

How to best collaborate on DAQ Hardware / Software (Discussion)

Taku Gunji (QNSI/U-Tokyo)

Streaming DAQ and Computing milestones

2

FY25	FY26	FY27	FY28	FY29	FY30	FY31
PicoDAQ	MicroDAQ	MiniDAQ	Full DAQ-v-1	Production DAQ		DAQ
Streaming Orchestration			Streaming Challenges			
AI-Empowered Streaming Data Processing			Analysis Challenges			Computing
				Distributed Data Challenges		
AI-Driven Autonomous Calibration			AI-Driven Autonomous Alignment, Calibration, and Control			AI

Streaming DAQ Milestones and Deliverables

- FY26Q1: PicoDAQ: Readout test setups
- **FY26Q4: MicroDAQ: Readout detector data in test stand using engineering articles**
- **FY28Q1: MiniDAQ: Readout detector data using full hardware, timing chain, and orchestration systems**
- FY29Q2: Full DAQ-v1: Full functionality DAQ ready for full system integration & testing
- FY31Q3: Production DAQ: Ready for cosmics

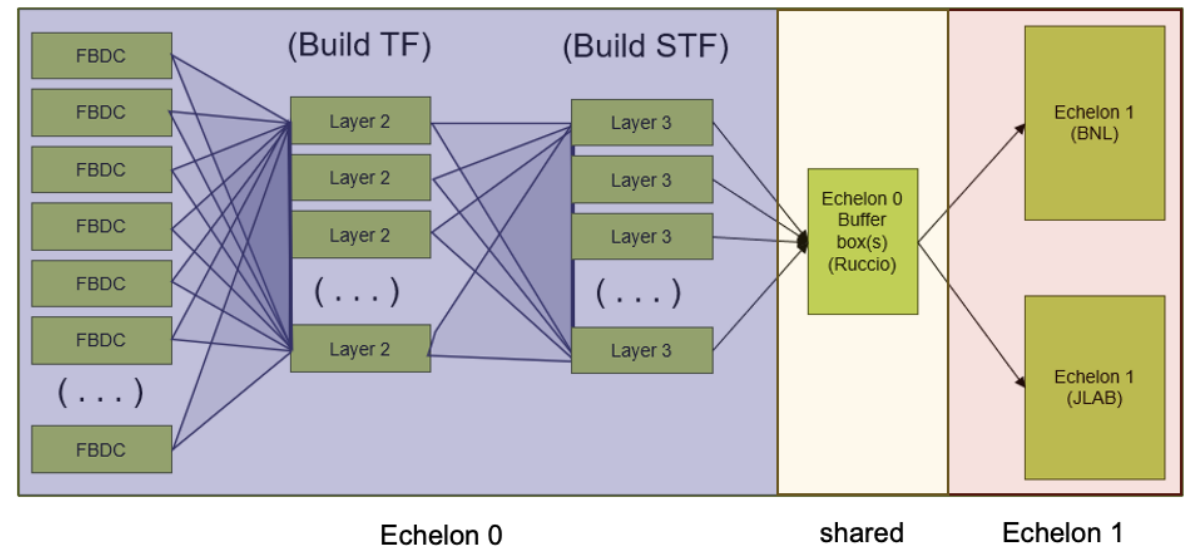
What do we need to connect E0-E1

3

Establish a middleware that seamlessly connects E0 and E1, enabling detector-independent streaming DAQ and distributed online computing.

Streaming DAQ Middleware

- Read out detector data and build detector payloads (e.g. GTU, FELIX)
- Construct TimeFrames based on heartbeat triggers
- Aggregate detector TimeFrames into Super TimeFrames (STFs)
- Buffer STFs/TFs at Echelon-0 (Sink)
- Transfer STFs/TFs to Echelon-1 for processing (asynchronous or synchronous)
- Interface to Run-Control, etc



Online Processing in E1 and E2

- STF Processing, Fast TF Processing
- Monitoring
- Interface to PanDA / Rucio / orchestration

Current Software and Candidate Roles

4

- ▶ **What we have now and those strong points/candidate roles:**
 - ▶ **RCDAQ / CODA** – Detector readout and hardware interface
 - ▶ **NestDAQ** – Streaming middleware (TimeFrame / STF transport)
 - ▶ **JANA2 / ERSAP** – Online processing (filtering, calibration, prompt reconstruction)
 - ▶ **PanDA / Rucio / Orchestration** – Distributed workflow and resource management

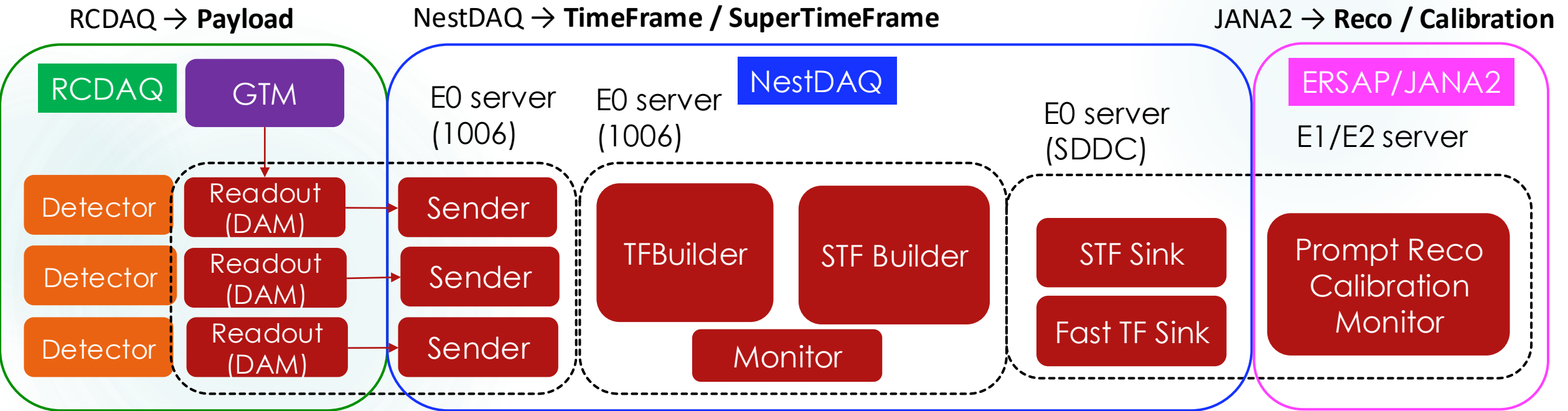
- ▶ **Motivation**

To realize the **Echelon-0 → Echelon-1 streaming DAQ architecture**, we need contributions from multiple software frameworks and detector groups.
Rather than replacing existing frameworks, we should **integrate their strengths**.

Example of the integration

5

► One possible integration of existing software frameworks



+ Run-Control, DB, Monitoring, Logbook, bookkeeping ...

Discussion Points

▶ Architecture

- ▶ Is this layering reasonable?
- ▶ Are the responsibilities of each layer well defined?
- ▶ What interfaces (or data formats) are needed between layers?

▶ Collaboration

- ▶ We will need discussions among experts to define the interfaces, data flow, and responsibilities of each software layer.
- ▶ Can we organize joint hands-on integration sessions at **BNL, JLab, or Japan** to integrate these software components into a common streaming DAQ system?
- ▶ If so: **When should we start? Who should participate? What infrastructure (hardware, test stands, networking, computing resources) is required?**