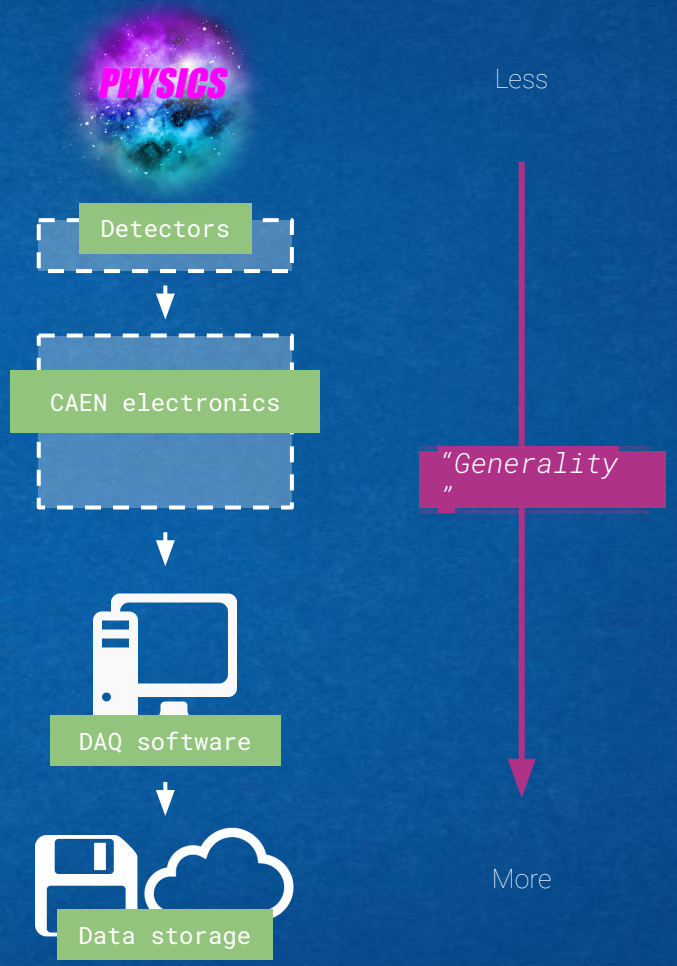
INTENSE *Monthly Meeting - Nov/2022*

Development of a data acquisition platform based on CAEN digital electronics

Matías ***Simonetto***

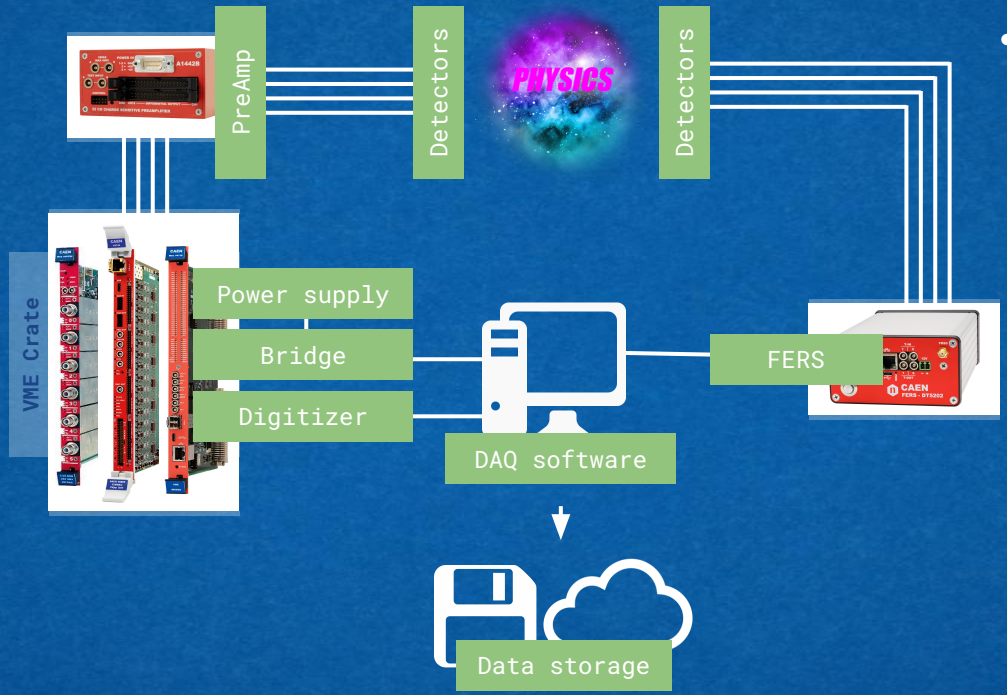
Data acquisition platform

- From detectors to data storage.
- CAEN electronics
  - Power supply.
  - Signal conditioning.
  - Digitalization.
  - Communication.
- DAQ software
  - Device configuration and control.
  - Data readout and storage (eventually in a cloud database).
  - Integrated, versatile, high performance and easy-to-use.



< Digitizer-based | ASIC-based >

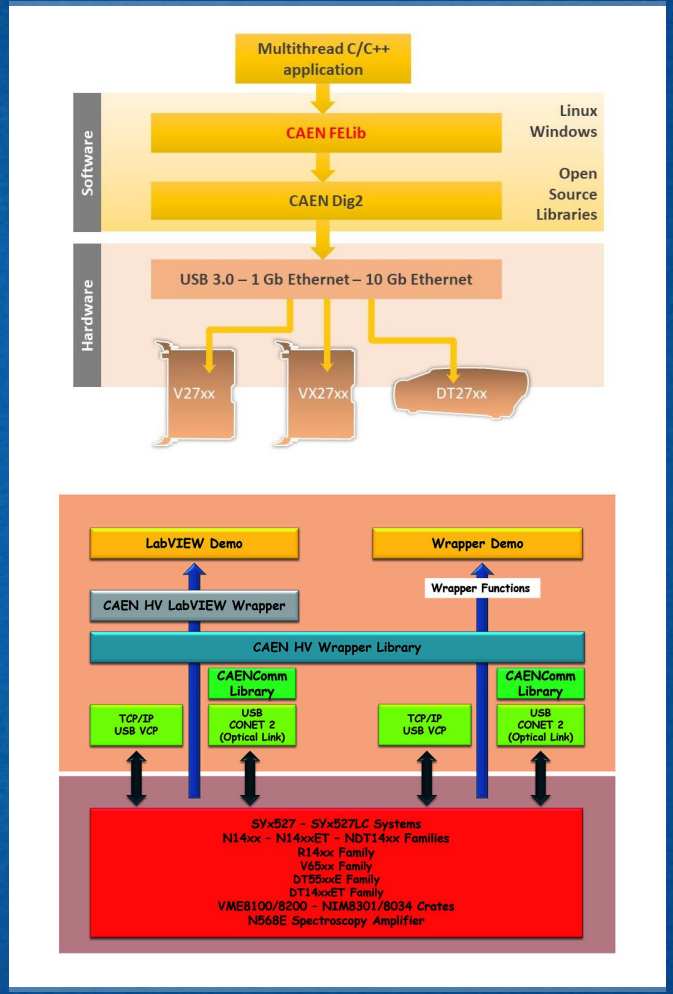
- *Signal conditioning*
  CAEN A1442
  16/32 Channel charge sensitive preamplifier.
- *Power supply*
  CAEN V6519
  6 Channel 500 V/3 mA VME
- *Communication*
  CAEN V4718
  VME to USB 3.0/Ethernet/Optical Link Bridge
- *Digitizer*
  CAEN V2740
  64 Channel 16 bit 125 MS/s.



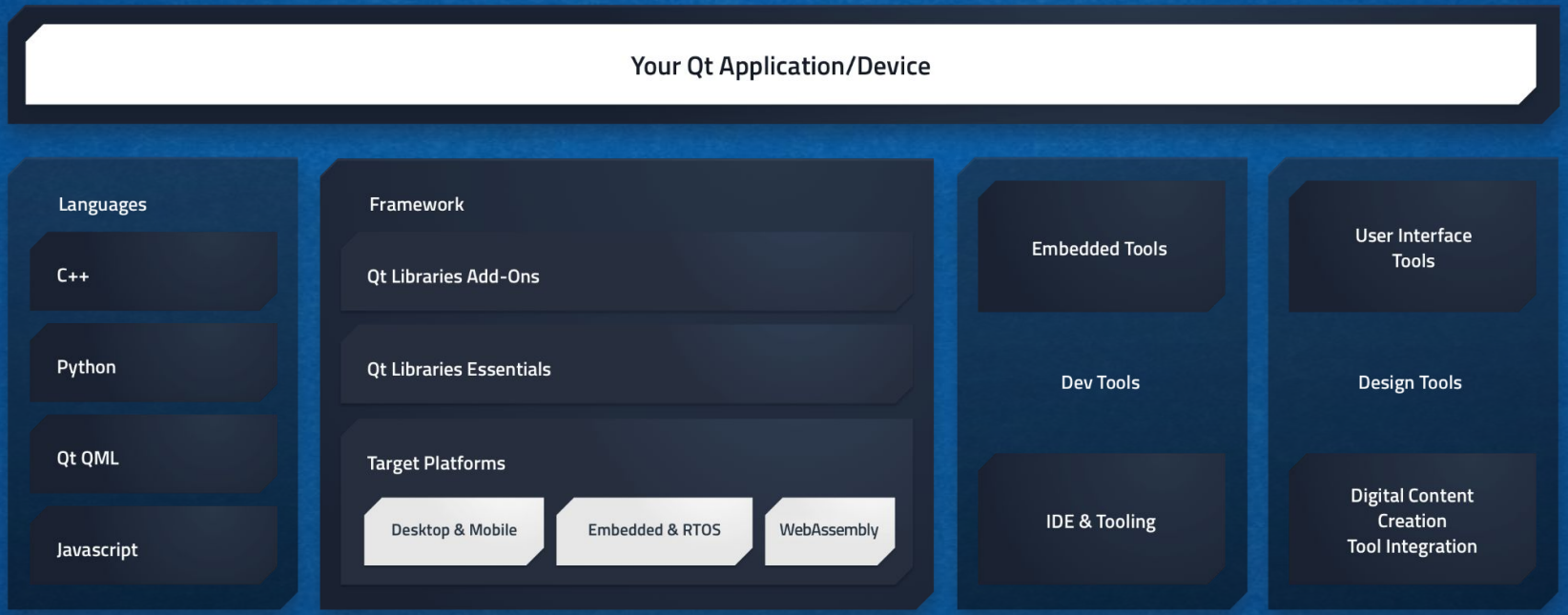
- **CAEN FERS 5202**: *Front-End Readout System*
  - Citiroc 1A 32-channel front-end ASIC (x2), working in conjunction with a ADC.
  - Onboard power supply: CAEN A7585D +85 V/10 mA.
  - Several communication interfaces: USB, Ethernet and TDlink.

DAQ software

- Current CAEN GUI softwares
  - Geco, Compass, WaveDump, Janus.
  - Communication (device control and data readout) in a simple and complete way with the *different* components of an acquisition system.
- CAEN intermediate level libraries
  - FELib library, HV Wrapper Library, FERSLib.
  - Easy development of application softwares.



Qt Framework: overview



Qt Framework: small example

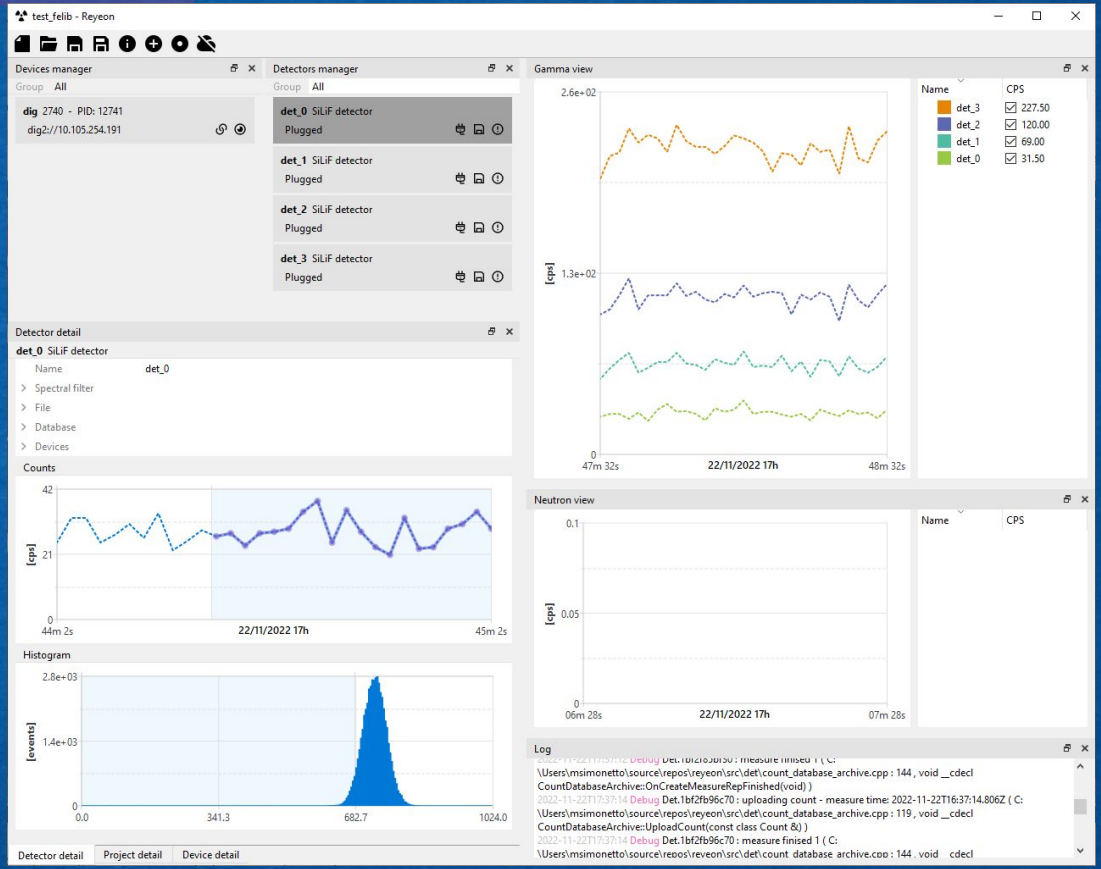
```cpp
#ifndef INCLUDE_DEV
#define INCLUDE_DEV

#include <QObject>

#include "utils.hpp"

class Dev : public QObject
{
    Q_OBJECT
public:
    void longOperation()
    {
        CUR_QTHREAD_OUT << "processing...";
        emit processing();

        // Slow operation...
        QThread::currentThread()->sleep(2);

        QString a = "42";
        CUR_QTHREAD_OUT << "finished... answer is " << a;
        emit finished(a);
    }

signals:
    void processing();
    void finished(const QString& a);
};

#endif
```

```cpp
#include <QCoreApplication>
#include <QThread>

#include "dev_qt.hpp"
#include "utils.hpp"

int main(int argc, char* argv[])
{
    QCoreApplication ap(argc, argv);

    QThread t;
    Dev d;
    t.start();
    d.moveToThread(&t);

    QMetaObject::invokeMethod(&d, &Dev::longOperation);

    QObject::connect(&d, &Dev::processing, &t,
        [] {
            CUR_QTHREAD_OUT << "ok... I'll do other things in the meantime...";
            QThread::currentThread()->sleep(1);
        });

    QObject::connect(&d, &Dev::finished, &t,
        [&ap, &t](const QString& a) {
            CUR_QTHREAD_OUT << a << "??";
            t.quit();
            t.wait();
            ap.quit();
        });

    ap.exec();
}
```

```
[ QThread(0xd265aff7f8) ]    processing...
[ QThread(0x25897889a10) ]   ok... I'll do other things in the meantime...
[ QThread(0xd265aff7f8) ]    finished... answer is  "42"
[ QThread(0x25897889a10) ]   "42" ??
```

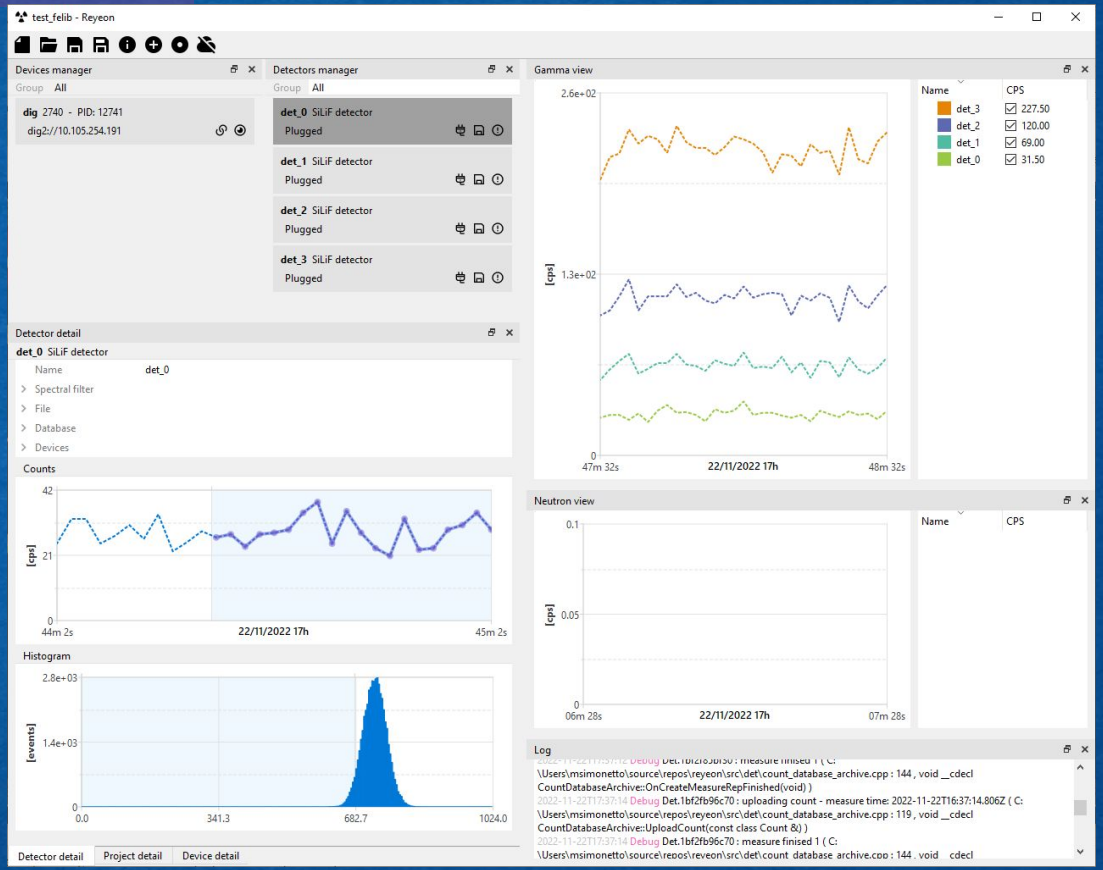
New DAQ software

- Modular design
  - Detectors and devices of different type can be easily added/removed.
- Device management
  - All devices of the platform can be configured and controlled from within the software. No need of additional programs.
  - Configurations are saved and properly reapplied on each run.
- Detector management
  - Simple and clear identification of the detectors and their relations with the devices.
  - Straightforward visualization and saving of the read data.

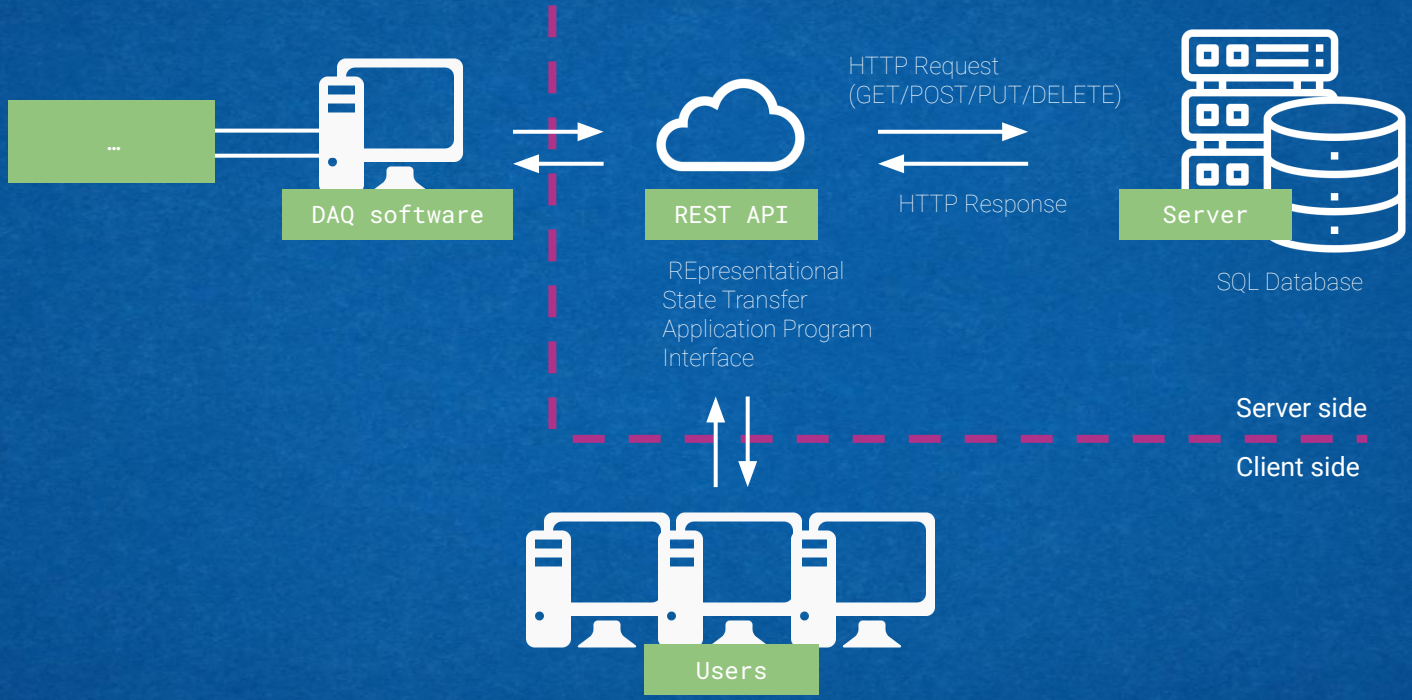


(Previous) Future work

- Bug fix
  - New DAQ software.
  - Underlying libraries. (Including Qt)
- Finish implementations
  - Cloud database communication.
  - Alarms.
- Improve user experience.



Database communication



Database communication

- Client side (DAQ Software):
  - **HTTP client** capabilities: implemented using **Qt Network** module.
  - Added support for authentication.
- Server side:
  - Starting point: **RadBASE**.
    - Java (Spring framework).
    - REST API endpoints (including authentication).
    - Web application.
  - Added features necessary for the platform.