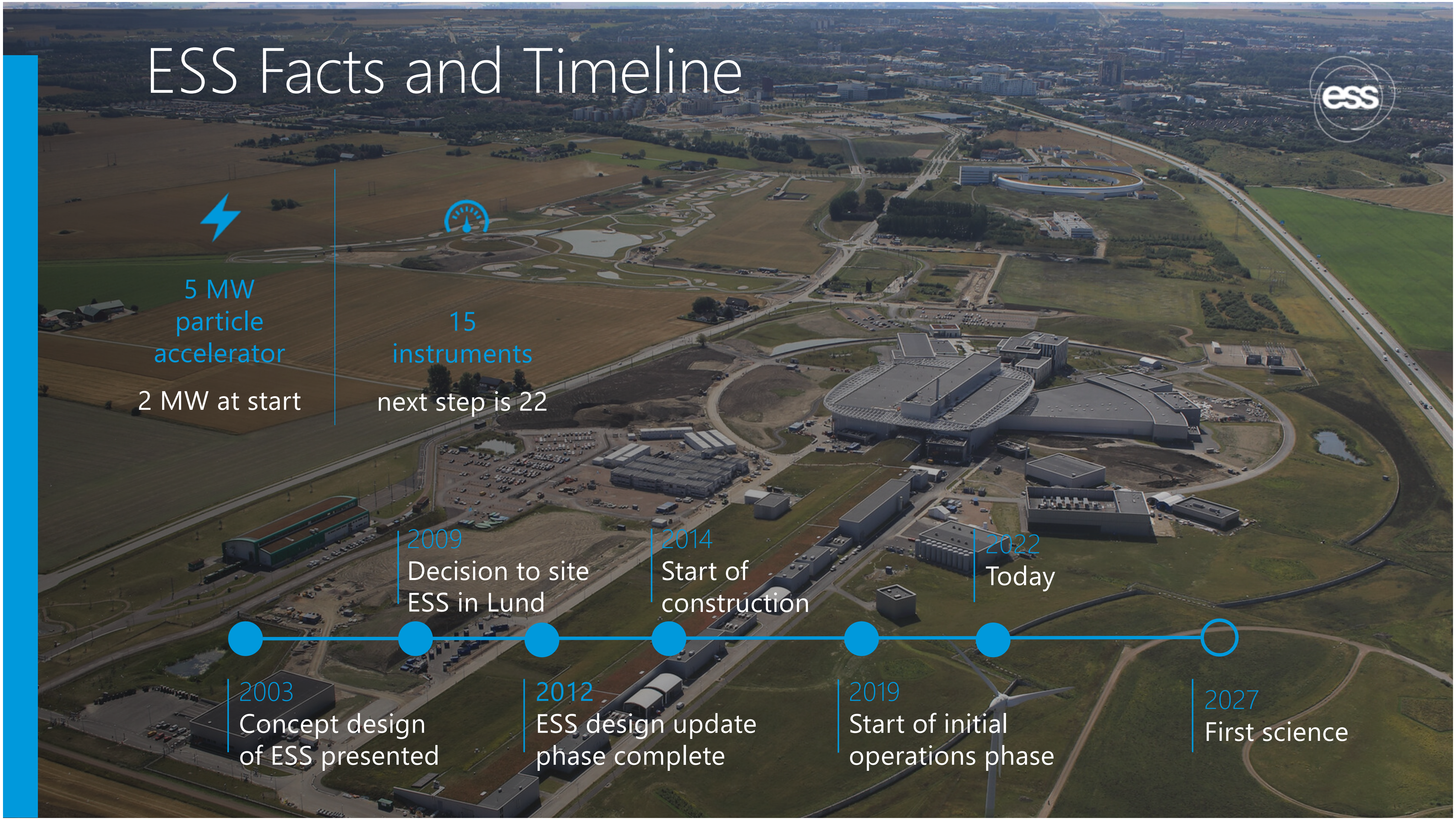
EUROPEAN
SPALLATION
SOURCE

DAQ Architecture for Instruments at the European Spallation Source

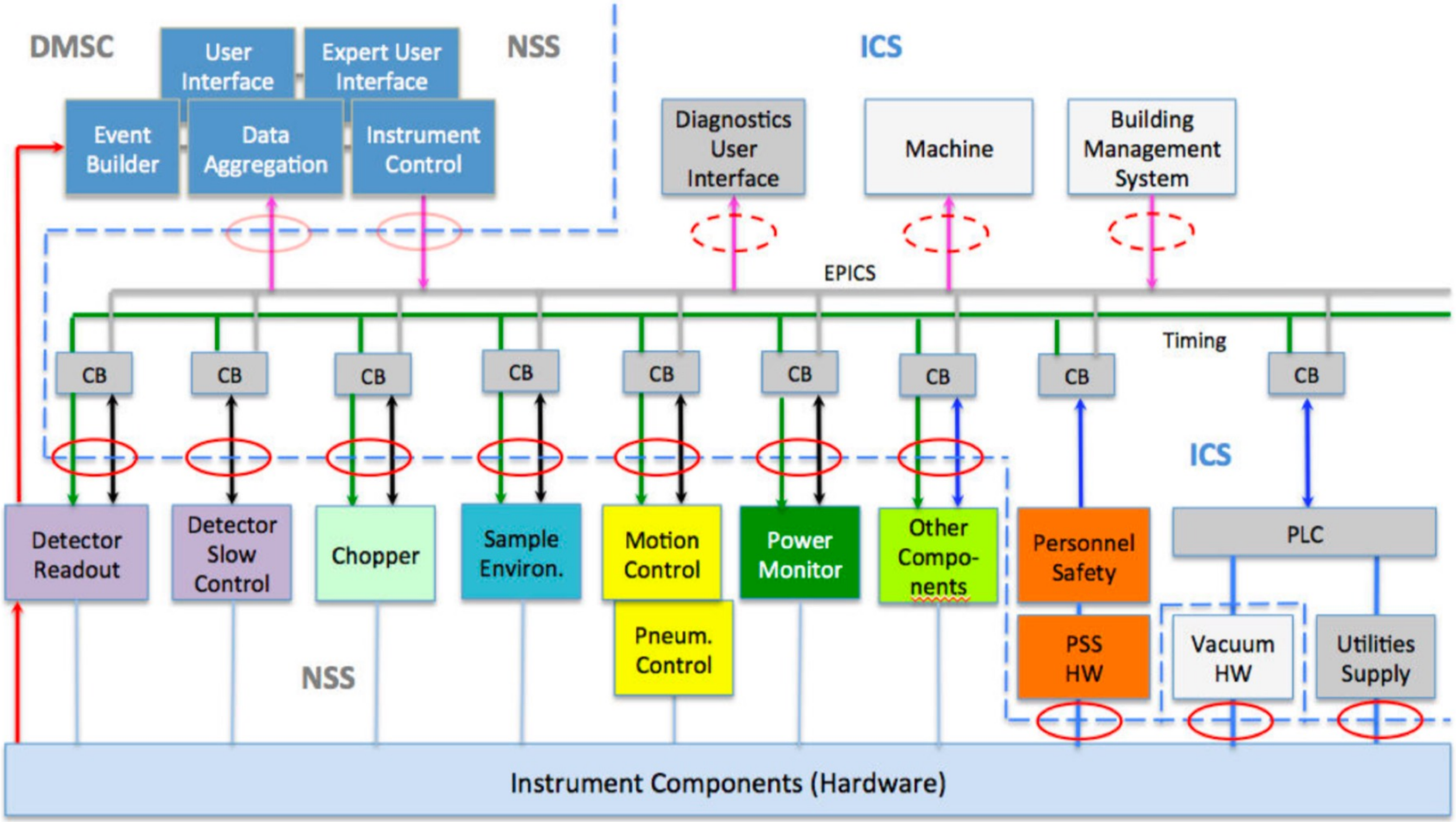
PRESENTED BY TOBIAS RICHTER - EXPERIMENT CONTROL AND DATA CURATION GROUP

2022-06


ESS Facts and Timeline
5 MW
particle
accelerator
2 MW at start
15
instruments
next step is 22
2009
Decision to site
ESS in Lund
2014
Start of
construction
2022
Today
2003
Concept design
of ESS presented
2012
ESS design update
phase complete
2019
Start of initial
operations phase
2027
First science

ESS Controls and Readout Architecture

Based on EPICS – Part of the Ancient Revelations of ESS



Low Level Control and Engineering Interfaces

Maintenance and Operations

Three types of interfaces:

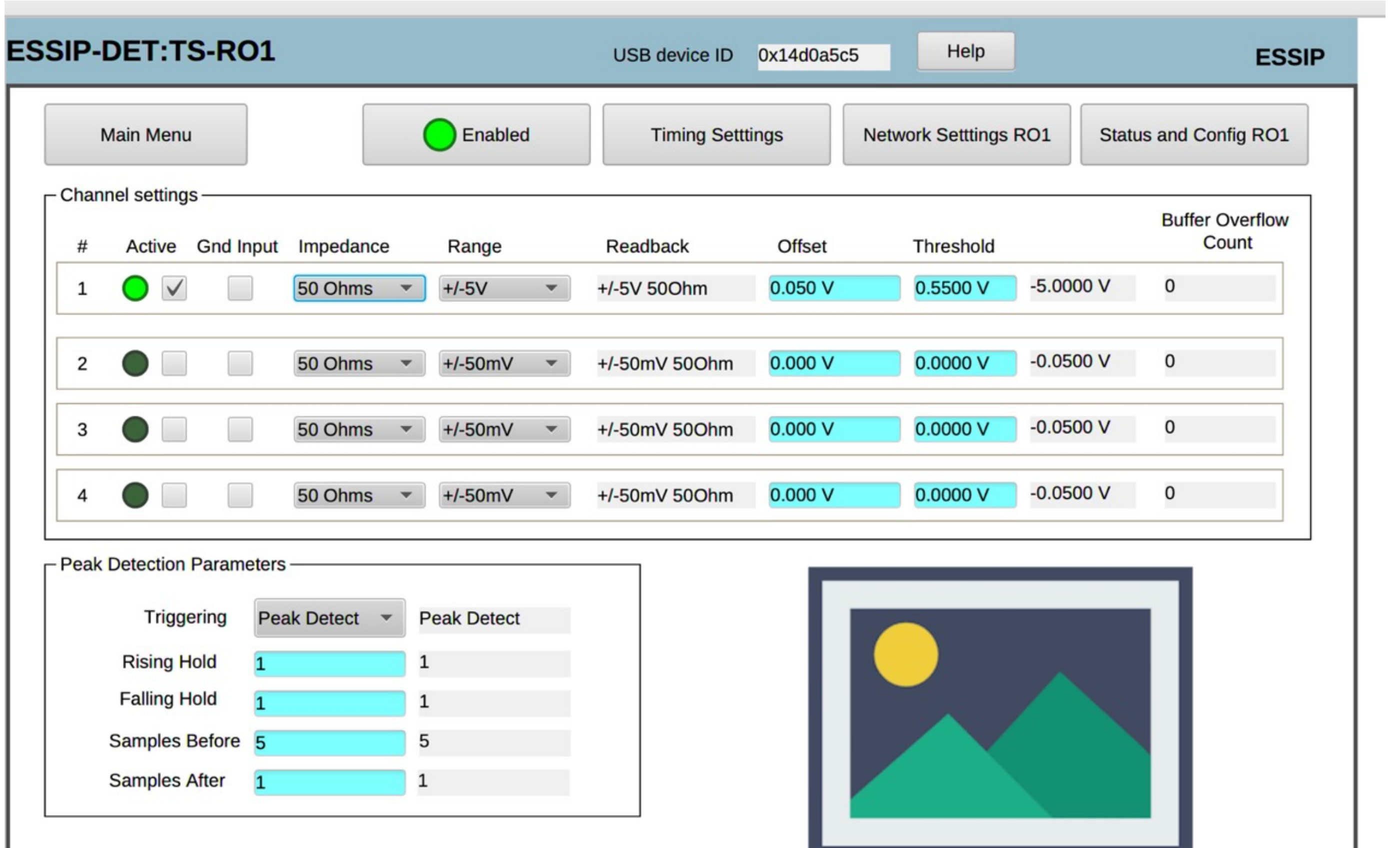
User
The average facility user

Expert
An expert user, also often the instrument scientist

Engineering
ESS support staff with special needs outside of running an experiment

We aim to cater for users and experts in NICOS.

The ideal way to support engineering tasks for a specific device depends on many factors.



CSS/Phoebus/EPICS interface example

Experiment Control Programme

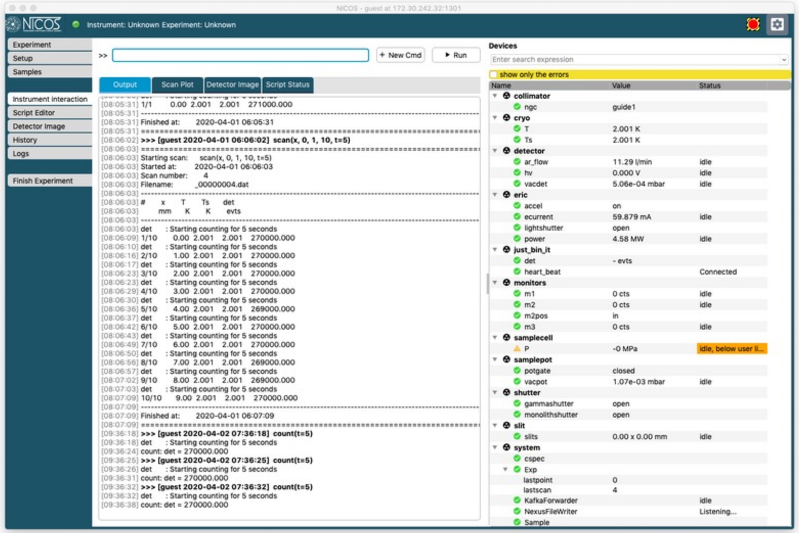
NICOS

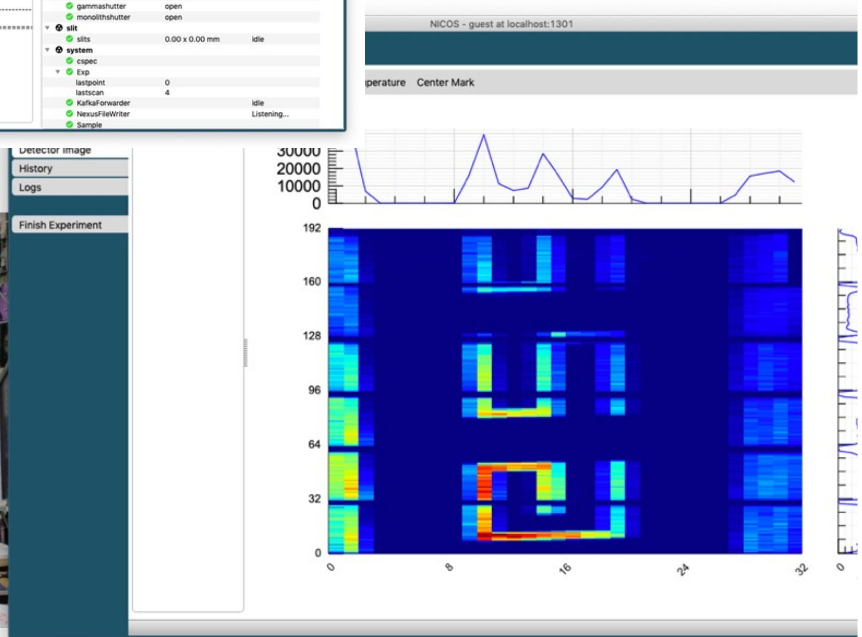
Integrated high level interface and device abstraction to low level EPICS controls.

- Python CLI
- Qt user interface
- Extensible and customizable

NOT an integral part of the data path.

- Interacts with devices via EPICS
- Controls file writing via Kafka and sends data there for recording
- Can visualise live data from Kafka





FRM II
Forschungs-Neutronenquelle
Heinz Maier-Leibnitz

Networked Instrument COntrol System

PAUL SCHERRER INSTITUT

Detector Data Acquisition

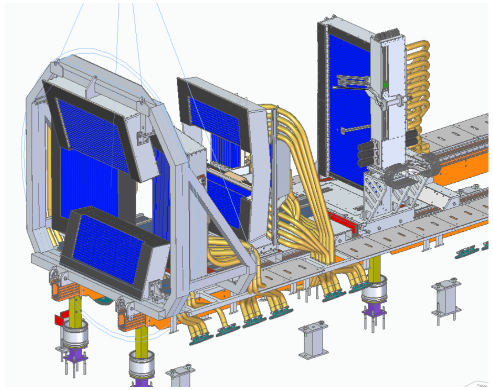
Time-of-flight event-based

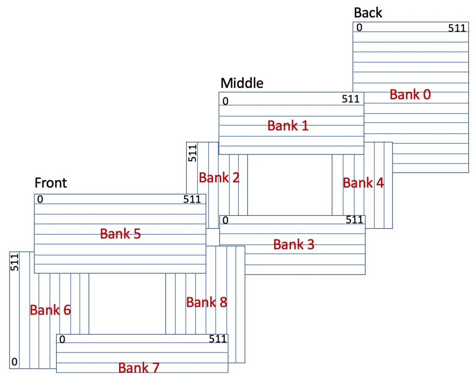
Standardised interface:
Detector “backend master” operates as the interface to the outside world. Talks to frontend electronics via a number of rings.

Provides:

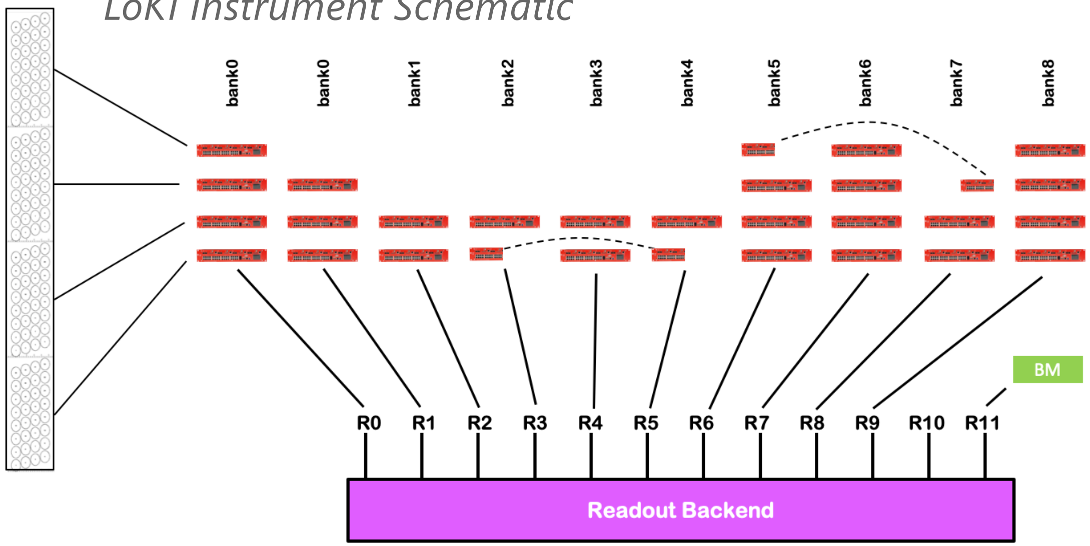
- Timing integration with frontends electronics
- High rate data path – direct to DMSC/ECDC event formation
- Interface for configuration via EPICS (slow control)

Beam Monitors (single pixel detectors) use the same infrastructure.





LoKI Instrument Schematic



Software for Detectors

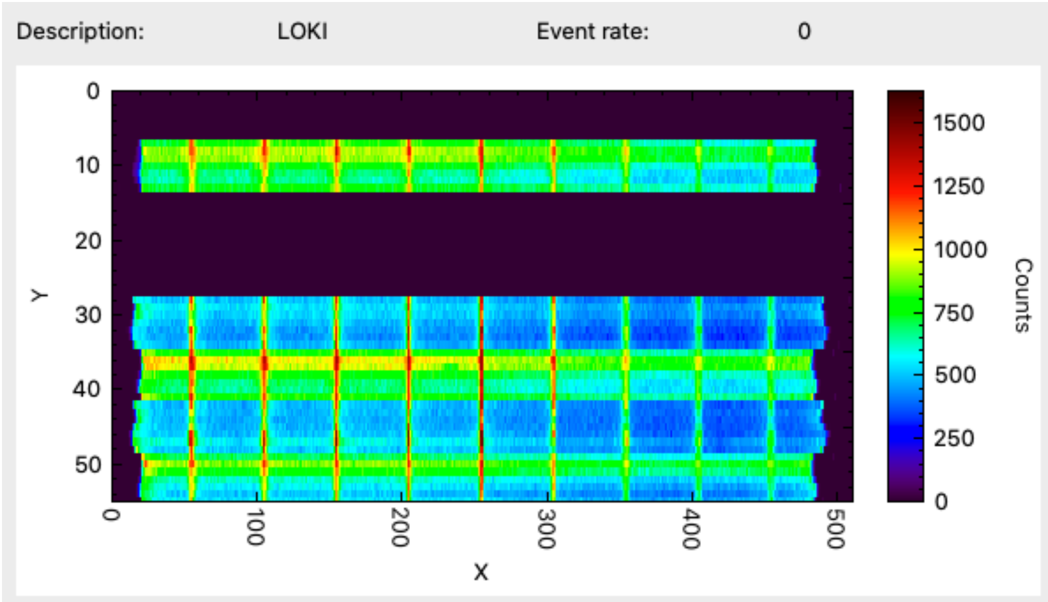
Event Formation and Slow Control

Event Formation

ESS Detectors run software on commercial off the shelf computers to perform online processing of detector output. That builds the coincidences, establishes neutron positions and refines the time-stamping. It translates from detector specific readout format to a standardised event serialisation format.

In addition

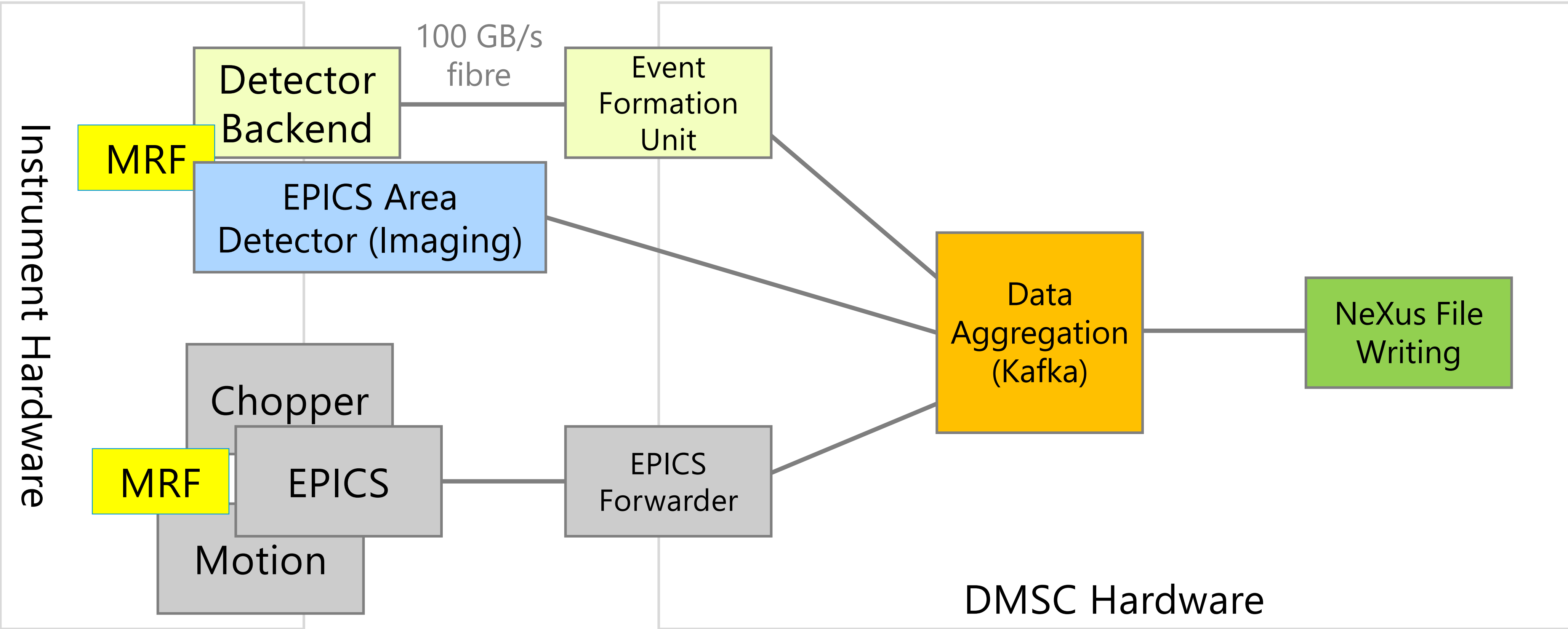
- Control of the detector electronics, power supplies and other infrastructure (gases, etc)
- Imaging (ODIN) will use cameras (EPICS Area Detector with Kafka plugin) as well as time-of-flight detectors



EFU processed (and corrected/calibrated) data from LoKI tubes,

Instrument Readout Architecture

Main Data Path



BrightnESS is funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 676548

Apache Kafka

Apache Kafka is a general purpose message broker system. Built for real-time data streaming and to be scalable, fault-tolerant and fast.

Scaling happens in clusters across hardware boundaries with individual brokers.

Data is divides in a number of independent topics, with different producers, consumers, retention policy and replication.

We use it as a chronological log that is always on and trigger file writing when needed (file writing can go back in time).

Kafka is agnostic to the payload of the messages.

We use a number of standardised Google Flatbuffer schemas as agreed serialisation interface between producers and receivers. For example:

- Neutron event data
- Chopper timestamps
- EPICS updates
  (motion, sample environment, etc)
- File writer control
  (including NeXus structure)

https://github.com/ess-dmsc/streaming-data-types

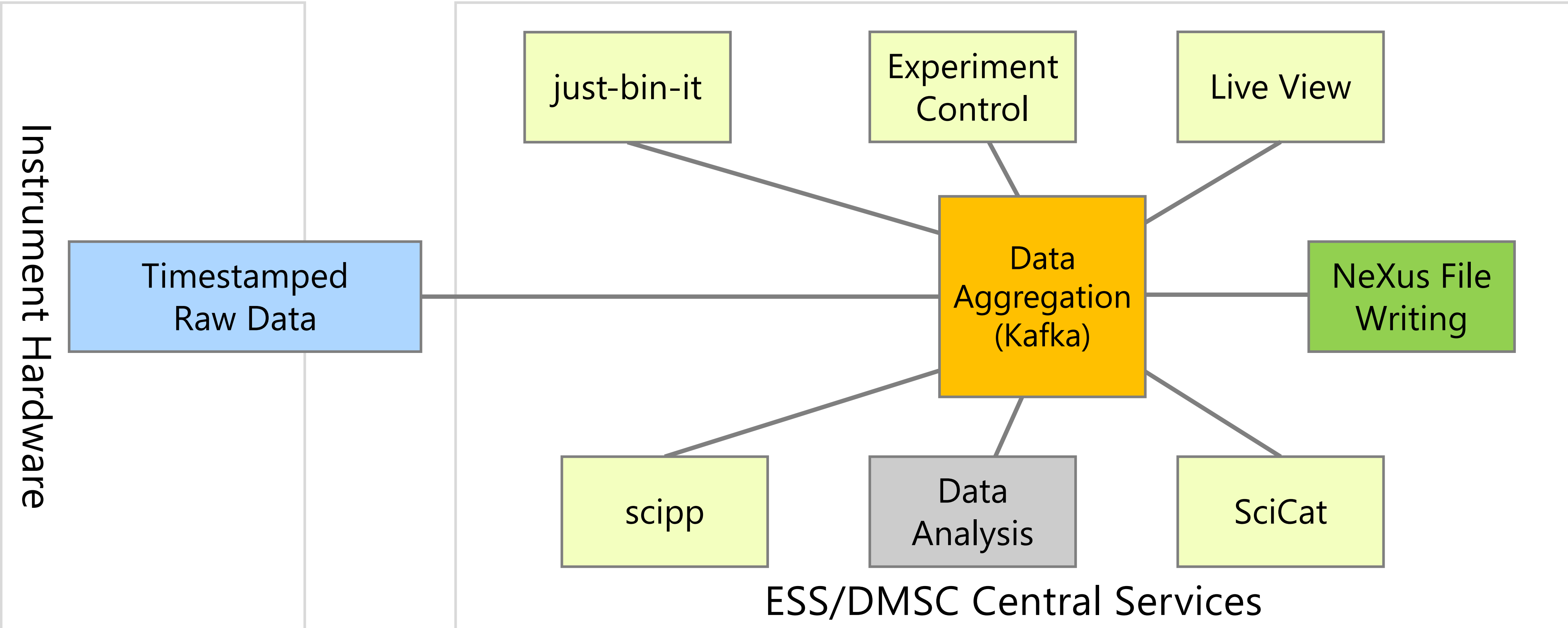
DAQ: Modular Setup

Central, scalable infrastructure with multiple services

- Controls and DAQ are (almost) completely separated
- File Writing makes no assumptions about control
- Neutron event data is matched to other information (choppers, motion, sample environment) via (asynchronous) timestamps
- Timestamping happens as close to the hardware as appropriate
- Forwarding of EPICS monitor data to Kafka is performed by one dedicated process per instrument
- File Writing is done by a pool of file writers not dedicated to any instrument
- Live histogramming and computing for feedback to the experiment happens via Kafka topics
- Data Processing, Analysis, etc can tap into the live data stream or be triggered by data lifecycle “events” and operate on data residing on disk

Downstream Data Architecture

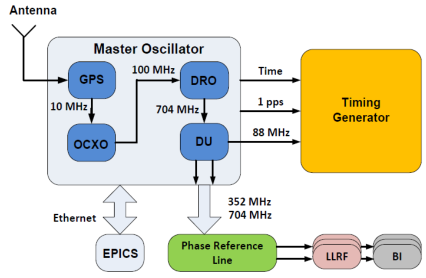
Data Flow Control and Processing



BrightnESS is funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 676548

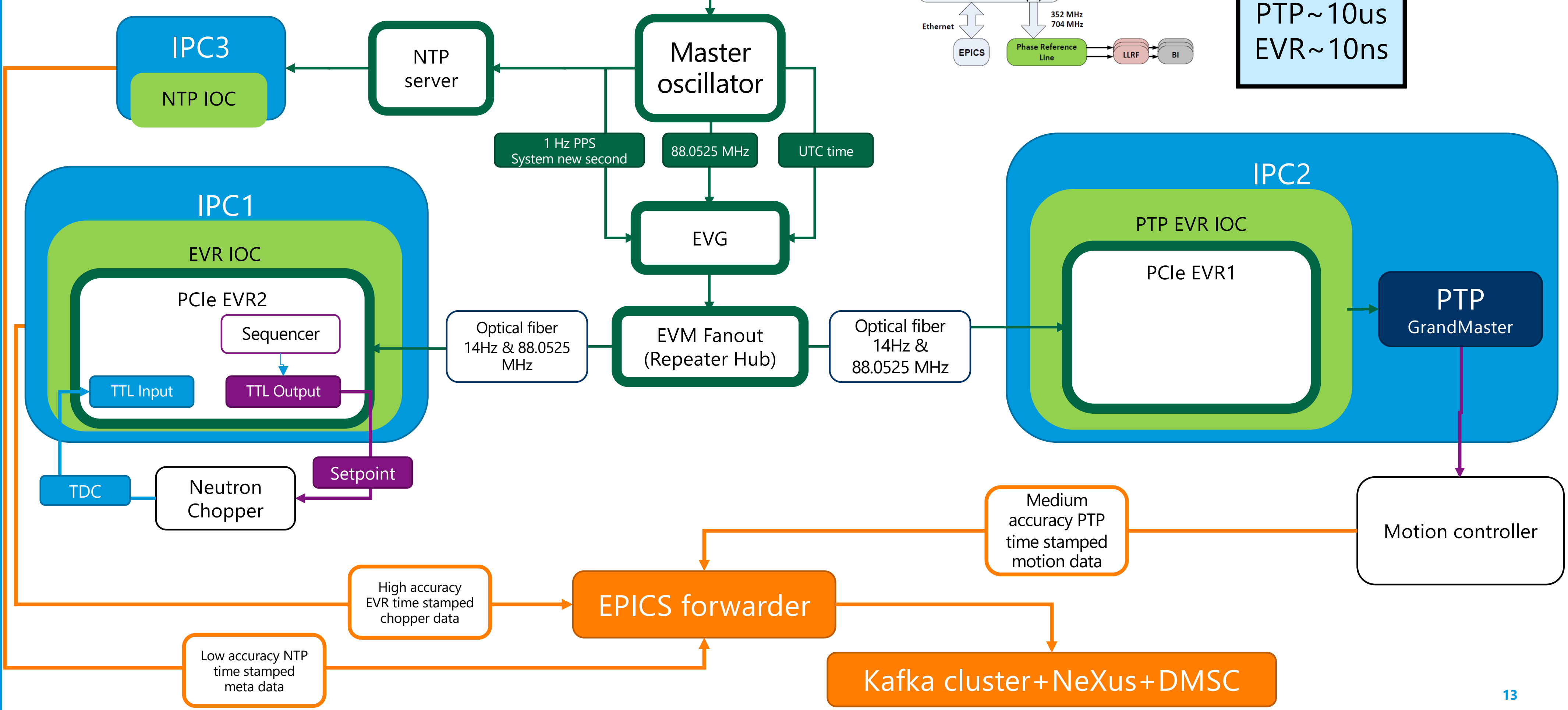
Timing system

One oscillator!



Accuracy

NTP~1ms
PTP~10us
EVR~10ns



Data Acquisition

Asynchronous and Timestamped

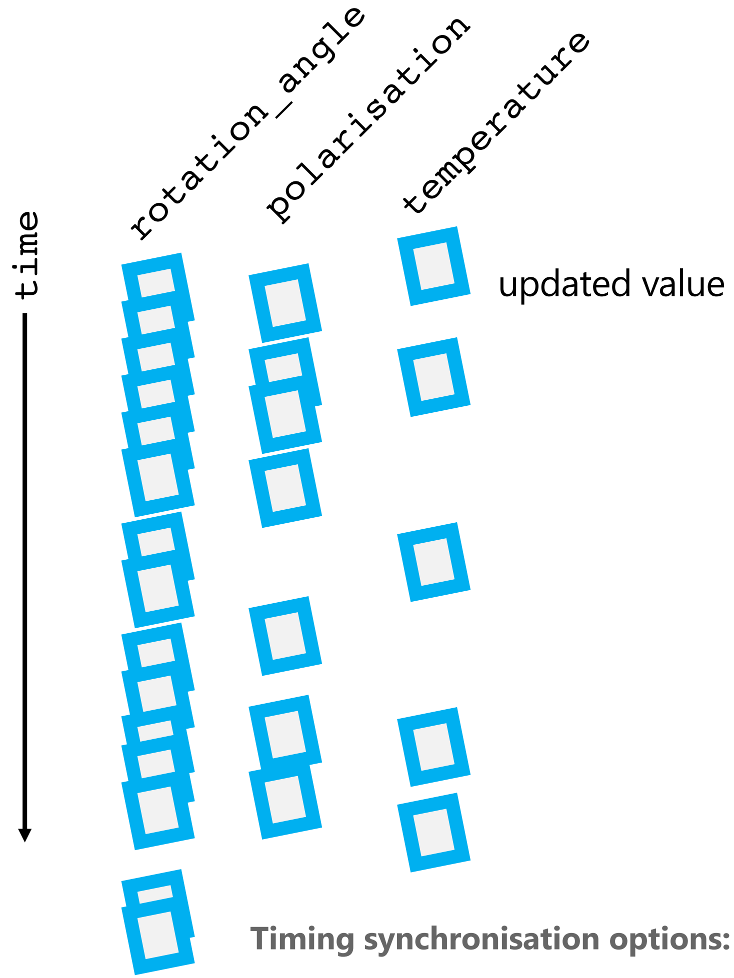
Every data source sends their information independently asynchronously:

- detectors & beam monitors & cameras
- choppers
- motion
- sample environment (temperature, pressure, fields)

Data only updated on change.

Flexible, sparse efficient storage, little hardware support needed, adequate timing synchronisation is critical.

Post processing needed in most cases.

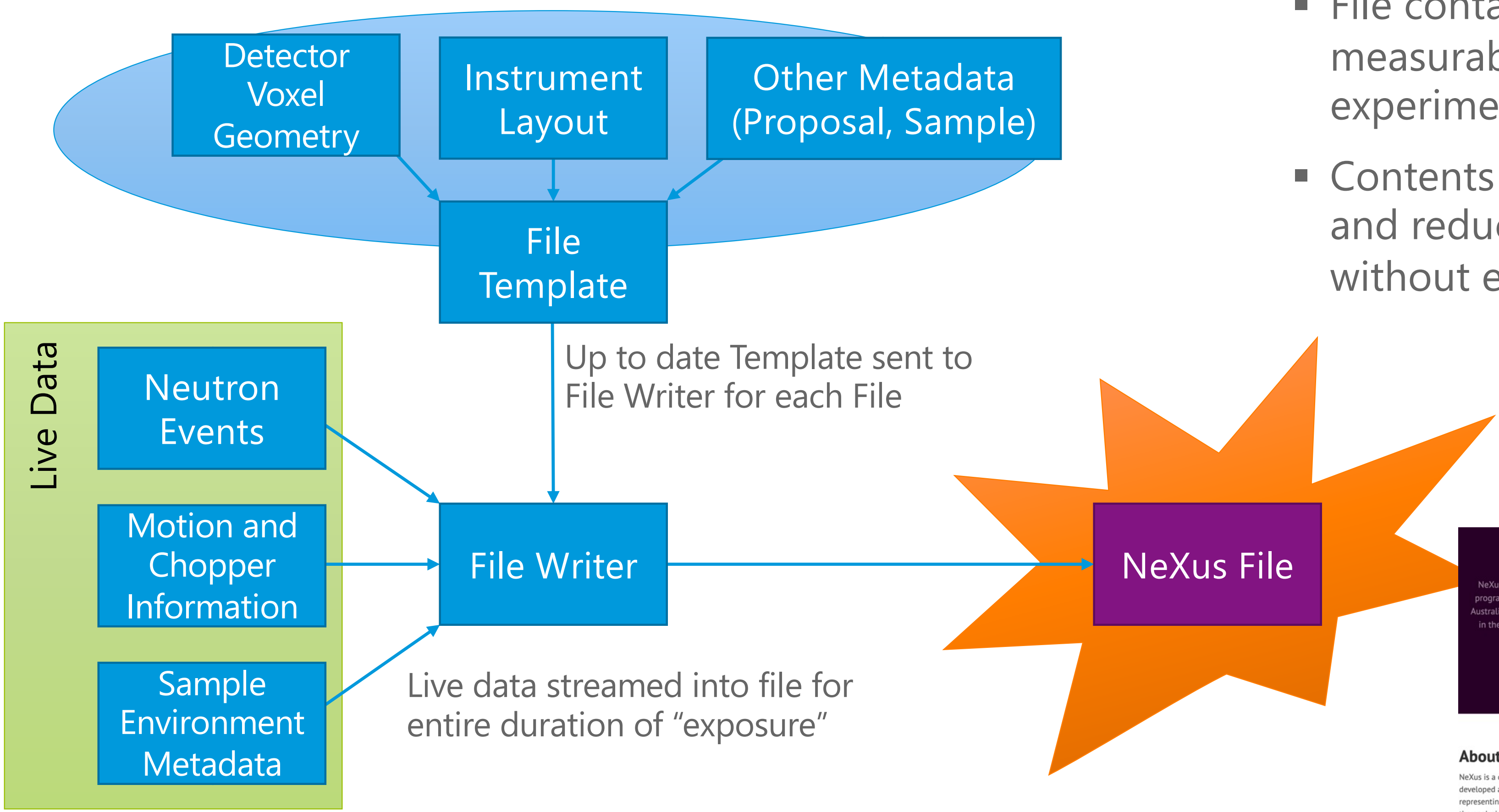


Timing synchronisation options:

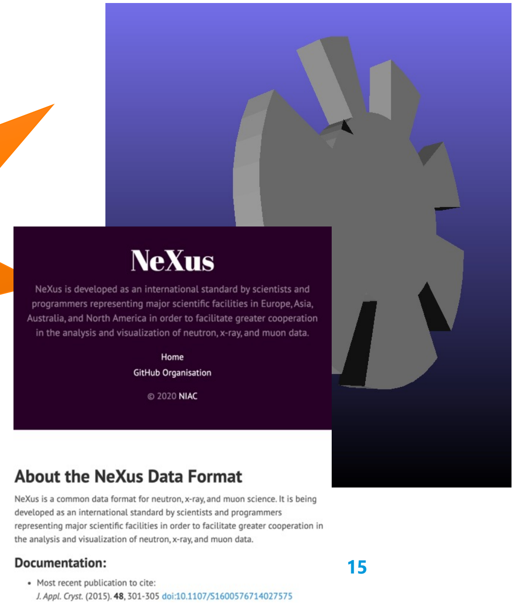
- **MRF EVR (Detectors, Choppers)**
- **PTP (Beckhoff: Motion, Sample Environment)**
- **NTP (Slow Sample Env, e.g. temperature)**

How We Write Files

Creating NeXus Templates “With Ease”



- File contains all pertinent measurable facts about the experiment
- Contents can be interpreted and reduced (processed) without external information



Conclusion

- Apache Kafka provides scalability and fault tolerance
- Modular setup allows scaling at various levels, if needed
- Interfaces between components defined
  - NeXus (data at rest)
  - NeXus mapped to JSON (scipp, data catalogue)
  - FlatBuffer schemas
- Data Sources can be added or Experiment Control Software be replaced with limited integration effort
- Kafka also enables "live" ("real-time") processing as well as triggering asynchronous operations. Both options are open to users or user code.
- scipp and various data analysis tools are provided and supported by the facility.
- Data Curation is performed by the facility. Policy guarantees long term storage and provides offline processing resources.
- Deployed at ANSTO (DAQ Pipeline) & PSI (Pipeline with NICOS)

Thank you