DAQ Frameworks

Daron Chabot, BNL
EPICS Collaboration Meeting
NSRRC, Hsinchu, Taiwan
6/13/11

U.S. DEPARTMENT OF ENERGY
BROOKHAVEN NATIONAL LABORATORY
BROOKHAVEN SCIENCE

Outline

- General use cases
- Tour of existing systems & tools
- General DAQ components & scan types
- Next steps

DAQ Users & Scenarios
(Who needs this stuff??)

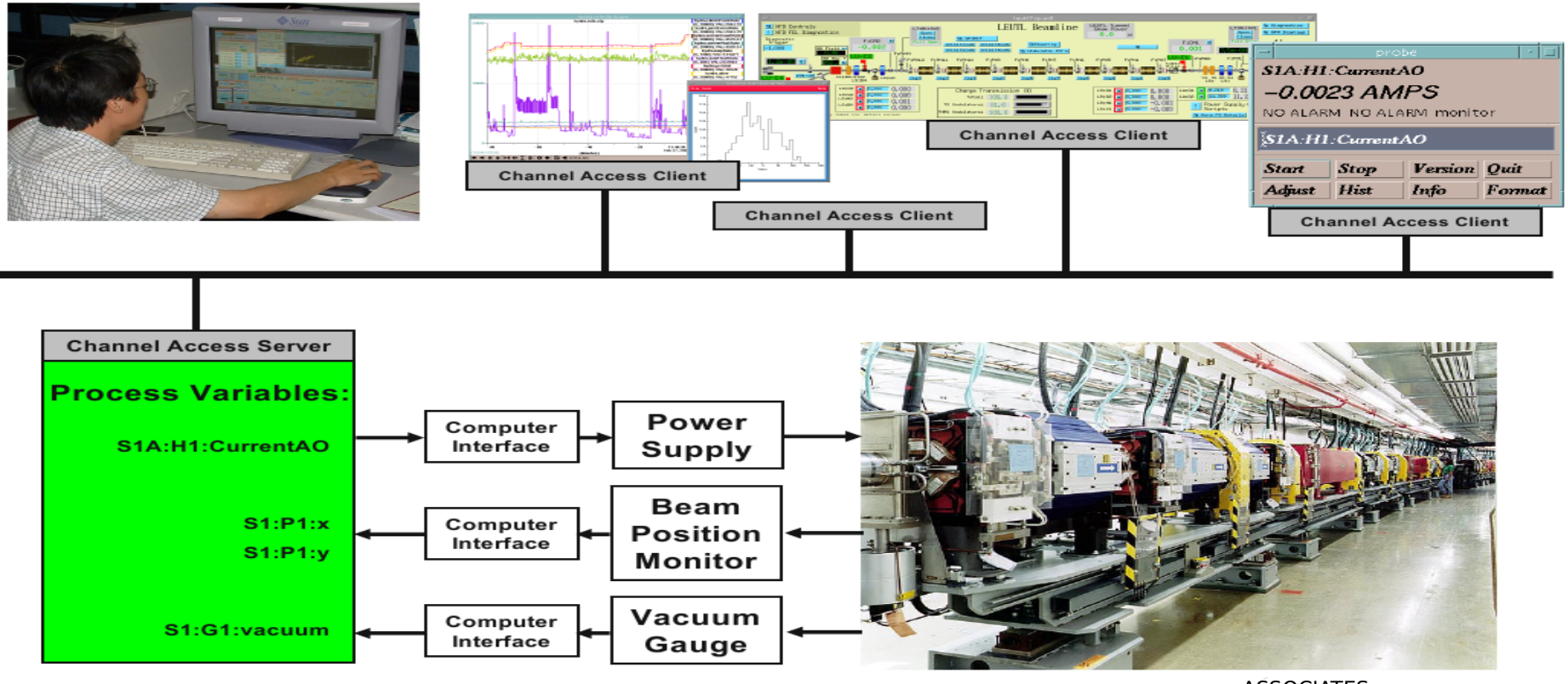
- **Accelerator physicist**: determine Response Matrix
- **Accelerator physicist**: determine ID correction-coil LUT
- **Beamline user**: reciprocal-space scan with 2-d array
- **Beamline user**: XAFS or EXAFS data collection
- **Beamline user**: 2-d, on-the-fly fluorescence scan
- **Beamline scientist**: Measure intensity versus ID-gap

Experiment Control & Data Collection

- ho·lis·tic/hōˈlistik/
  - Adjective: Characterized by comprehension of the parts of something as intimately interconnected and explicable only by reference to the whole.
- Beamline interface must be ***holistic***
- Experiment Control
  - Is sample- *and* data-centric
  - The *super-set* of functionality
  - Sample mgmt, data collection, data management (access/storage)
- Data acquisition
  - Is *only* data-centric (naturally!)

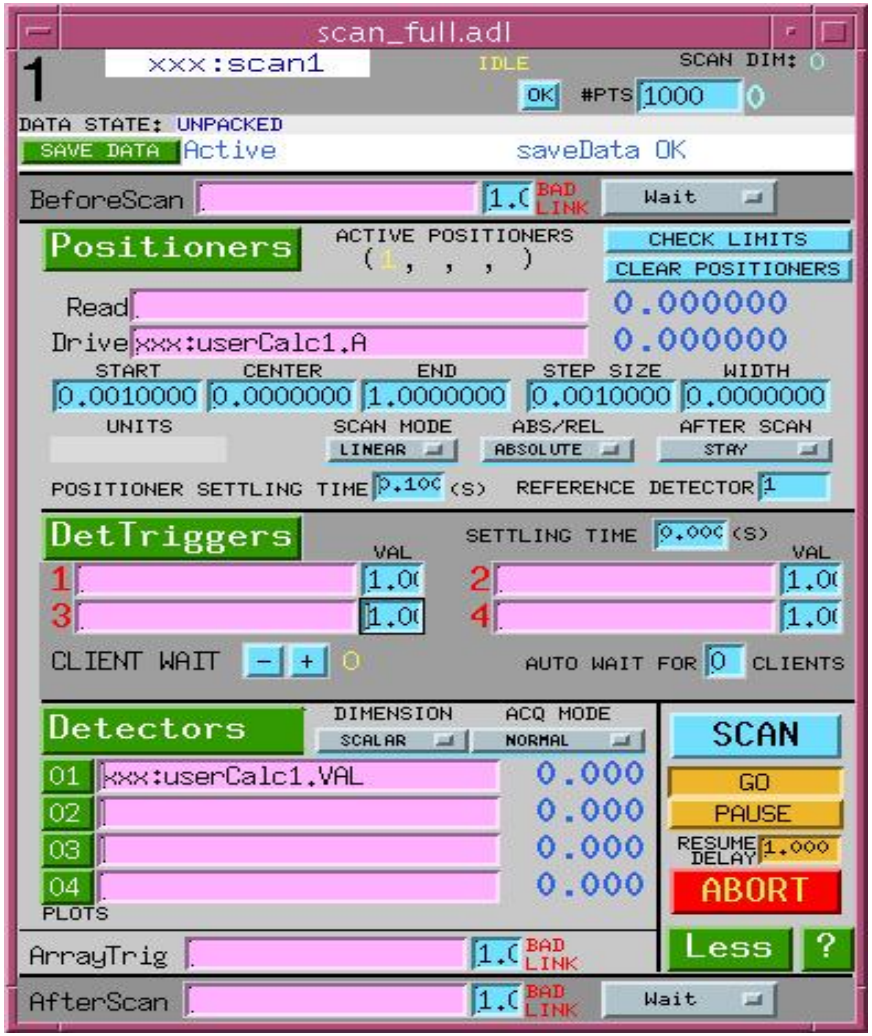
EPICS

- EPICS is ***NOT*** a DAQ/Experiment Control system
  - It is a ***foundation*** component and interface
- Large, open source community



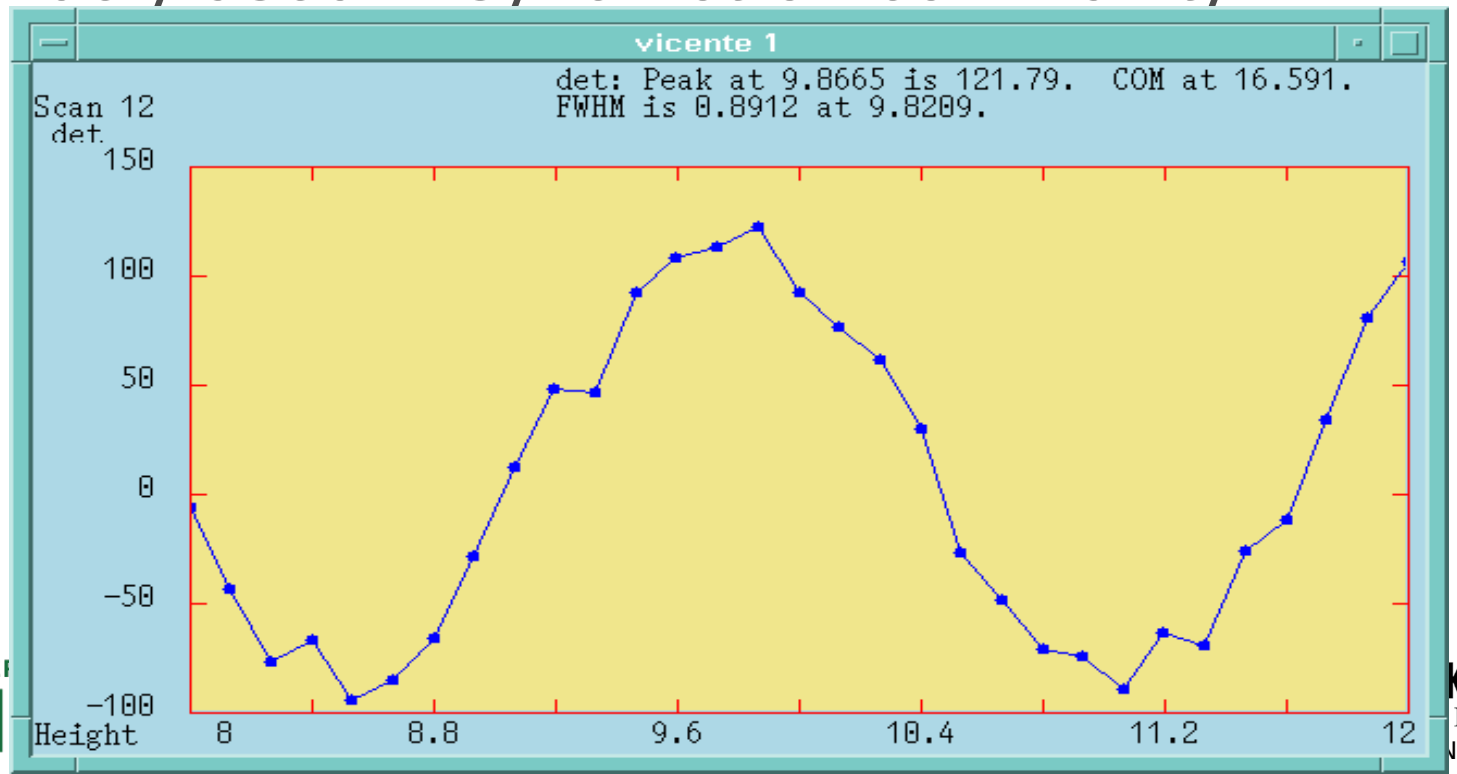
EPICS

- sscan record
  - AKA: “The Band-Aid”
- 4 “positioners”, 70 “detectors”
- N-dimensional scans
- Linear, table, fly modes
- Post-move ops:
  - Go to peak, + edge
- Binary hierarchical files



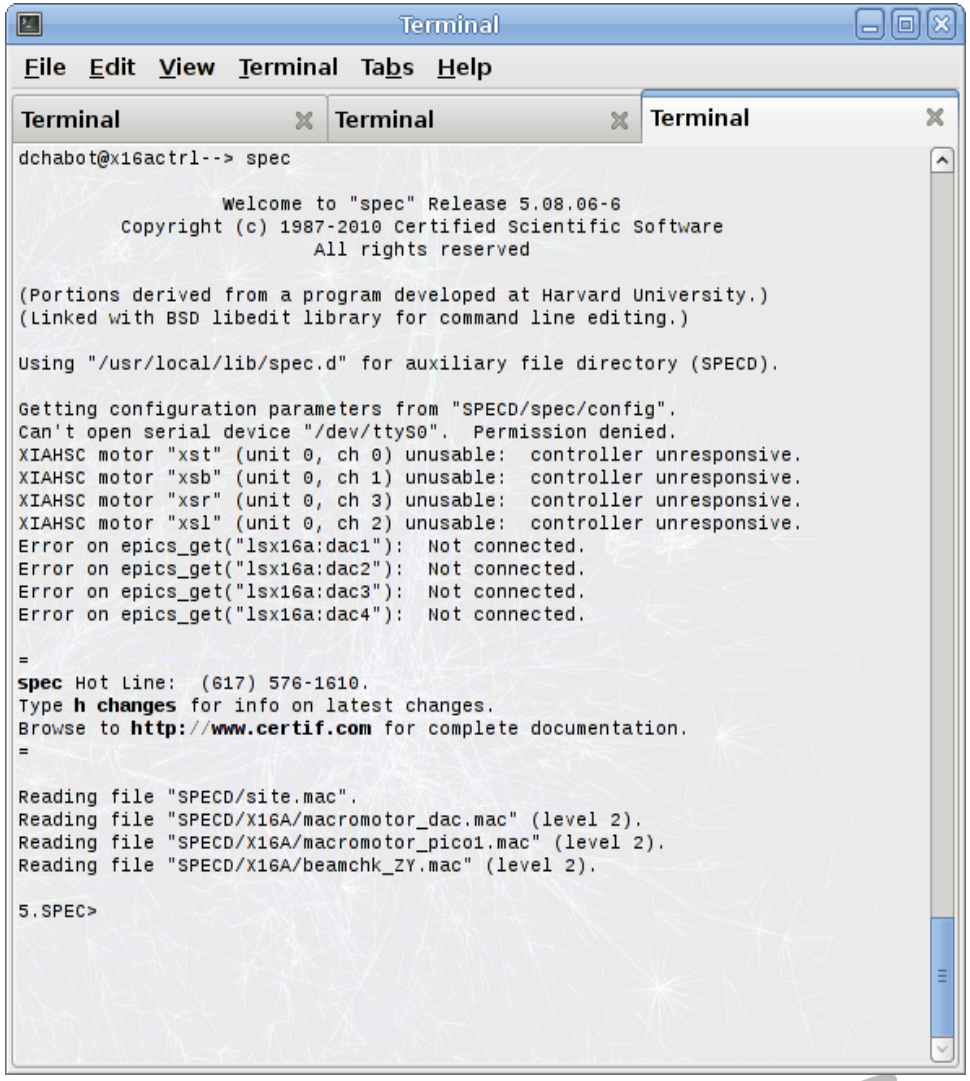
Spec

- Spec is ***NOT*** a control system
  - Highly ***configurable*** system for data acquisition
- Commercial product (v. small company)
- Widely used in synchrotron community



Spec

- CL interface
- Motor/counter-centric
  - Pseudo-motors
- Diffraction routines
- Script-able
- ASCII data format
- Basic plots/analyses

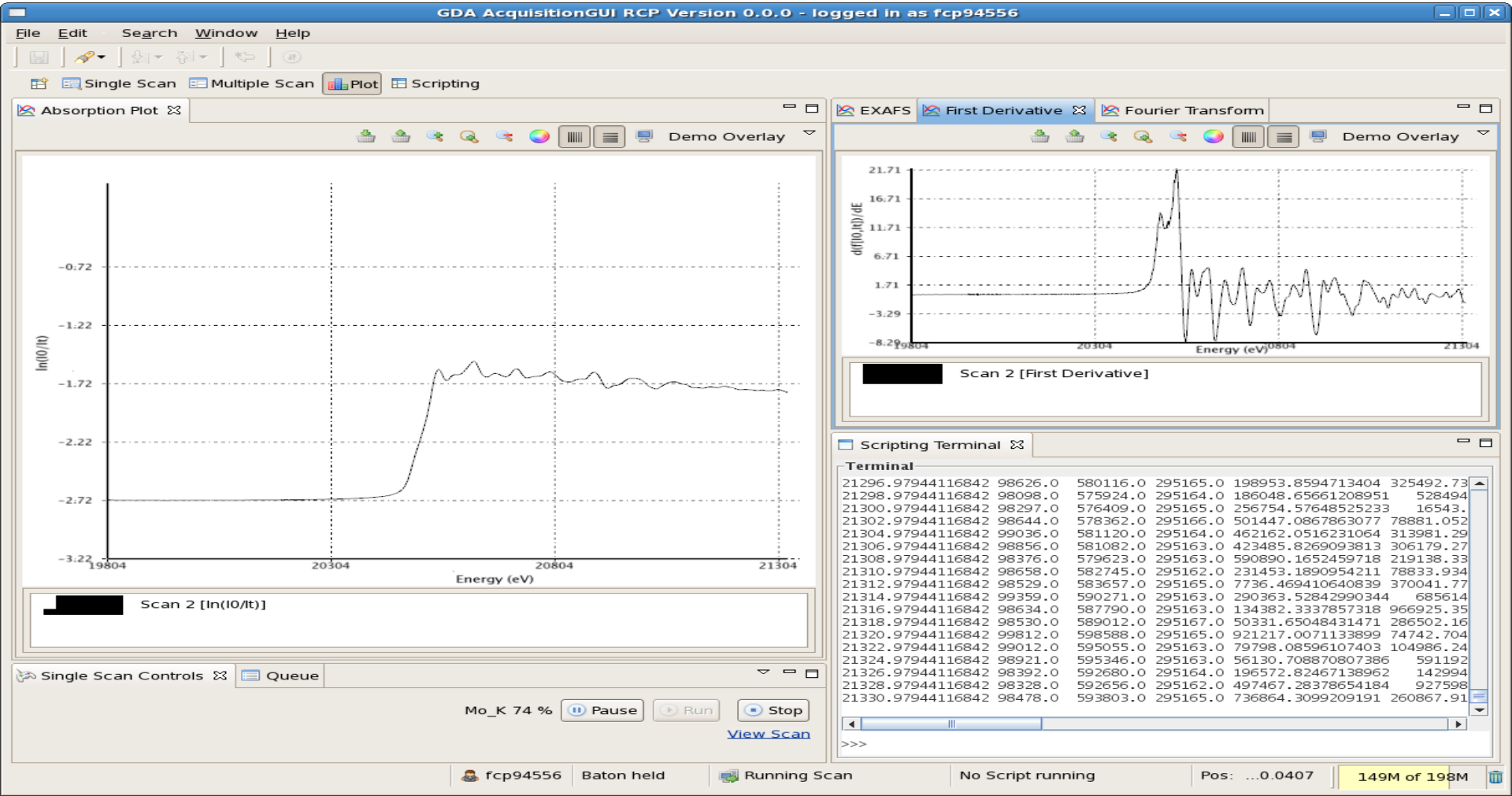


Spec

- Design is not easily extensible
  - Doesn't play well with others
- Domain-specific languages are difficult to justify.
- ASCII data
- No debugging procedures

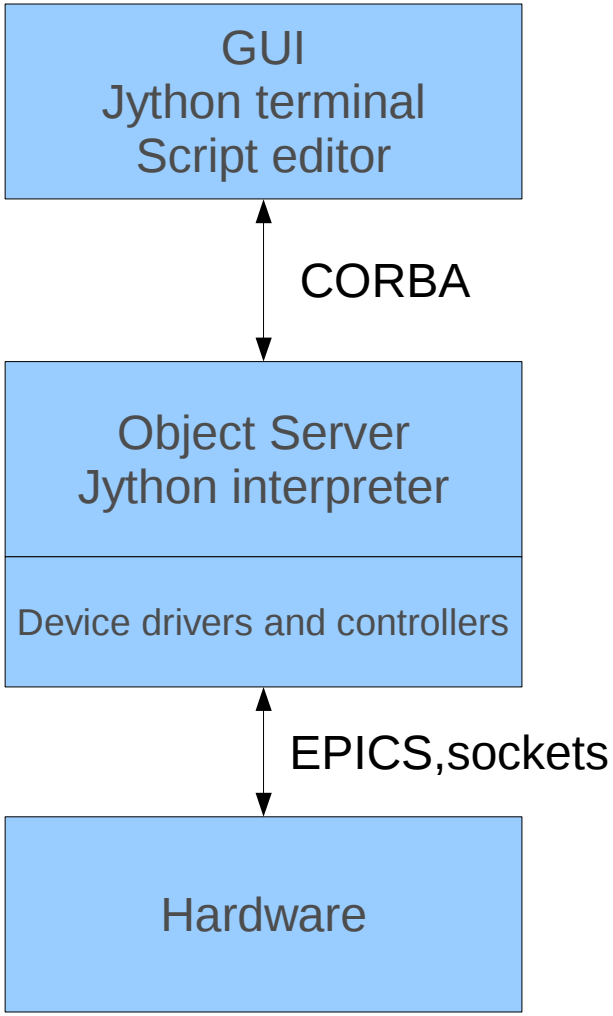
GDA (Generic Data Acquisition)

- Java-based “control and data collection software”
- Developed at Daresbury, now by DLS



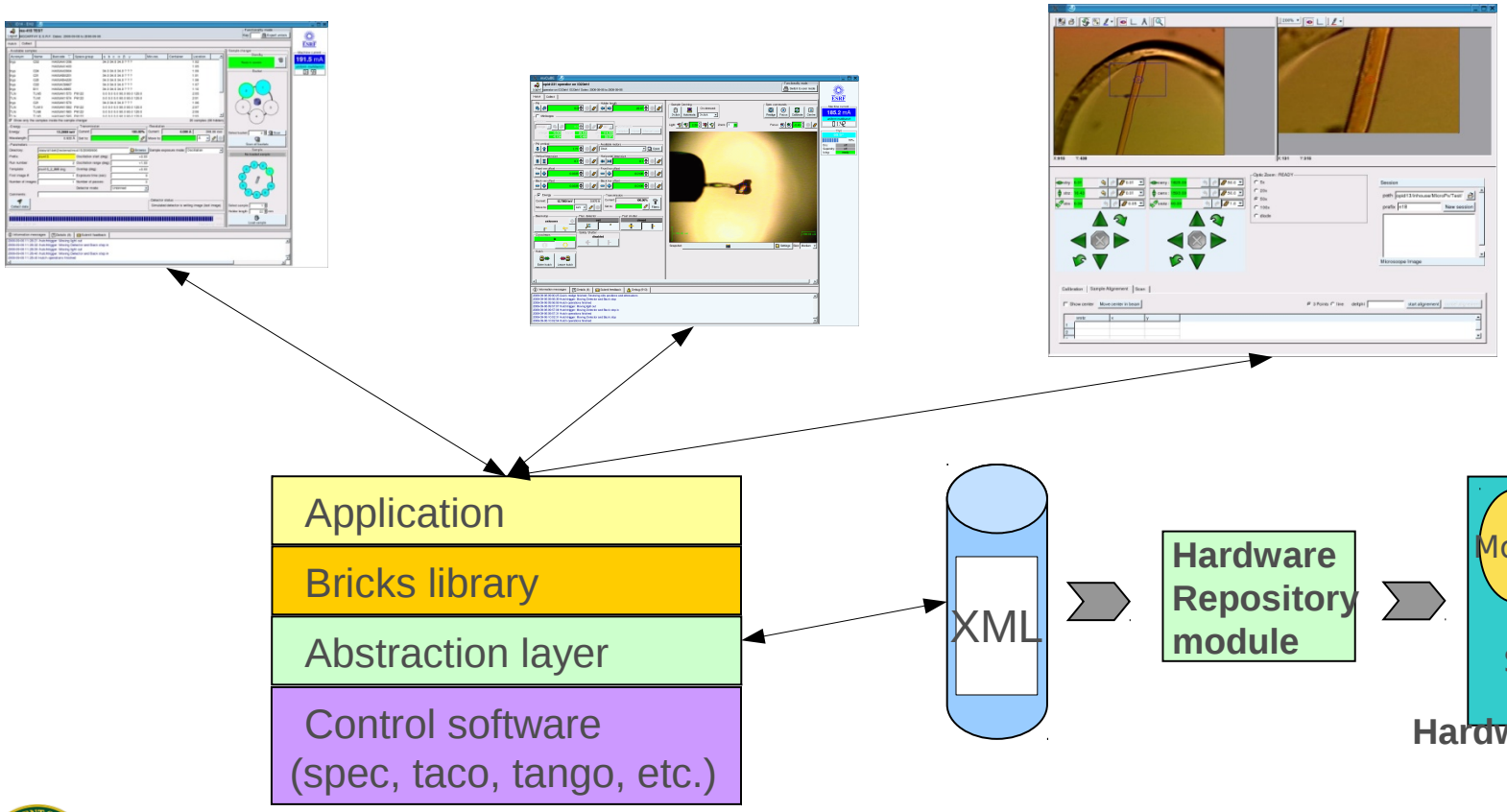
GDA

- Java, Eclipse RCP, CORBA
- Control system independent
- Script-able (Jython)
- Diffraction (DiffCalc)
- SciSoft analysis pkg
- NeXus files
- Authorization/Authentication
  - Integration with User Office
- Logging (log4j)



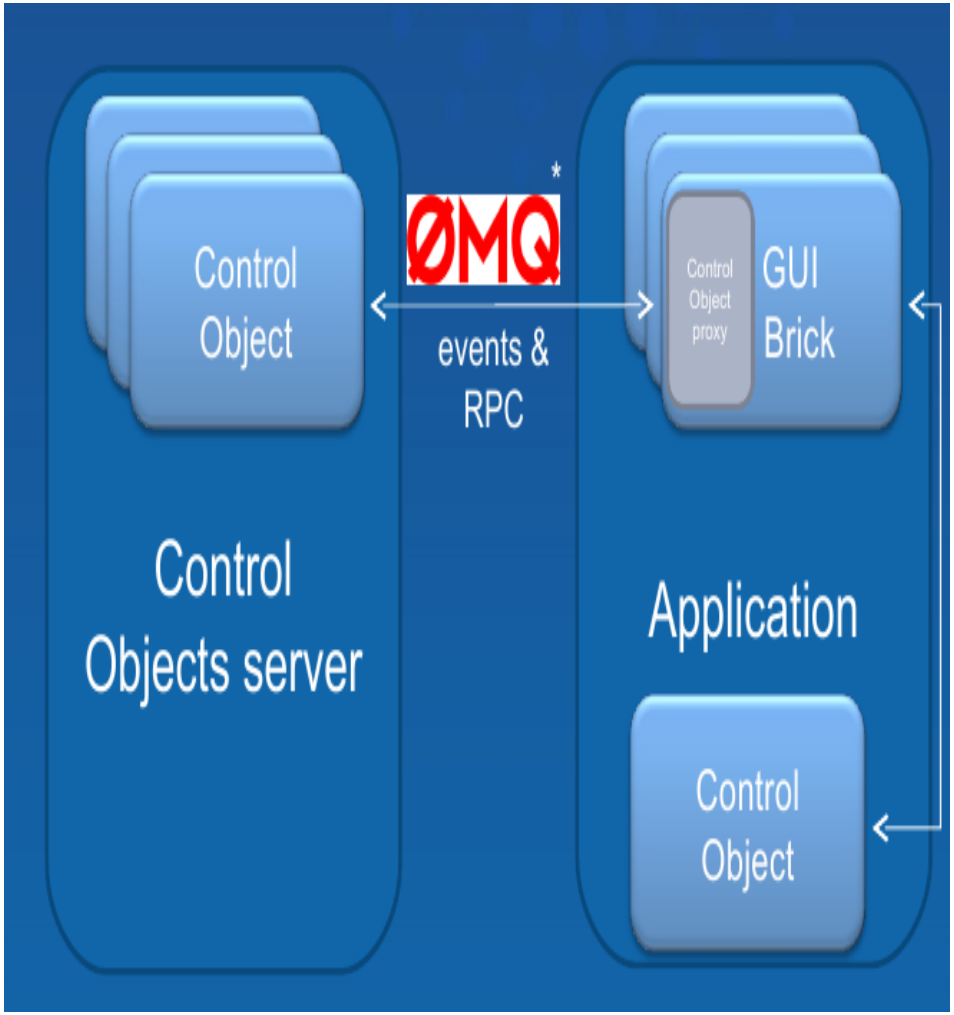
Bliss

- Python, QT4 DAQ framework
  - Developed at ESRF



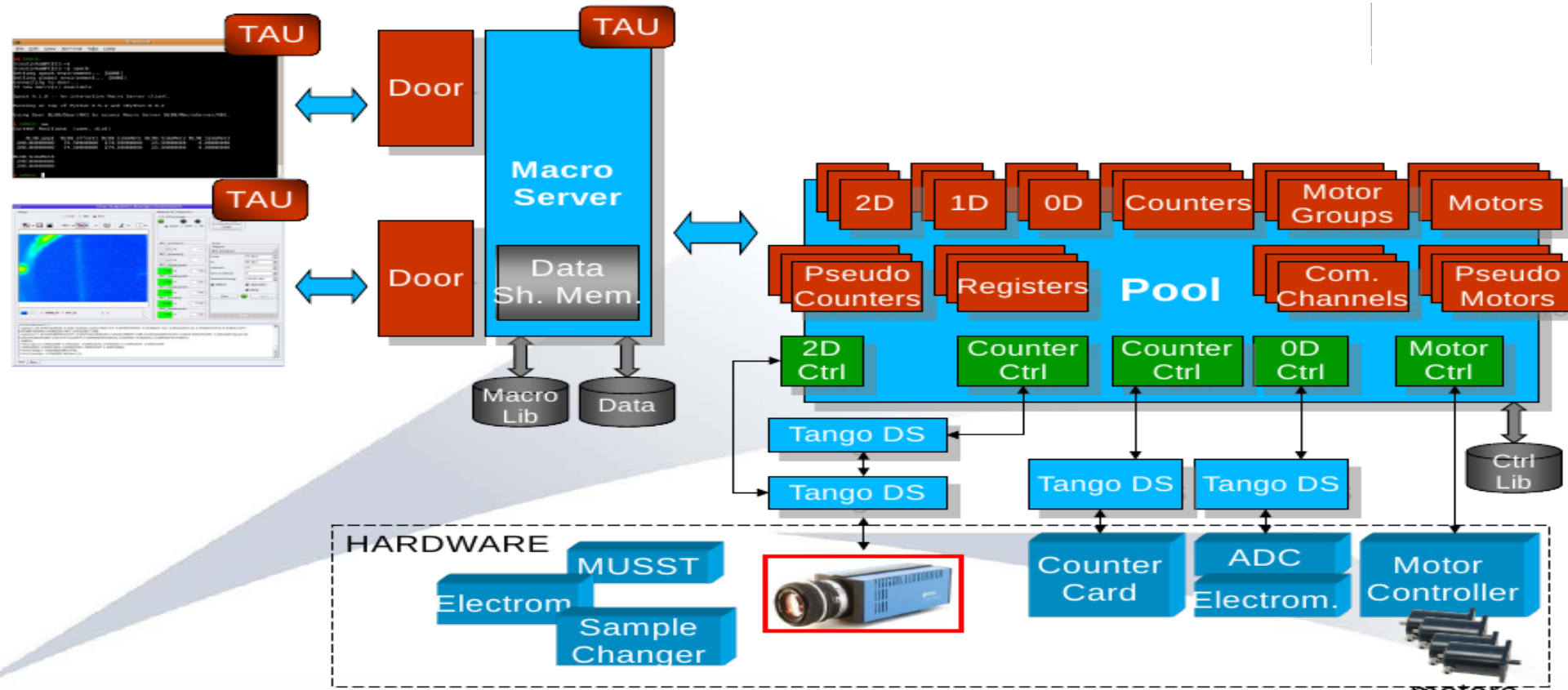
Bliss

- GUI “Bricks”
  - Application Builder
- Control Objects
  - Commands
    - Spec macros
  - Channels
    - Tango “Attributes”
- XML config per beamline



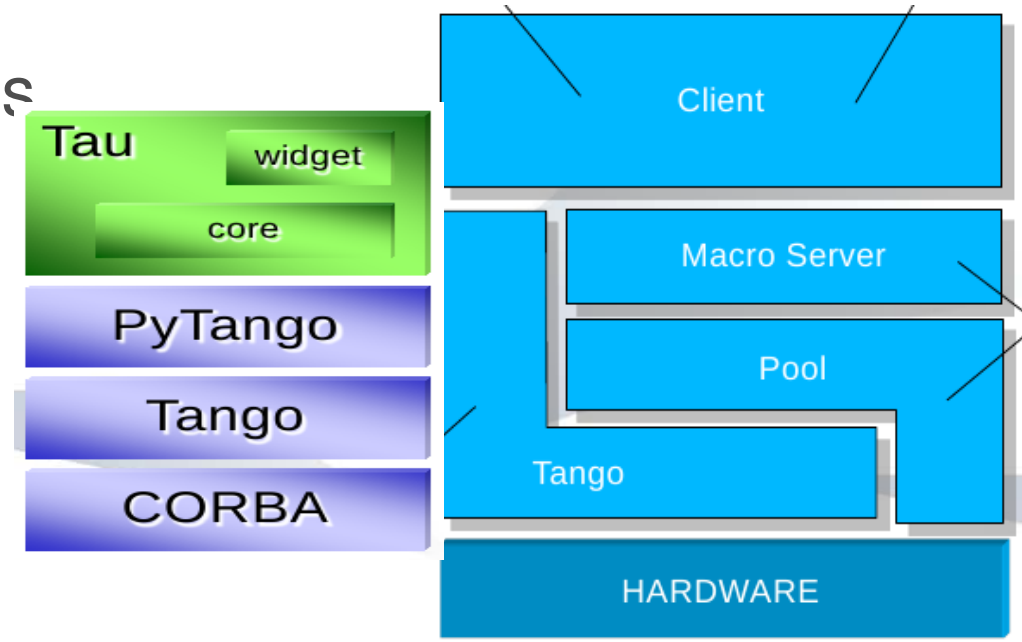
Sardana

- (another) Python DAQ framework
  - Developed at ALBA
    - Used at ALBA, ESRF, Petra III ?

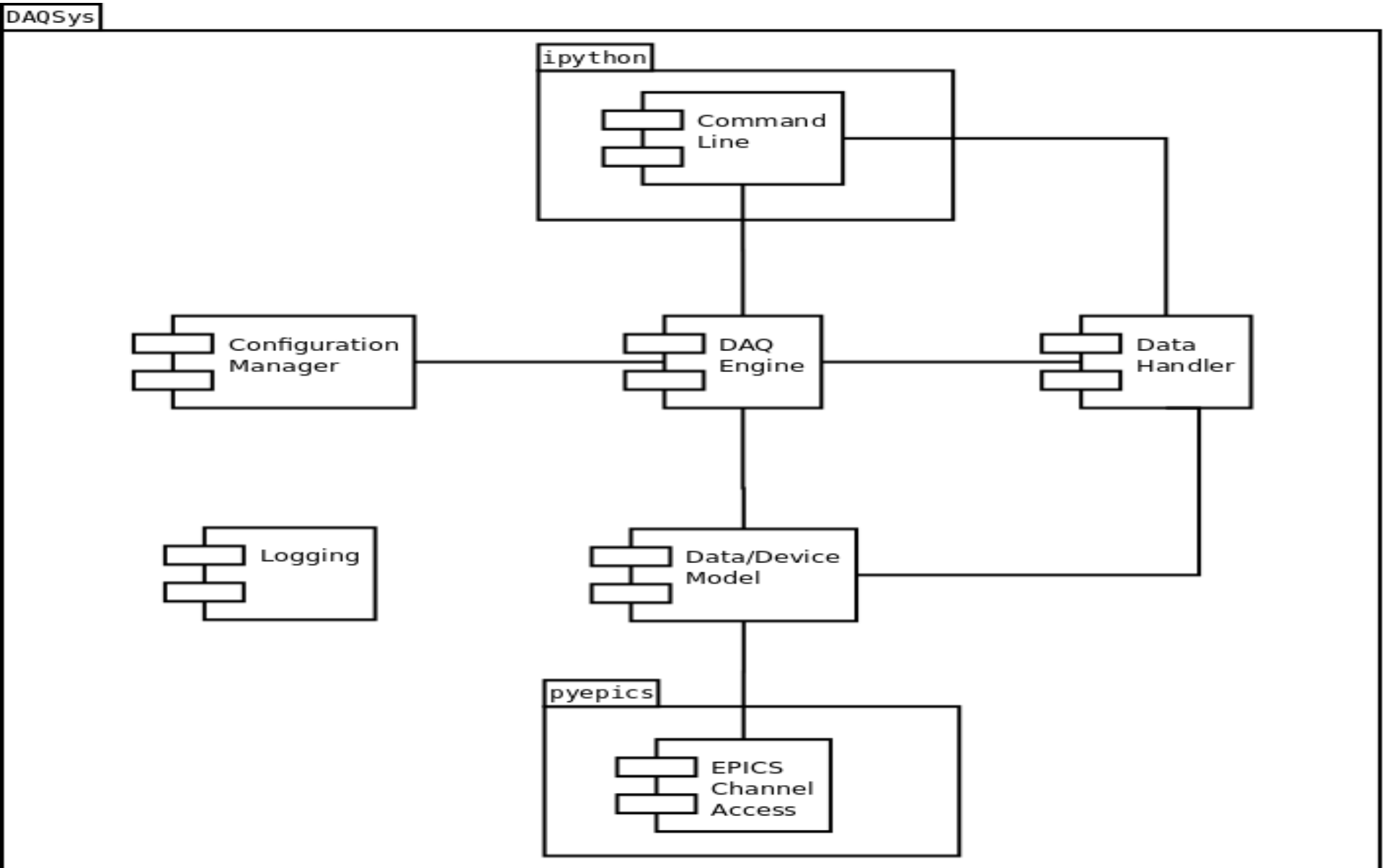


Sardana

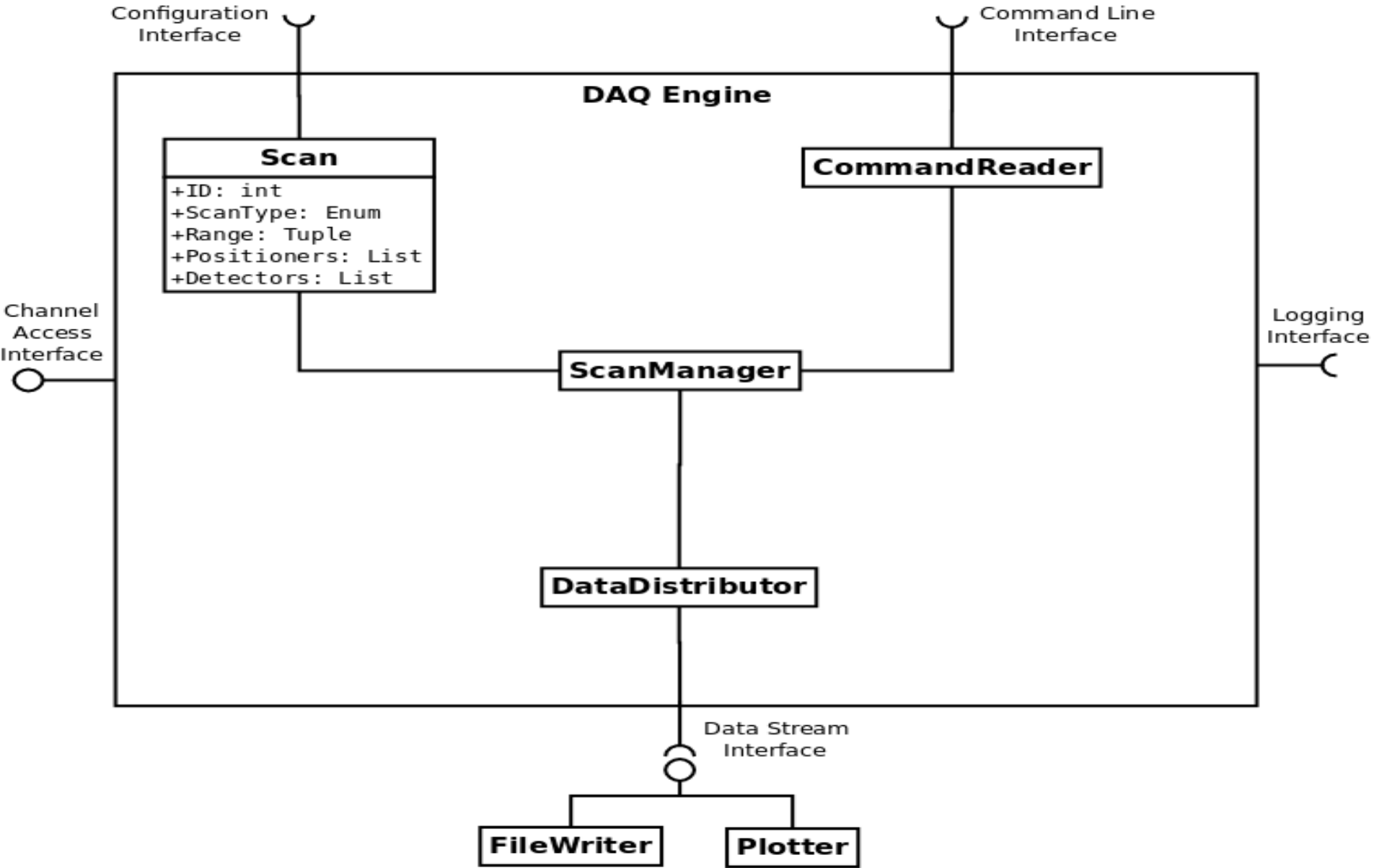
- Spec 'clone' (Spock)
  - Interface via **ipython**
  - ascan, dscan macros
- Tango UI (Tau)
- TauPlot
  - Basic plotting utils
- Scan Executor UI
- NeXus files



DAQ Components



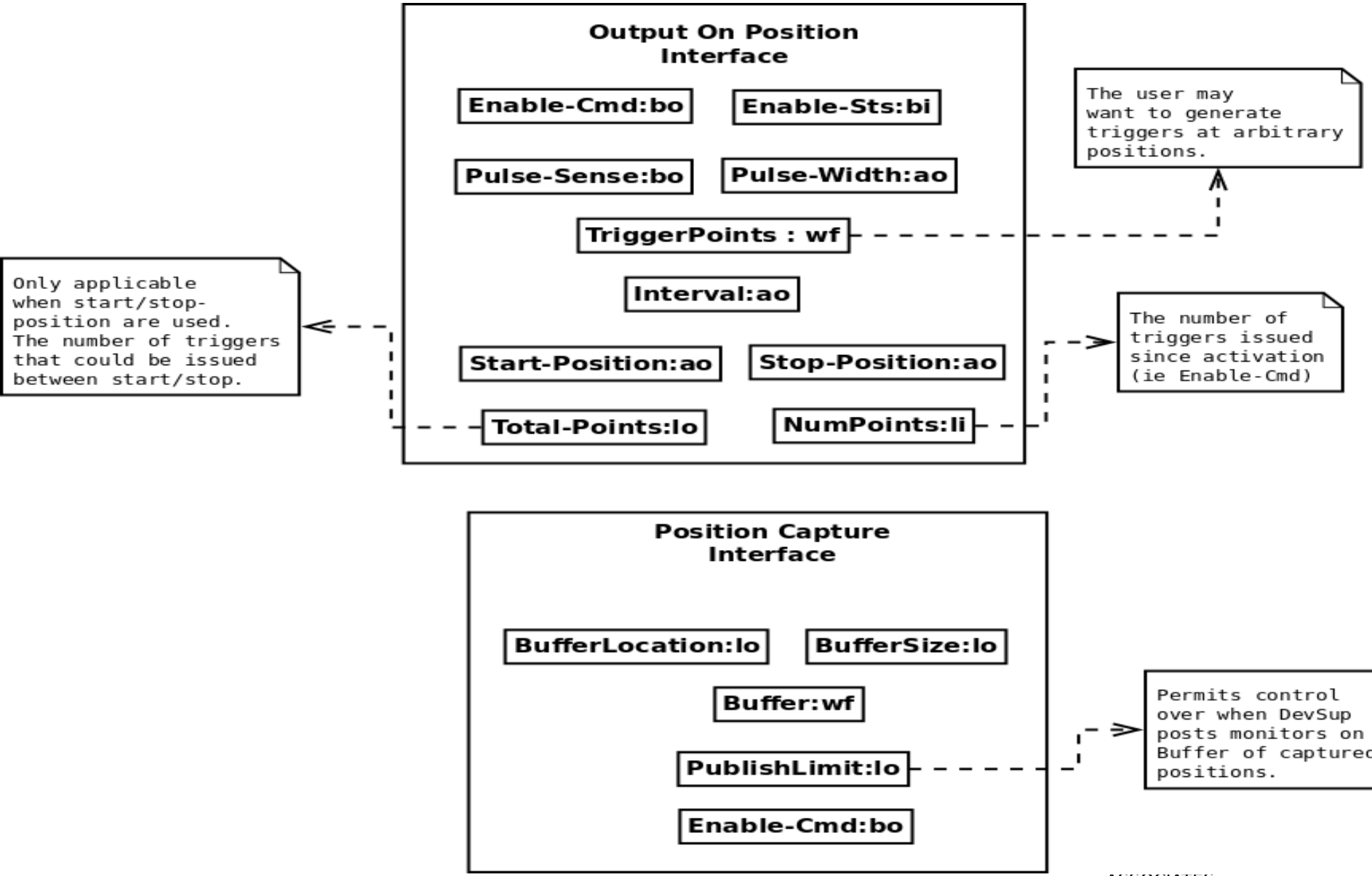
Scanning Engine



Scan Types/Algorithms

- Step scans (discrete)
  - Simple
  - Loose coupling (H/W, S/W)
  - H/W is time-limiting factor (mechanics)
- Fly scans (continuous)
  - More complex
  - Requires hardware support (buffering)
    - Speeds become too fast for net communication
  - S/W configures and manages. Hands off!

Fly Scanning: EPICS



Next Steps

- Collaborate. Collaborate. Collaborate.
- Personnel
- Salvage components/concepts

Next Steps (2)

- "The Solution" must be a framework
  - There is no *one-size-fits-all*
  - Concepts are "similar enough" to abstract into components
  - Configuration-based, not compilation-based
- Python will almost certainly be a core technology
  - Simple, flexible, powerful
  - Toolkits: graphics, communication, math & visualization

Acknowledgements

- NSLS-II Beamline-Controls Working Group
- Stuart Wilkins
- Paul Gibbons & Richard Fearn (DLS)
- Matias Guijarro (ESRF)
- Jorg Klora & Tiago Coutinho (ALBA)

Thanks for your attention!