Integrating Track Lab with Constellation for distributed DAQ of Timepix3 and Timepix4 detector networks

Petr Mánek[1,2], Benedikt Bergmann, Stephan Lachnit, Simon Spannagel, Peter Švihra

[1] Institute of Experimental and Applied Physics, Czech Technical University, Prague

[2] Department of Physics and Astronomy, University College London

On the menu today…

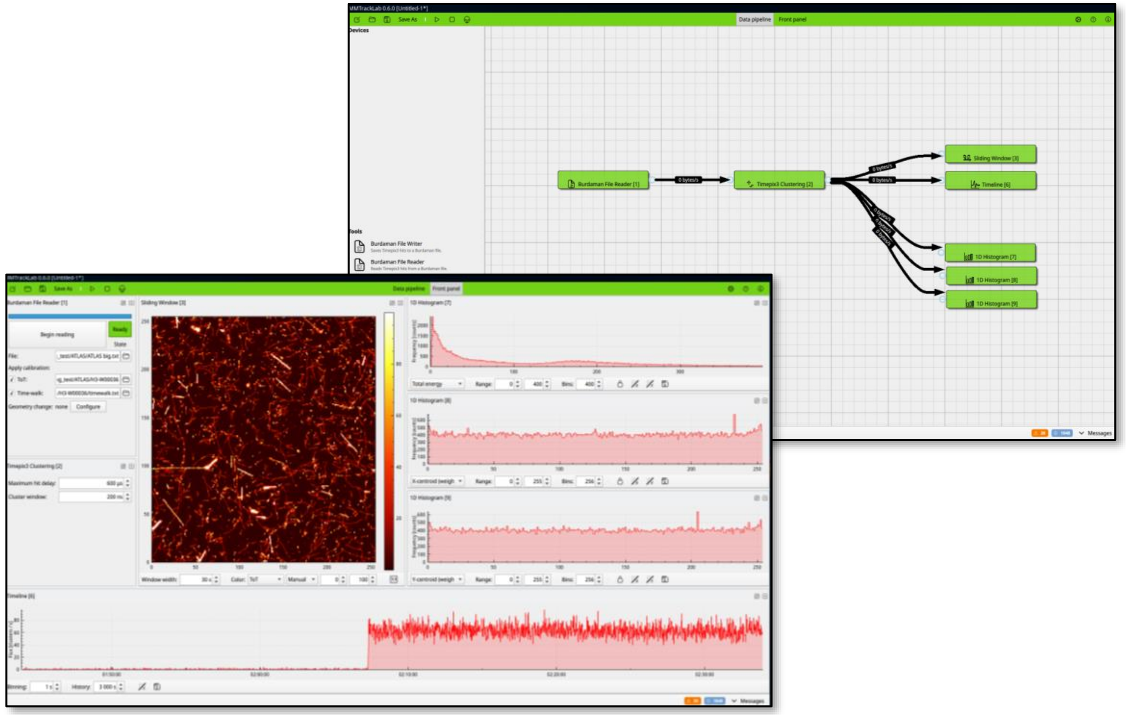
- Recap of the Track Lab concept
- Current hardware support
- Proposal to integrate with Constellation:
  - Multi-threaded vs. multi-process architecture
  - Distributing computational load
  - Physically detached experiments
  - Interoperability between languages and platforms
  - Finite-state machines
- Summary, request for comments

Previously featured at DRD3:

- 06/2024: High-performance software package for Timepix3 data acquisition, online analysis and automation

Recap of the Track Lab concept

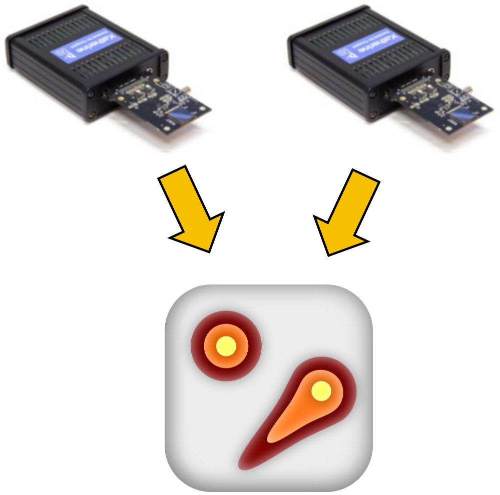
- Data acquisition (DAQ) and analysis software for pixel detectors
- Analysis composed of single-purpose building blocks



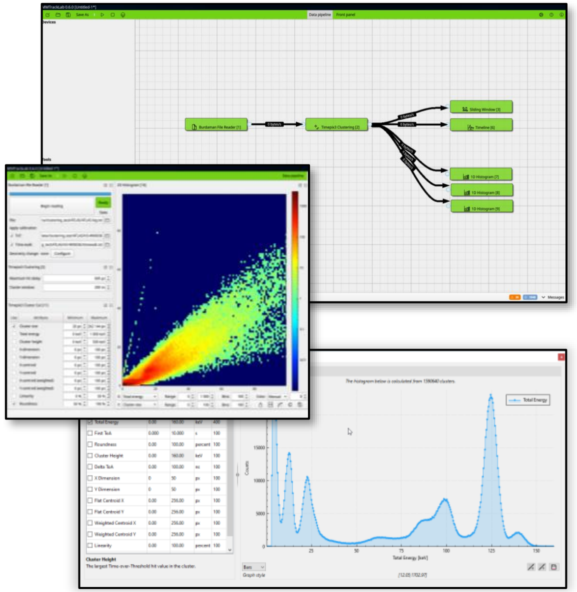
Aiming for:

- High performance: scaling
- Versatility: arbitrary topologies
- Extendibility: plug-in system

What can Track Lab do?



Take data from detectors

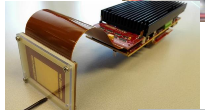


Run analysis in real-time

Automate repetitive tasks

Photo credit: University of West Bohemia, Standa LTD, Amptek Inc., Universal Robots A/S

Hardware: current support



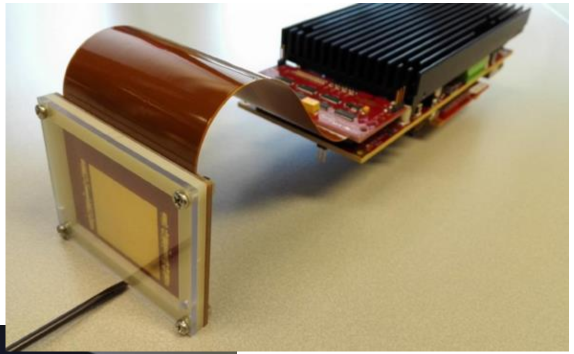
Readout	Sensor(s)	Connection
Katherine Gen1	1x Timepix3	1 Gbit Ethernet
	1x Timepix2	1 Gbit Ethernet
Katherine Gen2	8x Timepix3	1 Gbit Ethernet
		USB 3
		PCIe 3 x4
HardPix	1x Timepix3	USB 3
SPIDR4	1x Timepix4	10 Gbit Ethernet
COMBO+Spectrig	1x SiPM	USB 2
MicroDAQ	28x PMT	1 Gbit Ethernet

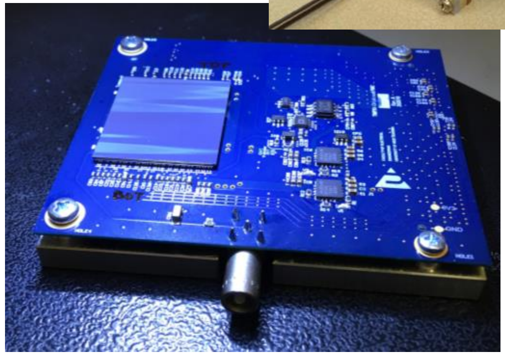
Readout	Sensor(s)	Conn.
Timepix2 Lite	1x Timepix2	USB 2
MiniPIX EDU	1x Timepix	USB 2
MiniPIX	1x Timepix	USB 2
	1x Timepix2	USB 2
	1x Timepix3	USB 2
AdvaPIX	1x Timepix	USB 3
	1x Timepix3	USB 3
WidePIX L	10x Medipix3	1 Gb. E.

Tested / in development / planned

Hardware: outlook for 2025

- Currently working on:
  - SPIDR4: API partially implemented, prototype expected in Jan 2025.
  - Katherine for Timepix4: before Nov 2024 final hardware unavailable, we plan to start software tests in Dec 2024.
- Also working on hardware plugins for:
  - Seifert high voltage generator for X-ray tubes
  - Stepper motor controllers by Phytron and PI
  - Advacam readouts: AdvaPIX, WidePIX





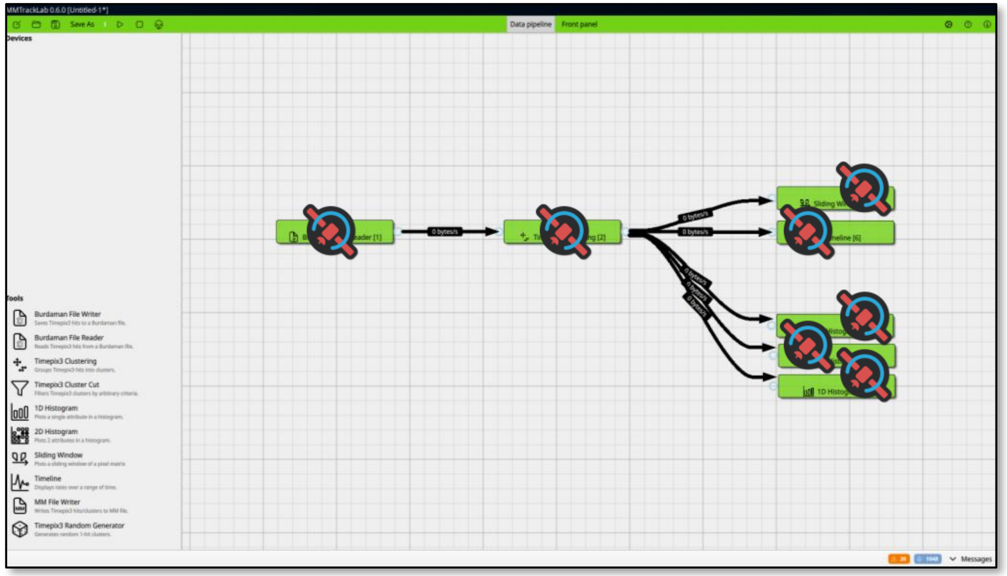
+

A proposal to integrate Track Lab with Constellation

Logo credit: DESY and the Constellation authors

How would it work?

- Satellites ↔ data processing elements in the workflow
- ZeroMQ data handling ↔ ZeroMQ data handling
- Satellite state machine ↔ Track Lab state machine
- Controller ↔ Track Lab core



Burdaman File Writer
Saves Timepix3 hits to a Burdaman file.

Burdaman File Reader
Reads Timepix3 hits from a Burdaman file.

Timepix3 Clustering
Groups Timepix3 hits into clusters.

Timepix3 Cluster Cut
Filters Timepix3 clusters by arbitrary criteria.

1D Histogram
Plots a single attribute in a histogram.

2D Histogram
Plots 2 attributes in a histogram.

Sliding Window
Plots a sliding window of a pixel matrix.

Timeline
Displays rates over a range of time.

MM File Writer
Writes Timepix3 hits/clusters to MM file.

Timepix3 Random Generator
Generates random 1-hit clusters.

Logo credit: DESY and the Constellation authors

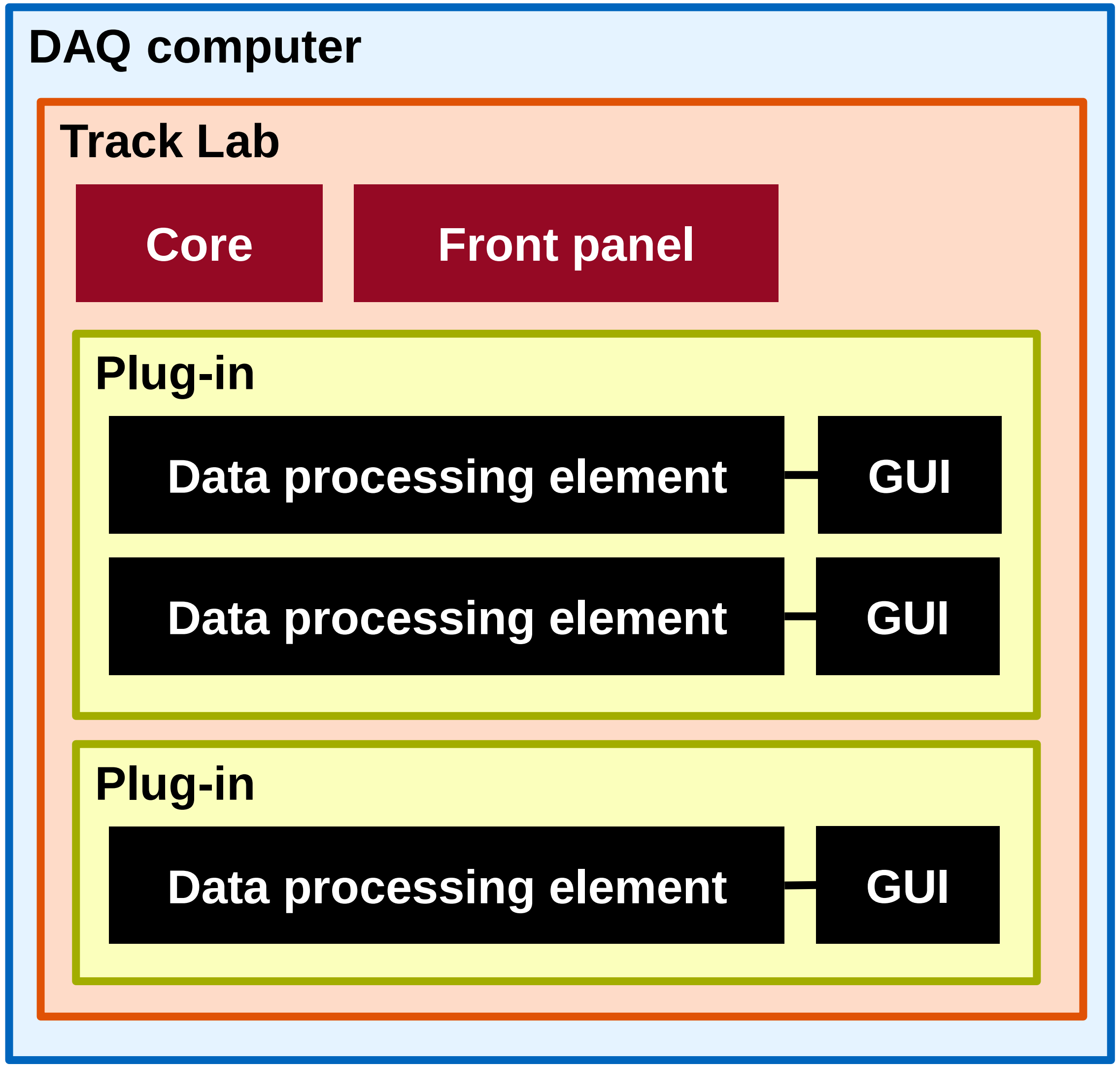
Multi-threaded vs. multi-process

- Now: all plug-ins run within the same process
  - Easy (and fast) to share memory between threads.
  - Easy to synchronize activities, start and stop.
  - Failure of a single plug-in takes down the entire program.

- Future: plug-ins instantiated in separate processes
  - Memory sharing not affected within plug-ins, otherwise a problem.
  - Still easy to synchronize thanks to the state machine.
  - Increased resiliency, but added overhead for error handling.

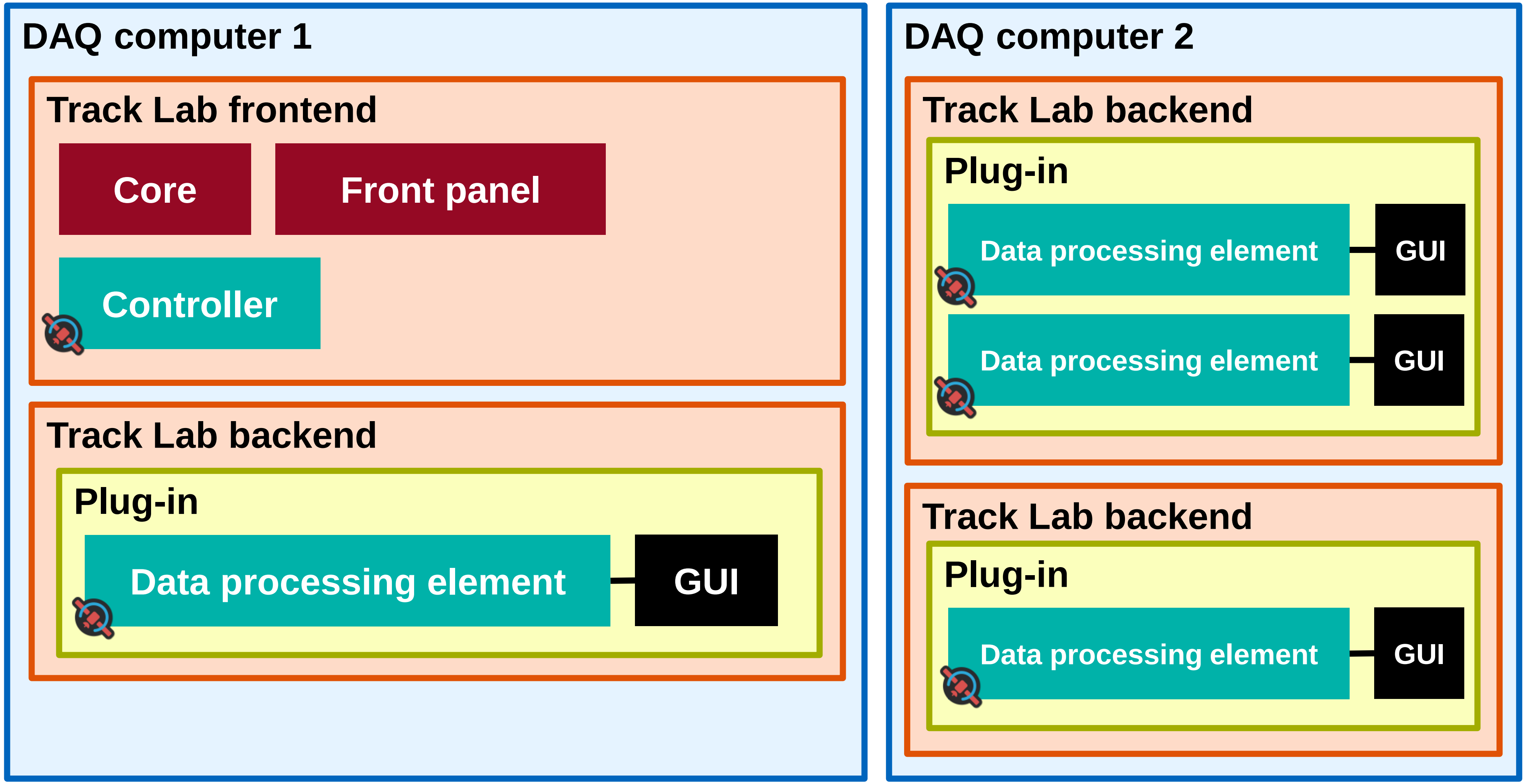
Multi-threaded vs. multi-process

- Now:



Multi-threaded vs. multi-process

- Future:



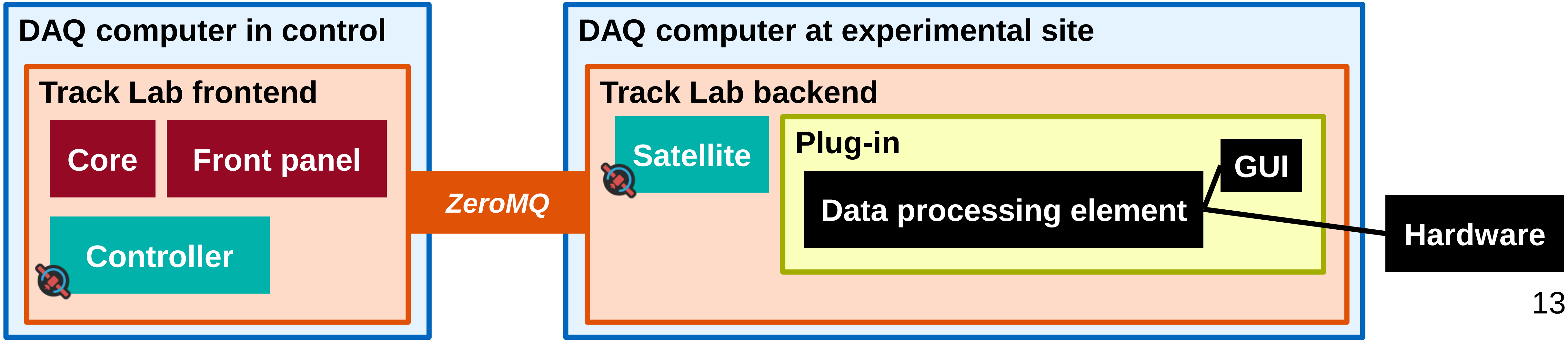
Logo credit: DESY and the Constellation authors

Distributing computational load

- Running plug-ins in processes allows to employ more CPUs.
- 2 main challenges:
  - Compatibility – will be verified and guaranteed by backend.
  - Lossy comm. channels – will need a reliable layer on top of ZeroMQ.
- What if a process exits/crashes:
  - Frontend – no problem, data acquisition continues in headless mode.
  - Backend – plug-ins hosted by the backend process fail.

Physically detached experiments

- This architecture can implement measurement networks.
  - Frontend operates at a console computer in the control room.
  - Backends are deployed close to hardware.
    …or even better: backends embedded inside instruments!
- For vast distances: tunnel ZeroMQ sockets over WireGuard.



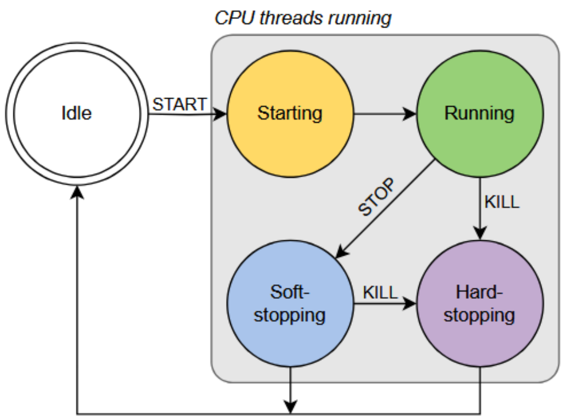
Interoperability between languages and platforms

- Now: core and all plug-ins are coded in C++
  - …required by running in the same process and sharing STL.
- Future: plug-ins could be developed in any language, as long as ZeroMQ data exchange conforms to specification.
- This is not granted, it will depend on interactions between the backend process and plug-ins.
- Ideally have a small C library with many bindings, which can be linked from major programming languages.
- Particularly interesting: Python scripts!

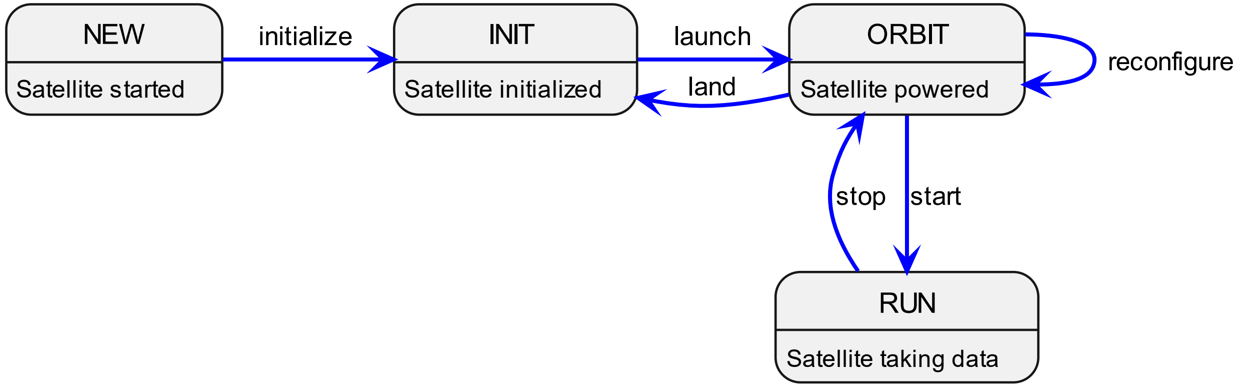
Finite state machines (FSM)

- Both programs use FSM to synchronize states of its elements.
- Track Lab’s FSM can be adapted for Constellation.
    - Potential issue when stopping: land vs. soft/hard stop.
    - We can interpret hard stop as ‘safe’ state.

Track Lab FSM:



Constellation FSM:



Credit: DESY and the Constellation authors

Technical footnotes

- Build systems:
  - Track Lab uses cmake
  - Constellation uses meson
- Remote GUI handling: need to deliver Qt signals over sockets
  - Should GUI component of each plug-in to be serviced in frontend?
  - Should it be serviced in backends?
- Platform compatibility checks out: Windows, Linux, macOS
- Long term: support for ‘alien’ satellites?

Summary, request for comments

- Track Lab has been growing steadily for 4 years, the current version is 1.5 (Oct 2024). We have lots of plans for 2025.
- This proposal is currently under consideration. If adopted, it can be viewed as a roadmap to the next major release (2.0).
- No illusion that this would be easy to implement, but:
  - We anticipate that our current architecture may become a bottleneck.
  - Track Lab was originally designed with this type of deployment in mind.
  - Constellation provides an environment favorable to scaling up.
- We would like to request feedback on this proposal.

Thank you for listening!

Petr Mánek, petr.manek@utef.cvut.cz

Try out Track Lab now:

Download v1.5 from
https://software.utef.cvut.cz

See article for details:

Available in J. Inst.
or arXiv:2310.08974

Minimum requirements:

- glibc 2.35 [x86_64, aarch64]
- Windows 10 [x86_64, arm64]
- macOS Monterey [x86_64, M1]

Prepared for submission to JINST
24th International Workshop on Radiation Imaging Detectors
June 25-29, 2023
Oslo, Norway

Track Lab: extensible data acquisition software for fast pixel detectors, online analysis and automation

P. Mánek,[a,b] P. Burian,[a,c] E. David-Bosne,[a] P. Smolyanskiy[a] and B. Bergmann[a]

[a] Institute of Experimental and Applied Physics, Czech Technical University in Prague, Husova 240/5, Prague, 110 00, Czech Republic
[b] Department of Physics and Astronomy, University College London, Gower Street, London, WC1E 6BT, United Kingdom
[c] Faculty of Electrical Engineering, University of West Bohemia, Univerzitní 2795/26, Pilsen, 301 00, Czech Republic

E-mail: petr.manek@utef.cvut.cz

Abstract: Fast, incremental evolution of physics instrumentation raises the question of efficient software abstraction and transferability of algorithms across similar technologies. This contribution aims to provide an answer by introducing Track Lab, a modern data acquisition program focusing on extensibility and high performance. Shipping with documented API and more than 20 standard modules, Track Lab allows complex analysis pipelines to be constructed from simple, reusable building blocks. Thanks to multi-threaded infrastructure, data can be clustered, filtered, aggregated and plotted concurrently in real-time. In addition, full hardware support for Timepix2, Timepix3 pixel detectors and embedded photomultiplier systems enables such analysis to be carried out online during data acquisition. Repetitive procedures can be automated with support for motorized stages and X-ray tubes. Freely distributed on 7 popular operating systems and 2 CPU architectures, Track Lab is a versatile tool for high energy physics research.

Keywords: Data acquisition concepts; Data processing methods; Pattern recognition, cluster finding, calibration and fitting methods; Detector control systems (detector and experiment monitoring and

arXiv:2310.08974v1 [physics.ins-det] 13 Oct 2023