

TGenBase - general purpose virtual database engine for HEP

Evgeny Lavrik¹, for CBM Collaboration

¹ Facility for Antiproton and Ion Research, Darmstadt, Germany

Compressed Baryonic Matter experiment at FAIR

What is TGenBase?

- Virtual DB engine
- Made for scientists, not programmers
- Rich data visualization in Web
- Language bindings
- Templated code generation

Key Aspects

- Visual data description
- Allows storage of user defined data, ROOT objects, binary data
- Special logics in addition to CRUD (create, read, update, delete)
 - Versioned insert-only → nothing is lost
 - Audited → keeps track of user activity

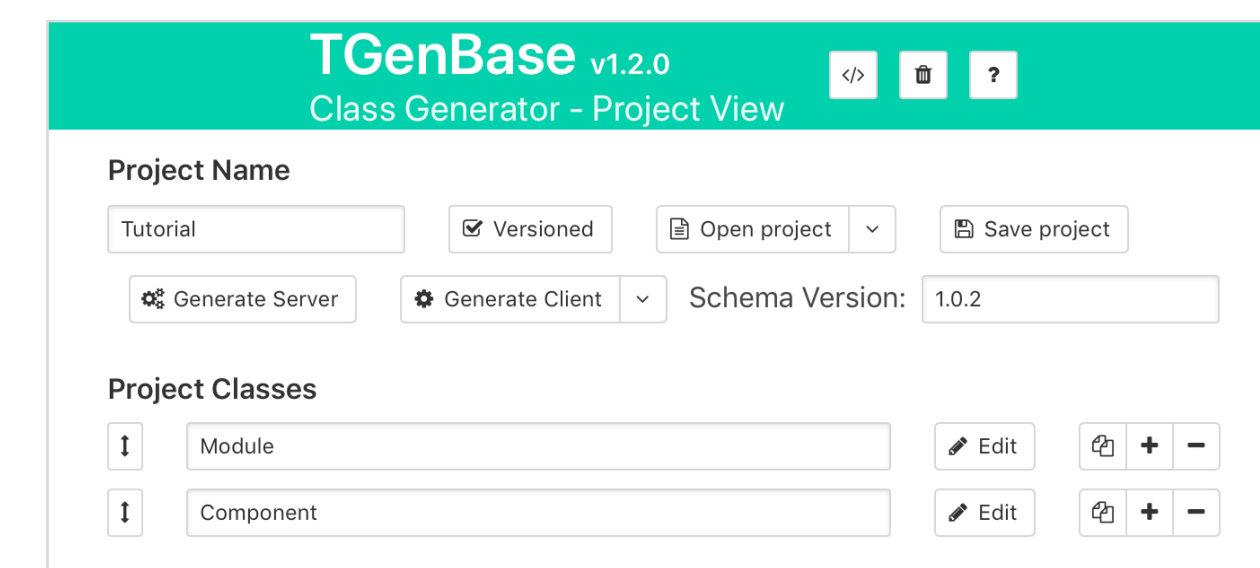
Visual data description

- Describe data classes in Web UI
- Establish relations between classes (one-to-one, one-to-many, many-to-many)
- Many built in class property types, including arrays, collections, (ROOT-)objects using serialization
- Properties can have custom default values, validation rules, semantic subtypes (URI, LaTeX, DOI)
- Dictionary properties allow for easy Id-Name lookup
- Static data description for initial DB seeding
- Saving and loading of projects, fetching from git

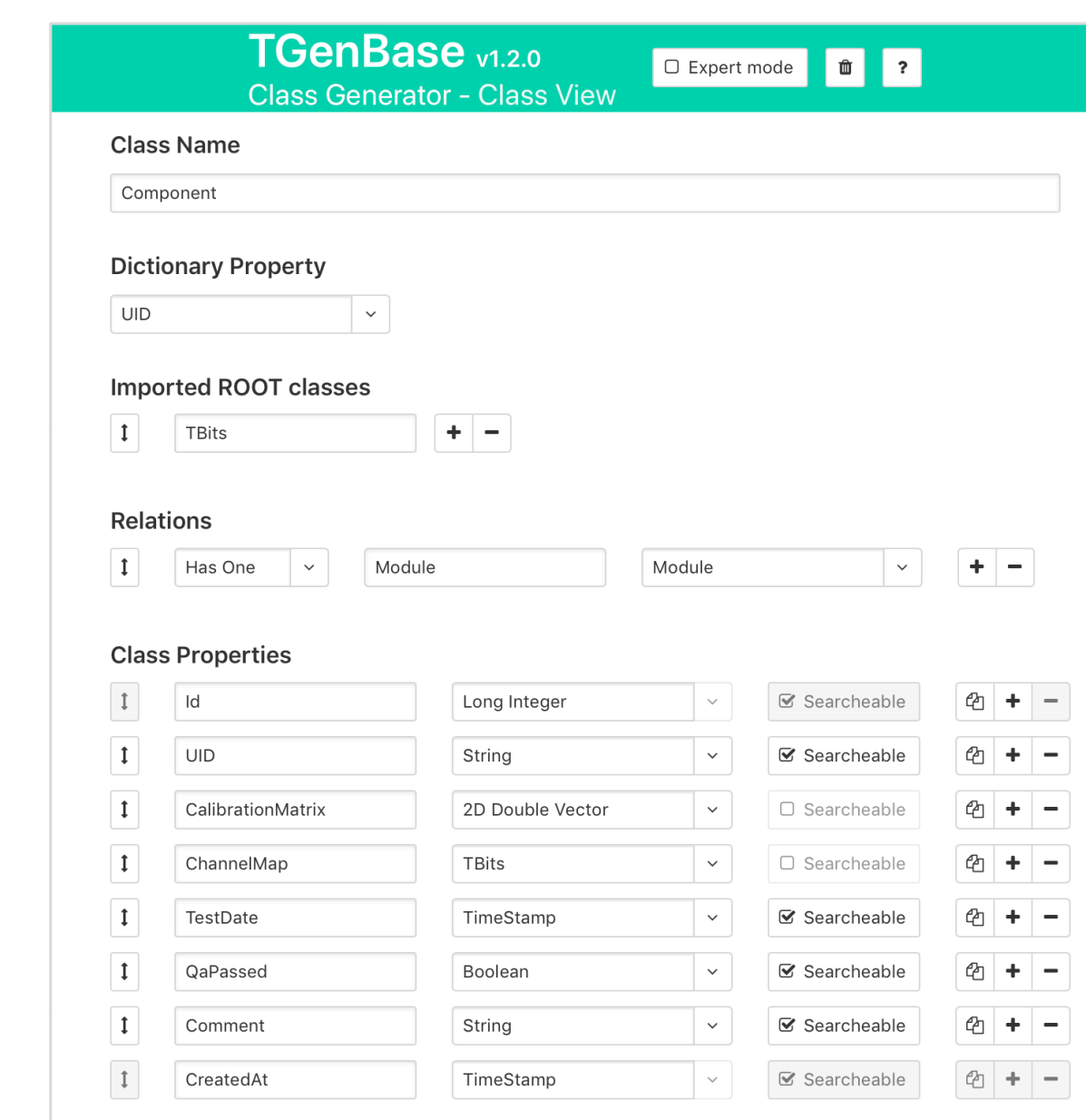
☞ Same description used to generate the database layout, server to host data, and clients to fetch data objects (C++, Python, LabVIEW)

☞ Rich data visualization in a full fledged CMS

Project description and management interface



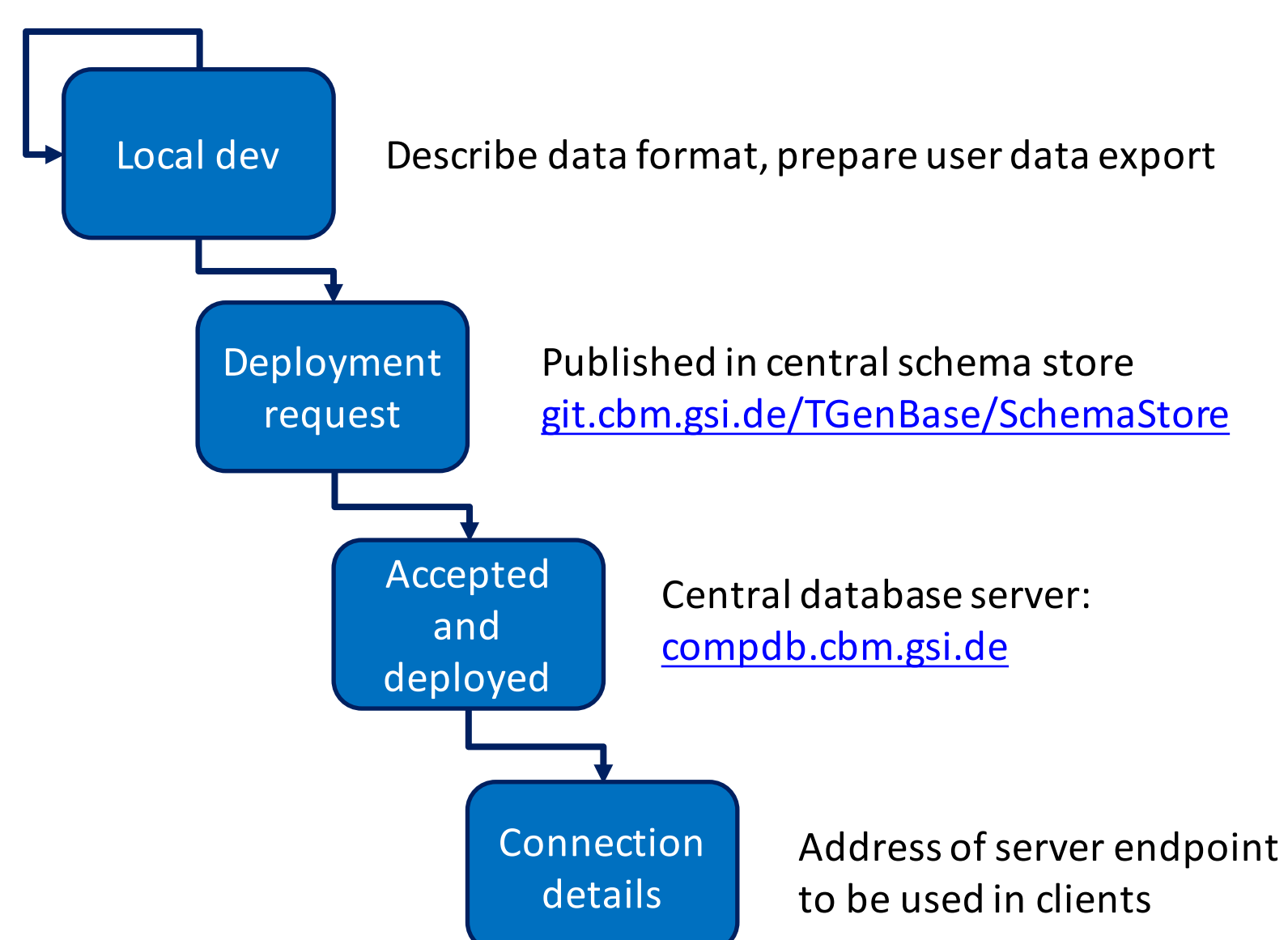
Class and class properties configuration



Data description is a Schema

- Defines the contract between the client and server
- Client and server exchange and check schema versions
- Schema evolution based on migrations

Schema development workflow

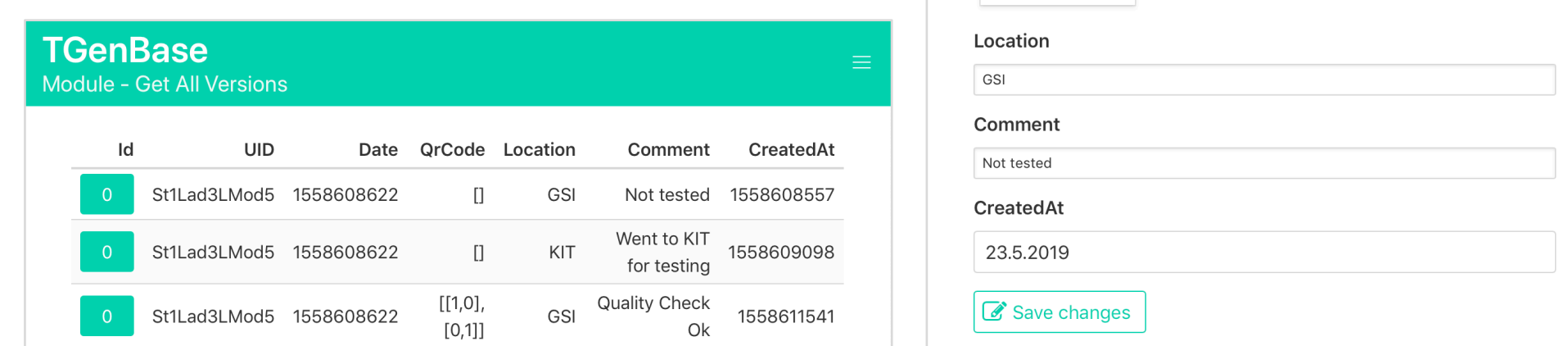


Data visualization

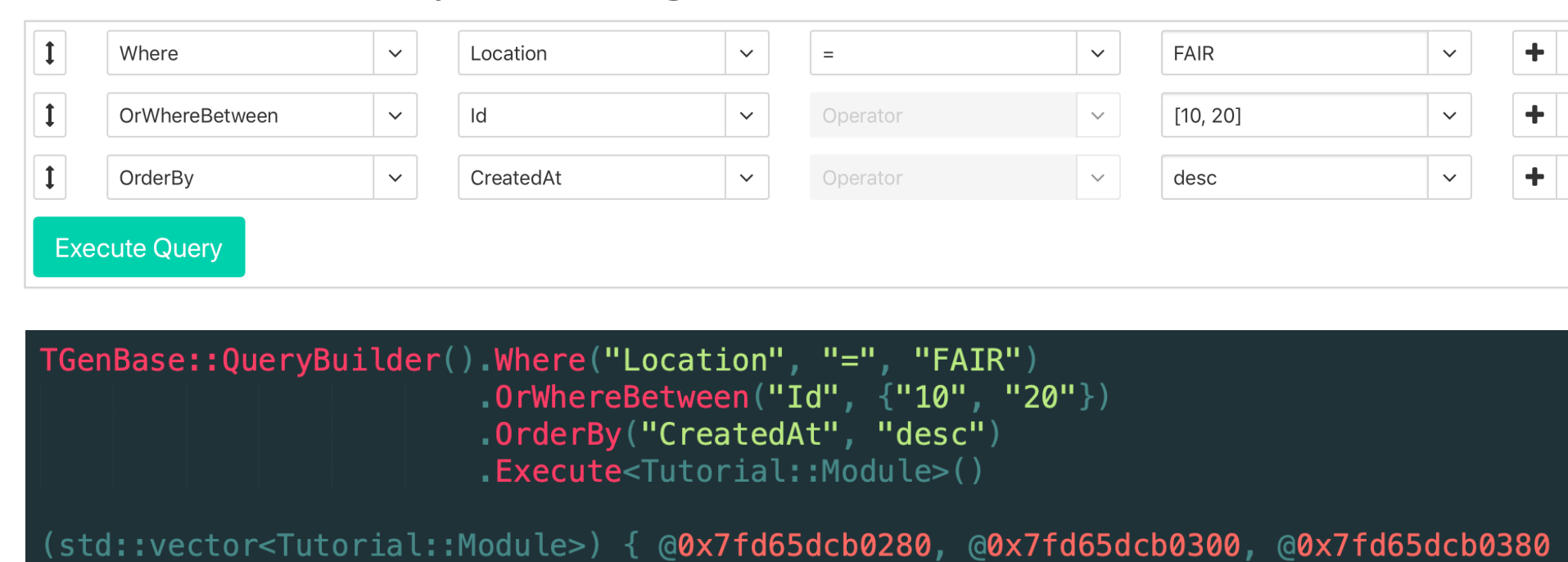
- A Content Management System (CMS) is generated from schema
- Allows for navigation between different classes and their relations
- Provides search functionality by single fields or using a dynamic query builder interface
- Data display customization
- Data visualization: ROOT objects (JSROOT), images, PDFs, spreadsheets
- Operator's workspace to create new entries and edit existing, upload files, etc.
- Admin panel to customize server behavior, manage users, etc.

Data visualization UI in the CMS generated from schema

Left: version history (logistics)
Right: view of a single object properties



Query building interfaces in UI and C++



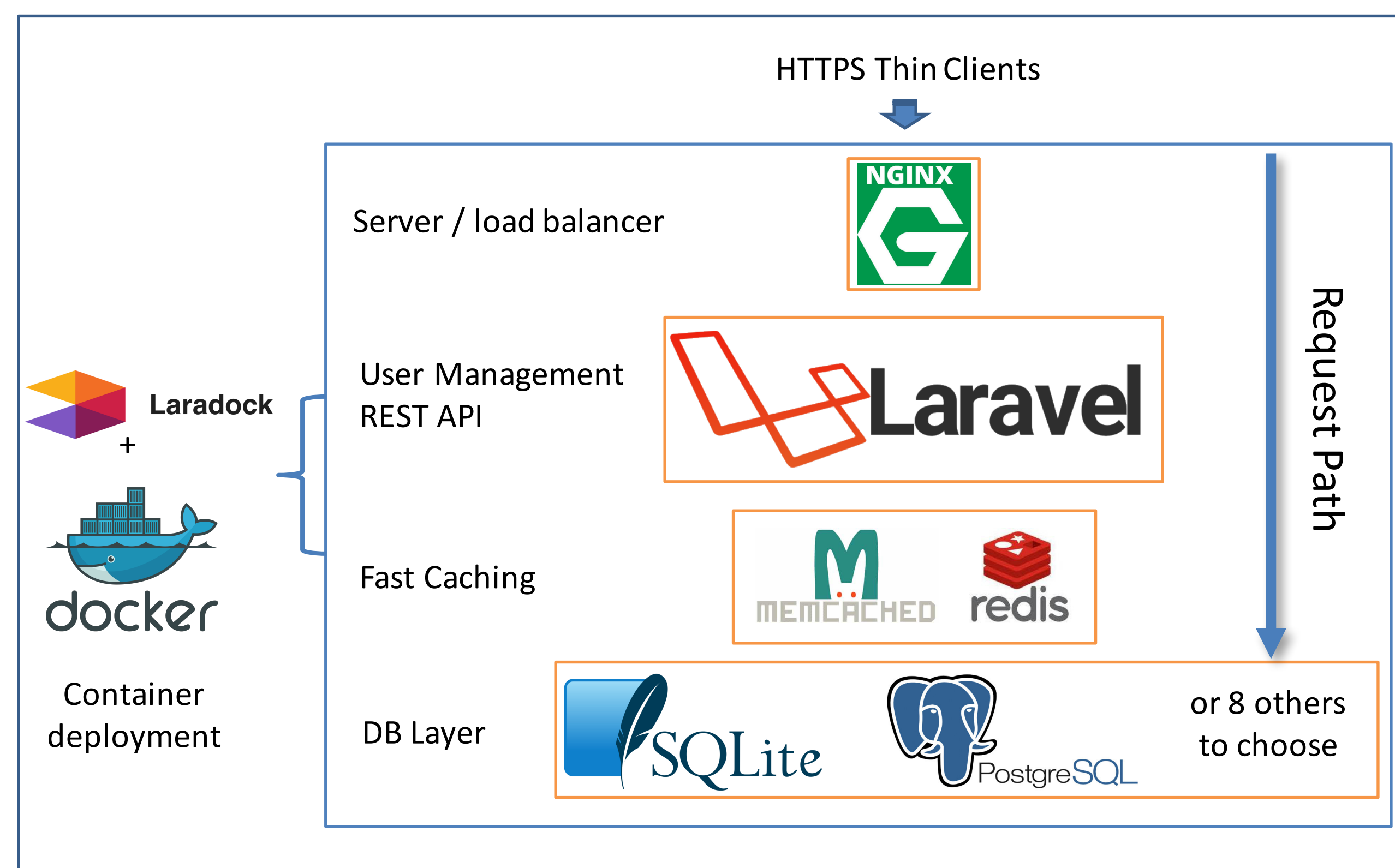
Server

- Production server runs inside docker container configured with docker-compose using Laradock as a recipe book
- Provides standard RESTful API to access data by JSON message exchange with clients using token authentication
- Provides configurable role-based user permission management, user import from LDAP directories
- Caches data for frequent access in key-value storages
- Communicates with underlying DB Management Systems
- Ongoing work towards push-to-deploy

Uses in CBM experiment at future facility FAIR are

- Detector component, logistics, QA data storage
- Parameter, conditions, configuration management

Production DB server software stack



Currently 6 different instances are hosted for CBM needs



Check out the data description UI <https://tgenbase.com>
Source code <https://tgenbase.com/#/git>
Tutorial example online <https://tgenbase.com:5050>

