Streaming Computing Model: Status and Plans

Markus Diefenthaler (Jefferson Lab) on behalf of ePIC Software & Computing

ePIC Software & Computing

Coordinators and WG Conveners

Development

Dmitry Kalinkin




Simulation WG: Chao Peng, Sakib Rahman. **Reconstruction WG**: Derek Anderson, Shujie Li.

Infrastructure

Torre Wenaus



Streaming Computing WG: Marco Battaglieri, Taku Gunji, Jeff Landgraf.

Operations

Wouter Deconinck




Production WG: Thomas Britton, Sakib Rahman. **User Learning WG**: Stephen Kay, Holly Szumila-Vance.

Map

Attending in person

Review Charge Questions

1. **ePIC Computing and Software**: Assess the progress towards TDR readiness and implementing the ePIC computing model. As appropriate, assess the plans and outcomes for data challenges and other prototypes/testbeds. Assess the interfaces between the EIC detector DAQ and the ePIC computing and software organization as appropriate.

This Talk

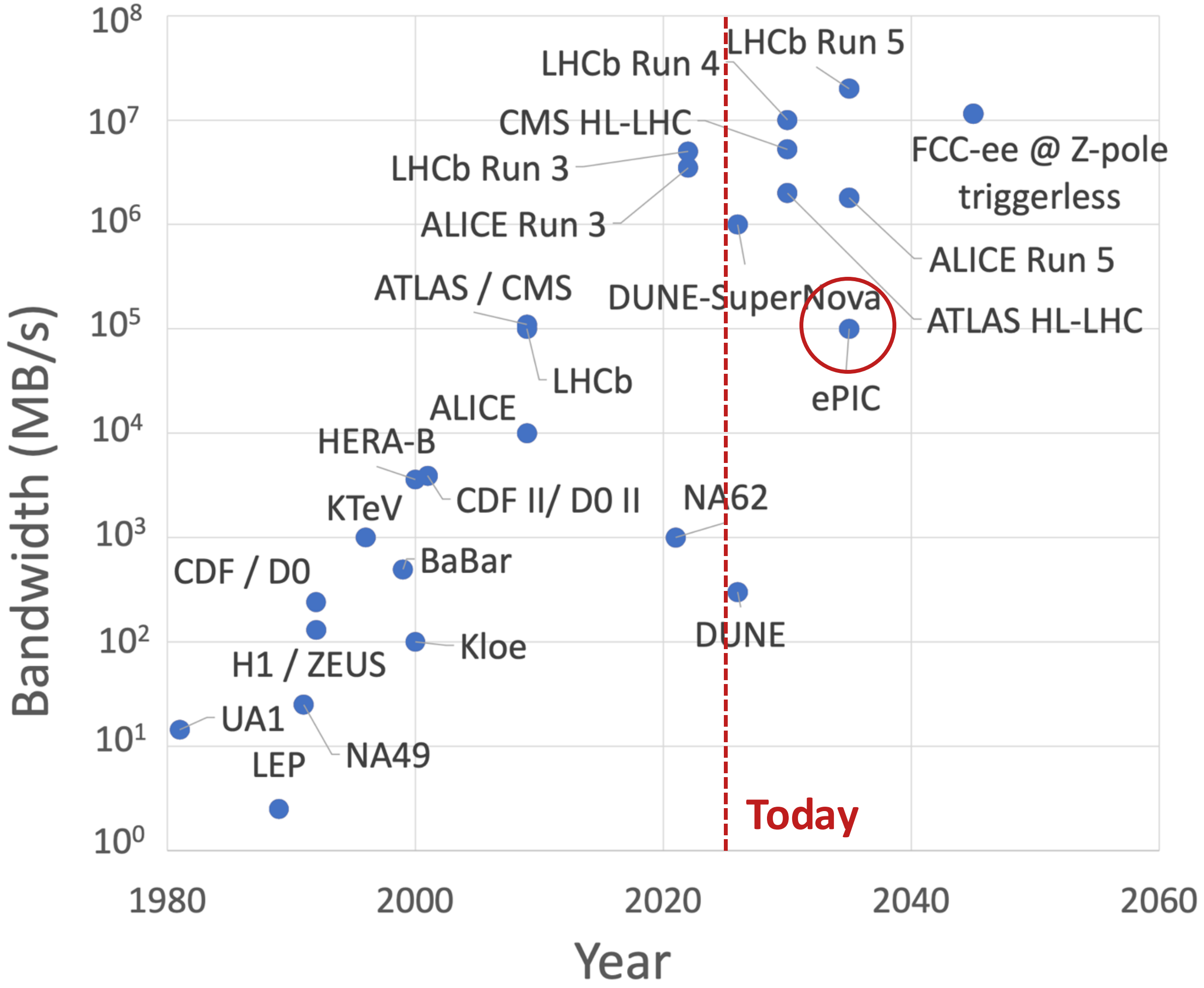
2. **Resources**: Assess the short-term and long-term resource planning relative to the near-term outcomes and long-term goals. As appropriate, comment on the delivery of resources by international partners.

3. **ECSJI**: Evaluate short and long terms support resource planning and delivery by the host laboratories through the ECSJI. As appropriate, evaluate the collaboration and division of responsibilities between the two labs.

4. **Broader Community**: Assess as appropriate engagements with aligned Software and Computing organizations across nuclear physics and particle physics communities and other relevant communities.

5. **Overall**: Assess the ePIC and ECSJI communication and collaboration. Assess the response to previous review recommendations.

ePIC Within the Global Particle Physics Experiments Landscape



Streaming Readout

Data rate of up to 100 Gbit/s

after low-level data reduction in the Streaming DAQ

Aarrestad, Thea, and Dorothea vom Bruch. *Trigger and Data Acquisition: Challenges and Perspectives*. Presentation at the Open Symposium on the European Strategy for Particle Physics, Venice, Italy, June 23, 2025. https://agenda.infn.it/event/44943/contributions/265988/

Science Drivers for Streaming Readout at ePIC

Broad ePIC Science Program:

- Plethora of observables, with less distinct topologies where every event is significant.

Moderate Signal Rate:

	EIC	RHIC	LHC → HL-LHC
Collision species	$\vec{e}+\vec{p}, \vec{e}+A$	$\vec{p}+\vec{p}/A, A+A$	$p+p/A, A+A$
Maximum x-N C.M. energy	140 GeV	510 GeV	13 TeV
Peak x-N luminosity	10^{34} cm^{-2} s^{-1}	10^{32} cm^{-2} s^{-1}	$10^{34} \to 10^{35}$ cm^{-2} s^{-1}
x-N cross section	**50 µb**	**40 mb**	**80 mb**
Maximum collision rate	500 kHz	10 MHz	1-6 GHz
$dN_{ch}/d\eta$	0.1-Few	~3	~6
Charged particle rate	**4×10^6 N_{ch}/s**	**6×10^7 N_{ch}/s**	**3×10^{10}+ N_{ch}/s**

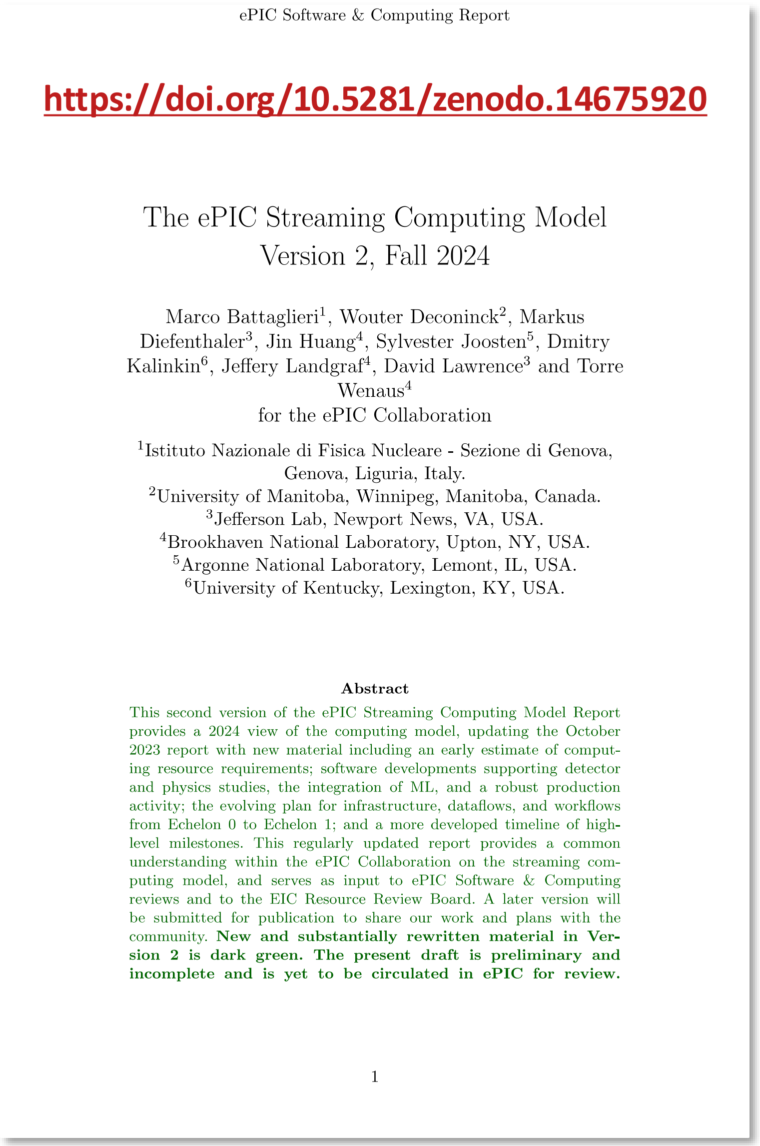
Enabling Next-Generation Compute-Detector Integration

- **Maximize Science**: **Capture every collision signal, including background.**
  - **High-precision measurements**: Control of systematic uncertainties is critical.
  - Event selection using all available detector data for **holistic reconstruction**:
    - **Eliminate trigger bias** and provide accurate estimation of uncertainties during event selection.
  - Streaming background estimates ideal to **reduce background** and related systematic uncertainties.
- **Accelerate Science**: **Rapid turnaround of two weeks for data for physics analyses**.
  - Timeline driven by alignment and calibration.
  - Subsystem experts indicate a two-week turnaround is feasible.
- **Technologies**: Compute-detector integration using:

Streaming Readout for continuous data flow of the full detector information.	**Artificial Intelligence** for rapid processing (autonomous alignment, calibration, and validation).	**Heterogeneous Computing** for acceleration (CPU, GPU).

The ePIC Streaming Computing Model

Included in Indico review materials


ePIC Software & Computing Report

https://doi.org/10.5281/zenodo.14675920

The ePIC Streaming Computing Model
Version 2, Fall 2024

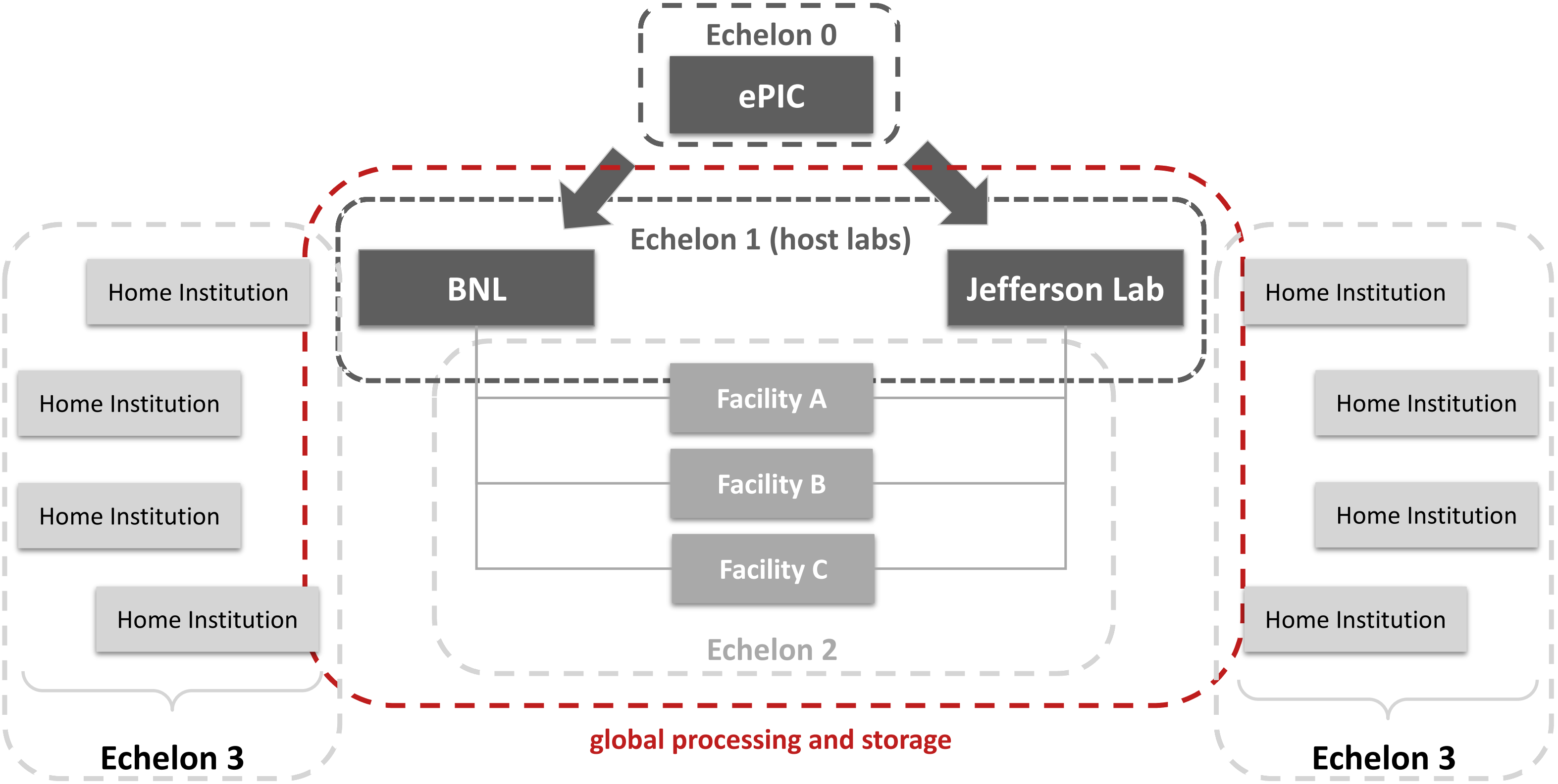
Marco Battaglieri[1], Wouter Deconinck[2], Markus Diefenthaler[3], Jin Huang[4], Sylvester Joosten[5], Dmitry Kalinkin[6], Jeffery Landgraf[4], David Lawrence[3] and Torre Wenaus[4]
for the ePIC Collaboration

[1]Istituto Nazionale di Fisica Nucleare - Sezione di Genova, Genova, Liguria, Italy.
[2]University of Manitoba, Winnipeg, Manitoba, Canada.
[3]Jefferson Lab, Newport News, VA, USA.
[4]Brookhaven National Laboratory, Upton, NY, USA.
[5]Argonne National Laboratory, Lemont, IL, USA.
[6]University of Kentucky, Lexington, KY, USA.

Abstract

This second version of the ePIC Streaming Computing Model Report provides a 2024 view of the computing model, updating the October 2023 report with new material including an early estimate of computing resource requirements; software developments supporting detector and physics studies, the integration of ML, and a robust production activity; the evolving plan for infrastructure, dataflows, and workflows from Echelon 0 to Echelon 1; and a more developed timeline of high-level milestones. This regularly updated report provides a common understanding within the ePIC Collaboration on the streaming computing model, and serves as input to ePIC Software & Computing reviews and to the EIC Resource Review Board. A later version will be submitted for publication to share our work and plans with the community. **New and substantially rewritten material in Version 2 is dark green. The present draft is preliminary and incomplete and is yet to be circulated in ePIC for review.**

1



We developed the ePIC Streaming Computing Model to accelerate the pace of discovery and enhance scientific precision through improved management of systematic uncertainties. The model is documented in a detailed report and was reviewed during the 2023 and 2024 ESCAC reviews.

Computing Resource Needs (2034) and Their Implications

Processing by Use Case [cores]	Echelon 1	Echelon 2
Streaming Data Storage and Monitoring	-	-
Alignment and Calibration	6,004	6,004
Prompt Reconstruction	60,037	-
First Full Reconstruction	72,045	48,030
Reprocessing	144,089	216,134
Simulation	123,326	369,979
Total estimate processing	**405,501**	**640,147**

Storage Estimates by Use Case [PB]	Echelon 1	Echelon 2
Streaming Data Storage and Monitoring	71	35
Alignment and Calibration	1.8	1.8
Prompt Reconstruction	4.4	-
First Full Reconstruction	8.9	3.0
Reprocessing	9	9
Simulation	107	107
Total estimate storage	**201**	**156**

O(1M) core-years to process a year of data:

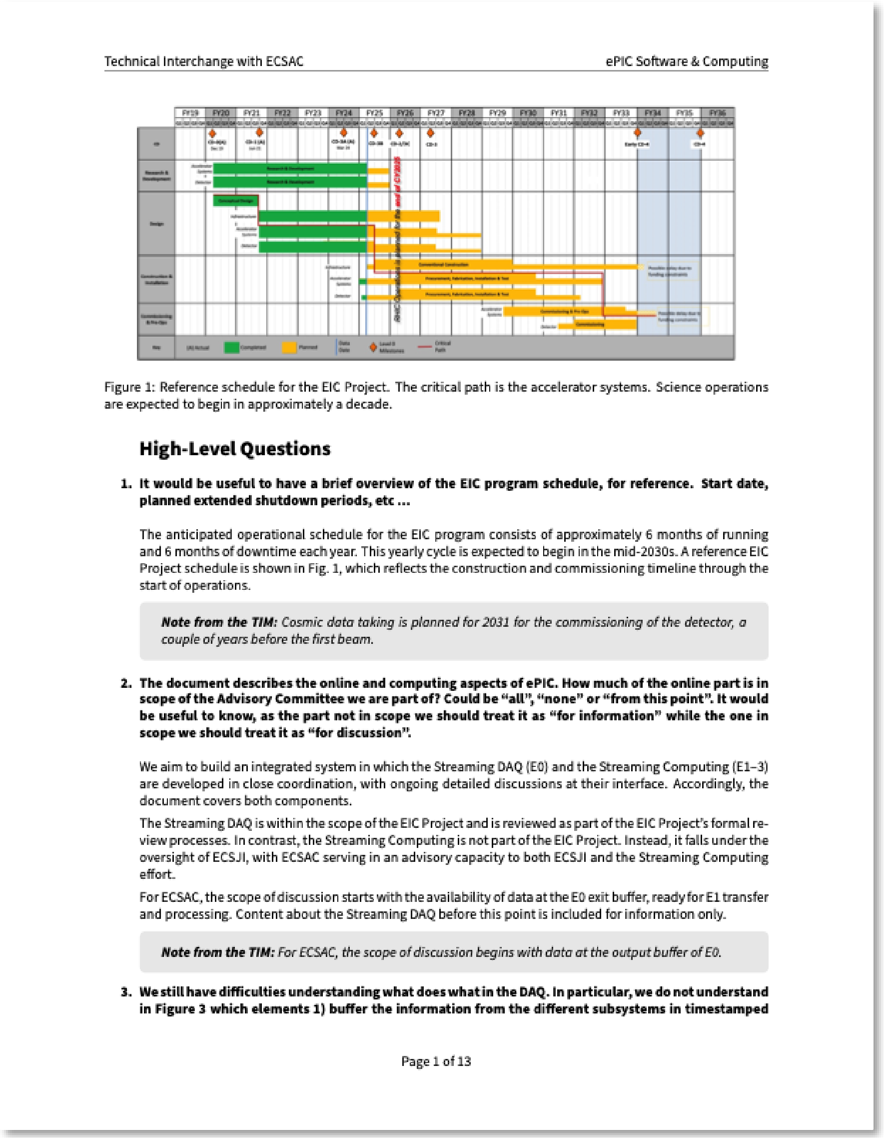
- Even with performance gains over the years, the required processing scale remains substantial.
- Highlights the need to leverage distributed and opportunistic resources from the outset.

~350 PB to store data of one year.

We presented our **compute resource needs in the 2024 ECSAC review.** ePIC is a compute-intensive experiment. Its science must not be limited by computing constraints.

Technical Interchange Meeting

Included in Indico review materials



Technical Interchange with ECSAC ePIC Software & Computing

Figure 1: Reference schedule for the EIC Project. The critical path is the accelerator systems. Science operations are expected to begin in approximately a decade.

High-Level Questions

1. **It would be useful to have a brief overview of the EIC program schedule, for reference. Start date, planned extended shutdown periods, etc ...**

   The anticipated operational schedule for the EIC program consists of approximately 6 months of running and 6 months of downtime each year. This yearly cycle is expected to begin in the mid-2030s. A reference EIC Project schedule is shown in Fig. 1, which reflects the construction and commissioning timeline through the start of operations.

   Note from the TIM: *Cosmic data taking is planned for 2031 for the commissioning of the detector, a couple of years before the first beam.*

2. **The document describes the online and computing aspects of ePIC. How much of the online part is in scope of the Advisory Committee we are part of? Could be "all", "none" or "from this point". It would be useful to know, as the part not in scope we should treat it as "for information" while the one in scope we should treat it as "for discussion".**

   We aim to build an integrated system in which the Streaming DAQ (E0) and the Streaming Computing (E1–3) are developed in close coordination, with ongoing detailed discussions at their interface. Accordingly, the document covers both components.

   The Streaming DAQ is within the scope of the EIC Project and is reviewed as part of the EIC Project's formal review processes. In contrast, the Streaming Computing is not part of the EIC Project. Instead, it falls under the oversight of ECSJI, with ECSAC serving in an advisory capacity to both ECSJI and the Streaming Computing effort.

   For ECSAC, the scope of discussion starts with the availability of data at the E0 exit buffer, ready for E1 transfer and processing. Content about the Streaming DAQ before this point is included for information only.

   Note from the TIM: *For ECSAC, the scope of discussion begins with data at the output buffer of E0.*

3. **We still have difficulties understanding what does what in the DAQ. In particular, we do not understand in Figure 3 which elements 1) buffer the information from the different subsystems in timestamped**

Page 1 of 13

Purpose:
- Discussed the ePIC (Streaming) Computing Model paper, as presented to the EIC Computing & Software Advisory Committee (ECSAC) in 2023 and 2024,

Outcome:
- Documented ECSAC questions, ePIC responses, and comments made during the meeting (13 pages).
- Meeting helped improve the manuscript by clarifying key points and addressing inconsistencies.

Key Discussion Points:
- **ECSAC Role**: Scope of the ECSAC review begins with data at the output buffer of E0.
- **Interface Between E0 and E1**: Discussion focused on data aggregation, transfer, and low-level processing.
- **Echelon 2**: Discussed role of E2 in the distributed computing model.
- **Analysis Infrastructure**: Discussed analysis model involving E1, E2, E3 sites.

Prototyping Streaming Computing: Requirements and Testbed Development

Computing Model and E0—E1 Dataflow and Processing

We have our ePIC Streaming Computing Model documented and an evolving conception of E0-E1 dataflow and workflows (see Technical Interchange Meeting), developed in an active Streaming Computing WG meeting series.

Now moving from reports and schematics to concrete testbed developments.

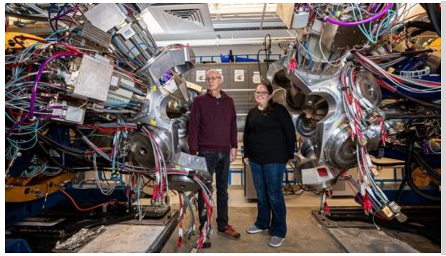
- Emphasis is now is moving from reports and schematics to the specifics.
- Prototyping ideas and tools in testbeds guided by requirements.

Lessons Learned From Other Experiments

The streaming readout and computing developments at Jefferson Lab and LHCb have been discussed in detail during the **Streaming Readout I–XII** workshop series and have informed our design. In particular, the pioneering work by **LHCb** has been a **major inspiration from the beginning**. We have discussed these topics in detail previously, so for this Lessons Learned session, we focused on projects that had not yet been covered—except for an updated presentation on ERSAP from Jefferson Lab.

ALICE O2 (Piotr Konopka, Lubos Krcal)

- State machines are central to orchestration but require careful design and maintenance.
- Mixed experiences highlight the importance of robust implementation.
- *"Software is not a croissant – to be baked, eaten, and forgotten."*
- Continuous maintenance and lifecycle planning are essential to avoid abandonware.



GRETA (Mario Cromaz)

- Early prototyping is vital.
- Effective orchestration is required for system testing and performance.
- A dedicated local, mobile compute cluster was chosen for reliability and portability.

Vera C. Rubin Observatory (Andy Hanushevsky, Wei Yang)

- Synchronization tools are necessary when integrating different data management systems.
- Infrastructure should be designed with security constraints in mind from the start.
- Kubernetes provides flexibility but adds significant complexity and maintenance overhead.

Requirements for Streaming Orchestration

Included in Indico review materials

Requirements for an ePIC Distributed Workflow Management System

Markus Diefenthaler, Torre Wenaus
and the ePIC Streaming Computing Model Working Group

Version 1.0 – September 2025

Contents

1

Requirements Document

- Builds upon and guides further development of the **ePIC Streaming Computing Model.**
- Developed collaboratively by the **ePIC Streaming Computing Model WG**.
- Informed by **lessons learned from other experiments** and streaming systems.

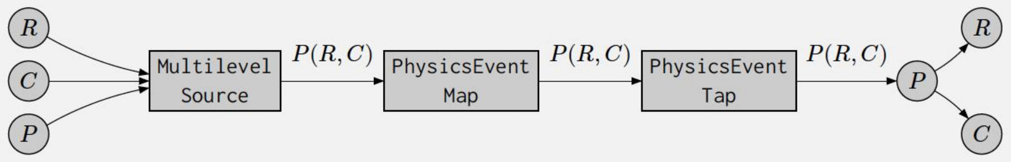
Key Themes

- **Scalable and Automated Workflows**: Low overhead, automated orchestration, and real-time processing across E1–2.
- **Streaming-First Design**: Native support for near real-time processing.
- **Integrated Data Management**: Tight coupling with Rucio-based DDM for data-driven workflows and provenance tracking.
- **Flexible & User-Centric Interfaces**: CLI, REST, and web interfaces with support for custom dashboards and diagnostics.
- **Robust Monitoring & Resilience**: Real-time analytics, fault tolerance, and automated recovery mechanisms.
- **Community and Documentation Focus**: Open development, transparent processes, and collaborative design.

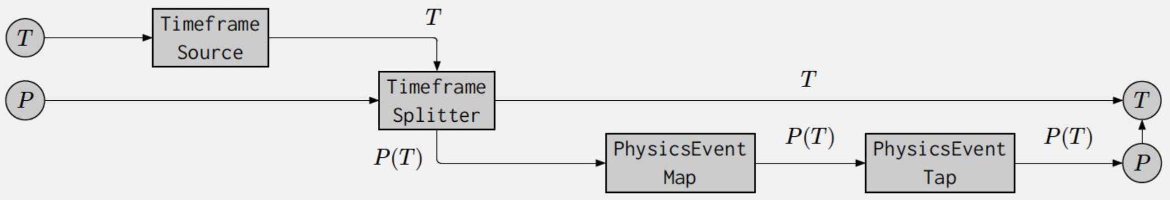
JANA2 for Streaming Processing

- Multithreaded JANA2 framework provides a component-level hierarchical decomposition of data boundaries into **Run**, **Timeframe**, **PhysicsEvent**, and **Subevent** levels. This is essential for streaming processing.
- The **Folder** and **Unfolder** component interfaces enable traversal of this hierarchy by supporting operations such as splitting and merging data streams. This functionality has been tested and validated within **EICrecon**.

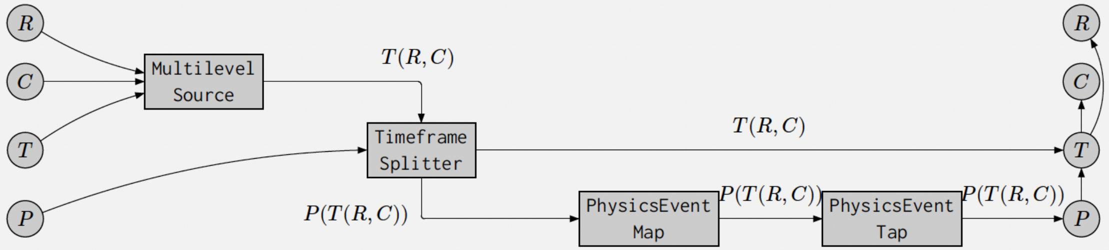
Introducing multilevel sources



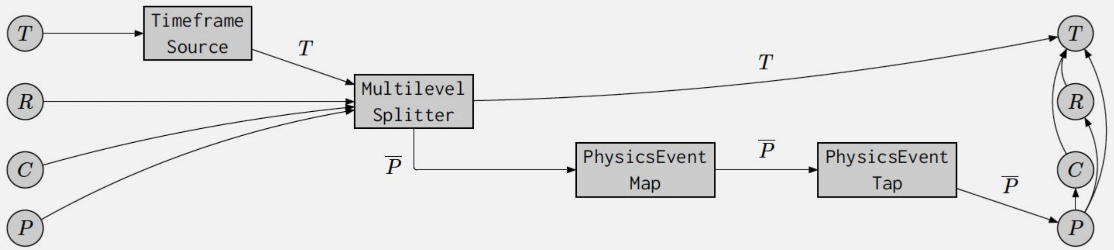
EICrecon timeframe splitting



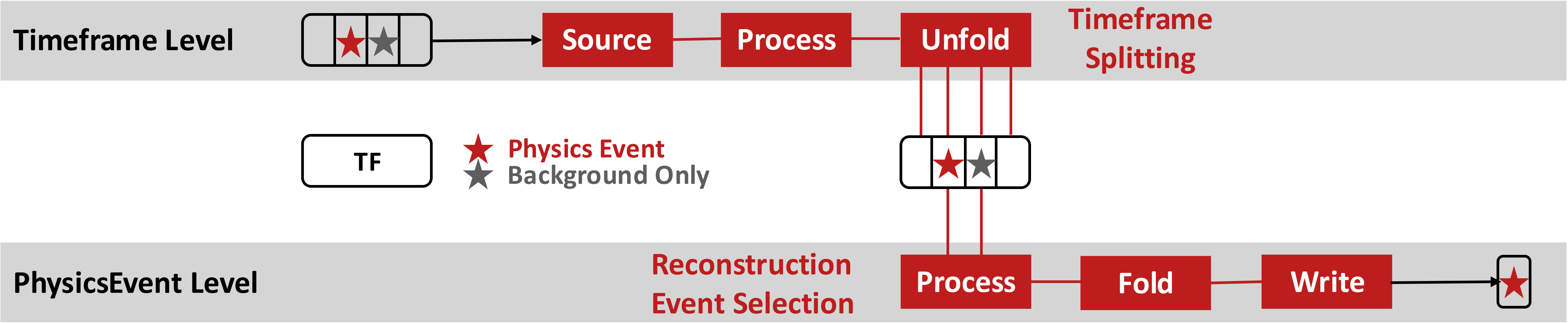
Multilevel sources with timeframe splitting

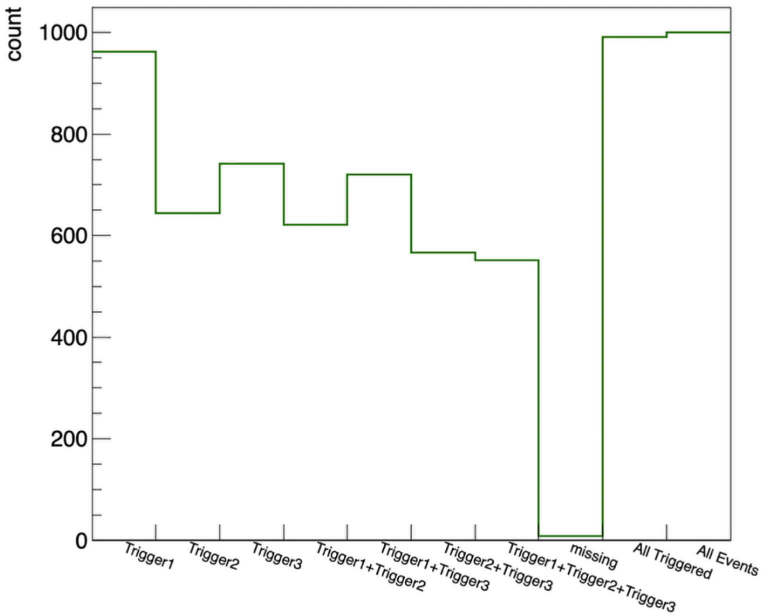


Timeframe sources with multilevel splitting



Event-Building from Streaming Data in JANA2





Trigger 1: Coincidence hits in the SVT Endcap and Forward MPGD Endcap.
Trigger 2: Coincidence hits in the TOF Barrel and MPGD Barrel.
Trigger 3: Coincidence hits in the SVT Endcap and Backward MPGD Endcap.

A straightforward event selection based on raw simulated hits in the SVT, MPGD, and TOF detector systems achieves greater than 99 % efficiency and less than 1 % background in identifying physics events in TFs.

Testbed for Streaming Orchestration

Motivation:
- Evaluate how well existing distributed computing tools support streaming orchestration.
- Focus on practical deployment and performance in realistic environments.

- **Design Precepts**:
  - Robust geographical distribution across real-world networks
  - Full automation of data processing workflows
  - Complete exposure of system status and operational analytics
- **Approach**:
  - PanDA and Rucio align with the stated design precepts.
  - Both are deployed in live testbed instances at BNL:
    - Other sites can participate in collaborative testing and development: https://github.com/BNLNPPS/swf-testbed
  - Assume that data is delivered in STFs, each consisting of 1000 aggregated TFs, with a size of ~2 GB at a rate of ~1 Hz.
- **Streaming in Action (Testbed Observations)**:
  - Each STF contains approximately 45,000 events and takes approximately 19 hours to process on a single serial core
  - This latency is too high for timely detector status feedback
  - STF data is distributed across multiple workers at sub-file granularity
  - Sub-file fan-out and parallel processing enable true streaming behavior in the testbed
  - Data-driven logic automatically triggers E1 transfers and prompt processing upon file appearance.

System Architecture: Simulated DAQ with Distributed Agents and Monitoring

A **set of collaborating agents** communicate via ActiveMQ and form the core of the system:
- **Data handling agent**: executes Rucio actions based on data and conditions.
- **Processing agent**: executes PanDA actions based on available data and conditions.
- **Monitoring agent**: skims STF data for fast feedback.

The **driver simulates the DAQ** and other external components to evaluate their impact:
- Single agent instance models a real DAQ system.
- Simulation templates define and track performance and conditions.
- System state evolves over time following a state machine.

A **monitoring backend** with a database aggregates and exposes system state:
- Browser-based UI and REST API access (authenticated)
- LLM integration (via MCP)

Testbed Objectives: ePIC Streaming Workflow Development

Preview of current capabilities and development direction:

Data delivery via STFs from DAQ:
- Automated workflows transfer data to E1s for archiving and prompt processing.
- Rucio/FTS-based DAQ-to-E1 data flows are functional and being integrated.

Low-latency streaming workflows for continuous accelerator and detector monitoring:
- Fast monitoring skims incoming STFs (implemented in simulation workflows).
- Brainstorming use of PanDA’s fine-grained processing (event service, iDDS).
- Exploring applicability to E2s as well as E1s.

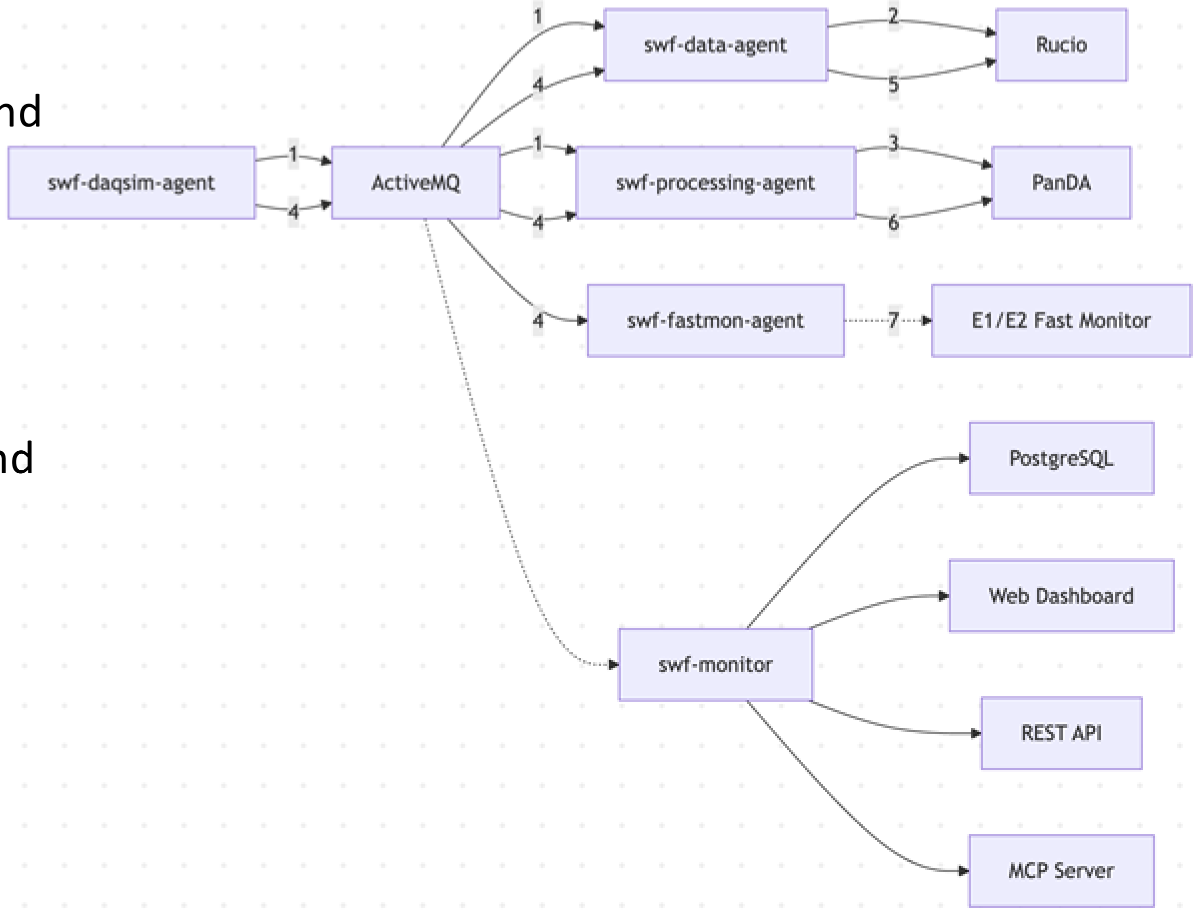
Detector/data state model tracks stream contents in real time:
- Initial version of the model is in place.

Calibration workflow orchestration alongside collision data:
- Planned for future development.

End-to-End Workflow: From Run Start to Monitoring

Workflow driven by DAQ simulator and 3 agents communicating via ActiveMQ.
All system activity and state recorded in the database via REST and displayed in the monitor.

1. **Run Start** DAQ simulator broadcasts a message signaling the start of a new data-taking run (**working**).
2. **Dataset Creation** The data agent receives the message and instructs Rucio to create a dataset for the run (**working**).
3. **Processing Task** Processing agent sets up a PanDA task based on the run start message (**working**).
4. **STF Available** DAQ simulator broadcasts availability of a new STF file; this continues while the run is active.
5. **STF Transfer** The data agent triggers Rucio registration and transfers the STF to E1 storage (**not yet integrated**).
6. **STF Processing** PanDA detects the new STF at E1, transferred via Rucio, and launches jobs to process it (**working**).
7. **Fast Monitoring** The fastmon agent sees the STF broadcast, performs a partial read, and injects a data sample into the E1/E2 monitoring stream (**fast monitor emulation working based on STF metadata**).



State Machine: Stream-Oriented Workflow States and Substates

States

- no_beam
  - Collider not operating
- beam
  - Collider operating
- run
  - Physics running
- calib
  - Dedicated calibration period
- test
  - Testing, debugging
  - Any substates can be present during test

Substates

- not_ready
  - detector not ready for physics datataking
  - occurs during states: no_beam, beam, calib
- ready
  - collider and detector ready for physics, but not declared as good for physics
  - when declared good for physics, transitions from beam/ready to run/physics
  - occurs during states: beam
- physics
  - collider and detector declared good for physics
  - if collider or detector drop out of good for physics, state transitions out of 'run' to 'beam' or 'off'
  - occurs during states: run
- standby
  - collider and detector still good for physics, but standing by
  - occurs during states: run
- lumi
  - detector, machine data that is input to luminosity calculations
  - occurs during states: beam, run
- eic
  - machine data, machine configuration
  - occurs during states: all
- epic
  - detector configuration, data
  - occurs during states: all
- daq
  - info, config transmitted from DAQ
  - occurs during states: all
- calib
  - a catch-all for a great many calib data types, we can start small
  - occurs during states:

The baseline workflow exercises nearly all states, with the exception of detector and machine configurations.

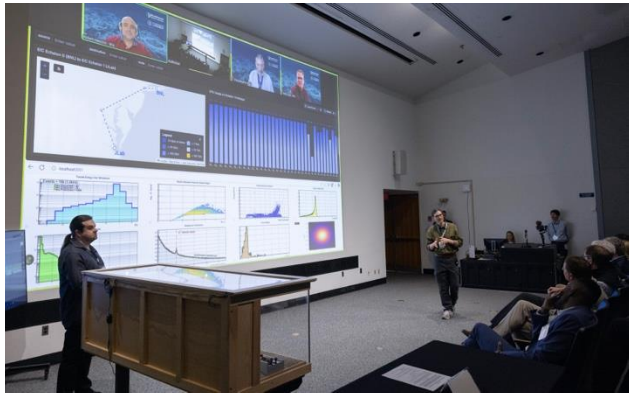
Fast Detector Feedback via PanDA Fine-Grained Processing

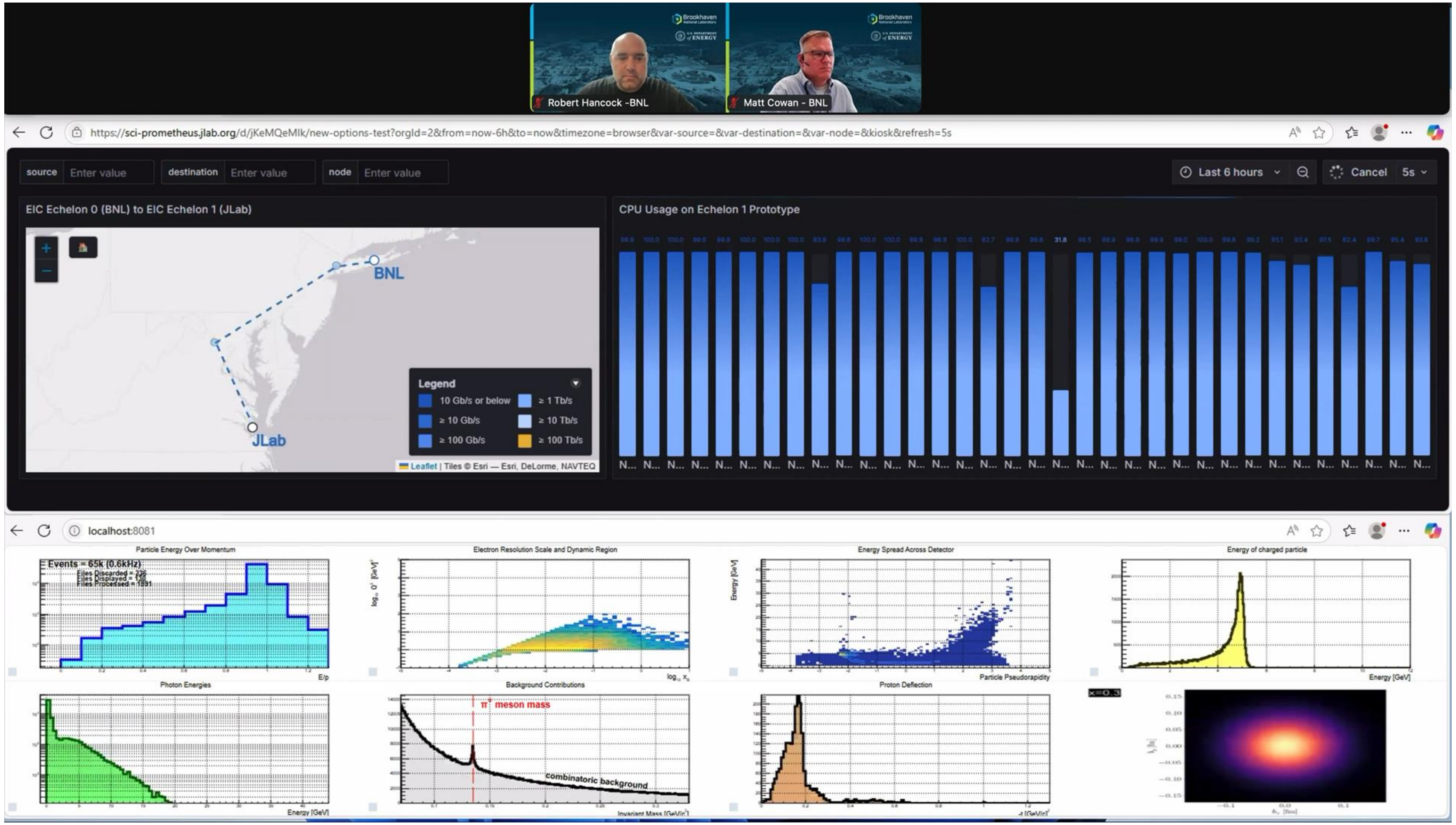
Objective: Enable E1-based monitoring to display near-live detector status, starting immediately after run start

- Rucio STF transfers take minutes and are not used in this fast sampling path.
- STFs arrive from DAQ every ~0.5 seconds, each containing ~45,000 events.
  - Rapid sampling is required to support quick feedback on high-statistics reconstructions.
  - A speculative target: process one STF (~45k events) every 30 seconds (vs. 19 hours serially).
- This is the role of the STF-sample-based fast monitoring stream from the fastmon agent.
- STF samples are fanned out in near real-time across E1 workers.
  - Each worker processes fractions of hundreds of STFs during its batch lifetime.
- PanDA's fine-grained processing services were built for this model:
  - **Event Service and iDDS** (intelligent Distributed Dispatch and Scheduling)
  - Workers act as semi-persistent agents accepting fine-grained tasks.
  - iDDS assigns STF samples from fastmon to workers.
  - PanDA handles bookkeeping and retry mechanisms.
- This is the **next major priority** for the testbed.

Streaming Data Processing Demonstration

Secretary Wright's Visit to Jefferson Lab on August 21





We successfully showcased **real-time data transfer from BNL to JLab** and its **processing at JLab**. It also prompted collaboration with BNL to test current tools and establish a clear network path.

Streaming Data Processing Demonstration

- **Pipeline with filesystem communication**

- **EJFAT** is a joint project between Jefferson Lab, ESnet, ASCR, and NP to enable **real-time experimental data processing via high-speed networks.**
  - It intelligently redirects data packets in transit to optimal compute nodes, dynamically managing resources and ensuring full detector views per time slice.
  - Sophisticated load balancer logic can be implemented to automatically allocate and deallocate compute nodes.
  - The design foresees using AI agents to implement more complex redirection for applications requiring large and dynamic resources.

- The test used a **34-node compute farm** built from older hardware (2014–2018), providing approximately 1,000–1,500 logical cores.

Prototyping Ideas and Tools in Testbeds

- Developed **requirements document for streaming orchestration**.
- **Testbed plans** are taking concrete shape:

  Streaming reconstruction: Raw data stream to collision event identification to reconstruction and analysis.

  Streaming orchestration: Developing E0-E2 streaming workflows in the testbed, utilizing Rucio and PanDA.

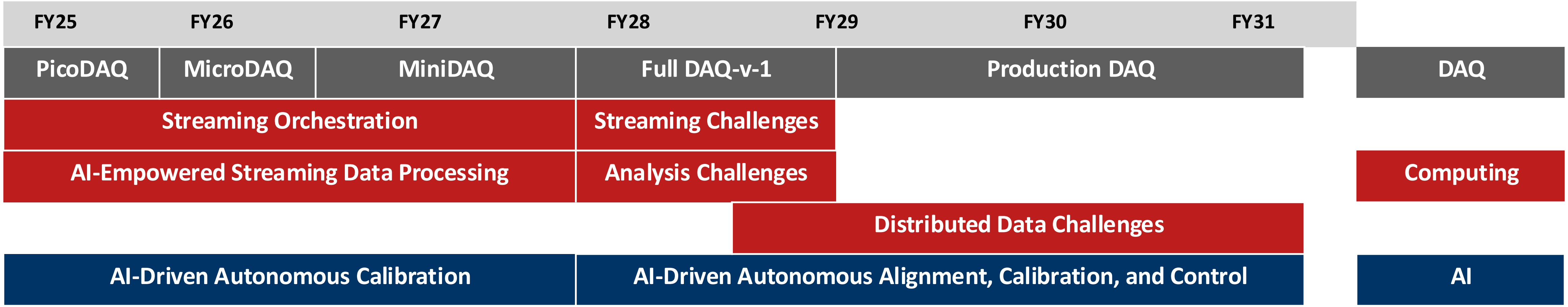
  Streaming processing: Developing E0-E2 streaming workflows using EJFAT.

- Not covered in this presentation but starting efforts:

  Streaming analysis: Demonstrate simulation data production streaming to E2 site.

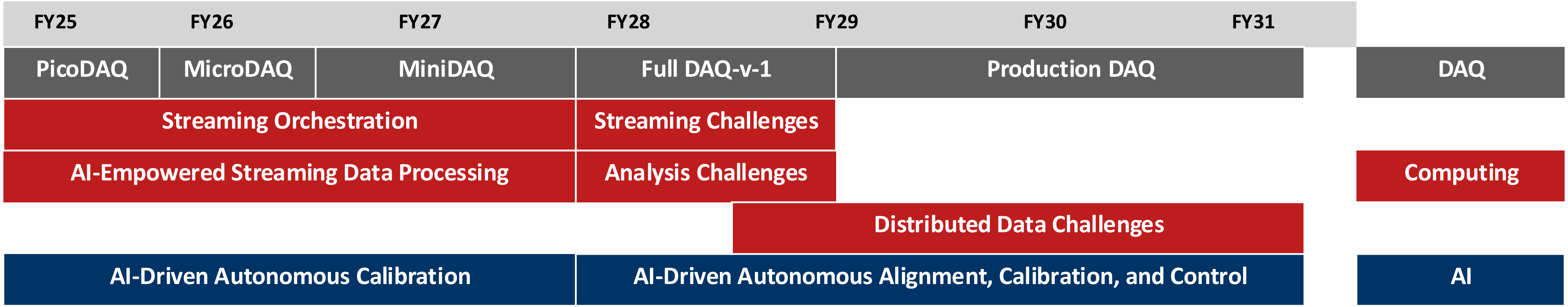
  Rapid data processing: Describing and executing complex calibration workflows with their dependencies.

Streaming DAQ and Computing Milestones



- **Compute-Detector Integration**:
  - Joint deliverables between **DAQ** and **computing** to develop integrated systems for detector readout, data processing, and ultimately physics analysis.
  - **Key role of AI(/ML)**: Empowering data processing and enabling autonomous experimentation and control.

Streaming DAQ and Computing Milestones



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 and timing chain
FY29Q2: Full DAQ-v1: Full functionality DAQ ready for full system integration & testing
FY31Q3: Production DAQ: Ready for cosmics

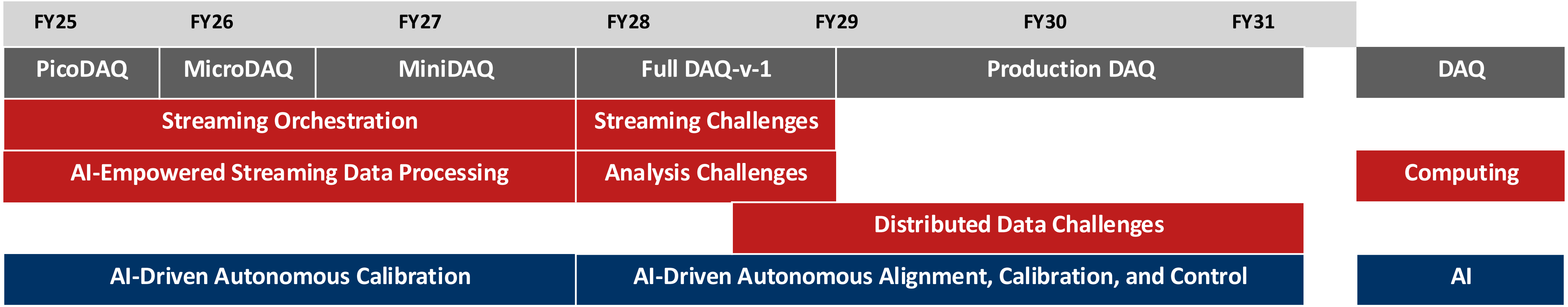
Streaming DAQ and Computing Milestones

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 Orchestration Milestones and Deliverables

- ✓ **Requirement documents** for streaming orchestration developed.
- **FY28 Q1 Goal**: Deliver a functional testbed for calibrating one detector system using simulated streaming data.
- Progress is ongoing in testbed development:
  - We are evaluating streaming orchestration using **PanDA + Rucio** (slides 15–20).
  - We have demonstrated streaming data processing using **EJFAT** (slides 21–22).
  - Additional prototypes under consideration: LHCb Allen, SPADI Alliance.

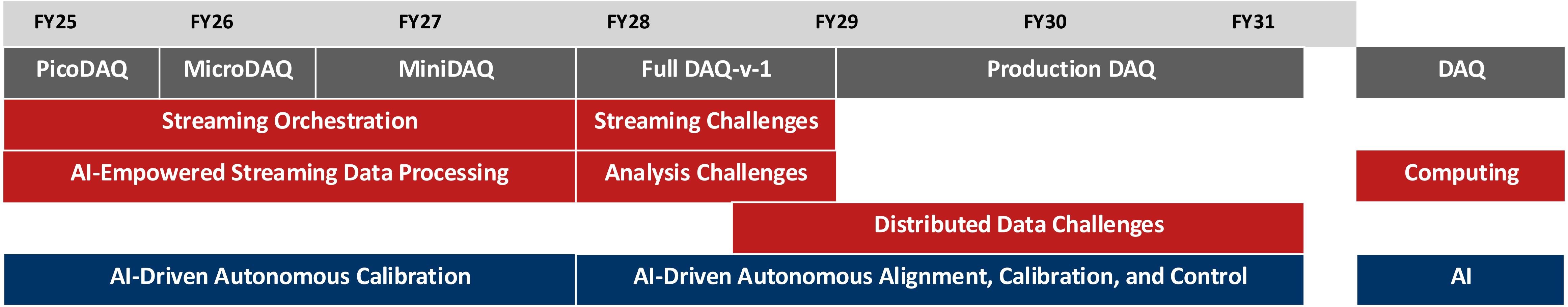
Streaming DAQ and Computing Milestones



Streaming Data Processing Milestones and Deliverables

- ✓ **JANA2 enables data processing at the timeframe, event, and sub-event levels.**
- **FY28 Q1 Goal**: Achieve streaming data reconstruction with high efficiency in identifying physics collision events in simulations, including varying levels of background. This includes an AI/ML challenge focused on developing algorithms for distinguishing physics events from background.
- Progress is ongoing in streaming data reconstruction (slides 13–14).

Streaming DAQ and Computing Milestones

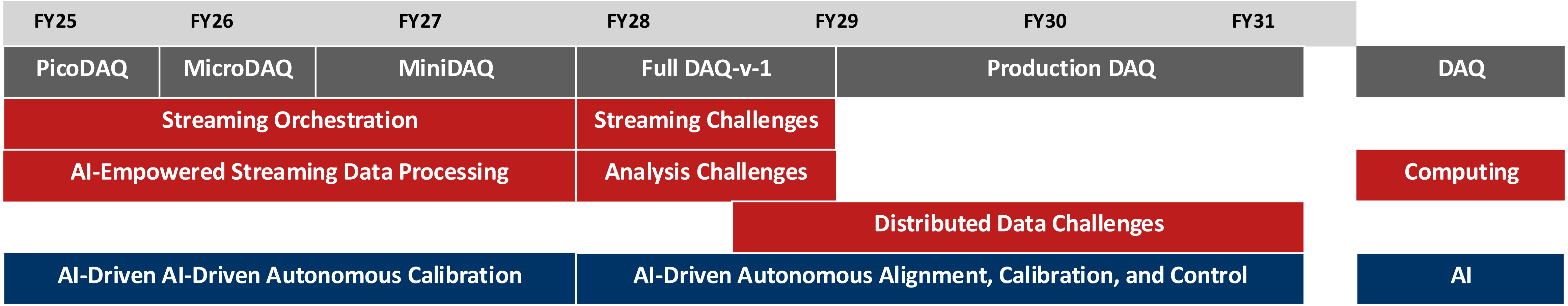


Streaming, Analysis, and Distributed Data Challenges

Streaming Data Challenge	E0+E1	Archiving and monitoring of TFs / STFs. Streaming reconstruction based on TFs / STFs.
Analysis Challenge	E1+E2	Autonomous alignment and calibration based on STFs Exercising end-to-end workflows from STFs to physics analysis.
Distributed Data Challenge	E1+E2	Exercising scaling and capability tests.

- Preparations focus on streaming orchestration, data processing, and monthly simulation campaigns that support detector and physics analysis for the preTDR and Early Science Program, while also driving near-term software development goals.

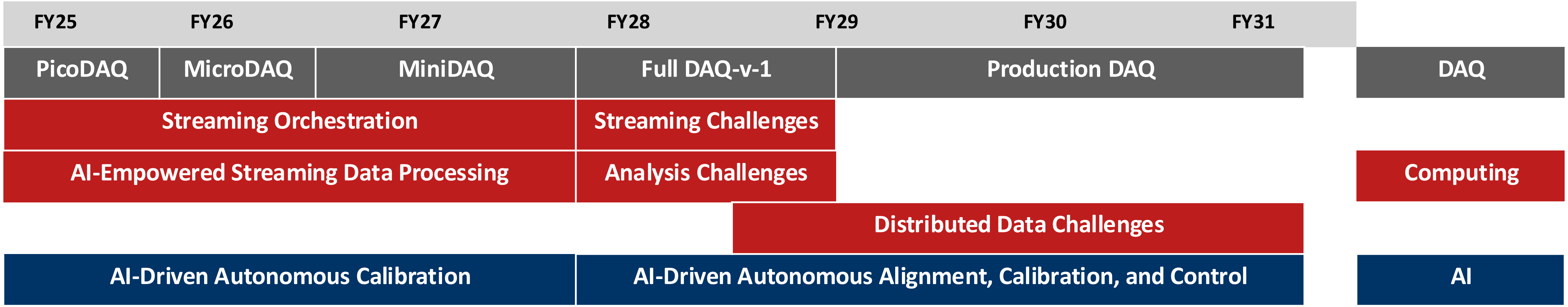
Streaming DAQ and Computing Milestones



AI-Driven Autonomous Calibration

- Progress continues on understanding alignment and calibration workflows in collaboration with subsystem experts, with a focus on identifying timelines and interdependencies.
- The strategy for autonomy involves algorithms for change detection and agentic workflows.
- **FY28 Q1 Goal**: Autonomous calibration of one detector system using simulated streaming data.

Streaming DAQ and Computing Milestones



AI-Driven Autonomous Alignment, Calibration, and Control

- Initial efforts focus on enabling fast access to detector data in a format suitable for real-time AI processing.
- Develop tooling to couple data stream processing with control systems with an emphasis on safety and reliability.

Summary

The **Streaming Computing Model** is defined, reviewed, and actively guiding ongoing developments. Its integration of computing and detector systems of ePIC experiment **maximizes scientific output** and **accelerates scientific discovery**.

ePIC's **computing and storage needs** are substantial with distributed computing playing a critical role.

The effort is **transitioning from design to implementation**, with active testbeds on streaming orchestration (PanDA and Rucio) and functional prototypes on streaming processing (EJFAT) and streaming reconstruction (JANA2/EICrecon).

Deliverables are aligned with EIC Project / Streaming DAQ milestones, enabling integrated development of detector readout, data processing, and analysis systems. FY28 Q1 streaming computing deliverables include: a fully functional **testbed for streaming orchestration**, an **autonomous calibration workflow for one detector system**, and **AI/ML-empowered streaming reconstruction**.

With a computing model, active testbeds and prototypes, and implementation milestones in place, the ePIC Software & Computing effort is fully aligned with (pre)TDR goals and with the development and implementation of the DAQ.