A modular approach to software in ATLAS: Microservices framework and configuration database for ATLAS ITk

Conference on Computing in High Energy and Nuclear Physics 2024

Gerhard Brandt[1], Daniele Dal Santo[2], Marianna Glazewska[2], Lucas Mollier[2], Dominik Schlothane[1], **Jonas Schmeing**[1], Maren Stratmann[1], Wolfgang Wagner[1]

jonas.schmeing@cern.ch

[1]Bergische Universität Wuppertal [2]Universität Bern

BERGISCHE UNIVERSITÄT WUPPERTAL

ATLAS EXPERIMENT

Introduction

During the Long Shutdown 3 (2026-2028) the ATLAS detector will be upgraded for the HL-LHC. Its Inner Detector will be replaced with the Inner Tracker (ITk[a]). To operate the ITk system tests and later the final detector, a graphical operation and configuration system is needed. For this a flexible and scalable framework based on distributed microservices (MS) has been introduced. Different MS are responsible for configuration or operation of all parts of the readout chain. The configuration database microservice provides the configuration files needed to configure the hardware components of the readout chain. Scans can be performed using the DAQ software standalone or in a ATLAS Trigger and Data Acquisition (TDAQ[b]) partition.

[a] Nucl. Instrum. Methods Phys. Res., A 1045 (2023) 167597

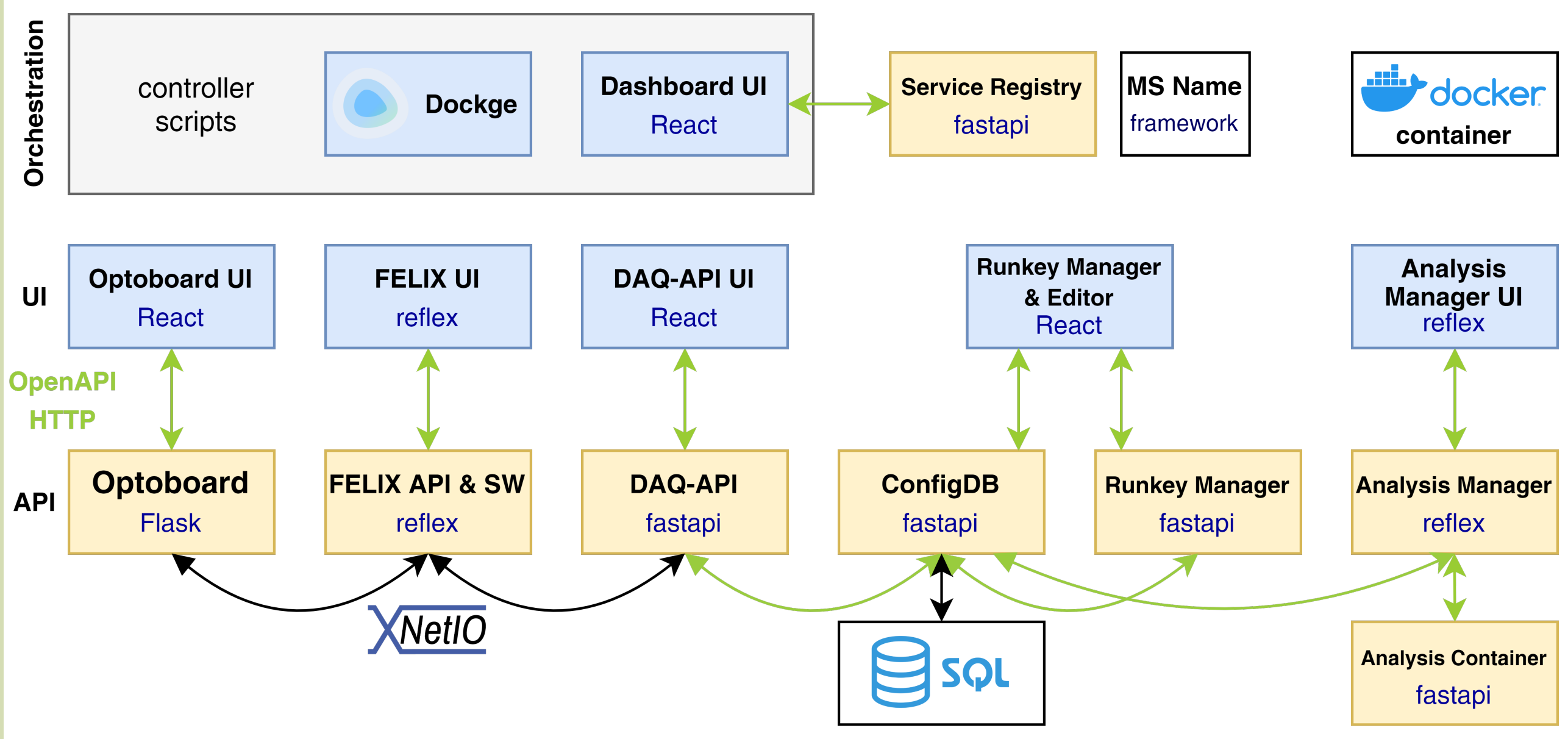
[b] CERN-LHCC-2017-020; ATLAS-TDR-029

What are Microservices?

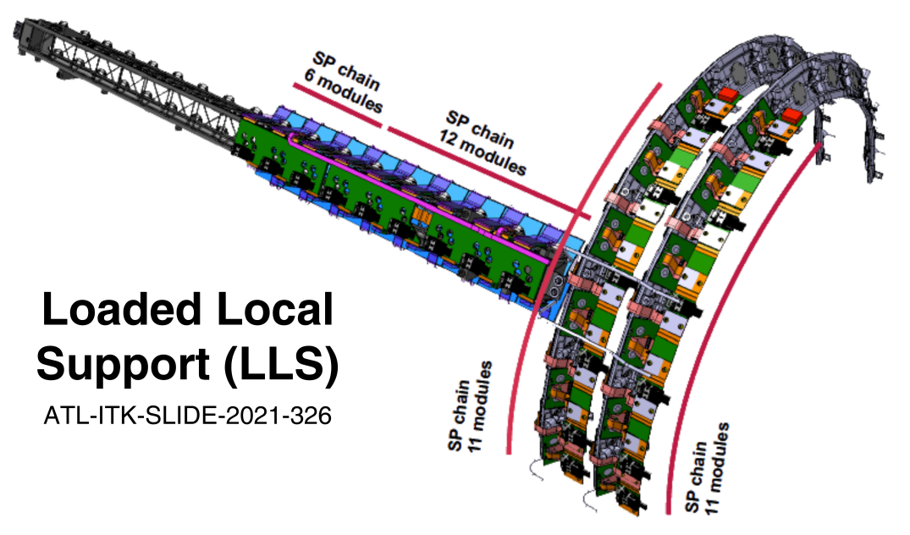
Architectural style that structures a system as a collection of microservices:

- Focus on one specific function (single-responsiblity principle)
- Independently deployable and scalable
- Loosely coupled, strongly cohesive
- Communciate via HTTP in the frontend, utilize message queueing (via AMQP)
- Developed by a small team, facilitating contributions from multiple groups in the collaboration

SR1 LLS Microservice Deployment



Overview of microservice deployment for qualification tests of detector elements at CERN. Microservices and additional tools are deployed using docker containers, which are orchestrated using compose files and easily managed via third-party and custom dashboard UIs. The ConfigDB is a central part of the system, storing connectivity, hardware configuration and more.



Loaded Local Support (LLS)

ATL-ITK-SLIDE-2021-326

List of Microservices

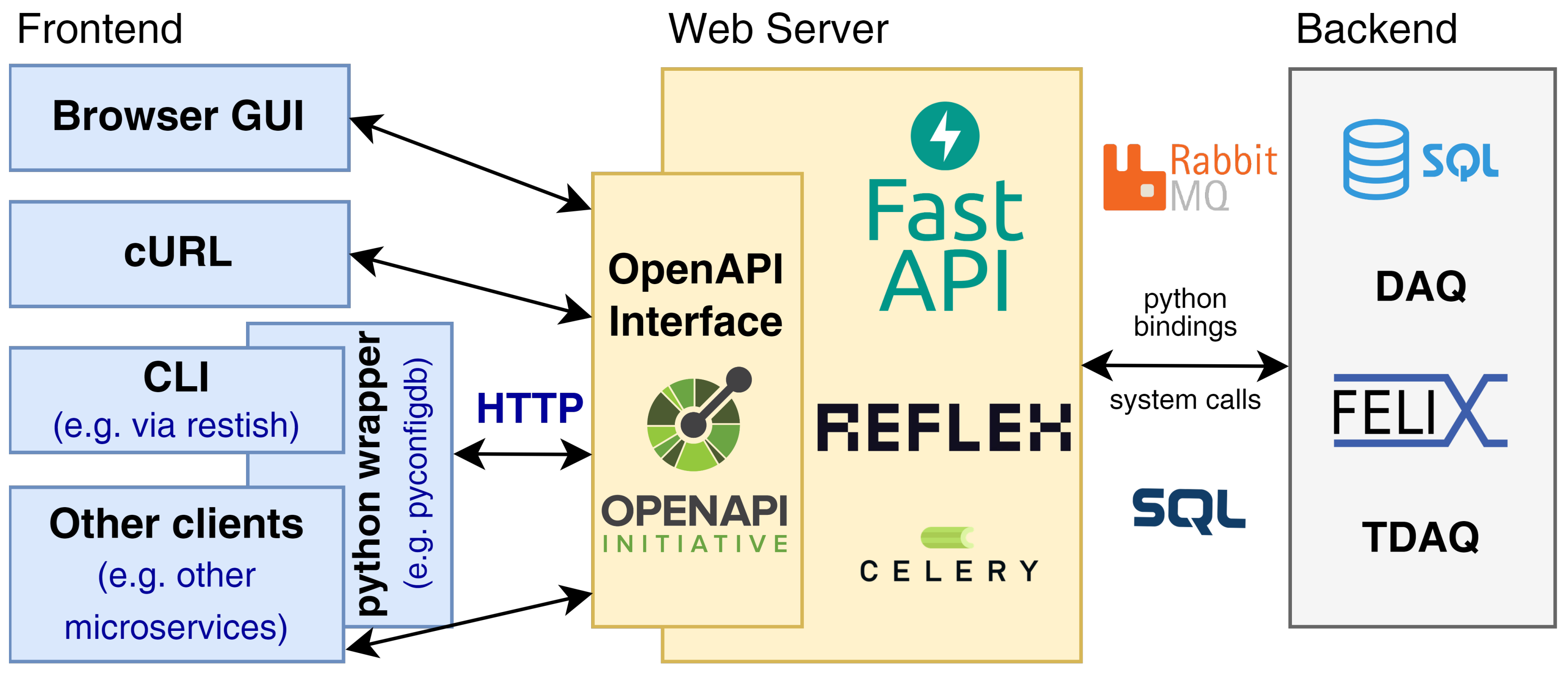
Dashboard Runkey-Manager	Overview and top-level control of system Creation and deployment of runkeys
Felix Optoboard DAQ-API Analysis-Manager	Operation of FELIX (Front-End Link eXchange)[a] Operation of the optical readout components[b] Interface to DAQ to run scans Run analyses on scan results
ConfigDB Service-Registry PDB-API SSO-API	Interface to local db storing runkeys and other data Stores information on all running microservices Interface to global ATLAS production database[c] Access management and logging

[a] EPJ Web Conf. 251 (2021) 04006

[b] J. Phys.: Conf. Ser. 2374 (2022) 012105

[c] ICHEP 42 (2024) 47

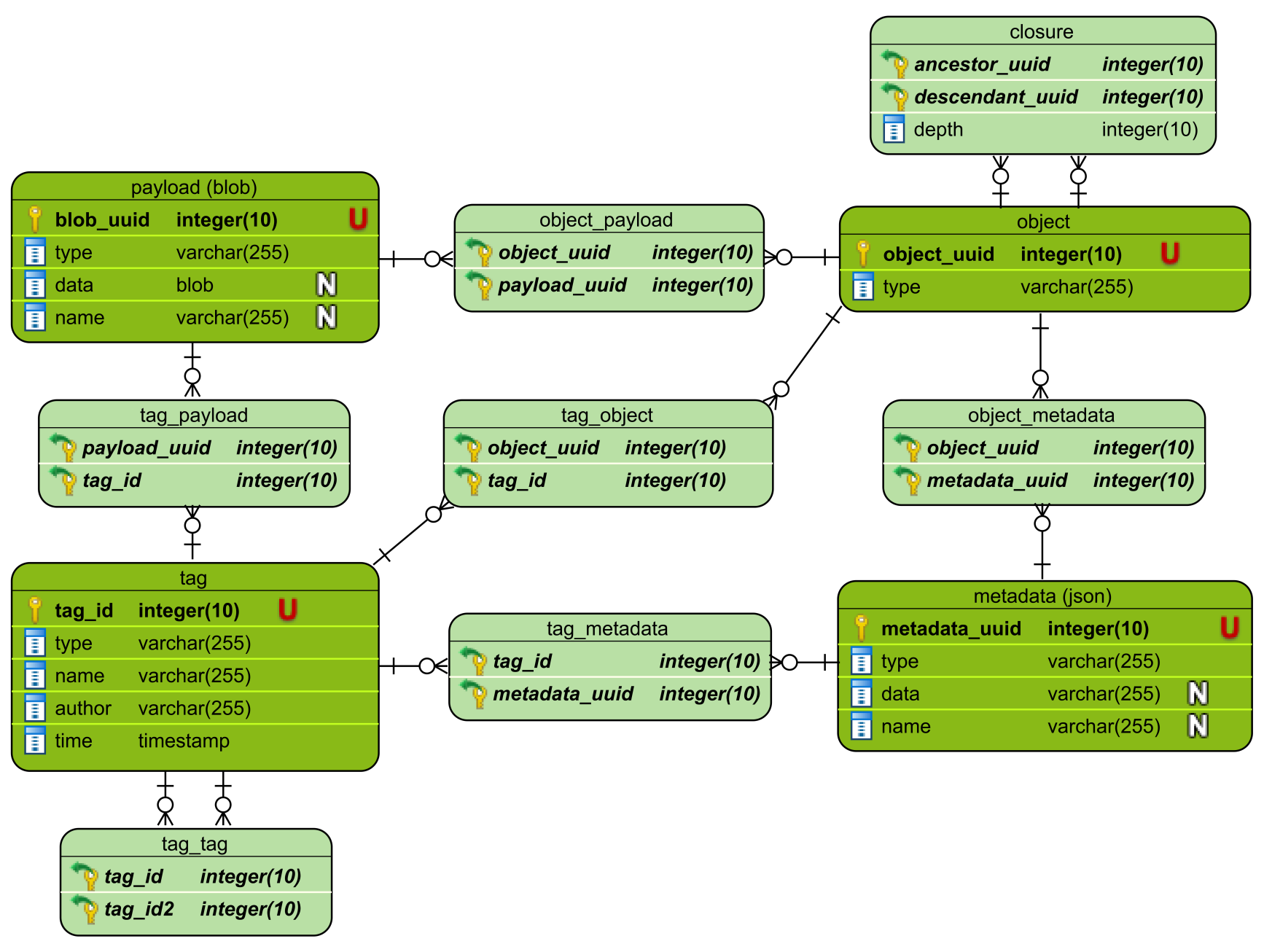
Microservice Architectural Overview



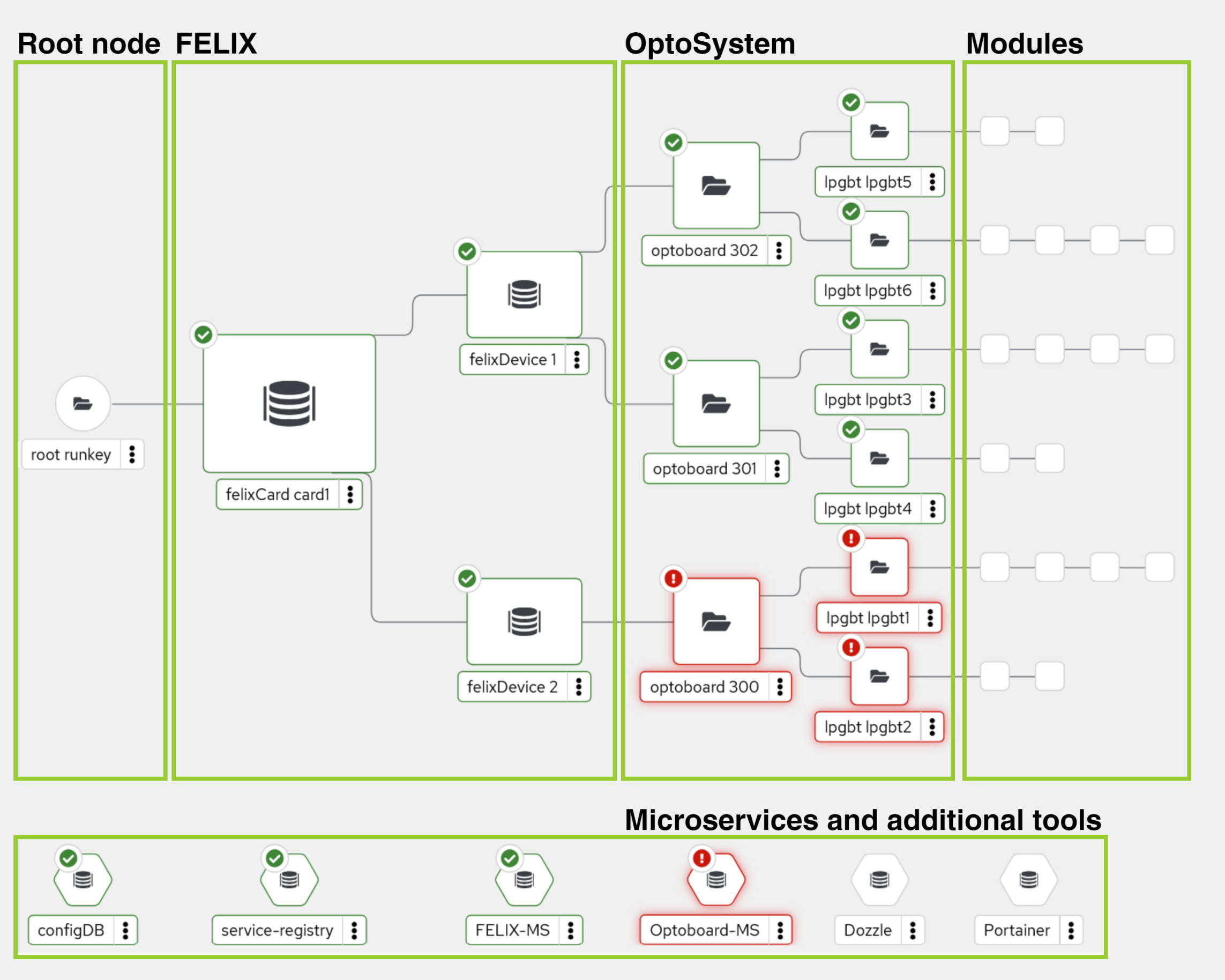
ConfigDB Database Model

The ConfigDB database model consists of four main tables that allow saving tree-like data structures:

- Object: Hardware (e.g. felix) and more (like scans/analysis)
- Payload: Config data for objects (compressed blob)
- Metadata: metadata for objects (queryable json)
- Tag: Groups trees, payloads and/or tags to make them easily accessible
- Associative tables for many-to-many relations



Dashboard UI - Runkey Topology



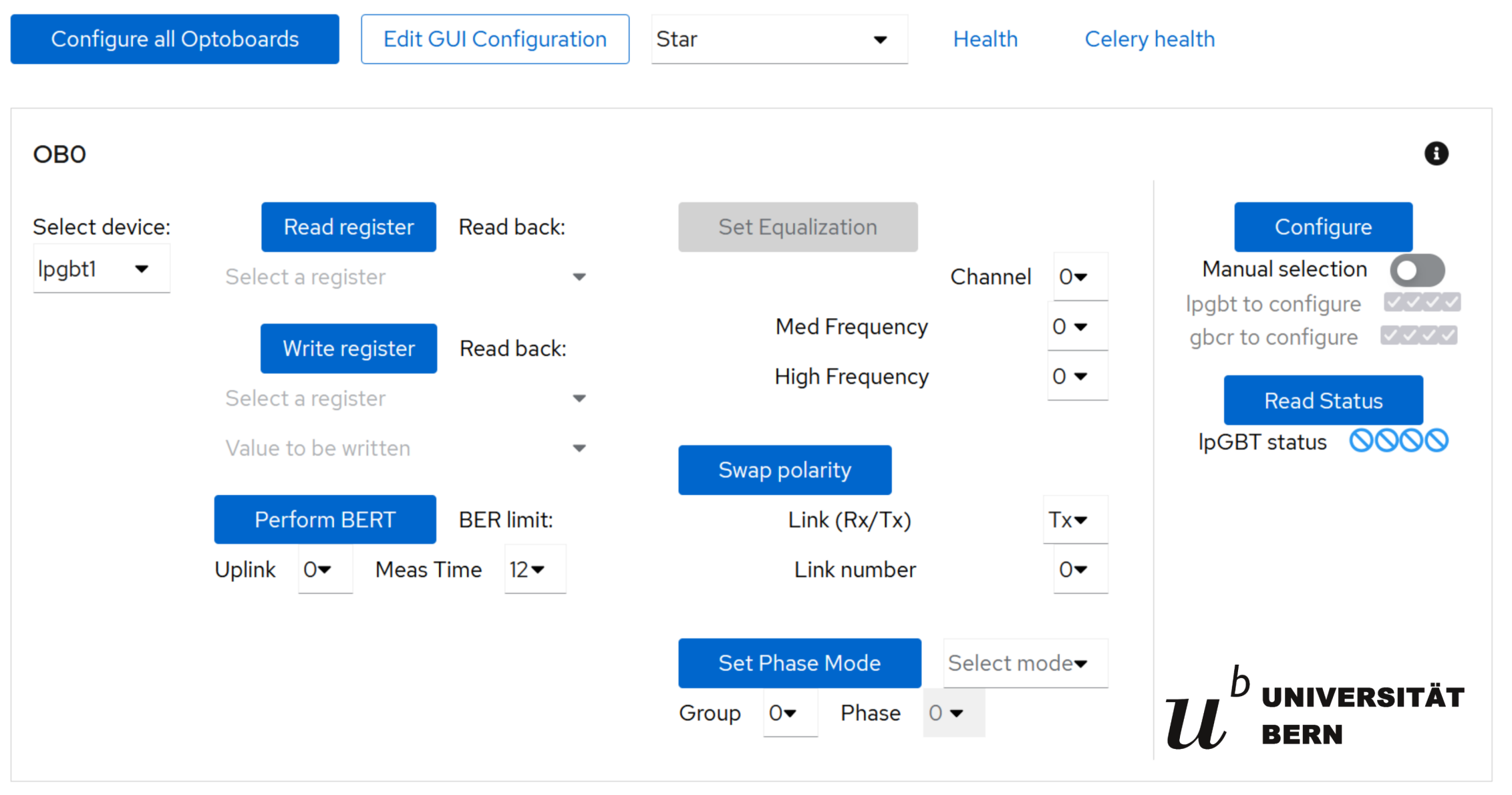
The dashboard UI shows a graphical representation of a runkey read from the ConfigDB. A runkey consists of two parts:

- Connectivity: Connected objects spanning a tree
- Configuration: Payload datasets of the objects

The status of runkey-nodes and the info on running MS are polled from the service-registry.

Optoboard UI

GUI for operation of optical readout components



Star

OBO

Select device: lpgbt1

Read back:

Select a register

Read back:

Select a register

Value to be written

BER limit:

Uplink 0 Meas Time 12

Channel 0

Med Frequency 0

High Frequency 0

Link (Rx/Tx) Tx

Link number 0

Select mode

Group 0 Phase 0

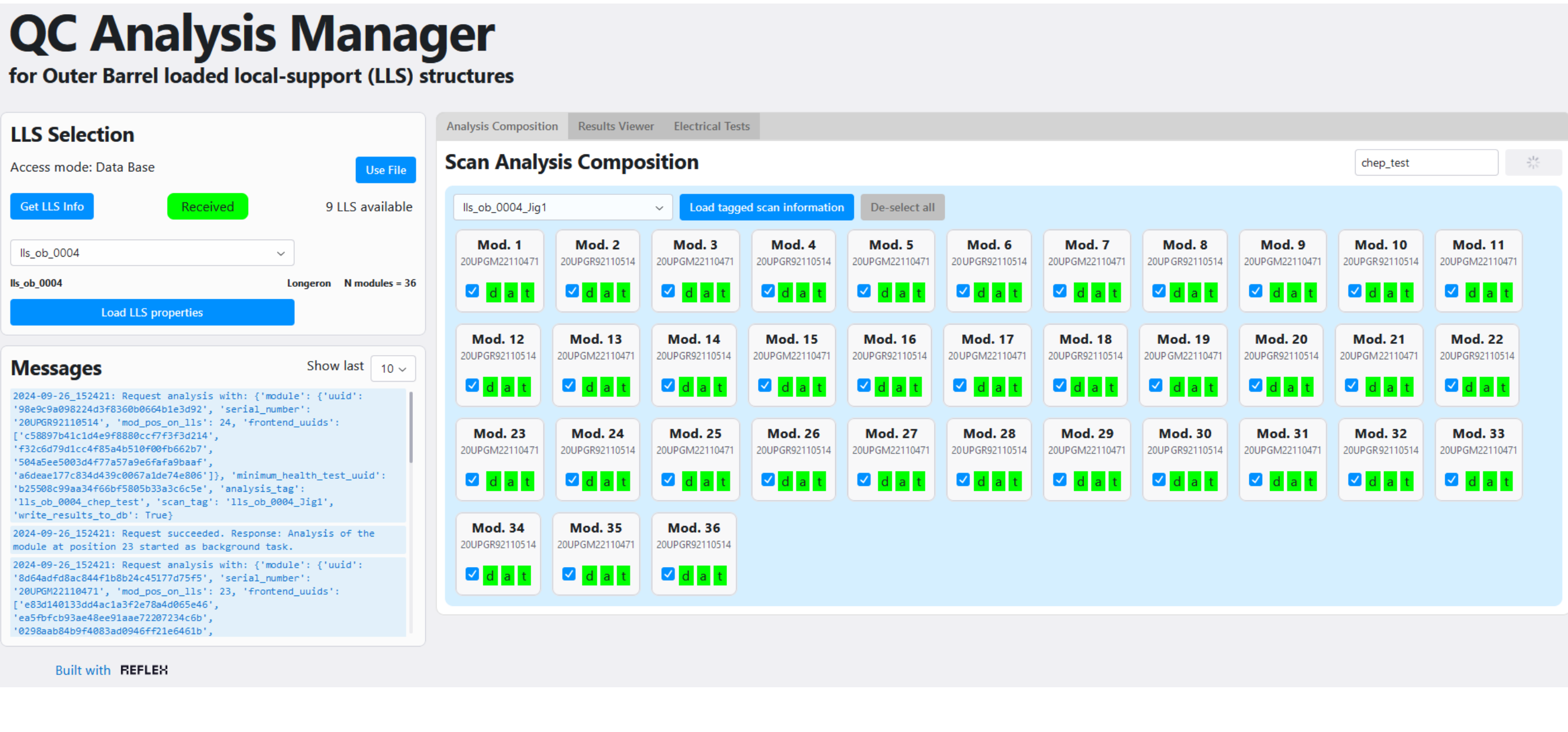
Manual selection

lpgbt to configure ☑☑☑☑

gbcr to configure ☑☑☑☑

lpGBT status

Analysis Manager



QC Analysis Manager

for Outer Barrel loaded local-support (LLS) structures

LLS Selection

Access mode: Data Base

9 LLS available

lls_ob_0004

lls_ob_0004 Longeron N modules = 36

Messages Show last 10

```
2024-09-26_152421: Request analysis with: {'module': {'uuid':
'98e9c9a098224d3f8360b0664b1e3d92', 'serial_number':
'20UPGR92110514', 'mod_pos_on_lls': 24, 'frontend_uuids':
['c58897b41c1d4e9f8880ccf7f3f3d214',
'f32c6d79d1cc4f85a4b510f00fb662b7',
'504a5ee5003d4f77a57a9e6fafa9baaf',
'a6deae177c834d439c0067a1de74e806']}, 'minimum_health_test_uuid':
'b25508c99aa34f66bf5805b33a3c6c5e', 'analysis_tag':
'lls_ob_0004_chep_test', 'scan_tag': 'lls_ob_0004_Jig1',
'write_results_to_db': True}
2024-09-26_152421: Request succeeded. Response: Analysis of the
module at position 23 started as background task.
2024-09-26_152421: Request analysis with: {'module': {'uuid':
'8d64adfd8ac844f1b8b24c45177d75f5', 'serial_number':
'20UPGM22110471', 'mod_pos_on_lls': 23, 'frontend_uuids':
['e83d140133dd4ac1a3f2e78a4d065e46',
'ea5fbfcb93ae48ee91aae72207234c6b',
'0298aab84b9f4083ad0946ff21e6461b',
```

Scan Analysis Composition

chep_test

lls_ob_0004_Jig1

Mod. 1 20UPGM22110471 ☑ d a t
Mod. 2 20UPGR92110514 ☑ d a t
Mod. 3 20UPGM22110471 ☑ d a t
Mod. 4 20UPGR92110514 ☑ d a t
Mod. 5 20UPGM22110471 ☑ d a t
Mod. 6 20UPGR92110514 ☑ d a t
Mod. 7 20UPGM22110471 ☑ d a t
Mod. 8 20UPGR92110514 ☑ d a t
Mod. 9 20UPGM22110471 ☑ d a t
Mod. 10 20UPGR92110514 ☑ d a t
Mod. 11 20UPGM22110471 ☑ d a t
Mod. 12 20UPGR92110514 ☑ d a t
Mod. 13 20UPGM22110471 ☑ d a t
Mod. 14 20UPGR92110514 ☑ d a t
Mod. 15 20UPGM22110471 ☑ d a t
Mod. 16 20UPGR92110514 ☑ d a t
Mod. 17 20UPGM22110471 ☑ d a t
Mod. 18 20UPGR92110514 ☑ d a t
Mod. 19 20UPGM22110471 ☑ d a t
Mod. 20 20UPGR92110514 ☑ d a t
Mod. 21 20UPGM22110471 ☑ d a t
Mod. 22 20UPGR92110514 ☑ d a t
Mod. 23 20UPGM22110471 ☑ d a t
Mod. 24 20UPGR92110514 ☑ d a t
Mod. 25 20UPGM22110471 ☑ d a t
Mod. 26 20UPGR92110514 ☑ d a t
Mod. 27 20UPGM22110471 ☑ d a t
Mod. 28 20UPGR92110514 ☑ d a t
Mod. 29 20UPGM22110471 ☑ d a t
Mod. 30 20UPGR92110514 ☑ d a t
Mod. 31 20UPGM22110471 ☑ d a t
Mod. 32 20UPGR92110514 ☑ d a t
Mod. 33 20UPGM22110471 ☑ d a t
Mod. 34 20UPGR92110514 ☑ d a t
Mod. 35 20UPGM22110471 ☑ d a t
Mod. 36 20UPGR92110514 ☑ d a t

GUI to analyse scan results in LLS QC:

- Reads LLS structure from production DB
- Loads selected LLS from configDB
- Shows available scans per module
- Sends selected modules to analysis container
- Container reads scan results from configDB
- Writes analysis results to DB
- Analysis results can be viewed in Result Panel
- Chosen results are saved in production DB

docs