The ePIC E0–E1 Interface

Markus Diefenthaler Torre Wenaus

Version 0.7, October 8, 2026

1 Introduction

The E0–E1 interface describes the boundary between Echelon 0 (E0), the ePIC detector and its data acquisition system (DAQ) within the experimental facility, and Echelon 1 (E1), the computing sites at BNL and JLab, in the ePIC Computing Model. Across this boundary, streaming data, data-taking state, and operational control are exchanged between the detector and computing domains, with workflow management system (WFMS) responsibility beginning on the E1 side of the interface. This document defines the E0–E1 interface, including the agreed architecture, interface definitions and mechanisms, and the open questions and development path.

At a high level, the E0–E1 interface supports a unidirectional flow of detector data from E0 to E1, accompanied by the state and metadata required to interpret and process that data. Control, status, and monitoring information may flow in both directions. E0 remains responsible for detector operation and data acquisition, while E1 assumes responsibility for downstream data handling, processing, monitoring, and workflow execution once data cross the interface. The principal information flows across the interface are:

- Detector data from E0 to E1;
- Data-taking state and operational metadata from E0 to E1;
- Status, acknowledgements, and monitoring information between E0 and E1;
- Calibration and conditions information as required by processing; and
- Control requests or intents between the two domains where explicitly permitted by the interface.

The E0–E1 interface is a critical element of the ePIC Computing Model. A streaming experiment depends on reliable and well-defined transfer of data, state, and control information between the detector and computing domains. The interface therefore needs to define not only the data products that cross the boundary, but also their state, timing, provenance, availability, and control semantics. The document also describes a non-normative reference implementation used to exercise selected interface concepts. The principal areas covered by the interface document are:

1. Raw data transfer in time frames and super time frames;
2. State model and transition rules;
3. Latency requirements;
4. Information interfaces between E0 and E1;
5. Calibration and conditions databases;
6. Control interfaces between E0 and E1; and
7. AI readiness and AI-enabled control.

The following sections first establish the architecture in Section 2 and E0 context in Section 3, then address each interface area in Sections 4–10. A reference implementation of selected interface concepts is described in Section 11, followed by the open issues and roadmap in Section 12.

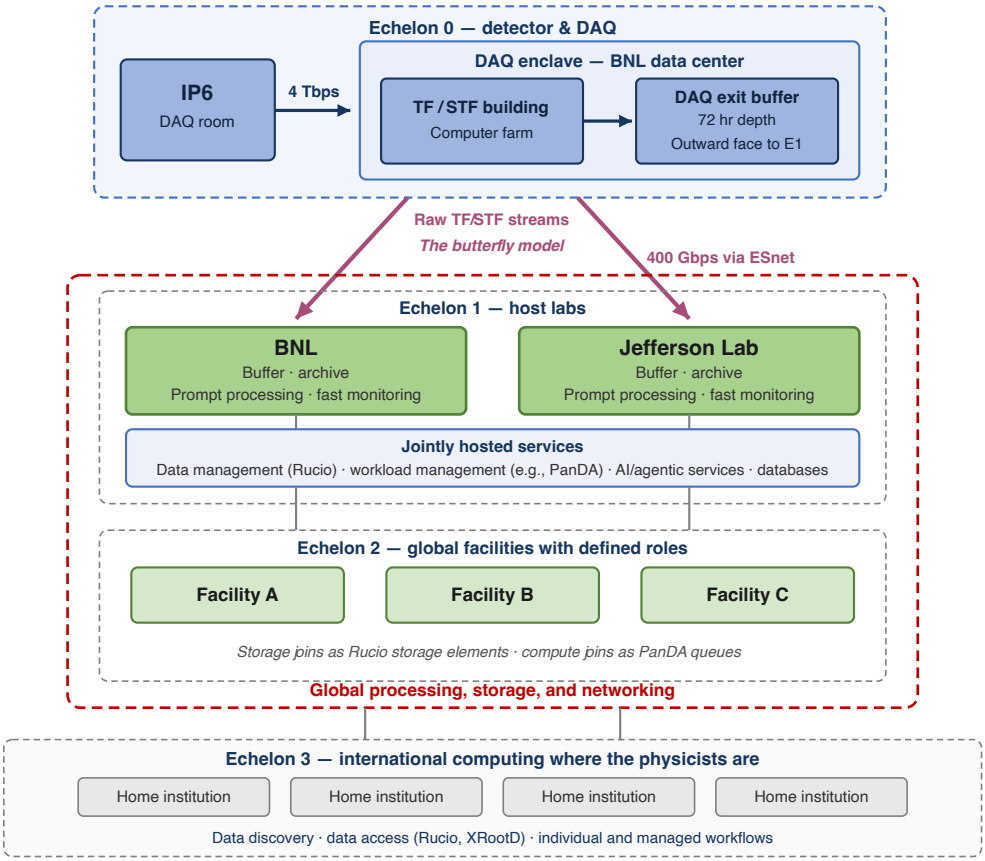

Figure 1: Overview of the ePIC Computing Model, showing the E0–E1 interface, the DAQ exit buffer, TF/STF data flow to the BNL and Jefferson Lab E1 sites, and the relationship to the wider computing model.

2 Architecture

The DAQ is a single system spanning two facilities: the DAQ room at IP6 and the DAQ enclave at the Scientific Data and Computing Facility (SDCF) at BNL, connected at $4\,\mathrm{Tbit/s}$. Time frames (TFs) are built in the DAQ enclave and may be aggregated into super time frames (STFs). Both TFs and STFs are written to the DAQ exit buffer, which is sized for approximately $72\,\mathrm{h}$ of data-taking and extends onto an external subnet for delivery to E1. The transfer mechanism across the E0–E1 boundary remains an open design choice, with both streaming and file-based delivery under consideration. This outward-facing interface is the portion of E0 visible to the post-DAQ computing system and defines the point at which E1 receives TFs and STFs from the data-taking stream, as illustrated in the overall ePIC Computing Model in Figure 1.

E1 comprises the sites at the two host laboratories, each providing buffer, archive, prompt processing, and fast monitoring, with jointly hosted services—Rucio data management, WFMS, AI services, and databases—spanning both sites. Raw data are delivered to both E1 sites (the butterfly model), with JLab connected over ESnet at $400\,\mathrm{Gbit/s}$. The WFMS requirements specify that the two E1 sites operate as one integrated distributed system with failover. If one E1 site becomes unavailable, the other site provides full functional replacement and assumes its responsibilities. Two data flows, together with control messaging, cross the E0–E1 boundary; they are described in the following sections.

The digitized detector data rate is roughly $2\,\mathrm{Tbit/s}$, with a recorded output requirement of about $100\,\mathrm{Gbit/s}$. The maximum deep inelastic scattering rate is approximately $500\,\mathrm{kHz}$, compared with the $98.5\,\mathrm{MHz}$ bunch-crossing clock. Noise and backgrounds can dominate the input data rate: synchrotron radiation alone may reach several

hundred $\mathrm{Gbit/s}$, while dRICH dark-current rates after radiation damage require a high-level filter.

These data rates motivate data reduction in E0 or early in E1, including the possible use of a software trigger. Where practical, such reduction should preferentially be performed in E1 so that E0 can provide an unbiased aggregated data stream and systematic effects introduced by data selection can be better controlled. The location at which raw data are finalized therefore remains an aspect of the E0–E1 interface that must be defined.

3 Echelon 0

E0 comprises about 30 detectors read out through approximately 2500 readout boards (RDOs), approximately 120 data aggregation boards (DAMs), and a global timing unit (GTU). E0 computing consists of $O(100)$ readout computers in the DAQ room and $O(100)$ computers at SDCF. The E0 systems at SDCF provide time frame building, high-level filtering, archiving, monitoring, logging, and quality assurance (QA).

The E0 run-control model incorporates both run-dependent and continuously running components, such as scalers. The DAQ operates continuously; a run denotes the start and end of data-taking with a given configuration. A run is therefore a logical data-taking interval rather than a file boundary. A run configuration selects and configures the participating detectors, while slow-controls status forms part of the overall state model. This E0 state information provides the basis for integration with the E0–E1 state model described in Section 5.

Raw data produced by the readout are organized into data banks. Raw banks may be formatted but are not modified after creation. Subsequent processing, including data reduction or enrichment, produces new banks rather than altering the original data. A fraction of the raw banks is retained, providing data-level provenance between the original detector readout and derived data products.

Detector timing is referenced to the EIC bunch-crossing clock. ASICs report hit times relative to a synchronization signal, and the DAQ provides a time function expressed in bunch crossings relative to the start of a time frame. Detector-specific functions map channel, clock cycle, and fine-counter information to a corresponding time value. Bunch-dependent polarization requires event data and luminosity monitoring to be resolvable to the individual bunch crossing for each of the 1260 bunches.

The electronics and DAQ program is working toward a final design review in early 2027. The planned DAQ releases progress from PicoDAQ (FY27Q3), providing basic readout test setups, through MicroDAQ (FY27Q4), supporting engineering-article test stands and E0 computing at SDCF, and MiniDAQ (FY29Q3), providing the full hardware and timing chain. FullDAQ v1 is planned for FY30Q4, followed by the Production DAQ in FY33Q1. PicoDAQ and MicroDAQ do not provide a direct interface between detector components and E1, but they can support development and testing of the E0–E1 interface, including implementation of the DAQ side of transfer interfaces used by the orchestration testbed.

4 Raw Data Transfer in Time Frames and Super Time Frames

A TF contains the multiplexed data of the entire detector for 2^{16} bunch crossings, corresponding to approximately $0.665\,\mathrm{ms}$. TF boundaries are deliberately not synchronized to bunch 0 so that boundary effects are decorrelated from spin states. Data from all participating detectors are incorporated into each TF.

An STF is an ordered and contiguous sequence of $O(1000)$ TFs, corresponding to approximately $0.6\,\mathrm{s}$ of detector data and a nominal data volume of order $2\,\mathrm{GB}$. STFs provide the larger-granularity unit for registration, bookkeeping, transfer, and bulk processing.

Both TFs and STFs are expected to cross the E0–E1 interface, as illustrated in Figure 2. TFs are logical time-ordered data units and are not necessarily materialized as individual files. STFs are larger-granularity data units that may be serialized as approximately $2\,\mathrm{GB}$ files for transfer, processing, and archival. The STF path provides the bulk-data path, while the TF path provides finer-granularity access to recent data and may use a streaming transfer mechanism. The final transport mechanisms remain to be defined and will be evaluated in testbeds and prototypes. The final TF

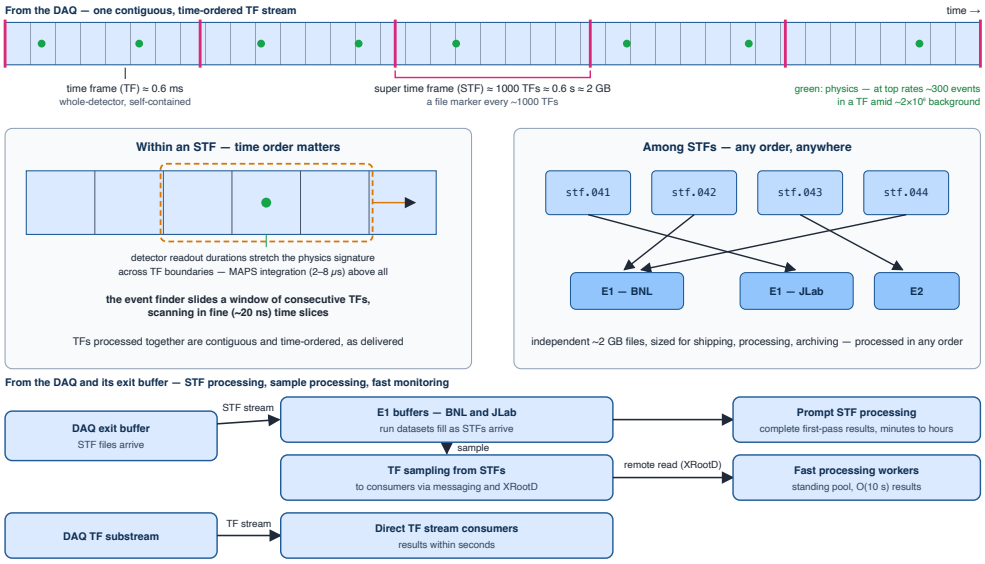

Figure 2: Relationship between time frames and super time frames and their use in the E0–E1 data flows.

and STF formats, including zero suppression and nominal data sizing, also remain to be defined.

Two complementary raw-data flows are currently under discussion. The STF flow would provide the complete raw-data stream for reliable delivery to the E1 buffers at both host laboratories, with a target latency of order minutes. The TF flow would provide finer-granularity access to recent data for fast monitoring and control, with a target latency of order seconds.

The two flows serve different consumers and have different completeness and retention semantics. Consumers must therefore be able to distinguish between data obtained through the fast TF path and a complete, validated STF. The interface definition must specify what completeness guarantees apply to each flow and what conclusions downstream consumers may draw from the available data.

Polarimetry and luminosity detectors operate as independent DAQ systems alongside the primary ePIC detector stream and provide their own data streams. These contain $n \times 1260$ values, with polarimetry information provided less frequently and luminosity information provided once per second. Data from these special-purpose streams may also be incorporated into the main data stream while remaining independently available for their specific uses.

Archiving and processing are initiated independently as data are ingested into E1 so that processing cannot block archival. Complete copies delivered to the two E1 sites are validated after arrival, and data integrity between the sites is monitored using checksums. The data stream is also sampled and validated against independent DAQ and conditions metadata to provide early detection of corruption and stream faults.

The final STF format, including zero suppression and nominal file sizing, remains to be defined.

5 State Model and Transition Rules

The ePIC system will maintain a global operational state model covering the accelerator, detector, and computing systems. The E0–E1 interface state model is an integral part of this global state because accelerator, detector, and computing status must be exchanged across the E0–E1 boundary and interpreted consistently by systems on both sides.

The interface state model therefore defines the state information relevant to accelerator operations, experiment and data-taking status, and downstream processing. It also defines the authority responsible for each state element and the rules by which state changes are propagated across the interface. The authoritative source of a given state

element remains with the system that owns it; E0 and E1 maintain the information required to interpret data and coordinate processing without duplicating state authority.

One candidate evolution is a factorized state model in which operational state is represented by independently owned facets, with a single authority for each facet and a recorded transition history. Derived operational views can then be constructed from these facets without introducing additional state authorities.

Relevant accelerator, detector, and DAQ state information must remain available across the E0–E1 interface outside active data-taking periods. E1 workflows must therefore be able to respond to state changes, including suspending and resuming processing as appropriate.

6 Latency Requirements

Three latency regimes are relevant to the E0–E1 interface, ordered from shortest to longest timescale:

1. **Very low latency**: latencies of approximately $1\,\mathrm{s}$ and below may be required for experiment control, including change detection and associated alarms. It remains an open question whether these functions will be implemented entirely within the DAQ or whether some will extend beyond the DAQ into E1. In the latter case, meeting the latency requirement may require a streaming transfer mechanism. The placement of this latency regime with respect to the E0–E1 interface therefore remains to be defined.
2. **Fast path**: recent data should become available for fast monitoring and control with a target latency of order $10\,\mathrm{s}$ to $30\,\mathrm{s}$. This may be implemented through fine-grained processing of arriving STFs and may not necessarily require a separate TF streaming path.
3. **Bulk prompt processing**: complete STF data should become available at the E1 sites with a target latency of order minutes. Prompt processing results may become available over timescales from minutes to hours, depending on processing load. This regime is associated with the STF data flow.

Each latency requirement must be defined between explicit endpoints. For example, an E0–E1 latency may be measured from data egress from the DAQ to availability of the corresponding data or derived monitoring result in E1. The precise start and end conditions for each latency class remain to be defined.

Beyond the E0–E1 interface, the streaming readout model has an end-to-end goal of providing reconstructed and calibrated data for physics analysis within approximately two weeks of data-taking. This turnaround is driven in part by calibration latency. Calibration workflows introduce additional timescales, ranging from seconds for prompt feedback, through minutes for run-dependent updates, to hours for delayed or quasi-online processing. These latency classes must be mapped onto the E0–E1 data flows and the corresponding calibration states.

7 Information Interfaces Between E0 and E1

The E0–E1 information interface provides the operational information required to coordinate data taking, data transfer, processing, monitoring, and downstream workflows across the boundary. This information includes run-lifecycle events, data-availability notifications, state information, identifiers, timestamps, schema information, and data-integrity metadata.

Information originating in E0 is propagated to E1 as events and metadata associated with the data-taking stream. Run-lifecycle information includes transitions such as run preparation, start, pause, resume, and end. A run represents a logical data-taking interval associated with a given ePIC configuration and state; it does not define a file boundary. Data-availability events indicate when TFs or STFs become available for downstream handling. These events allow E1 services to initiate data management, processing, monitoring, and other workflow activity in response to changes in the state of the experiment and the arrival of data.

The information interface must define the contract for each event type, including the authority that emits the event, its identity, ordering, timestamp, schema version, and delivery semantics. Consumers must also know whether an

event is durable, whether repeated delivery is possible, and what completeness guarantees apply. Stable identifiers and sequence information are required so that consumers can detect missing, duplicated, or out-of-order events.

TF and bunch-crossing identifiers must be monotonically increasing and unique for the lifetime of the ePIC experiment. The mapping between TFs and bunch-crossing ranges must be available so that data requested at either granularity can be associated with the corresponding TF. Written TFs are immutable, while run-period records may overlap; consequently, a TF is not necessarily associated with a single run period.

State information exchanged through the information interface follows the state model defined in Section 5. The information interface exposes state and state transitions to consumers but does not change the authority responsible for the underlying state. Consumers on both sides of the boundary must interpret state, identifiers, and lifecycle events consistently.

The detailed transport mechanisms used for these information exchanges remain to be defined. Possible implementations include messaging systems, service interfaces, or messages that reference externally stored payloads. The interface definition should specify the information contract independently of the transport technology used to carry it.

8 Calibration and Conditions Databases

Calibration and conditions information forms part of the E0–E1 interface because calibration workflows may consume data from E0, execute in E1, and produce updated constants or conditions information that must be made available to downstream processing and, where required, returned to E0.

The interface must define the data produced and consumed by calibration workflows, together with the format, metadata, versioning, and validity intervals of calibration objects. It must also define the mechanisms by which calibration and conditions information is exchanged between E0 and E1 and made available to processing workflows.

Calibration workflows fall into three broad classes: standalone calibrations, such as laser or pulser measurements performed between fills; physics-driven calibrations that depend on reconstructed tracks, vertices, or particle-identification information; and iterative calibrations that depend on previous constants or reconstruction–calibration feedback loops. Dependencies between calibration workflows must be represented explicitly, including circular dependencies where they occur. Tracking provides a prerequisite for many of the physics-driven calibrations.

The requirements for conditions information differ between E0 and E1. E0 requires mechanisms for gathering conditions, selectively populating conditions information, fast loading, and access to E0-local sources such as raw conditions and slow controls. E1 requires robust distributed read access for many clients and controlled write access for a smaller number of authorized producers. Conditions information may include calibration measurements, selected slow-controls parameters, detector alignment information, and other data required to interpret or reconstruct detector data.

It remains to be determined whether E0 and E1 will use a single distributed conditions database or separate but synchronized services. The mechanism by which calibration results produced in E1 are propagated back to E0 also remains to be defined. Regardless of the implementation, ownership, versioning, validity intervals, and provenance of calibration and conditions objects must be preserved across the interface.

Calibration workflows may execute in E0 or E1 depending on their latency, data-access, and resource requirements. Where practical, calibration processing should be placed in E1 so that it can operate on the data available through the E0–E1 interface without introducing unnecessary processing into E0. Detector experts must retain appropriate control over the calibration procedures and constants associated with their detector systems, independent of where the processing executes.

Calibration latency spans several timescales, from prompt feedback on the order of seconds, through run-dependent updates on the order of minutes, to delayed or quasi-online processing on the order of hours. These workflows must therefore be coordinated with the E0–E1 data flows, state model, and latency requirements defined in Sections 4, 5, and 6.

9 Control Interfaces Between E0 and E1

The E0–E1 control interface governs requests originating in E1 that may affect detector or DAQ operation in E0. Typical use cases include adjusting detector thresholds or configuration in response to monitoring, calibration, or processing results produced in E1.

Control across the E0–E1 boundary should be expressed as attributable requests or intents rather than as direct writes to E0 control systems. E0 remains authoritative for detector and DAQ operation and is responsible for validating whether a requested action is permitted in the current operational state.

A control request should therefore pass through a sequence of distinct stages, including policy evaluation, approval where required, precondition checking, execution, and readback of the resulting state. The interface must preserve the identity of the requester, the requested action, the decision made by E0, the execution status, and the observed outcome.

The communication mechanism used to carry control requests remains to be defined. Possible implementations include messaging systems, service interfaces, or messages that reference externally stored payloads. The control contract should be defined independently of the transport mechanism.

Security and authorization are integral to the control interface. Access to E0 control functions must remain subject to E0-local policy and enforcement, including authentication, authorization, and protection against invalid or unsafe actions. Control authority is not transferred to E1 by exposing an interface to E0 services.

The E1-to-E0 control path remains an open area of the interface design. Its definition must be coordinated with the state model in Section 5, the information interface in Section 7, and the requirements for calibration and conditions workflows.

10 AI Readiness and AI-Enabled Control

AI readiness for the E0–E1 interface is the capability of the detector, computing, data, and associated workflows to support the development, integration, and sustained operation of AI-enabled services. It therefore requires more than access to data or the ability to train an AI model. The interface must provide the information, state, control, provenance, and operational mechanisms needed for AI-enabled components to participate reliably in experiment workflows.

FAIR principles apply across the full data lifecycle represented by the E0–E1 interface. This includes data produced by the DAQ, the metadata and state required to interpret those data, their transfer and orchestration by the WFMS, calibration and conditions information, processing outputs, and the provenance needed to relate derived results back to their inputs and operational context.

The DAQ and WFMS therefore contribute directly to the FAIR data foundation of ePIC. The DAQ must preserve the identifiers, metadata, state, and provenance needed to interpret the data it produces, while the WFMS must preserve and propagate this context as data are transferred, processed, calibrated, and made available to downstream consumers.

The E0–E1 interface is particularly relevant to closed-loop workflows in which information derived from monitoring, calibration, or downstream processing in E1 can be used to improve subsequent detector operation and data taking in E0. Such feedback must follow the information and control interfaces defined in Sections 7 and 9; AI-enabled components do not bypass those interfaces or their authority and safety requirements.

AI readiness at the E0–E1 boundary includes access to experimental and operational data for inference and validation, inference capabilities that meet the required latency, mechanisms for uncertainty handling and fallback, clearly defined ownership and lifecycle management of AI services, safe control and actuation interfaces, and appropriate human oversight.

AI-facing services may use an access layer such as MCP to expose authoritative system information and permitted actions. Such an access layer does not own the underlying state or transfer control authority. State ownership,

authorization, policy enforcement, and execution remain with the systems responsible for those functions.

AI interactions with the E0–E1 boundary can be classified into four stages:

1. **Observation**: access to system state, monitoring information, logs, calibration data, and other operational information;
2. **Recommendation**: generation of a proposed action or configuration change based on the available information;
3. **Approval**: evaluation of the recommendation by a human, policy mechanism, or other authorized decision process; and
4. **Action**: execution of an approved request by the authoritative control system, with the result recorded and made available for readback.

Consequential actions affecting detector or DAQ operation remain subject to the control model defined in Section 9. AI-generated recommendations do not bypass E0-local policy, authorization, precondition checking, or deterministic safety enforcement.

All AI-assisted actions should be attributable and traceable. The interface should preserve the relationship between the observations used to form a recommendation, the recommendation itself, the approval decision, the resulting control request, the action taken, and the observed outcome. This provenance is required both for operational accountability and for evaluating the behavior of AI-enabled control systems.

AI readiness also requires sustained operational support for deployed AI services. This includes monitoring for model and data drift, handling out-of-distribution inputs, maintaining documented fallback behavior, defining responsibility for model development and deployment, and establishing procedures for monitoring, maintenance, and incident response. Human oversight remains an integral part of this operational model, particularly for actions that can affect detector or DAQ behavior.

11 Reference Implementation

The streaming workflow testbed provides a reference implementation of selected E0–E1 interface concepts described in this document. It is used to exercise data transfer, state propagation, workflow orchestration, monitoring, processing, and AI-enabled interaction with production-oriented services. The reference implementation is non-normative: the technologies and implementation choices used by the testbed do not constrain the final E0–E1 interface.

The testbed represents E0 with a simulated DAQ that provides the run lifecycle, TF and STF data, and associated state metadata. The E1 reference implementation is built around Rucio for data management, PanDA for workload management and workflow execution, and agents that use MCP to access system state, coordinate workflows, and invoke permitted operations across the testbed.

The reference implementation exercises both prompt and fast processing paths. In the prompt path, a run-imminent event creates the corresponding Rucio dataset, arriving STFs are registered and attached to the dataset, and processing tasks consume the data as they become available. The implementation can associate data with the individual E1 sites according to data-management policy.

The fast-processing path exercises finer-granularity access to recent data. TF-scale samples are derived from STFs, divided into slices, and distributed to a standing pool of persistent processing workers. The current testbed uses persistent EICrecon processing and streaming-oriented data delivery for this path. These implementation choices exercise the low-latency processing concept but do not prescribe the final TF transfer mechanism across the E0–E1 interface.

State information accompanies the data and operational messages in the testbed so that downstream components can associate processing activity with the corresponding data-taking state. This provides a practical environment for exercising the state semantics and transition rules defined in Section 5.

The reference implementation also exposes system information and permitted operations through an MCP-based service layer. The service provides access to information about runs, STFs, agents, workflows, logs, PanDA, and Rucio, and is used by both interactive clients and AI-enabled agents. This allows the testbed to exercise information access and control workflows without making MCP itself part of the normative E0–E1 interface.

The testbed further exercises the distinction between observation, recommendation, approval, and action described in Section 10. Current implementations include attributable action records, AI-generated proposals with human approval, deterministic execution, and credentialed operations. These mechanisms provide experience with provenance, authorization, and human oversight for AI-assisted control, while authority and safety enforcement remain with the systems responsible for the underlying operation.

The testbed has also exercised STF-scale processing using official ePIC DIS simulation samples and an EICrecon processing payload. The current samples contain approximately $200\,\mathrm{MB}$ and 40 events per file without zero suppression, with a processing time of approximately $5\,\mathrm{s}$ per event. These exercises provide experience with data organization and processing workflows while the final TF and STF formats remain under development.

The testbed is deployed in a reproducible containerized environment, allowing the interface concepts to be exercised outside its original installation. This provides a path toward integration with future DAQ implementations and toward replacing the simulated E0 components with DAQ-owned components as the E0–E1 interface matures.

12 Open Issues and Roadmap

The E0–E1 interface will be developed and validated as part of the broader Streaming Computing program in ePIC. The target is a bilateral implementation in which E0 and E1 each own and implement their respective side of the defined interface.

The immediate goal for 2026 is to define the requirements and interfaces for the WFMS for streaming data, informed by testbeds and prototypes. The E0–E1 interface definition in this document is one of the principal outputs of that activity. Several aspects of the interface remain to be completed:

- **Raw-data transfer**: We need to determine the transfer mechanisms for the TF and STF data flows. The current STF-based approach provides a working path for both bulk data transfer and fast monitoring and will continue to be evaluated in testbeds and prototypes. The 2027 end-to-end fast-processing workflow and the 2028 Streaming Computing Challenge will exercise these interfaces from simulated DAQ egress through downstream processing. An open question is whether experiment-control requirements, including AI-enabled feedback, impose latencies that cannot be met by the STF path alone. If so, finer-granularity access to TFs or parts of TFs and a streaming transfer mechanism may be required.
- **State model and transition rules**: The complete state-transition model is not part of the E0–E1 specification. It must, however, provide a path for reconciling the E0–E1 state model with the DAQ run-control model and ensure that the state model can be extended as operational requirements evolve. This includes defining how changes in detector or conditions state interact with run boundaries, including which state changes require a new logical run. The proposed factorized state model will be evaluated in 2027 in an operational end-to-end fast-processing workflow, from (S)TF acquisition through streaming reconstruction and monitoring of the reconstructed results. In 2028, the lessons learned from this work will be used to develop a flexible language for defining state-machine transitions.
- **Latency requirements**: We need to define the latency requirements for the different E0–E1 use cases and their measurement endpoints. In particular, the requirements for fast monitoring, experiment control, calibration feedback, and AI-enabled control need to be established. These requirements will determine whether the STF-based path is sufficient or whether finer-granularity TF access and streaming transfer are required.
- **Information interface**: We need to complete the contract for run-lifecycle and data-availability information exchanged across the E0–E1 boundary, including the required identifiers, ordering information, timestamps, schema versioning, delivery semantics, and completeness guarantees. The interface specification should define these semantics independently of the transport mechanism used to carry them. This interface will be exercised in the streaming-data challenges once a functional DAQ prototype becomes available with the

MiniDAQ in FY29 Q3.

- **Calibration and conditions**: We need to define the interfaces between calibration workflows and conditions services, including the mechanisms for propagating calibration results between E1 and E0 and the ownership and validity semantics of calibration objects. In 2027, autonomous triggering and integration with the WFMS for streaming data will be developed. In 2028, this work will be extended to an end-to-end calibration workflow of representative ePIC complexity.
- **Control interfaces between E0 and E1**: We need to complete the definition of the control interface, including the representation of control intents, authorization and policy evaluation, approval where required, precondition checking, execution, readback, and provenance. This work is planned for 2029 and beyond as part of the development of experiment-control capabilities, including AI-enabled control.
- **AI readiness and AI-enabled control**: We need to establish the common infrastructure required to expose information sources, services, and data-lifecycle operations across the streaming system. This work has started with the consolidation of a common computing platform and stable interfaces to AI services and will take time. In 2027, development will begin on an end-to-end AI and data backplane integrated with the streaming DAQ. FAIR principles provide the data foundation for this work and should apply across the full data lifecycle, including DAQ data, metadata, state, provenance, workflow orchestration, calibration and conditions information, and derived data products. Work on this FAIR foundation will continue through the 2027 integration of the AI and data backplane. What remains to be defined is how these FAIR properties are maintained consistently across the E0–E1 boundary and through the WFMS, including common identifiers, machine-readable metadata, provenance links, data and workflow state, and the relationships between raw data, calibration information, processing outputs, and derived products.

The planned development therefore progresses from interface definition and initial prototypes in 2026, through integrated end-to-end demonstrations in 2027, to first system-level validation in the 2028 Streaming Computing Challenge.