
 - git.astron.nl/lofar2.0/lofar-station-client
 - git.astron.nl/lofar2.0/attributewrapper
 - git.astron.nl/lofar2.0/grafana-station-dashboards
 - git.astron.nl/lofar2.0/operations-central-management
- dantalion.nl/knowledgetransfer.html
 - Corrections, questions? tango@dantalion.nl

Thanks to the team!

- Taya Snijder
- Jan David Mol
- Hannes Feldt
- Stefano Di Frischia (INAF)
- Jorrit Schaap
- Fanna Lautenbach
- Leonardo Pelonero (INAF)
- Jörn Künsemöller (Universität Bielefeld)
- Arno Schoenmakers
- Sander ter Veen
- Rene Lourens

