

# Considerations Involving the High Level Application Services for the EIC Project

EPICS Collaboration Meeting  
James Jamilkowski, Seth Nemesure, Ted D'Ottavio  
Apr 27th, 2023

Electron-Ion Collider

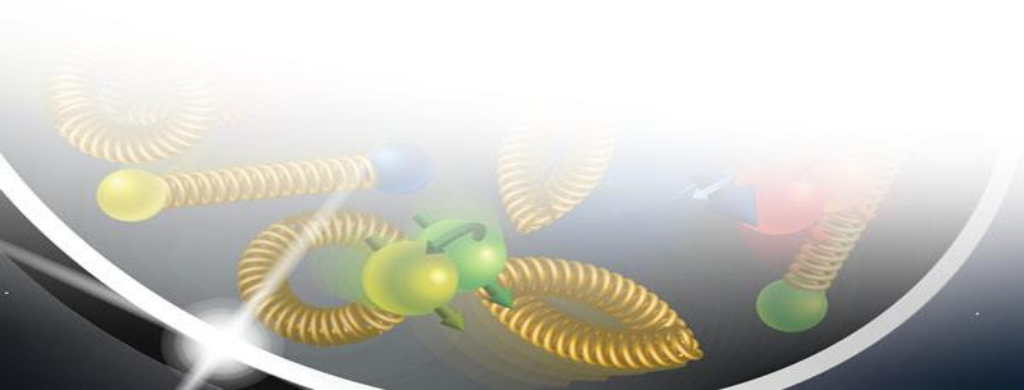


# Outline

- EIC Project Background and Basic Controls Requirements
  - How the EIC Project relates to RHIC
  - Scope of Device Interfaces
  - Data Publishing & Retention
  - Controls Framework Utilization
- Name Lookups
  - Lookup Services
  - Naming Convention(s)
- Time Series Data Logging & Retrieval
  - Logged Data Trends at RHIC
  - Logging Infrastructure Scaling
  - Predicting Retrieval Performance Expectations
- Alarm Notifications
  - Alarm Metadata Handling
  - User Expectations for Alarms
- Command Sequencing Tools
  - Intro to Tape Sequencer Application
  - Use of Tape at RHIC and NSRL Facilities
  - Ideas for Integration and Expansion
- Hybrid Controls Strategies
  - Hybrid Integration Environment Diagram
- Development Environment Strategies
  - Supporting Codebases
- Future Plans
  - Areas Requiring Attention
  - Additional Areas of Interest



# EIC Background and Basic Controls Requirements



# How the EIC Project Relates to RHIC

	RHIC	EIC
Operating Period	2000 - 2025	~2032 - 2050s
Machines	Blue & Yellow Rings (LReC, CeC)	HSR, ESR, RCS, Linac, SHC
Spin Physics Program	Part-time (p <sup>^</sup> )	Most of the time
Collisions	Hadrons, same or mixed species	Hadrons / electrons
Beam Cooling	Add-ons for injection and store	At injection and store
Footprint / Circumference	RHIC tunnel, 2.4 miles	>RHIC tunnel, 2.4 miles
Beam Experiments (Initial)	4	1*
Buildings (incl. Storage, Cooling)	44	62 <sup>t</sup>

\* "EPIC" Detector, on-project  
<sup>t</sup> as of April 2023

# Scope of Device Interfaces

- RHIC Blue and Yellow Rings, CeC, and LEReC (eCooling systems) currently support ~ 70k Accelerator Device Objects (ADOs)
  - Proprietary controls system interface format with similarities to EPICS
  - Each ADO instance hosts several to >1k of I/O parameters, comparable in function to EPICS Process Variables
  - Additional interfaces via CDEV objects for services (ex. Online modelling)
  - Total control points currently, ~29.5M, though approximately 1/6<sup>th</sup> or ~5M are parameter values that may be of interest for logging purposes
- EIC is expected to support
  - Hadron Storage Ring (HSR) will roughly be equivalent to RHIC Blue + Yellow Ring, even though segments will not be used due to increases in support equipment
  - We're adding new machines: Electron Storage Ring (ESR), Rapid Cycling Synchrotron (RCS), eLinac, and Strong Hadron Cooling (SHC)
  - Total device types, ~60
  - Total device instances, ~8000

# Data Publishing & Retention

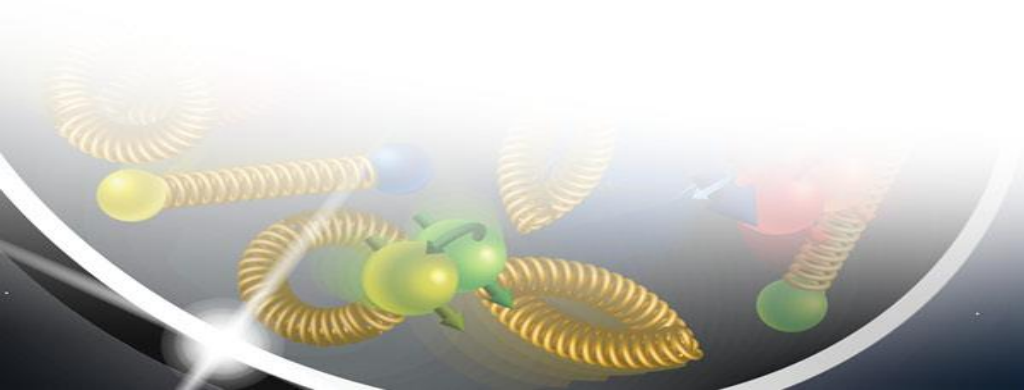


- RHIC systems
  - Time-series data storage for last run, **~247 TB with compression**
  - Logging system supports large volumes of scalar and array data
  - Use of data compression is a requirement
  - Includes a tiered data retention policy, though few systems are assigned to a category where data is removed or permanently culled
- EIC expectations
  - A factor of **~20** increase in raw data volume
  - **~5 PB** of data stored per year for long-term use
  - The system will run for 20 – 30 years
  - We must seriously consider applying **strict retention policies**
  - Concept of capturing "golden" datasets over very limited periods for documentation of optimal running conditions to support long-term comparisons of optimal conditions (ie. Opt-in)
  - Snapshot data for normal/off-normal event capture
  - AI/ML data processing during the collection process
  - Certain datasets will be considered mission critical, and others will not be (especially after a few weeks)
  - Streaming data from the Detector is not included

# Controls Framework Utilization

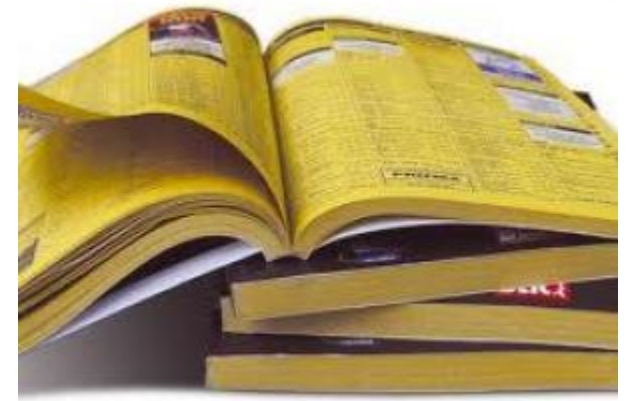
- For EIC, we're anticipating using EPICS for...
  - A turnkey eLinac
  - Most systems associated with the new Electron Storage Ring (ESR), Rapid Cycling Synchrotron (RCS), and Strong Hadron Cooling (SHC) machines
  - Our new Front-End "Common Platform" (though an interface for ADO format is also expected to be developed in parallel)
    - For more info, see the presentation on the topic, *Background and Development Status of the EIC Common Platform*, abstract #139 (Tues 4/25 @ 11:40)
  - While re-use of HSR equipment and software interfaces was part of the CDR, the latest plans call for upgrades to the newer HW/SW platform
- Certain legacy systems may still utilize the BNL C-AD ADO interface, though the overall scope has not been determined yet
  - Unlikely to include modelling resources or interfaces to RF, power supplies, instrumentation, etc.

# Name Lookups





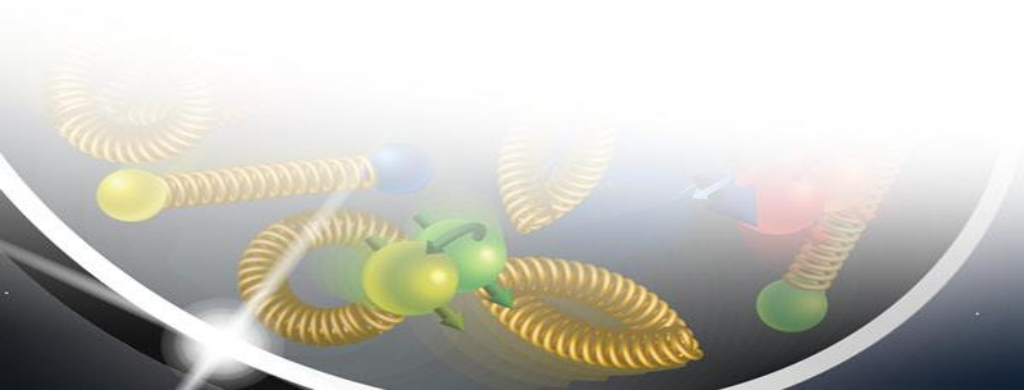
# Lookup Services



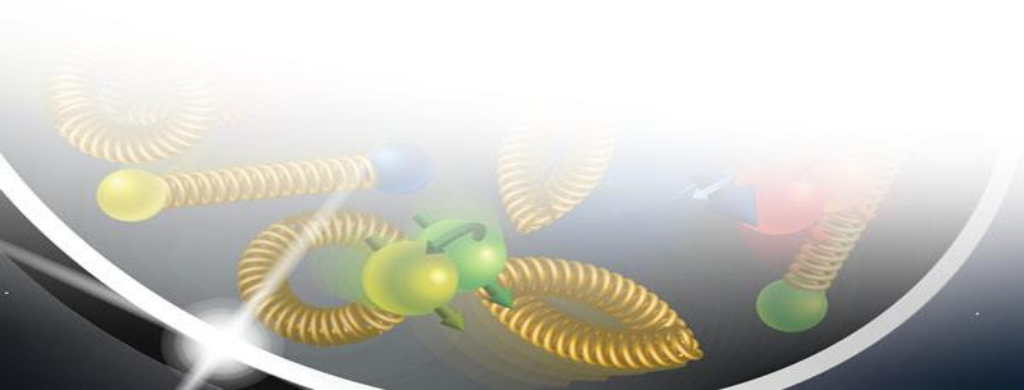
- How best can we handle a hybrid controls system?
  - Users deal with only one set of names and one naming convention that covers all EIC machines
- Options to consider
  - RHIC currently relies on the Controls Name Server, which supports ADOs and CDEV objects
    - This infrastructure has scaled well over the last 25 years, partly due to the improvements in server and networking performance
    - Unlike EPICS scheme, all I/O transactions in the ADO/CDEV environment involve interactions with CNS
    - Supports up to two unique names per device as a default, and can be extended using aliases
    - Can we reasonably extend CNS to support EPICS PVs, and how would that potentially scale to fit the project needs?
  - We have started exploring the capabilities of Channel Finder
    - If necessary, can we add a module to incorporate support for ADOs?
      - This would open us up to a wider integration with the tools and infrastructure supported by the EPICS community
    - Are there any scalability / performance concerns given the number of control points anticipated for EIC?

# Naming Convention(s)

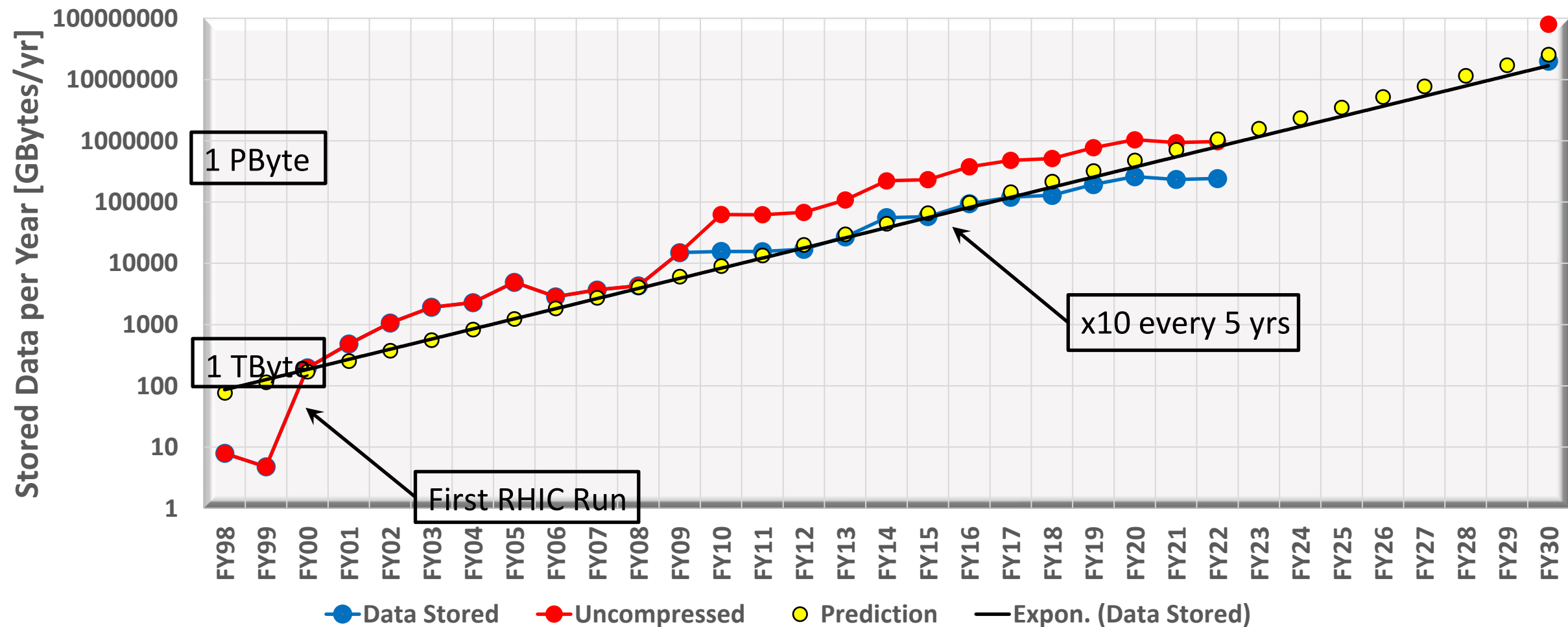
- ADO and EPICS names tend to follow different standards
  - This has been addressed in the separate workshop presentation, *Experiences Adopting EPICS from a New User Perspective for EIC*, abstract #140 on Mon 4/24 @ 3:00
- Technical differences limit the acceptable characters for ADO or PV names that might be shared by both frameworks
  - Resolving the impact of any conflicts could be costly in terms of development and testing, if this isn't truly necessary



# Time Series Data Logging & Retrieval



# Logged Data Trends at RHIC



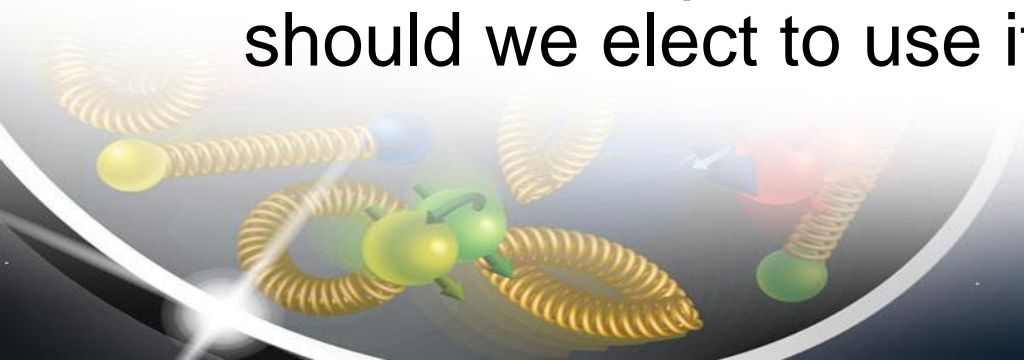
# Logging Infrastructure Scaling



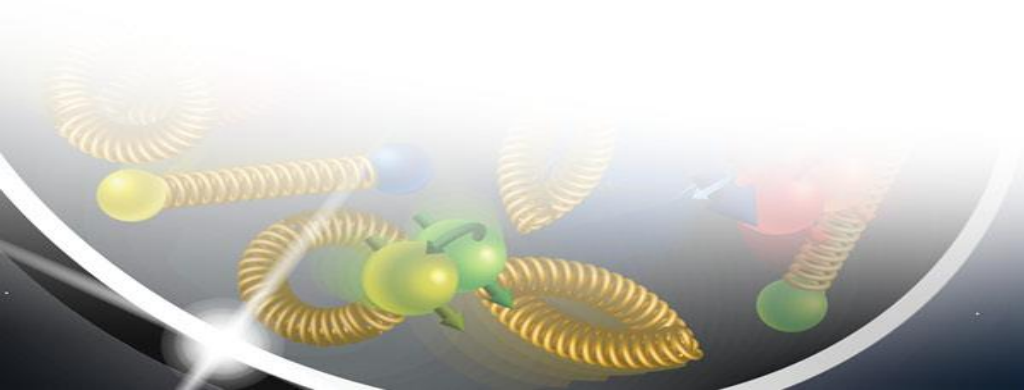
- What logging system design will be required in order to handle the number of PVs and a  $\sim 10\text{Hz}$  standard data collection rate?
  - Some datasets will be deemed critical, so some level of redundancy will likely be required at the logging infrastructure level
  - If we give users the flexibility to adjust data collection scope, rates, or retention policies, what would the risk be to maintaining reasonable levels of performance and limiting storage utilization?
- The Archiver service is an attractive option
  - It supports all basic needs: rate and retention policy control, three levels of configuration supporting data age and performance based data management mechanisms
  - Modular components make it easier to add-in updates for new features as needed (ex. ADO data collection support)
  - REST is a well-used API, already familiar to multiple stakeholders

# Predicting Data Retrieval Expectations

- Current RHIC performance, ~10 sec maximum response time for recent/limited datasets
  - We should aim to at least maintain a comparable retrieval performance for EIC users
- Archiver capabilities encourage optimization of both storage and retrieval performance for the most recent datasets which we will employ regardless of the implementation
- What kind of performance can we expect for cloud storage, should we elect to use it for older data?



# Alarm Notifications



# Alarm Metadata Handling

- EPICS and ADO alarms have slightly different implementations

EPICS	ADOs
NO_ALARM	OK
MINOR_ALARM	NOTIFY_DEBUG
MAJOR_ALARM	NOTIFY_INFO
INVALID	NOTIFY_NOTICE
	NOTIFY_WARNING
	NOTIFY_ERROR
	NOTIFY_CRITICAL
	NOTIFY_EMERGENCY

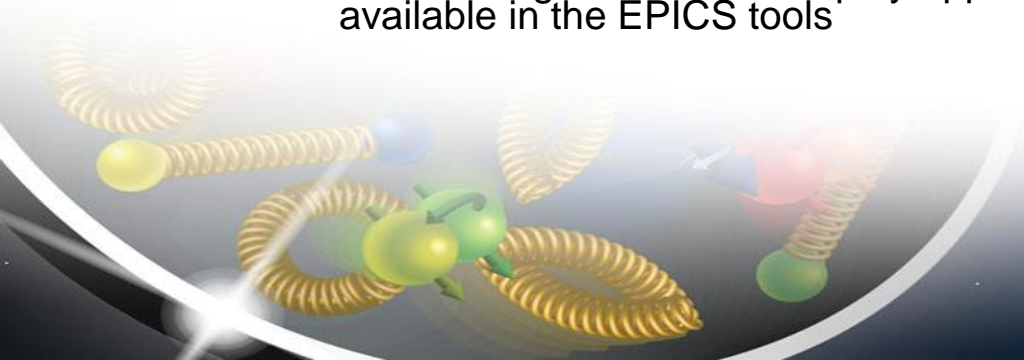
If needed, how might we map the two schemes without complicating functionality?



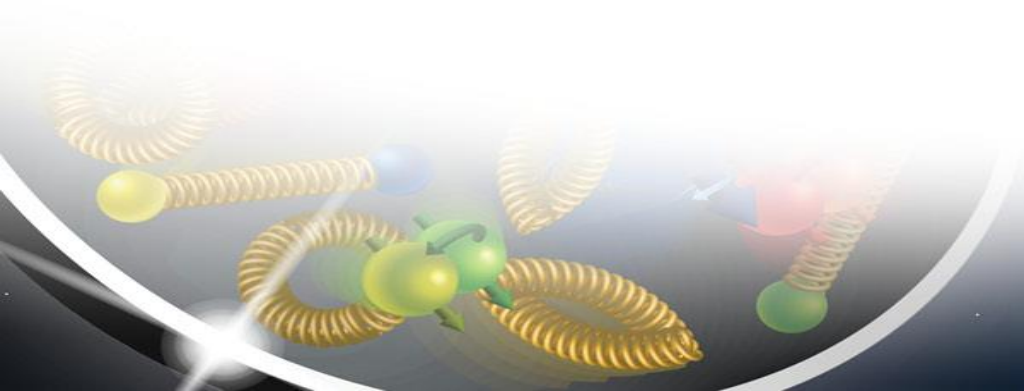
# User Expectations for Alarms



- The existing RHIC ADO alarm scheme does not meet the expectations of Operations
  - Way too many alarms are generated
  - **1.78M** separate annunciations In FY21, associated with **22,129** distinct control points
    - The majority of alarm activity was associated with a small portion of "chattering" cases
  - The main problem is that alarm properties need to be enabled at the ADO level (ie. Opt-in), which leads to inertia and an overreliance on developer-level management
    - Once alarming is enabled though, users can be given easy access to threshold settings
  - Filtering at the UI level is available, but it's not as practical as the users would like it to be
  - Tools exist to shunt alarms directly to responsible experts via an Opt-in email/texting notification service
    - effort is still required to scrub the Operations-level alarms on a per-ADO class basis
- We need to start with a better philosophy from day one at EIC
  - Dissociate alarms from the ADO / IOC processing level, which seems to be the standard treatment in EPICS
  - Collaborate with stakeholders to review alarm activity concerns on a periodic basis, and apply changes
  - Give stakeholders outside of Controls the tools to contribute to alarm management
  - The existing RHIC AlarmDisplay application lacks a hierarchical display mode, which is both attractive and readily available in the EPICS tools

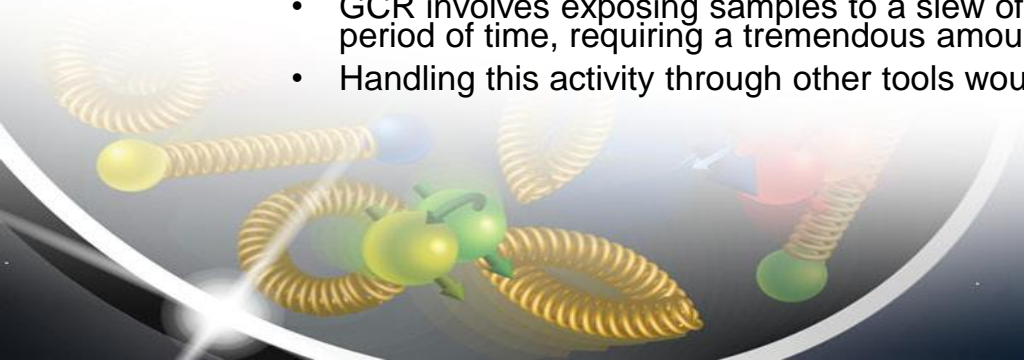


# Command Sequencing Tools



# Intro to Tape Sequencer App

- Early in the history of the RHIC project it was identified that machine management required the use of scripted events, that were modular in nature
  - The RHIC operating cycle has multiple phases (injection, ramp, store, dump, refill)
  - Each last from minutes to many hours and require complex sets of changes to be applied in a coordinated fashion at each transition
  - Plain scripts only get you so far
    - Poor visibility of statuses
    - Versioning concerns
    - Diagnostics require extra work
    - Modular design isn't necessarily encouraged
- Tape Sequencer was developed to fit the unmet needs of RHIC Operations
  - GUI for loading, running, pausing, stopping sequences of commands, as well as the ability to skip or pause on specific steps
  - Tree-based interface for grouping main sequences and sub-sequences, encouraging modularity
  - Provides run status, active step, error messaging, along with troubleshooting information in a connected log interface
  - Sequences can be edited via a text editor with a simple syntax OR using a GUI
  - Selected commands can be off-loaded to a server
- Demands from the NASA Space Radiation Laboratory program lead to further Tape sequence development related to their Galactic Cosmic Ray operating mode
  - GCR involves exposing samples to a slew of particle species and beam energies that might be encountered in space over a short period of time, requiring a tremendous amount of reconfiguration for elements of the CAD Injectors and Preinjectors at each step
  - Handling this activity through other tools would be virtually impossible



# Tape Sequencer Interface

```
File Edit Search
/operations/app_store/Tape/RHIC/Up.tape

Find: Replace with:

[TYPE] tape -version 5

NAME Pre-ramp checks

NAME Check if running low energy
TASK Language::Tree::Reference
TAGS -Tree_Reference RHIC/Ramp/NoRamp/RequireNotLowEnergy.tape.check

NAME Already "Prepped?"
TASK Language::Conditional::DeviceIf
TAGS -Device ssop.switches -Property prepStatus5 -Comparison not equals -value Prepp

NAME Notify
TASK UI::Popups::ShowMessage
TAGS -Message SSOP indicates that you may not have run PrepBeforeFill.tape.

NAME If Yes
TASK Language::Conditional::If
TAGS -Expression equalStr($BUTTON_SELECTED,Yes)

NAME PrepBeforeFill.tape
TASK Language::Tree::Reference
TAGS -Tree_Reference RHIC/PrepBeforeFill.tape

NAME End if
TASK Language::Conditional::EndIf

NAME End if
TASK Language::Conditional::EndIf

NAME Check LEReC
TASK Language::Tree::Reference
TAGS -Tree_Reference LEReC/StatusCheckForRHIC.tape

! NAME Check Fundamental Damper inserted
TASK Other::Device::Wait
TAGS -Device fundDamp56MHz -Property posLSStateM -value Fully Inserted -Timeout 2

NAME Check for beam
TASK Language::Tree::Reference

Loaded 30 Kbytes from /operations/app_store/tape/RHIC/Up.tape.
```

tape PPM User: NSRL\_jon\_U3

File PPM Mode Diagnostics Help

Sequence: Test/Controls/Demo.tape

Sequence Tree

- AGS
- Booster
- NSRL
- RHIC
- Tools
- Test
  - MCR
  - Controls

ConditionalTest.tape  
DelayedTasks.tape  
**Demo.tape**  
EnergyDoseChild2.tape  
EnergyDoseTest.tape  
LinkedHoldTest.tape  
LockBBQ.tape  
PSCheck.tape  
QDtest.tape  
RestoreBBQB1Conditions.tape  
RestoreBBQB2Conditions.tape  
RestoreBBQConditions.tape  
RestoreBBQY1Conditions.tape  
RestoreBBQY2Conditions.tape  
SaveState1.tape  
SaveState2.tape  
SetBBQB1Conditions.tape  
SetBBQB2Conditions.tape  
SetBBQConditions.tape  
SetBBQY1Conditions.tape  
SetBBQY2Conditions.tape  
StoreBBQB1Conditions.tape  
StoreBBQB2Conditions.tape

Filter: \*.tape

Run Retry Resume Cancel

Demo.tape

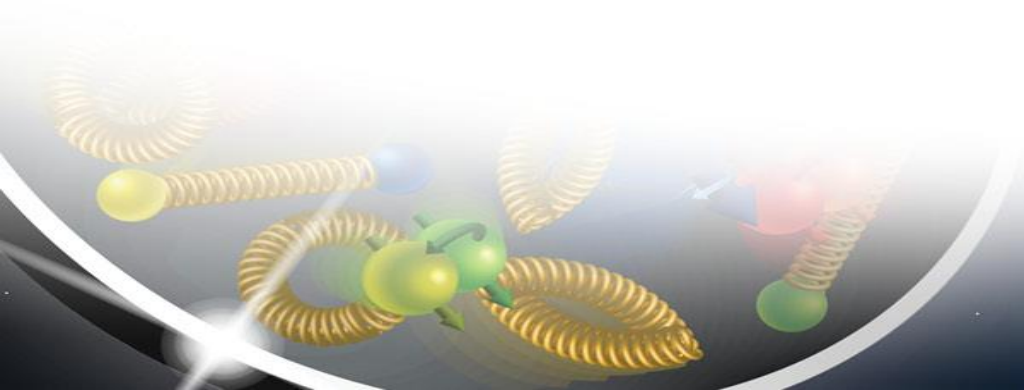
Open Device File	P ?
Load Device/Prop Names from File	P ?
1 Begin Loop	P ?
Wait for 1 Second	P ?
Read the Device1 Value	P ?
<b>Read the Device2 Value</b>	P ?
Dump Variable Values	P ?
Compare the 2 Device Values	P ?
Wait for 1 Second	P ?
End Loop	P ?
Indicate Finish	P ?
Diagnostic Variable Dump	P ?

Running Node: Read the Device1 Value (Sep 10 10:34:28)  
Running Node: Read the Device2 Value (Sep 10 10:34:28)  
**ERROR: Sequence Paused due to an error at: Read the Device2 Value (Sep 10 10:34:28)**

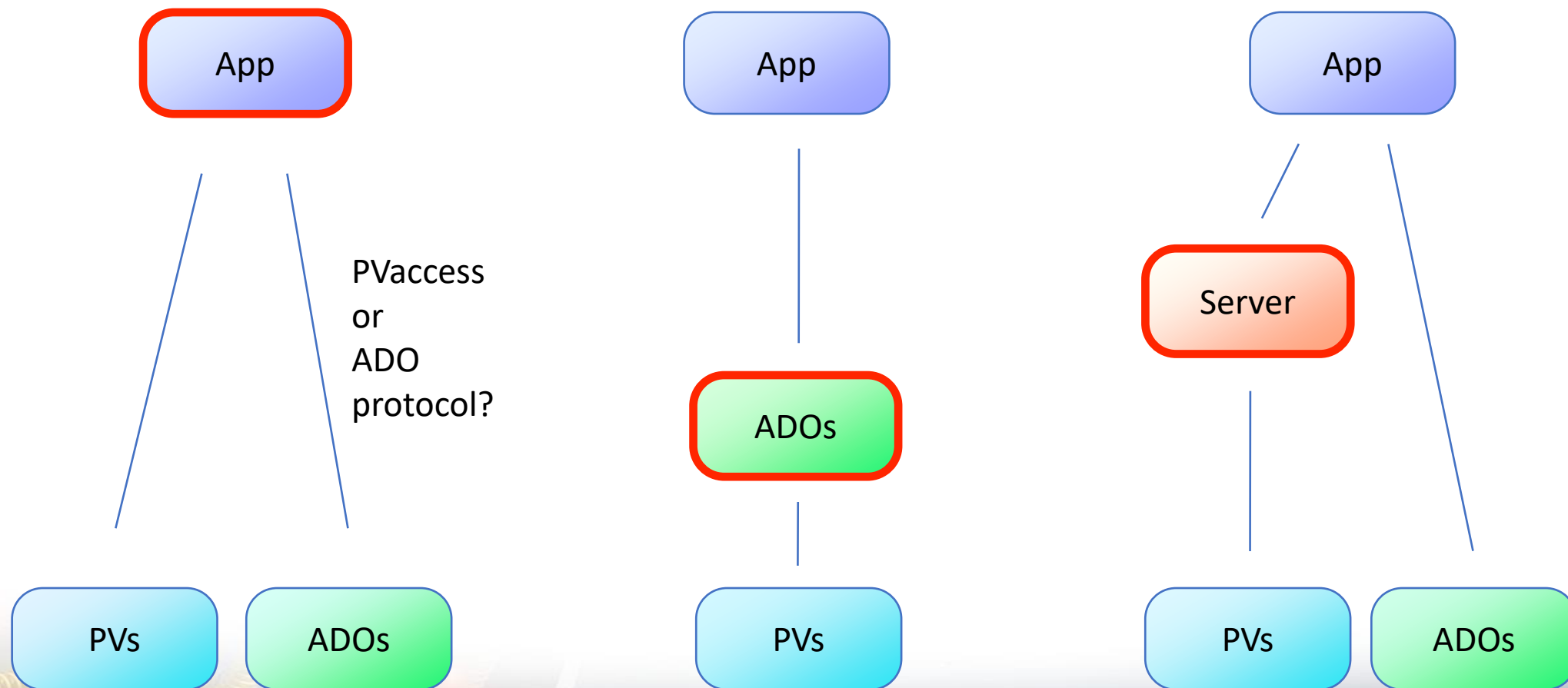
# Plans for Tape Development

- Move to an App/Server-backend operating model
  - App provides UI, and commands are processed by the server
  - Better support for multi-user environment
    - Management of active sequences from afar
    - Better handling of dueling sequence activation
  - Could support alternate interfaces, including web
- Consider adopting an existing interpreted language for composing sequences
  - Python is the leading candidate
  - Built-in functionality for math, logic, strings, arrays, other
  - Good tools for editing, building, debugging sequences
  - Can be extended to incorporate task-like functions and classes
- Sequencing will be critical for EIC, since the operating modes for each machine will be complex and intertwined
  - Our hope is that Tape Sequencer can be made available to the EPICS community
  - We're interested in learning about potential use cases that we may not have encountered yet that could be supported

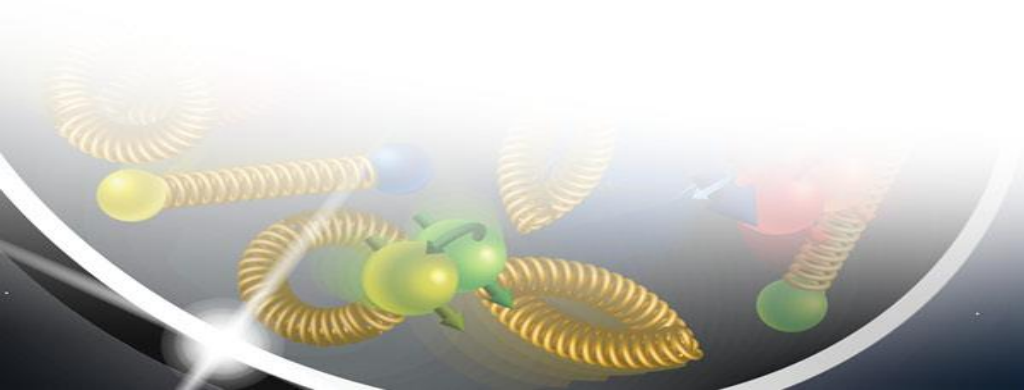
# Hybrid Controls Strategies



# Hybrid Controls Environment Diagram



# Development Environment Strategies



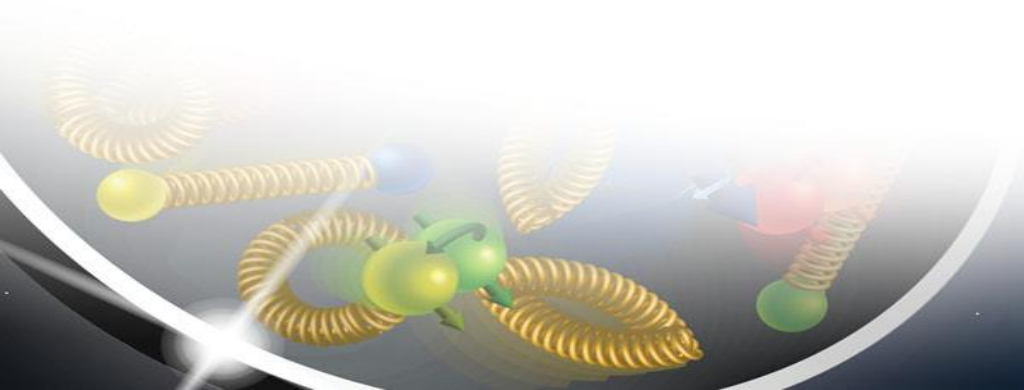


# Supporting Codebases



- The expectation is that EIC software/firmware version control will be handled using git
  - RHIC C++, Java, and Python software currently uses a combo of ClearCase and GitLab repos, though plans are being developed to transition fully to the latter
    - Treating applications as projects, and libraries as submodules has been demonstrated to be viable development model
    - Makefiles are still used to manage the build process
  - We're also receiving positive input on GitHub Enterprise, which may help with the CI/CD processes needed for EIC
- RHIC Python applications have been developed on a versioned packaging scheme in order to enforce policies and avoid issues related to Python, libraries, or OS updates
  - Templates available for developers to create new projects
  - This model provides suitable performance when Anaconda is locally installed
  - Are EPICS Python apps given a similar treatment?
- We also need to consider support for alternative languages, including Rust and Julia
- MatLab will need to be supported, perhaps using a HTTP gateway as is supported for RHIC ADOs

# Future Plans



# Areas Requiring Attention

- SW strategies are needed for the rest of the HLA areas
  - Parameter interface app
  - Synoptic display app
  - Timed Archives / Monitoring Setting Activity
  - Snapshot data handling
  - Inventory control / Assembly & Integration tracking
  - Electronic logs
  - Cross-app integration
- Documentation needs to be developed or extended
  - Prototype details
  - Performance and Interface Requirements
  - Performance testing

# Additional Areas of Interest

- Understand the AI/ML needs and possible implementations
  - FEC-level
  - Centralized Service-level
- Data storage and retrieval options
  - Local NAS + Cloud resources?
- Adding Pulse to Pulse Modulation (PPM) support to EPICS in a standardized fashion
  - Other facilities have a proprietary solution, but it might be advantageous to develop a format that is portable enough to be used in most environments where it might be useful
  - See my presentation, *Experiences Adopting EPICS from a New User Perspective for EIC*, abstract #140
- Developing a Virtual Accelerator framework

Thanks to...

Seth Nemesure, Ted D'Ottavio, Greg Marr,  
Sam Clark, Kevin Brown, Joe Piacentino Jr.  
(BNL/CAD)

Kunal Shroff, Anton Derbenev (BNL/NSLS-II)

Kay Kasemir (ORNL)

Bob Dalesio (Osprey)

Questions?

