

EIC Controls System Architecture Status and Plans

James Jamilkowski

EIC Controls Systems Group Leader, Brookhaven National Laboratory, US

ICALEPCS 2023, Cape Town, South Africa

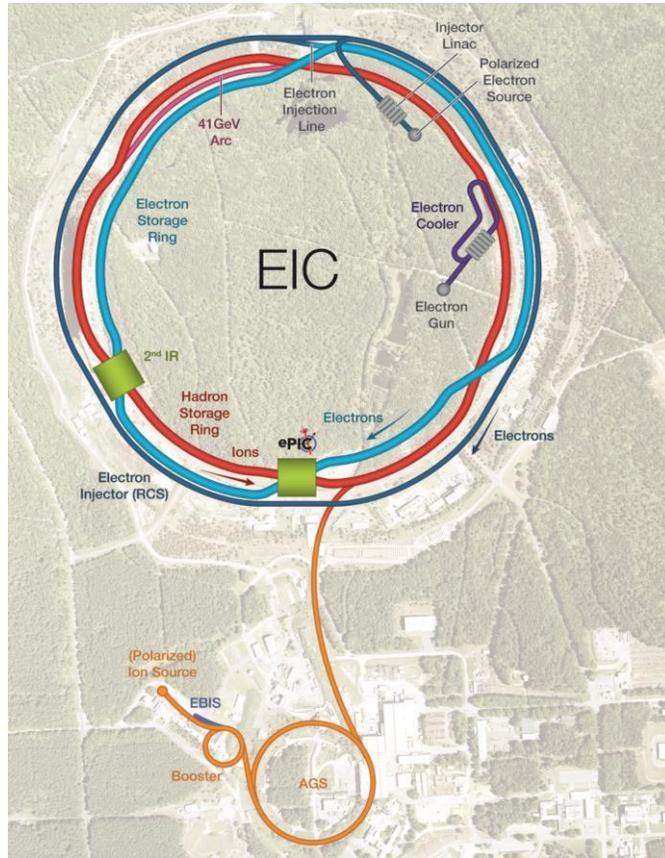
October 9, 2023



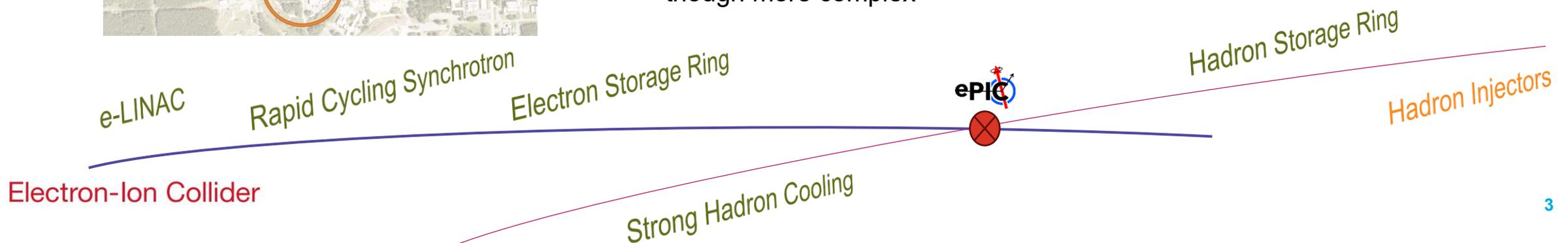
Outline

- What is the EIC?
- Reimagining the RHIC Controls System
- EIC Software Strategy
- Software Challenges
- Software Architecture
- EIC Hardware Strategy
- Hardware Challenges
- Common Platform Hardware Design
- Timing System Architecture
- Summary

What Is the Electron Ion Collider?

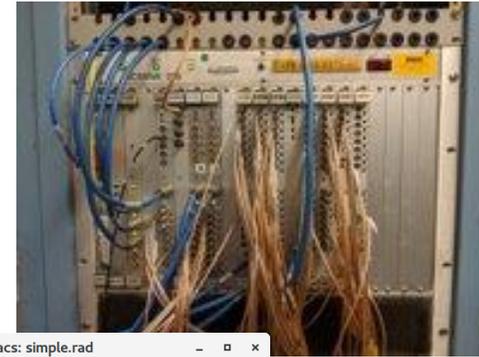


- Relativistic Heavy Ion Collider (RHIC) operations end in **FY25!**
- The EIC Project is based on the existing RHIC tunnel and certain equipment with many additions, extensions, and improvements
- Our goal is to provide a **tool for probing the nuclear structure**
- Polarized beams of hadrons and polarized beams of electrons will collide inside an experiment's detector, called **ePIC**, with an option for an additional detector in a second location
- **Electrons** generated in a new **turn-key LINAC**, boosted in the **Rapid Cycling Synchrotron**, and stored in the **Electron Storage Ring**
- **Hadrons** will be transported from the existing injector chain to the **Hadron Storage Ring**, the former of which is outside of the scope of the EIC
- A separate **Strong Hadron Cooling** electron machine commissioned later in the project will improve luminosity by shrinking the emittance of the hadron beams
- Start of construction decision currently expected in **FY25**
- Commissioning of first element starting in **FY28**, Project complete by **FY34**
- Machine Protection requirements will be comparable in speed to RHIC, though more complex



Reimagining the RHIC Controls System

- RHIC has a robust controls system that has met all beam operations objectives over the last 30 years
 - some software we may reuse/adapt
- The MCR is **not** planned to be replaced, **nor** will the injector chain for hadron beams receive an upgrade in a similar timeframe
- Much of the RHIC controls are based on the older VME standard, which significantly limits performance
 - VxWorks 5.5 and 6.5 are currently used on many Front-Ends, though a small portion run Linux
- The RHIC Controls System includes core proprietary elements like Accelerator Device Objects (ADOs)
 - Based on CDEV library
 - Provides the familiar Synchronous Get/Set, and Asynchronous Get protocol over TCP
 - The pool of experts is small, and **attrition is an important concern**



```
EMacs: simple.rad
File Edit View Cmds Tools Options Functions Buffers C++ Help
[Icons]
simple.rad
//!- simple.rad -----
ABSTRACT = a simple;
CONCRETE = simple;
VERSION = 1.8;
DESC = Examples of parameter types, features and techniques;

ADOTYPE = manager;

OVERVIEW = {
This ADO is a special example class.<br>
It contains examples of various types of parameters, features and techniques which
an ADO developer may find useful.<br>
It can be compiled for either manager or fec environments.<br>
Two instances are typically configured into the system.<br>
<B>simple.test</B> is a manager instance managed by the simpleMan manager and
<br>
<B>simple.fec</B> is an instance which can run in an FEC<br>
} // ENDCODE

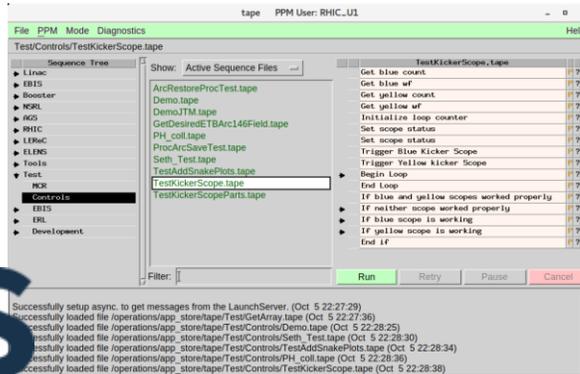
MEMBERDATA = {
unsigned int activatedPpmUsersCount;
unsigned int *activatedPpmUsers;
int outOfTolerance;
int alarmSent;
Adolf* serverMan;
AsyncSetup* setup;
ADO_RW_PROPERTY *something;
}

UTF8-99% 12 CO Top simple.rad (/vobs/ados/simple) (C++/1 Font Abbrev)
Scanning buffer... (100%) done
```

EIC Software Strategy

EPICS

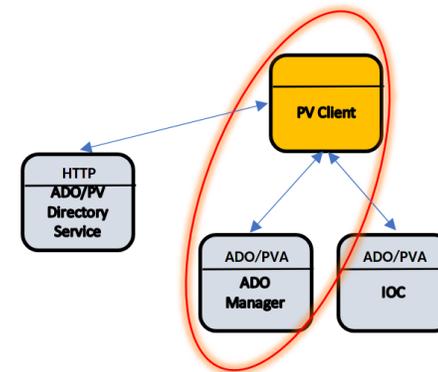
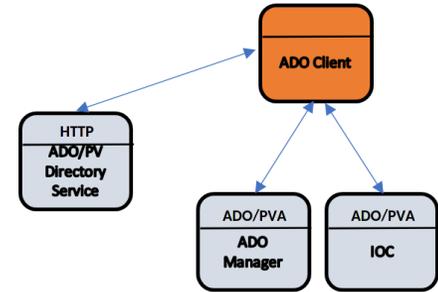
cs|studio



- **Identify RHIC software** that may continue to have utility for EIC and **update** those cases
- **Adopt EPICS v7** as part of the core infrastructure
 - streamline development
 - enable greater collaboration with the wider community
- **Evaluate** existing RHIC and EPICS tools for core services
 - Name Lookups, Alarm Distribution, Data Logging and Retrieval, Electronic Logkeeping etc.
- **Expand use of Python** at multiple levels of the stack
 - Popular amongst many developers on the team
 - Influenced by positive past experiences with package management
- **Incorporate AI/ML support** using a **non-proprietary design**

Software Challenges

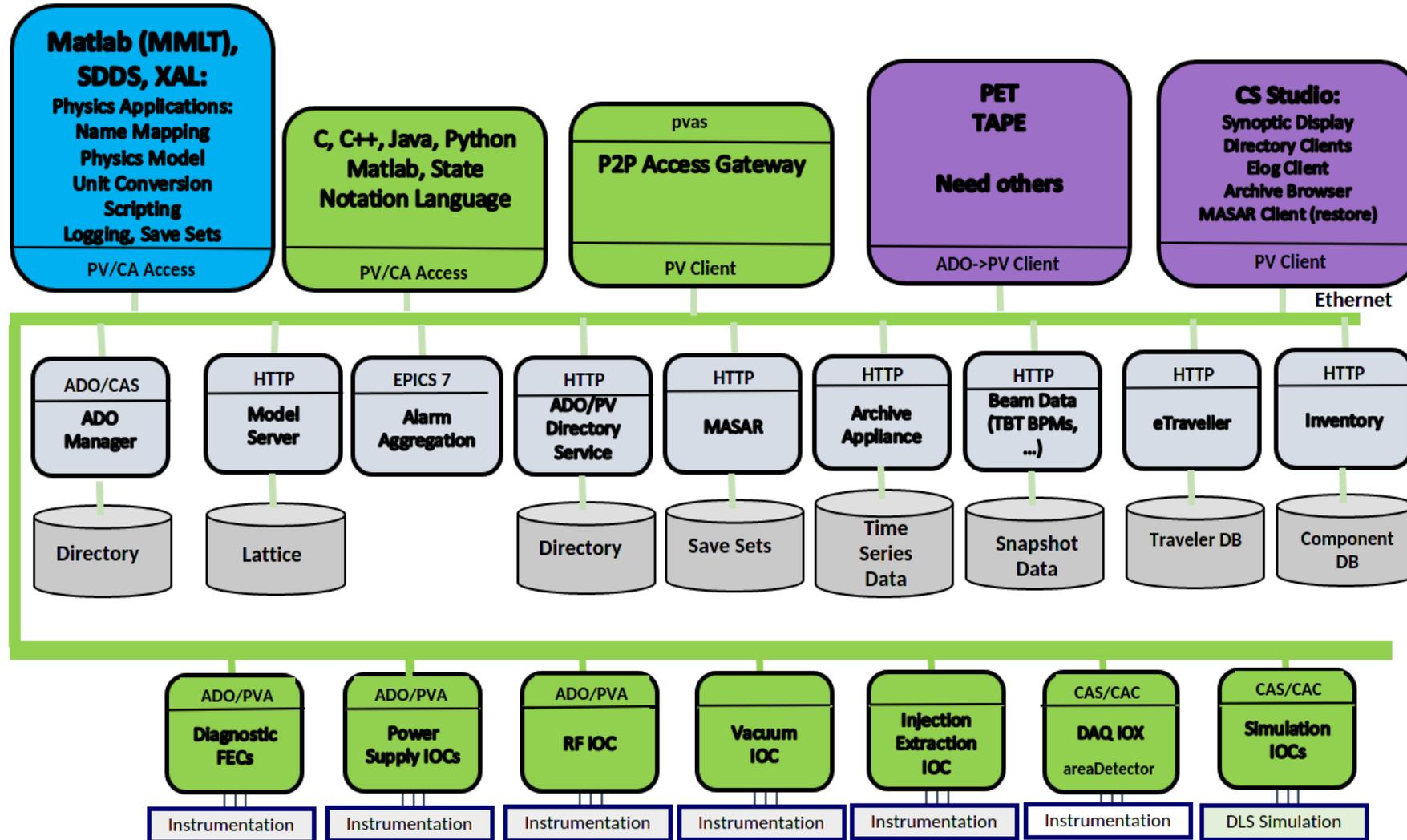
- **Evaluating** extending support for RHIC software interface to equipment (ADOs) to **provide bridge** to existing systems, where they could interoperate **using PVXS protocol**
 - basic demo in 2021
- ADOs and EPICS 7 tools share many common threads but not all edges line up neatly
 - Alarm metadata
- C, C++, and Java code will clearly **still need to be developed and maintained** for both the RHIC and EPICS codebases
 - Concerns about attrition amongst experienced core developers



A screenshot of the AlarmDisplay window. The window title is "AlarmDisplay". It has a search bar with "Name: gpm.lhp-snd35-max" and "All Fields:" next to it. There are buttons for "search", "Last 10 Minutes", and "Advanced Search". Below the search bar is a table with columns: Row, Timestamp, Name, Level, State, Description, Action, PPI, Machine, Type. The table contains two rows of data.

Row	Timestamp	Name	Level	State	Description	Action	PPI	Machine	Type
1	Oct 03 2023 13:47:07	gpm.lhp-snd35-max:dataM	1	cleared	range error	erase	2	Linac	ado
2	Oct 03 2023 10:50:34	gpm.lhp-snd35-max:dataM	1	assigned	range error	set	2	Linac	ado

Software Architecture



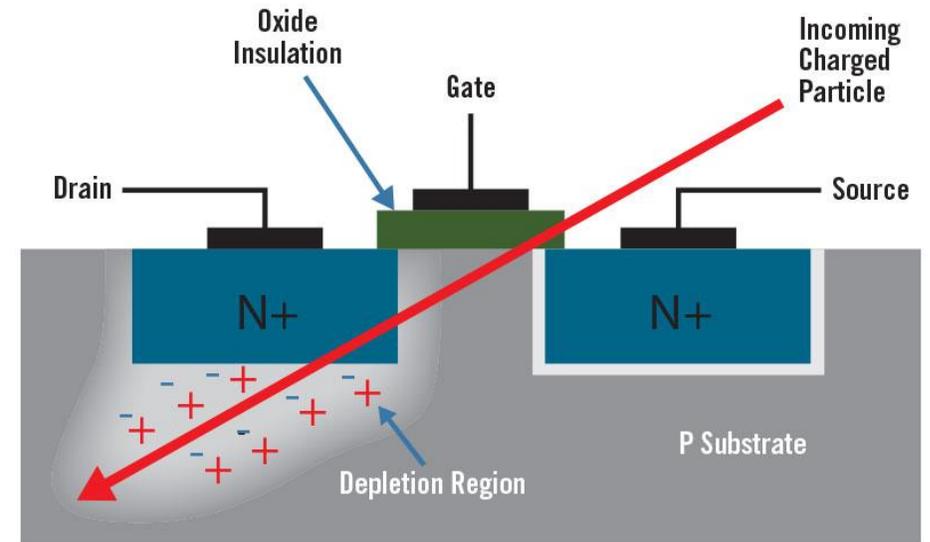
EIC Hardware Strategy



- **Avoid** legacy **VME** bus limitations!
- **Leverage** existing knowledgebase from FPGA-based Front-End Computer developed in 2009 with **modernized components**
- **Avoid using VxWorks** for the OS due to licensing concerns
 - We're reviewing the utility of using RTOSs in our future platform
- **Choose FPGA design** that **maximizes reusability** due to life cycle concerns
- **Build in external clock capability** so that **any crate** is capable of simple event/delay timing or beam synchronous timing on demand
- **Developing a middleware interface** for Front-Ends to **export client load** to scalable host machines
 - Based on RHIC era experience where many crates required workarounds for handling excessive data publishing loads for legacy VME or FPGA-based systems

Hardware Challenges

- EIC will continue to have **Alcove locations** distributed around the Tunnel, which **expose equipment to radiation and Single Event Upsets (SEUs)** during RHIC beam operations
 - Keep complex equipment out of the tunnel wherever possible
 - Some equipment may move to other locations
 - EIC redesigns will likely include external controllers for magnets that rely on fiber connections to equipment that cannot be relocated
- EIC will require **many types of controls hardware components** to be developed
 - Modularity of components can help mitigate the cost in money and effort though the reuse of hardware IP and software interface components
 - The Controls Group is responsible for the platform, and other stakeholders can extend it by developing Daughter and Signal Conditioning cards to meet more custom application requirements



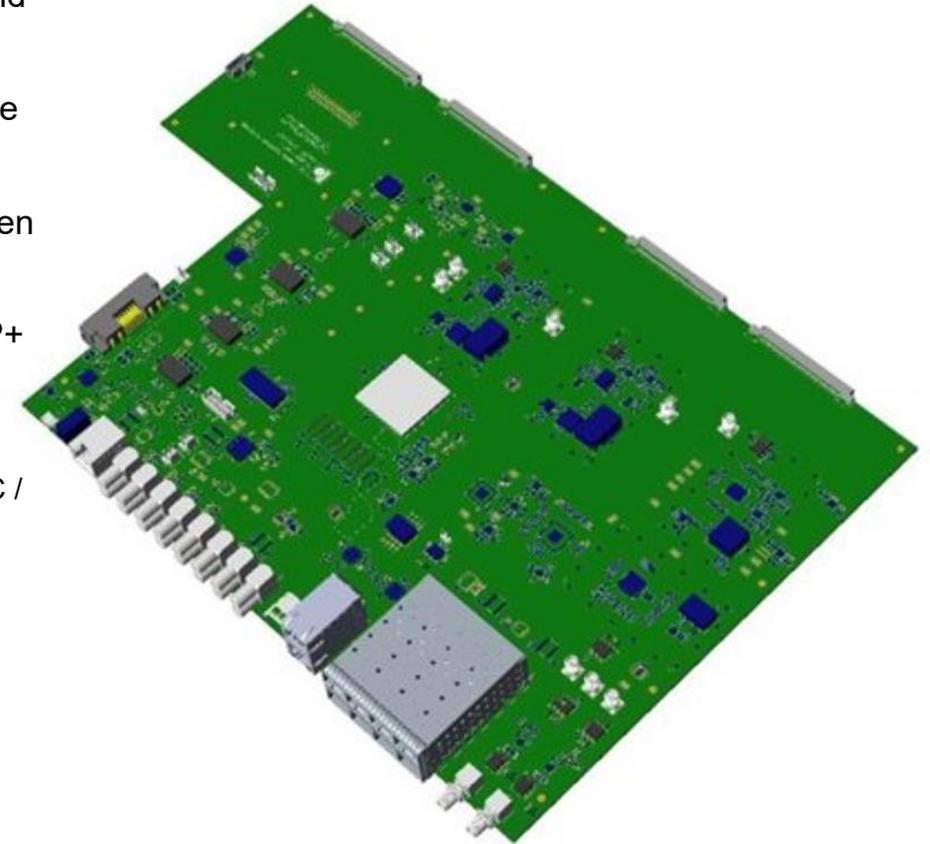
Common Platform Hardware Design



Thermal/Airflow Mockup Chassis



- Pizza-box **2U** form factor can support a wide variety of use cases and installations
- Carrier board with a **Zynq Ultrascale+** and **dual-daughter card** connector
- Daughter cards have optional signal conditioning modules for physical interface standardization and flexibility
- Leverage COTS boards from Daughter Cards using FMC standard connector when desired
- **First article expected in October** for Common Platform Carrier board and SFP+ Breakout Daughter Card
- Additional daughter cards under development so far
 - DIO, Baseband ADC/DAC, BPM, RF ADC / DAC
- Extended Carrier Interface (ECI) module under development
 - Link up to 3 extra chassis with a Carrier board for a total of 8 Daughter Cards
 - 1 Carrier + 1 ECI per extra crate
 - SFP+ connectivity
 - **Improves I/O density and reduces cost per channel** for certain applications



Common Platform Carrier

Timing System Architecture

- A new **serial link** will be distributed around the EIC Complex to Front-End Computers
- Data, event, and timestamp activity are included
- Represents a **higher bandwidth version** of an **existing serial link design**
- Please listen to a talk this afternoon for additional details...

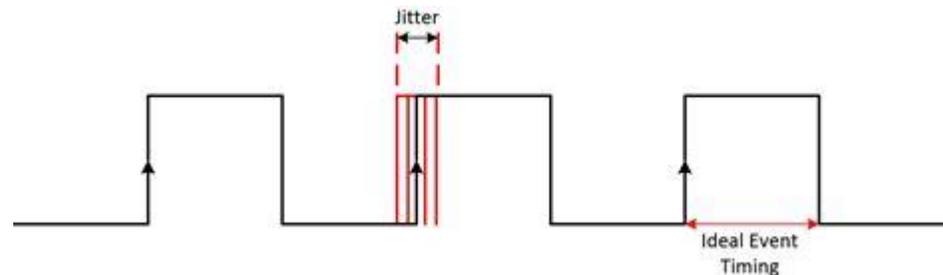
Time: 17:00

Code: MO4AO05

Location: Hall A

Presenter: Paul Bachek

Title: "Development of a Timing and Data Link for EIC Common Hardware Platform"



Summary

- RHIC operations come to an end in FY25
- EIC Project construction and commissioning will continue until FY34 once government approvals have been received
- We're considering significant changes to the software infrastructure for the EIC Controls System
 - EPICS 7 and Pheobus are being evaluated as core components
 - Prototyping support for the RHIC ADO interface via a bridge
 - Will continue evaluating available services, tools, and applications
- A new proprietary Common Platform design for Front-End Computers has been developed
 - We're preparing first article of our initial components

Thanks to:

Samuel Clark, Michael Costanzo, Ted D'Ottavio,
Margaret Harvey, Kevin Mernick, Seth Nemesure,
Freddy Severino, Kunal Shroff, Kyle Kulmatycki,
Christoph Montag, Vahid Ranjbar, Kevin S. Smith
(Brookhaven National Laboratory (BNL), Upton, New
York), Bob Dalesio (Osprey DCS LLC, Ocean City)

Questions?