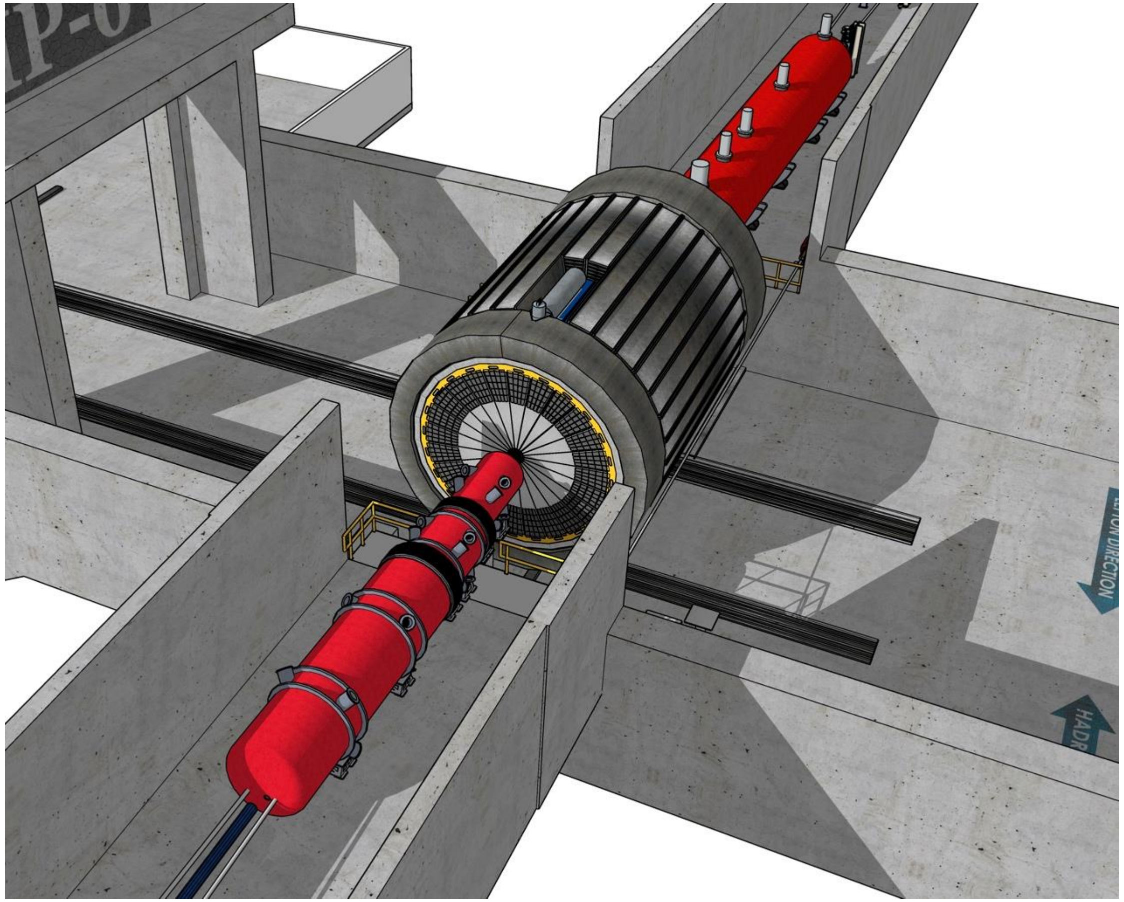
ePIC Computing Overview

Dr. Markus Diefenthaler
(Jefferson Lab, ePIC Software & Computing Coordinator)

Streaming Readout

The Highly-Integrated ePIC Experiment

Integrated Interaction and Detector Region (90 m)
Get close to full acceptance for all final state particles, and measure them with good resolution. All particles count!



Compute-Detector Integration
Seamless data processing from detector readout to analysis using streaming readout and streaming computing.

Definition of Streaming Readout
- Data is digitized at a fixed rate with thresholds and zero suppression applied locally.
- Data is read out in continuous parallel streams that are encoded with information about when and where the data was taken.
- Event building, filtering, monitoring, and other data processing is deferred to computing.

Advantages of Streaming Readout
- Simplification of readout (no custom trigger hardware and firmware) and increased flexibility.
- Event building from holistic detector information.
- Continuous data flow provides detailed knowledge of backgrounds and enhances control over systematics.

Compute-Detector Integration Using Streaming Readout and Computing

Broad ePIC Science Program:
- Plethora of observables, with less distinct topologies where every event is significant.

Streaming Readout Capability Due to Moderate Signal Rate:
- **Capture every collision signal**, including background.

	EIC	RHIC	LHC → HL-LHC
Collision species	$\vec{e}+\vec{p}$, $\vec{e}+A$	$\vec{p}+\vec{p}/A$, $A+A$	$p+p/A$, $A+A$
Top 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} → 10^{35} cm^{-2} s^{-1}
x-N cross section	**50 µb**	**40 mb**	**80 mb**
Top collision rate	500 kHz	10 MHz	1-6 GHz
$dN_{ch}/d\eta$	0.1-Few	~3	~6
Charged particle rate	**4M N_{ch}/s**	**60M N_{ch}/s**	**30G+ N_{ch}/s**

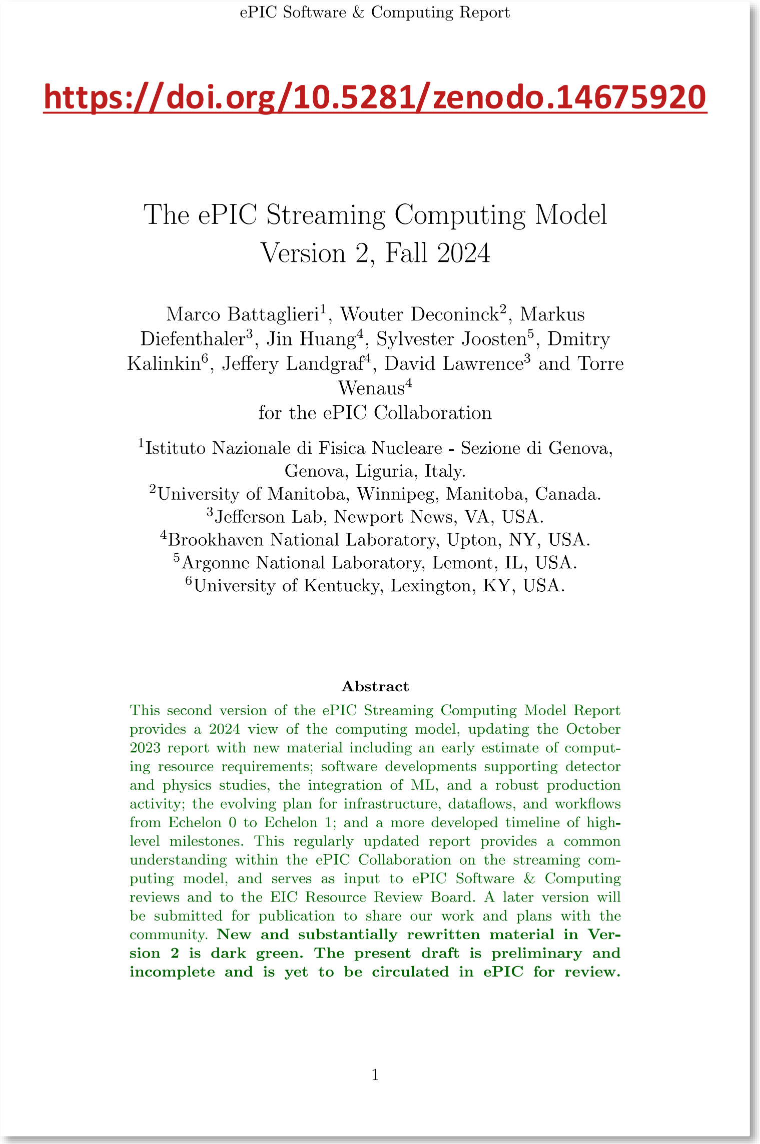
Compute-Detector Integration to Maximize and Accelerate Science

- **Maximize Science** **Capture every collision signal, including background.**
  - **High-precision measurements**: Control of systematic uncertainties of paramount importance.
  - 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 2-3 weeks for data for physics analyses**.
  - Timeline driven by alignment and calibration.
  - Preliminary information from detector subsystem experts indicates that 2-3 weeks are realistic.
- **Technologies** Compute-detector integration using:

Streaming readout for continuous data flow of the full detector information.	**AI** for rapid processing (autonomous alignment, calibration, and validation).	**Heterogeneous computing** for acceleration (CPU, GPU).

Computing Model

The ePIC Streaming Computing Model



ePIC Software & Computing Report

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

The ePIC Streaming Computing Model
Version 2, Fall 2024

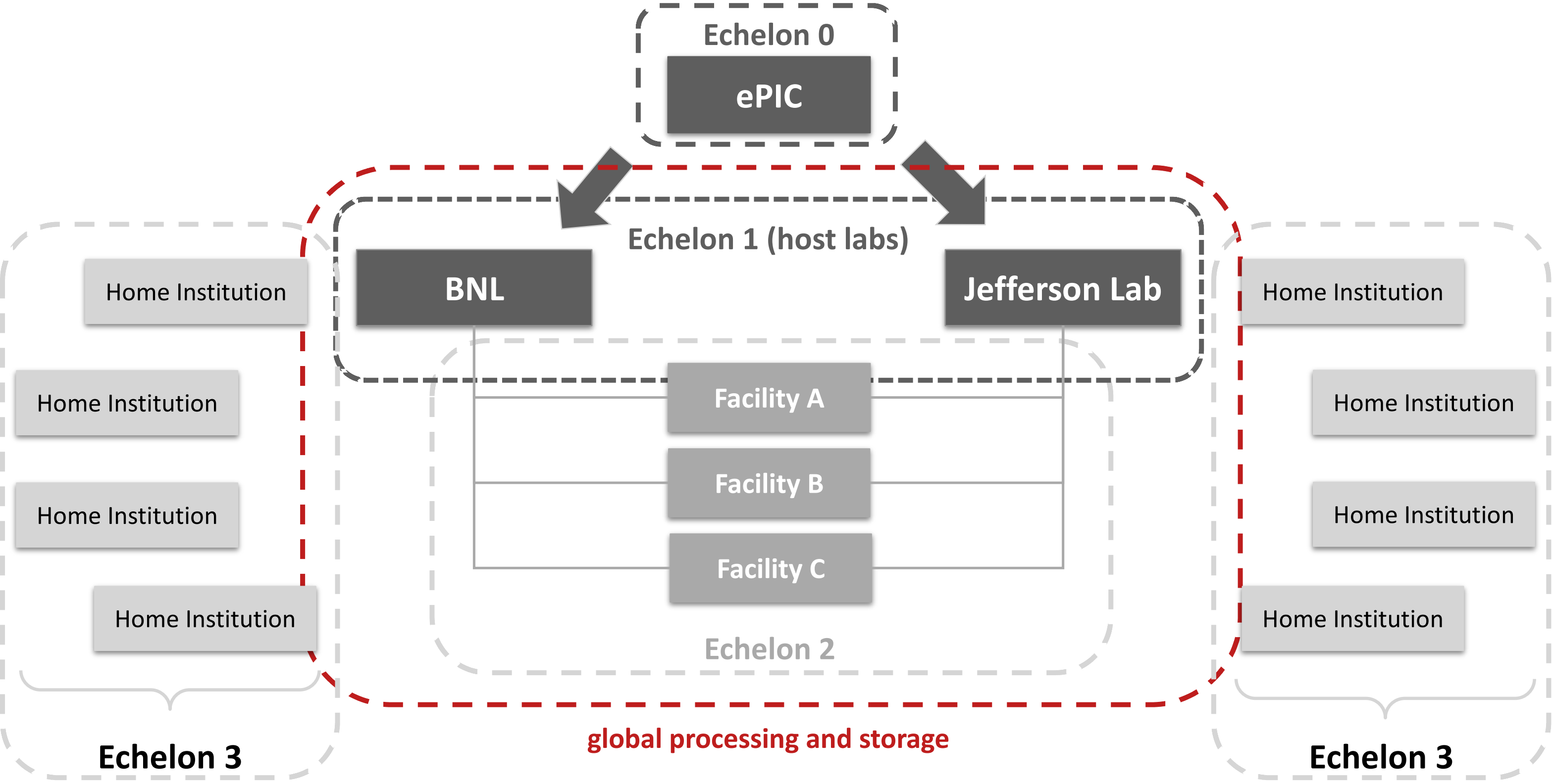
Marco Battaglieri[1], Wouter Deconinck[2], Markus Diefenthaler[3], Jin Huang[4], Sylvester Joosten[5], Dmitry Kalinkin[6], Jeffery Landgraf[4], David Lawrence[3] and Torre Wenaus[4]
for the ePIC Collaboration

[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

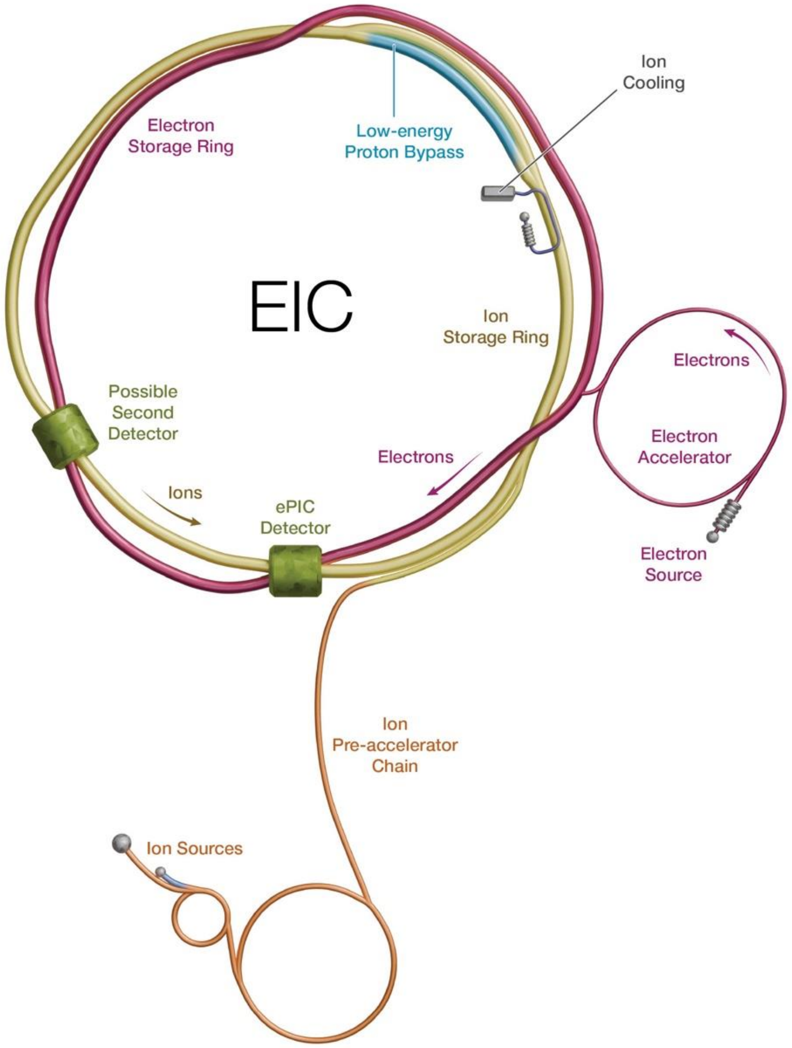
This second version of the ePIC Streaming Computing Model Report provides a 2024 view of the computing model, updating the October 2023 report with new material including an early estimate of computing resource requirements; software developments supporting detector and physics studies, the integration 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 ECSAC reviews.

The EIC and Event Rates



- **Versatile machine**: versatile range of beam polarizations, beam species, center of mass energies.
- **High luminosity** up to L = 10^{34} cm^{-2} s^{-1} = 10 kHz/µb.
  - The e-p cross section at peak luminosity is about 50 µb. This corresponds to a signal event rate of about 500 kHZ.
- The **bunch frequency** will be **98.5MHz**, which corresponds to a **bunch spacing** of about **10ns**.
  - For e-p collisions at peak luminosity, there will be in average 200 bunches or about 2µs between collisions (98.5MHz / 500 kHz).
- The EIC Project and ePIC collaboration are currently discussing the early science program of the EIC.
- For the computing resource estimate, we consider the luminosity scenario of L = 10^{33} cm^{-2} s^{-1} = 1 kHz/µb.

Rate Estimates from Streaming DAQ

- **Event size of in average 400 kbit**,
  - Including signal and background apart from detector noise,
  - Assuming that detector noise can be substantially reduced in early stages of processing.
  - Event sizes will decrease in later stages of data taking as detector thresholds are raised.

- **Data rate of in average 30 Gbit/s**,
  - Estimate of upper limit: 10Gbit/s for detector noise + event rate * event size.
  - Event rate = 50 KHz for EIC Phase 1 luminosity and maximum e-p cross section of *50 μb.*

- **Running 60% up-time for ½ year = 9,460,800 s**:
  - Data rate of 30 Gbit/s results in 710×10^9 events per year.
  - The data volume of 35.5 PB per year will be replicated between Echelon 1 facilities (71 PB in total).

Computing Resource Needs (EIC Phase I) 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:

- Optimistic scaling of constant-dollar performance gains would reduce the numbers about 5x:
  - Based on current WLCG measure of 15% per year.
  - But the trend is towards lower gains per year.
- Whatever the gains over time, processing scale is substantial!
- Motivates attention to leveraging distributed and opportunistic resources from the beginning.

~350 PB to store data of one year.

Computing resource needs at a scale of ATLAS and CMS today.

ePIC is compute-intensive experiment; must ensure ePIC is not compute-limited in its science.

ePIC

Distributed Computing for ePIC

- **Echelon 1** sites uniquely perform the **low-latency streaming workflows**:
  - Archiving and monitoring of the streaming data, prompt reconstruction and rapid diagnostics.
- Apart from low-latency, **Echelon 2** sites fully participate in use cases and **accelerate** them.
- **Priority**: Establishing EIC International Computing Organization (EICO):

ECSAC recommended a staged implementation starting with the Collaboration and Overview Boards. See ECSJI presentation for details on EICO.

Use Case	Echelon 0	Echelon 1	Echelon 2	Echelon 3
Streaming Data Storage and Monitoring	✓	✓		
Alignment and Calibration		✓	✓	
Prompt Reconstruction		✓		
First Full Reconstruction		✓	✓	
Reprocessing		✓	✓	
Simulation		✓	✓	
Physics Analysis		✓	✓	✓
AI Modeling and Digital Twin		✓	✓	

Substantial role for Echelon 2 in preliminary resource requirements model

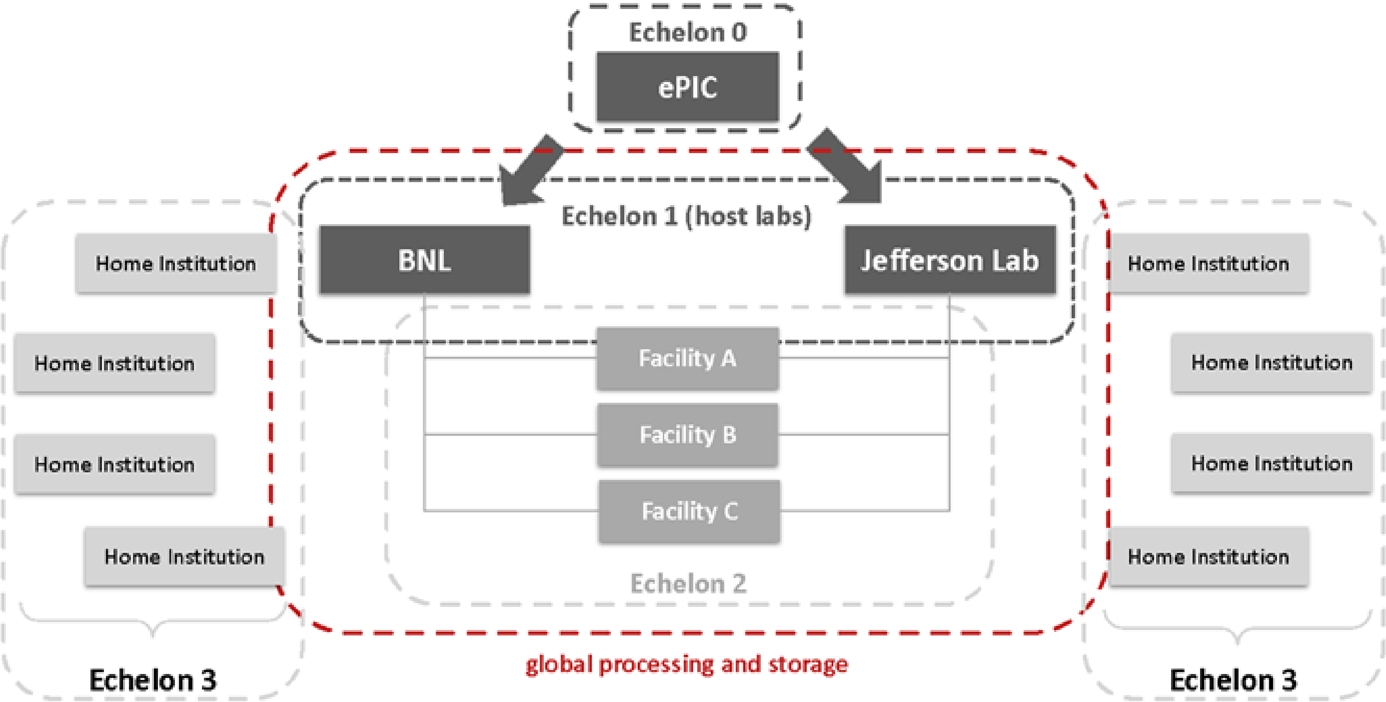
Assumed Fraction of Use Case Done Outside Echelon 1	
Alignment and Calibration	50%
First Full Reconstruction	40%
Reprocessing	60%
Simulation	75%

Distributed Computing Model

Building Echelon 2 capacity through early integration in simulation campaigns.

- Central infrastructure at **JLab**:
  - HTCondor submit node
  - Rucio main server
  - Rucio storage element
- Central infrastructure at **BNL**:
  - HTCondor submit node
  - Rucio storage element

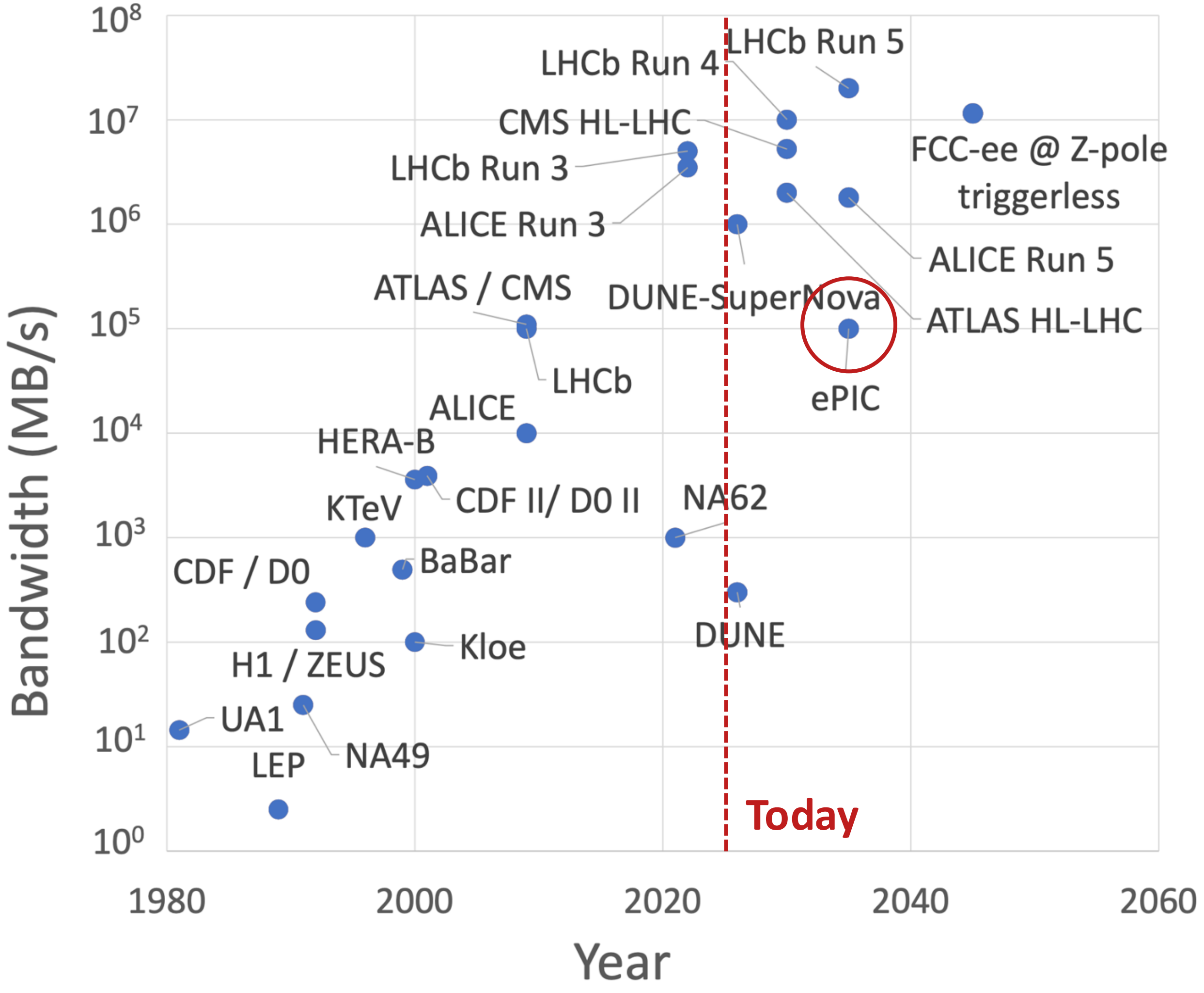
- Active integration of **Canada** and **Italy** as compute providers integrated through OSG.



- Commissioning of **Canada** as Rucio storage element provider.
- Planning in progress for **Japan** and **Taiwan** as storage providers.

- Discussions on potential compute and storage contributions are underway with **France** and the **United Kingdom**.

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/

Testbeds

Prototyping Ideas and Tools in Testbeds

With active testbeds and functional prototypes now in place, the effort is moving from design to implementation. These developments aim to define and test the interface between DAQ and computing, and to mitigate risks in the integrated DAQ-computing system.

- **Streaming orchestration**, i.e., a workflow and workload management system for streaming data—is essential for system testing. A requirements document has been developed and is now guiding testbed and prototype development.
  - **Testbed plans** are taking concrete shape:

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

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

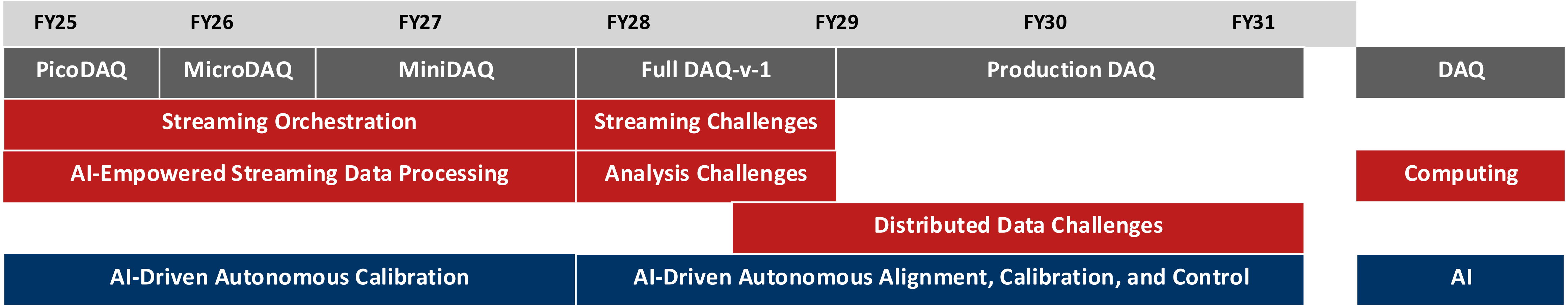
Streaming processing: Developing E0-E2 streaming workflows using EJFAT.

  - Not covered in this presentation but starting efforts:

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

Rapid data processing: Autonomous calibration workflow for one detector system.

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

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 20–22).
  - We have demonstrated streaming data processing using **EJFAT** (slides 23).
  - Additional prototypes under consideration: **Allen**, SPADI Alliance.

ePIC

Streaming Data Processing

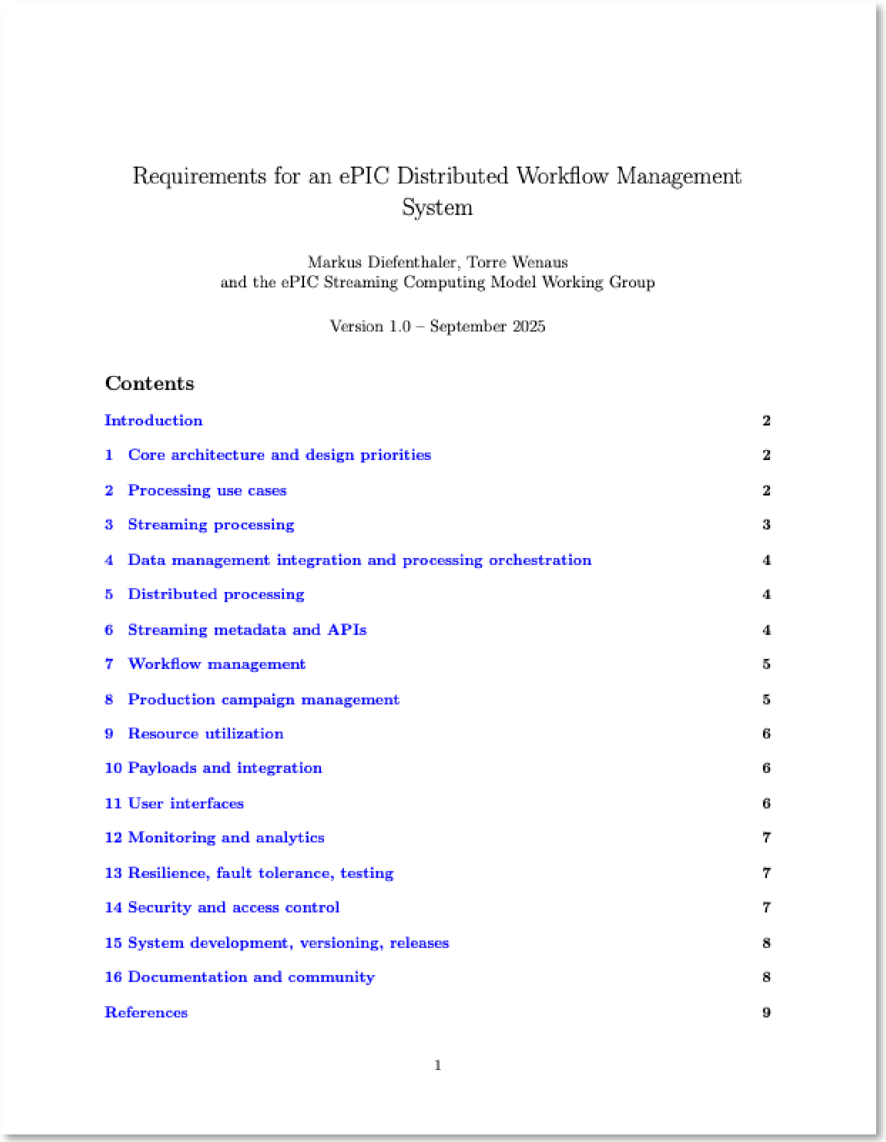
Traditional Workflow Characteristics in NP and HEP Experiments:
- Data is acquired in online workflows.
- Data is stored as large files in hierarchical storage.
- Offline workflows process the data, often with substantial latency.
- Batch queue-based resource provisioning is typical.
- Key features: discrete, coarse-grained processing units (files and datasets) and decoupling from real-time data acquisition.

ePIC Streaming Data Processing Characteristics

- Quasi-continuous flow of fine-grained data.
- Dynamic flexibility to match real-time data inflow.
- Prompt processing is crucial for data quality and detector integrity.
- Processing full data set quickly to minimize time for detector calibration and deliver analysis-ready data.

Challenging Characteristics of Streaming Data Processing:
- **Time critical**, proceeding in near real time.
- **Data driven**, consuming a fine-grained and quasi-continuous data flow across parallel streams.
- **Adaptive and highly automated**, in being flexible and robust against dynamic changes in data-taking patterns, resource availability and faults.
- **Inherently distributed** in its data sources and its processing resources.

Requirements for Streaming Orchestration


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.

Streaming Orchestration Testbed

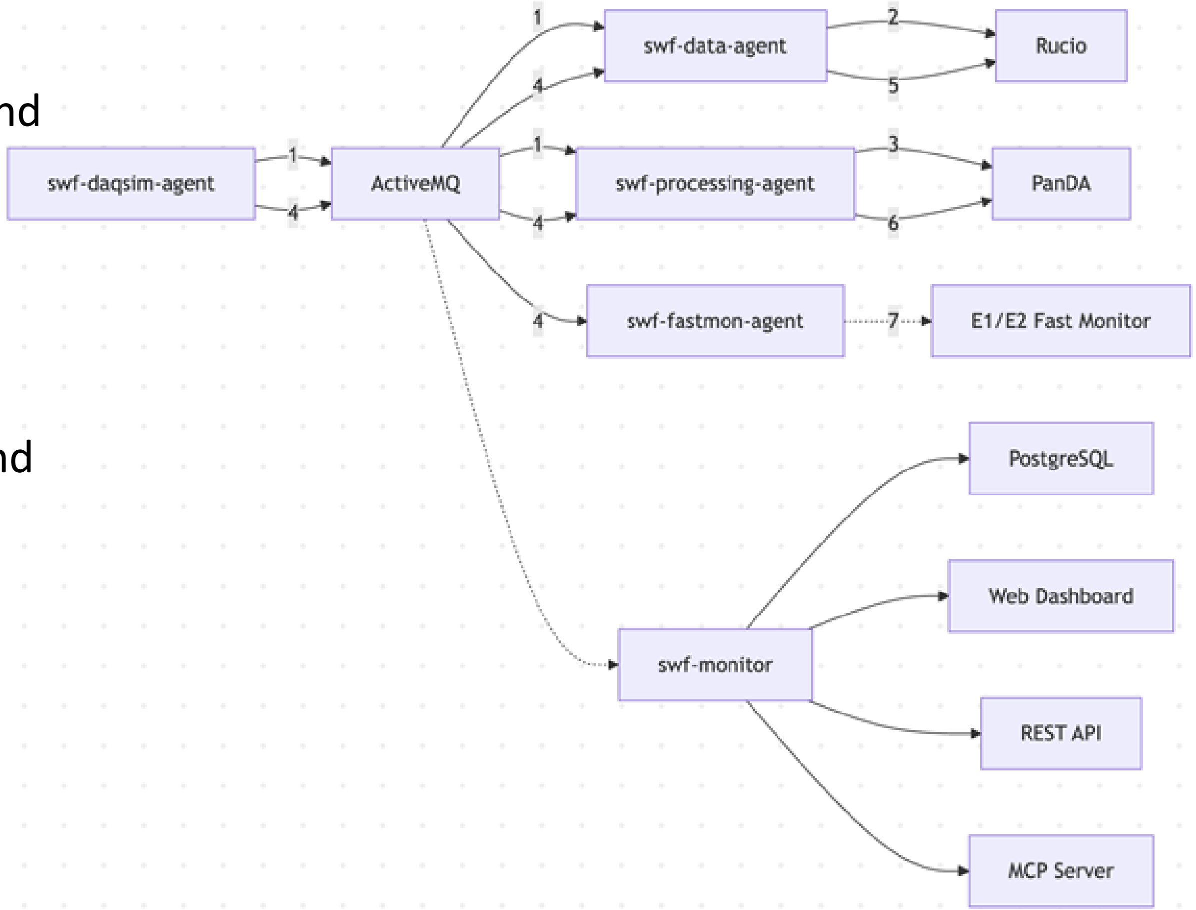
Motivation:
- Evaluate how well existing distributed computing tools support streaming orchestration.
- Focus on practical deployment and performance in realistic environments.

- **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 and are deployed in live testbed instances at BNL.
  - 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.
- **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.
  - A **monitoring backend** with a database aggregates and exposes system state.

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**).



ePIC

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

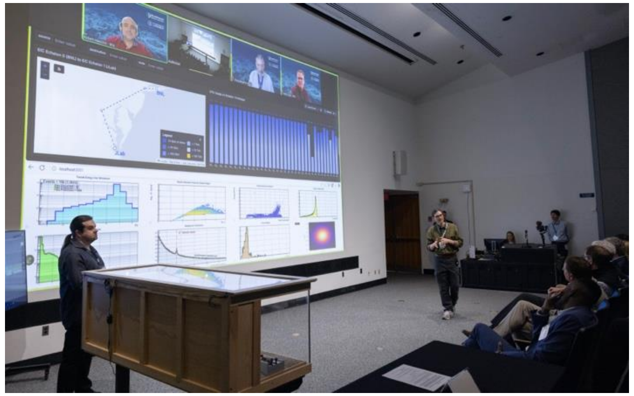
Substates

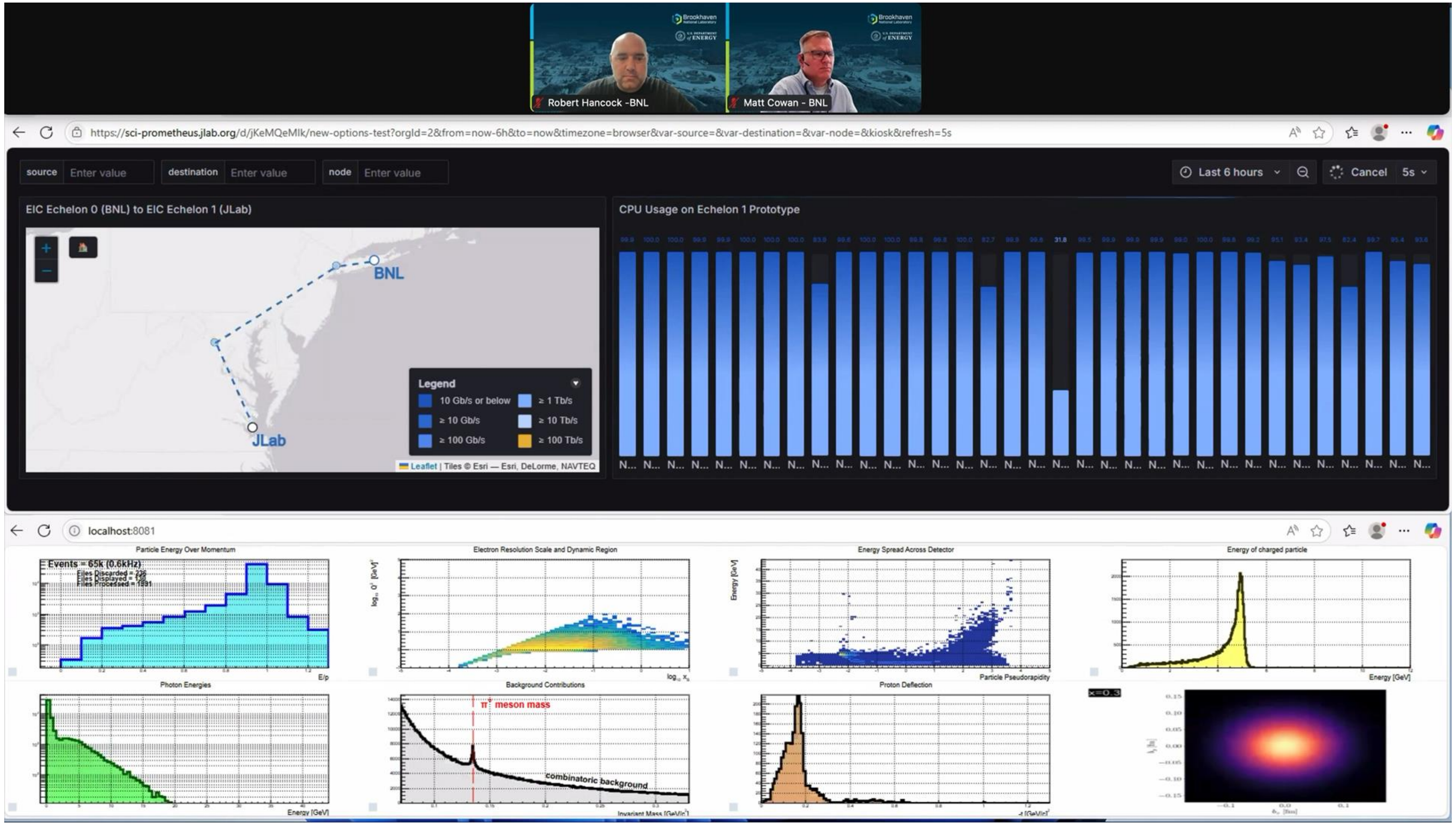
- not_ready
  - detector not ready for physics datataking
  - occurs during states: no_beam, beam, calib
- ready
  - collider and detector ready for physics, but not declared as good for physics
  - when declared good for physics, transitions from beam/ready to run/physics
  - occurs during states: beam
- 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.

Streaming Data Processing Demonstration

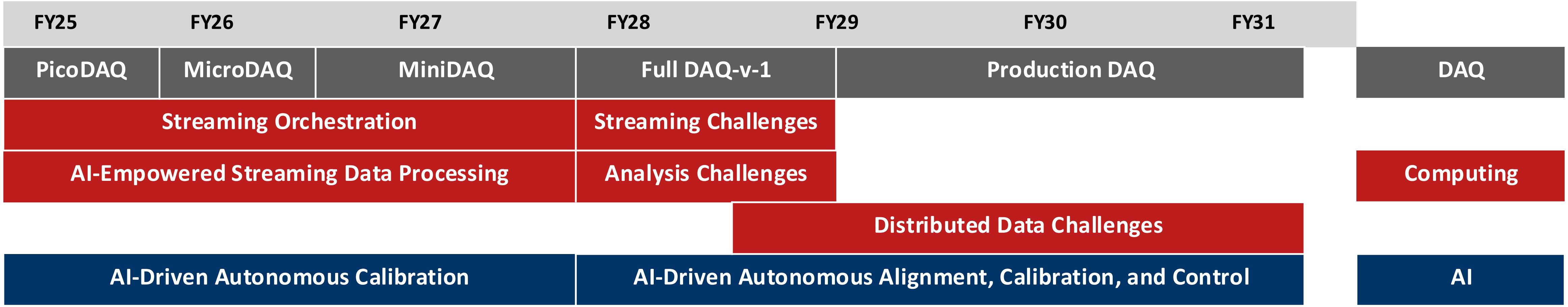
Secretary Wright's Visit to Jefferson Lab on August 21





We successfully showcased **real-time data transfer from BNL to JLab** using the ESnet-JLab FPGA Accelerated Transport (EJFAT) Load Balancer and its **processing at JLab**. It also prompted collaboration with BNL to test current tools and establish a clear network path.

Streaming DAQ and Computing Milestones



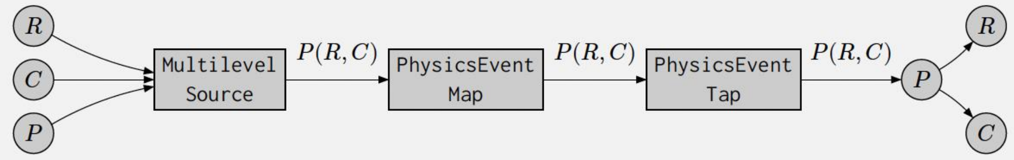
Streaming Data Processing Milestones and Deliverables

- ✓ **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 25–26).

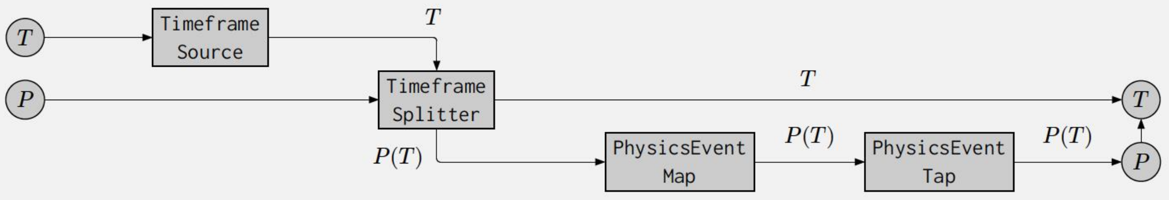
JANA2 for Streaming Processing

- 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**.

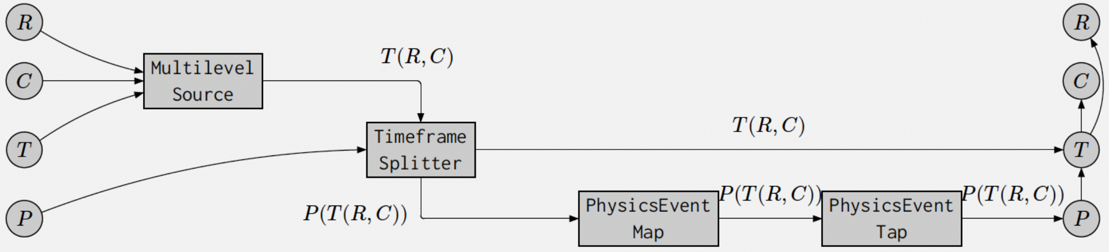
Introducing multilevel sources



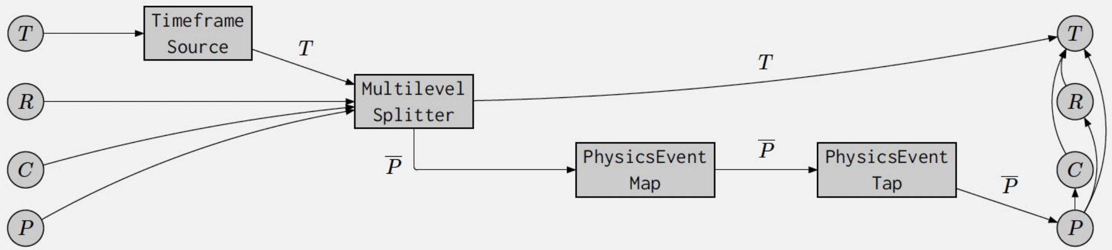
EICrecon timeframe splitting



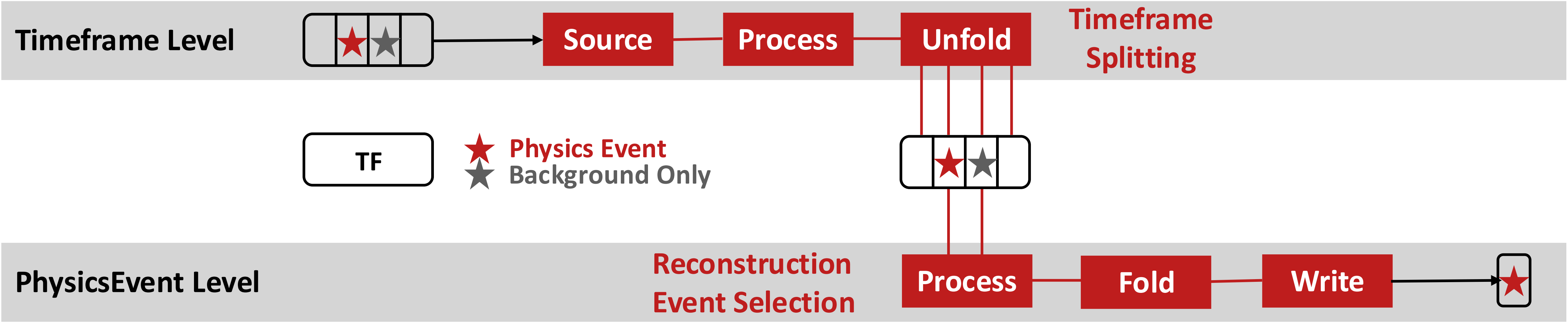
Multilevel sources with timeframe splitting

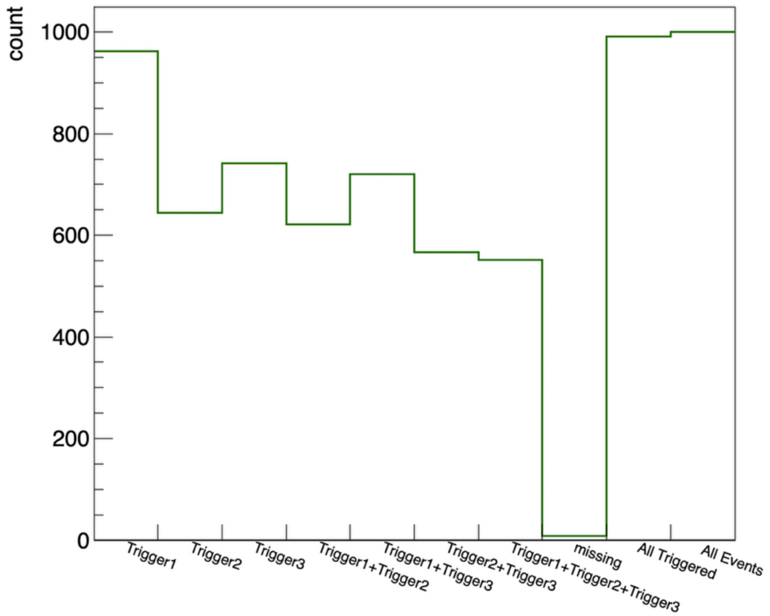


Timeframe sources with multilevel splitting



Streaming Reconstruction Prototype: Event-Building in JANA2

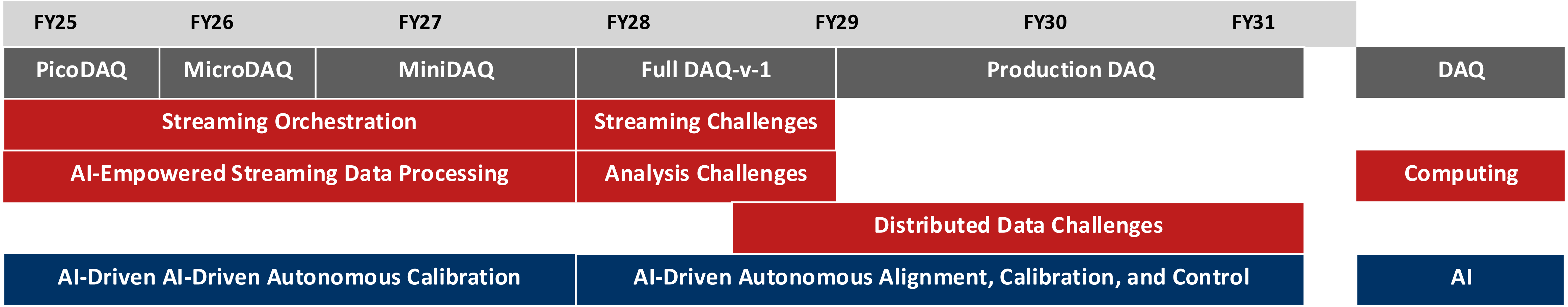




Trigger 1: Coincidence hits in the SVT Endcap and Forward MPGD Endcap.
Trigger 2: Coincidence hits in the TOF Barrel and MPGD Barrel.
Trigger 3: Coincidence hits in the SVT Endcap and Backward MPGD Endcap.

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.

Streaming DAQ and Computing Milestones



AI-Driven Autonomous Calibration

- Progress continues on understanding 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.
- Milestones and deliverables linked to:
  - Streaming Orchestration Milestones and Deliverables (slide 17)
  - Streaming Data Processing Milestones and Deliverables (slide 24)

AI-Driven Autonomous Calibration

Example: We will use the **Backward ECal (EEEMCal) Calibration** as an example. Carlos will provide the necessary calibration scripts and integrate them into JANA2 and EICrecon. As part of this integration, the data flow and required input data will be defined.

To achieve autonomy in this calibration case, we have three distinct tasks (**AI aspects** are highlighted in **dark blue**):

1. **Calibration Logic**: We need **software that detects when a new calibration is required** for the EEEMCal and updates the state machine accordingly.
2. **Calibration Integration Into State Machine**: The state machine must be connected with a calibrations/conditions database system to track calibration status and link to calibration data. Clarify who is reading and writing.
3. **Calibration Execution and Validation**: The **calibration script must recognize** when it has derived new constants, **validate these constants**, and then register them in the calibrations/conditions database.

There are also three boundary conditions:

- **The Online Condition**: All of this must function in an online context, meaning while data is still being accumulated and not yet available offline. We will start with file-based workflows to make immediate progress, while developing support for streaming.
- **The Human Condition**: We aim to automate these processes as much as possible. However, it is essential to clearly define where human oversight is required. This includes questions such as: When should a human intervene in the loop? Which decisions must be reviewed or approved manually? While this is a crucial discussion, this is a separate conversation. Our priority is to design and implement the autonomous system and then determine how and where to introduce the human-in-the-loop component.
- **The Cybersecurity Condition:** Cybersecurity requirements will be addressed in a later implementation phase.

Outlook

Why Allen Is Relevant for ePIC Streaming Computing

- LHCb has been a major inspiration for the ePIC computing model, and we have actively studied lessons learned from the operation of Allen in shaping our approach.
- Our goal is seamless data processing from detector readout through alignment, calibration, and reconstruction, and ultimately physics analysis in a streaming and heterogeneous computing environment.
- You have experience with autonomous alignment, calibration, and reconstruction on heterogeneous architectures, which closely aligns with the challenges ePIC faces in our streaming computing model.
- ePIC is at the stage where design choices are still open, and Allen experience can directly shape a next-generation streaming experiment.
- We have discussed exploring streaming orchestration and streaming calibrations using Allen Core as part of ePIC testbeds and prototyping efforts. These activities would inform the design of the ePIC streaming orchestration system through FY28 and provide a concrete opportunity for collaboration at both the algorithmic and systems levels.