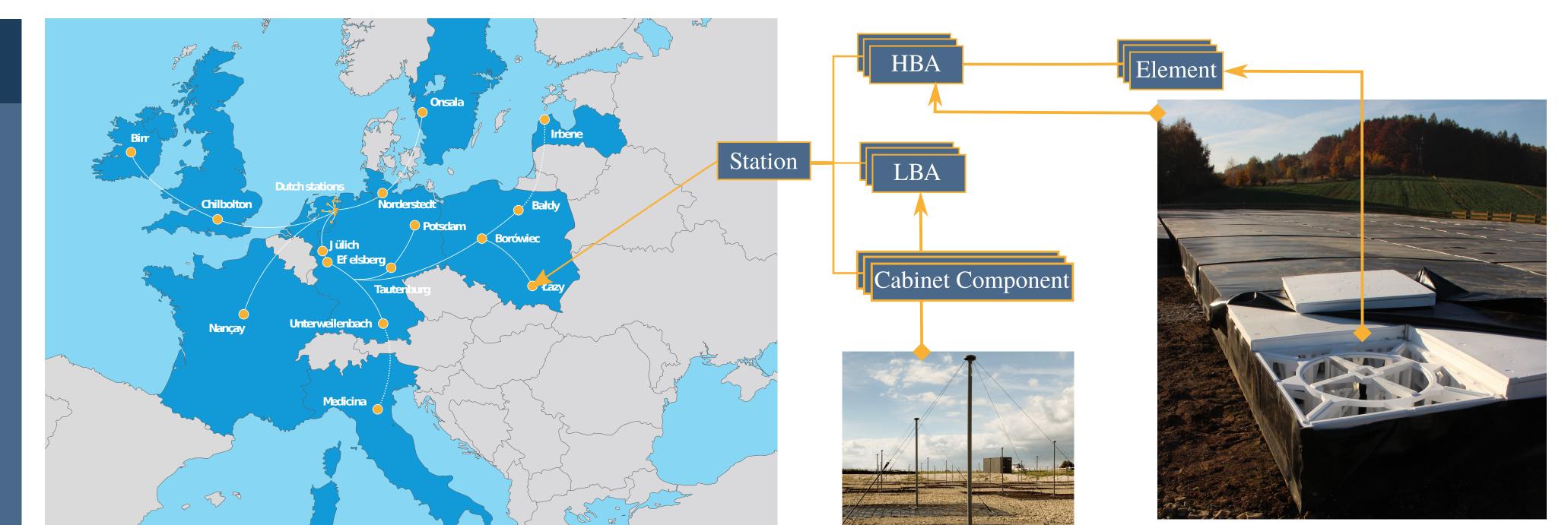
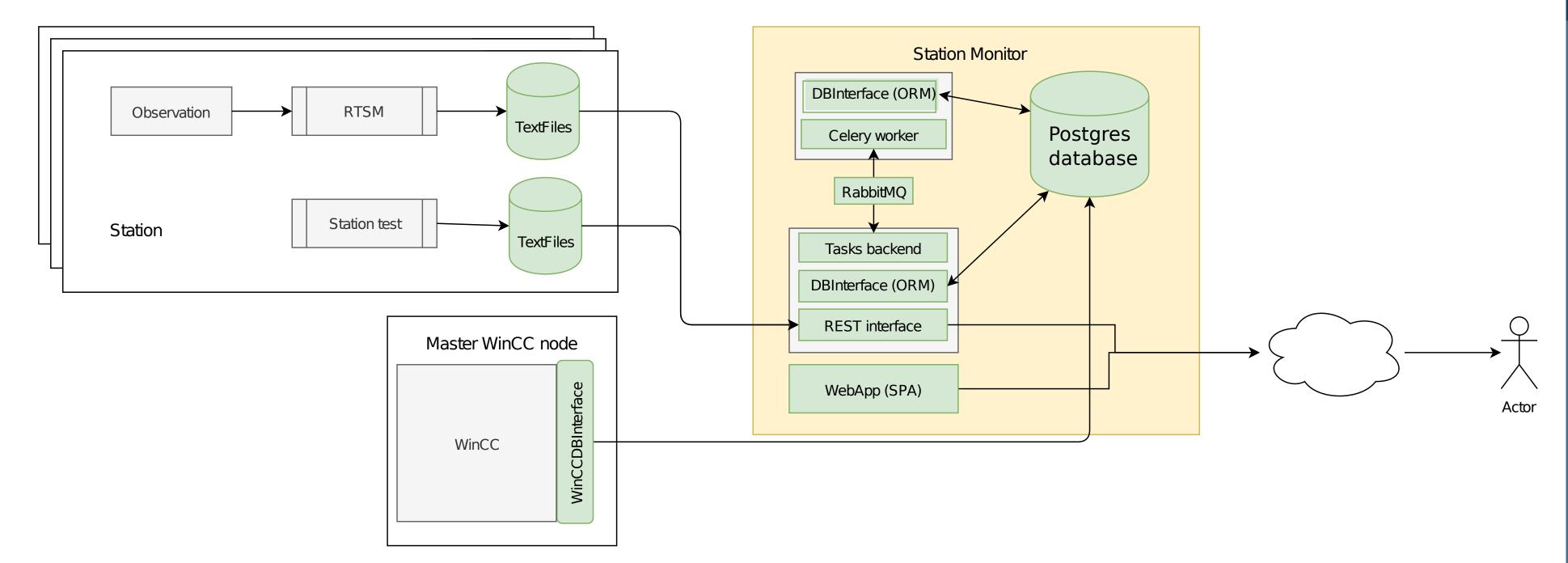
MMIS: StationMonitor

combining different tests data to provide an overview of the LOFAR array status

The LOFAR telescope

The LOFAR telescope is composed of several stations spread across Europe. A station consists of many components. However, only a subset of such components is tested by the station tests. For sake of simplicity, we will focus on the antennas only. There are two kind of antennas in a LOFAR station: the Low Band Antenna (LBA) and the High Band Antenna (HBA). LBA antennas are sensitive for frequency range ~10-90 MHz while HBA's are sensitive for ~100-200 MHz. An HBA antenna is composed of 16 elements. Each element hosts a couple of dipoles, one per polarization. The LBA antenna is considered a single component with a dipole per polarization.





view lile

Total errors

256

155

152

106

148

66

per error type \$

SUMMATOR_N

Implementation

ASTRON

To expose the WinCC data to the station monitor we developed a c++ application called WinCCDBInterface using the WinCC C++ API and the Qt5 to react on every datapoint change listed in a configuration file and store the datapoint value directly in a table on the Postgres database. The WinCCDBInterface can also be configured to do a checkpoint at a given time were all the value of the configured datapoints are stored regardless if their value changes.

The database interface, tasks back-end and **REST** interface are implemented using the **Django** python framework. We used the **Django** Object relational mapping (ORM) to manage the database interaction and the tables and to expose a **REST** API through the django rest framework plugin. The text files from the RTSM and Station Test are automatically transferred to the Station Monitor server after the test is done. By using file system events, a REST call is performed with the test file and the corresponding task is fired. The tasks are carried out in the task backend developed with the help of the **Celery** framework. We choose the **RabbitMQ** message broker as result backend **Celery**. Using a asynchronous task queue prevents exhausting server resources and long blocking web server requests. A disadvantage is that in some cases a polling mechanism is necessary to check if a task has completed.

The user interface

LOFA		SI	au				nte		0.1									Ducinoci		ation Ove						
Station grou	ip All	station	IS	÷	Errors	s only	Peri	od 1	wk 2	wk 3	3 wk 4	1 wk	Error	type	All -											
Station o	vervi	iew															Latest ob	servations								
	Station	1															Observation	Start date	Stations	Stations with er						
	tests		Late	st obse	ervatio	ns	-										745626	2019-09-19 12:40	6	6						
CS001C	146 11	12 10	74	5618 <mark>2</mark>	745	5158 4	745	380 <mark>2</mark>	7459	94 3	74585	50 3	74514	1 3			745622	2019-09-19 12:20	10	10						
CS002C	147 19	21 15	74	5622 <mark>3</mark>	745	5164 2	745	158 🠬	7453	80 3	74599	94 2	74585) 3			745618	2019-09-19 11:50	15	15						
CS003C	185 3	8 8	74	5622 <mark>2</mark>	745	5618 <mark>2</mark>	745	846 2	7450	84 <mark>2</mark>	73618	33 2	73616	3 2			742173	2019-09-18 18:11	1	1						
CS004C	138 18	17 12	74	5618 <mark>3</mark>	745	5164 <mark>2</mark>	745	158 <mark>2</mark>	7450	84 <mark>2</mark>	74514	14 3	73613	2			746230	2019-09-16 11:35	1	1						
CS005C	154 9	13 6	74	5144 <mark>2</mark>	736	5103 <mark>3</mark>	736	099 2	7448	50 <mark>2</mark>	74513	38 2	74496	2			745164	2019-09-14 10:55	14	14						
CS006C	148 21	22 21	74	5994 <mark>2</mark>	745	5850 0		004 0		nn n	74483	36 <mark>2</mark>	74484	5 2			745158	2019-09-14 06:54	39	39						
CS007C	147 11	16 12	74	745158 6 74515				745084				50 3	74514	1 2			745154	2019-09-14 06:43	16	16						
CS011C	121 16	16 16	74	5622 <mark>2</mark>	745	561 S	tart:	2019-09	9-13 10	:00	74514	14 3	74514	3			745380	2019-09-14 04:22	23	23						
CS013C	146 8	7 15	74	745158 3 74515 End: 2019-09-13 12:00					7450 4 3 745150 2						745994	2019-09-13 15:30	21	21								
CS017C	155 17	18 17	74	745626 5 74562 Mode: 745626 2 74562 Errors: HIGH_NOISE 3 745626 3 745618 4 740230 4 740104 4							745158 5 745848 13						745850	2019-09-13 15:17	19	19						
CS021C	166 32	36 29	74								745158 2 745154 2						745848	2019-09-13 15:06	27	27						
CS024C	155 18	16 14	74								745158 4 745848 4						745846	2019-09-13 15:00	12	12						
			_					-				_		-	-		7 745004	2010 00 12 10.00	10	10						
Station to	est si	umma	ary														Station st	atistics	Both test	types 🗢 day						
Date	Time	Stati on	Com p.	SM	VS	TE	SH	DW	CS	JI	vo	MY	LN	HN	мо	os	8,000			error_type						
2019-09-03	14:2 0	LV61 4C	-	CIN			on								me		6,000			DOWN FLAT HIGH_N JITTER						
2019-08-31	20:4 5	RS10 6C	HBA							1				48			s 6,000 b 4,000 aquuu 2,000		2							
			LBH					3						45						1 A A						
			LBL					2						46)8-22)8-24	08-26- 08-20- 08-30- 09-01- 09-03- 09-05	09-07- 09-09- 09-11- 09-13-	09-16- 09-19-						
	20:4 5	FR60 6C	HBA												1		2019-08-22 2019-08-22	2019-08-28 2019-08-28 2019-08-30 2019-09-03 2019-09-03 2019-09-05	2019-09-07 2019-09-09 2019-09-11 2019-09-13	2019-09-16 - 2019-09-19 -						

from the station to the single component



< CS003C	⊗ > ⁺	Type HBA LBH	LBL A	ntenna id	1 •	Type BOTH	I ST-TEST	RTSM	Period 1 w	k 2 wk 4 wk 2019-Aug-2	2 2019-Sep-19	
T Date S S R 27 obs 12d15h I I S 2019-08-31 20:45 I I R 13 obs 5d20h I I S 2019-08-25 20:00 I I R 17 obs 3d12h I I R 2019-08-25 11:46 I I R 2019-08-24 19:11 I I R 2019-08-24 19:10 I I R 2019-08-24 19:11 I I R 2019-08-24 19:10 I I R 2019-08-24 19:10 I I R 2019-08-24 12:12 I I R 2019-08-24 11:30 I I R 2019-08-23 11:20 I I R 2019-08-23 11:30 I I R 2019-08-22 12:33 I I R 2019-08-22 11:54 I I R 2019-08-22 11:30 I										StationTest 2019-08-31		

Future development

The first version is now operational. In the near future the Station Monitor will be extended with additional data types and functionalities. We will add the maintenance log containing repair actions executed by a field engineer and shown in the timeline of components and elements. This helps when interpreting the history of errors and to judge if a repair was successful. Moreover, we will add the possibility to create reports containing various statistics of the data containing in the station monitor. As this needs to be flexible we are currently experimenting with exporting the data to Elasticsearch and use Kibana as interface to create various reports and graphs. Finally, we would like to trigger on certain results of the Station Test or the RTSM so that problematic antennas can be automatically disabled or re-enabled once tested.







LOFAD Otalian

