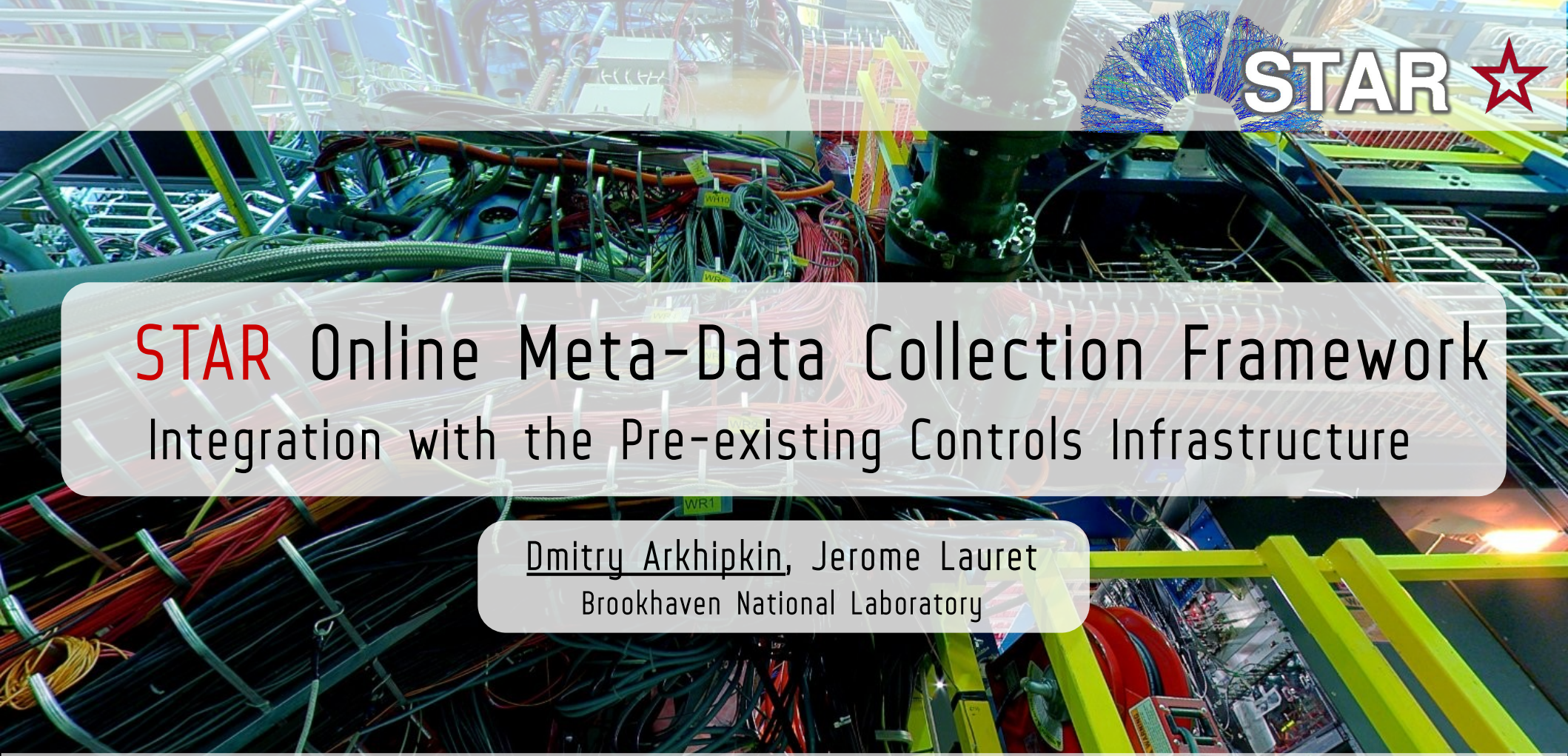


STAR Online Meta-Data Collection Framework

Integration with the Pre-existing Controls Infrastructure

Dmitry Arkhipkin, Jerome Lauret
Brookhaven National Laboratory

Office of Science U.S. DEPARTMENT OF ENERGY

U.S. DEPARTMENT OF ENERGY

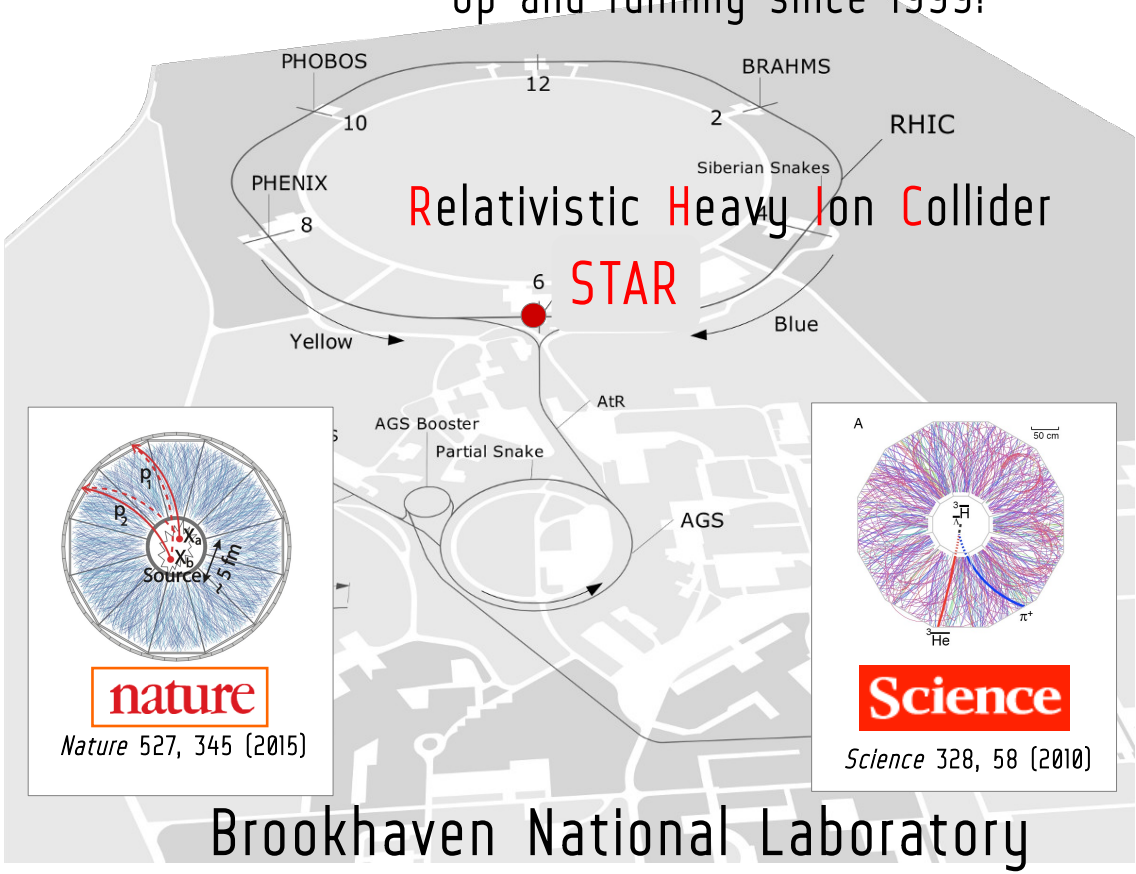
BROOKHAVEN NATIONAL LABORATORY

Outline

- STAR & MIRA Overview
- Controls Upgrade Process
- EPICS<->MQTT
  - Bridging two worlds
  - Deployment overview
- Use-Case Example
  - Alarm Handler Upgrade
  - ALH User Interface
- Summary & Outlook

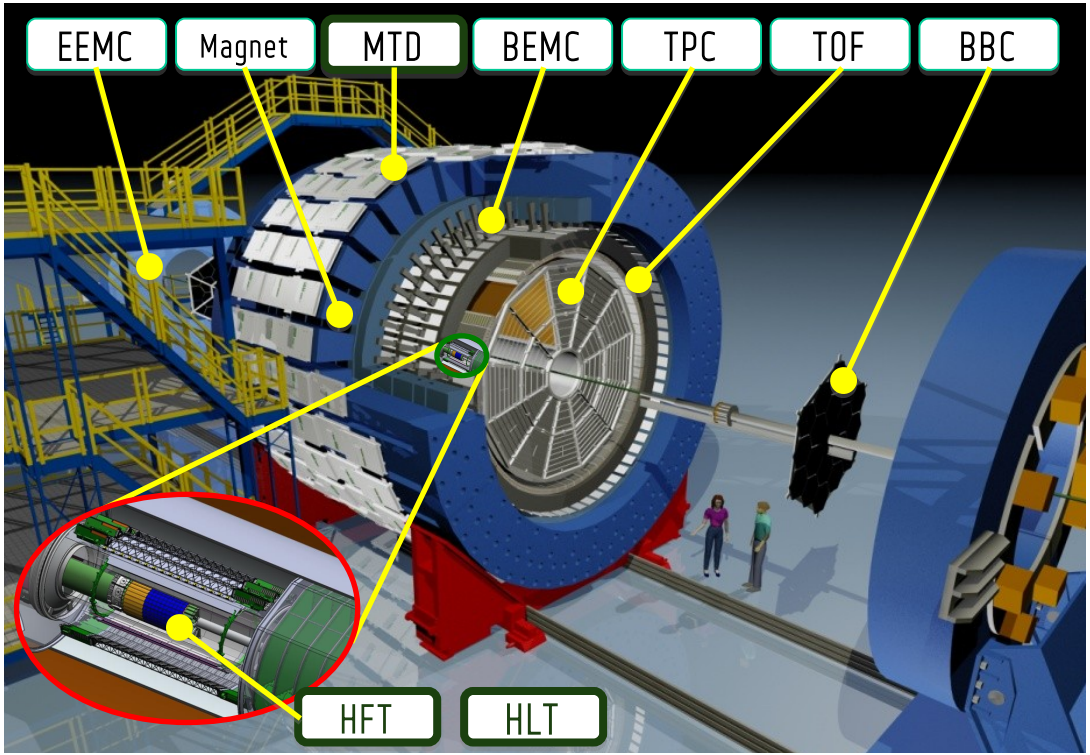
STAR

Up and running since 1999!



STAR at RHIC, Brookhaven Lab

Detector and Control System Evolution



Solenoidal Tracker at RHIC

STAR detector:

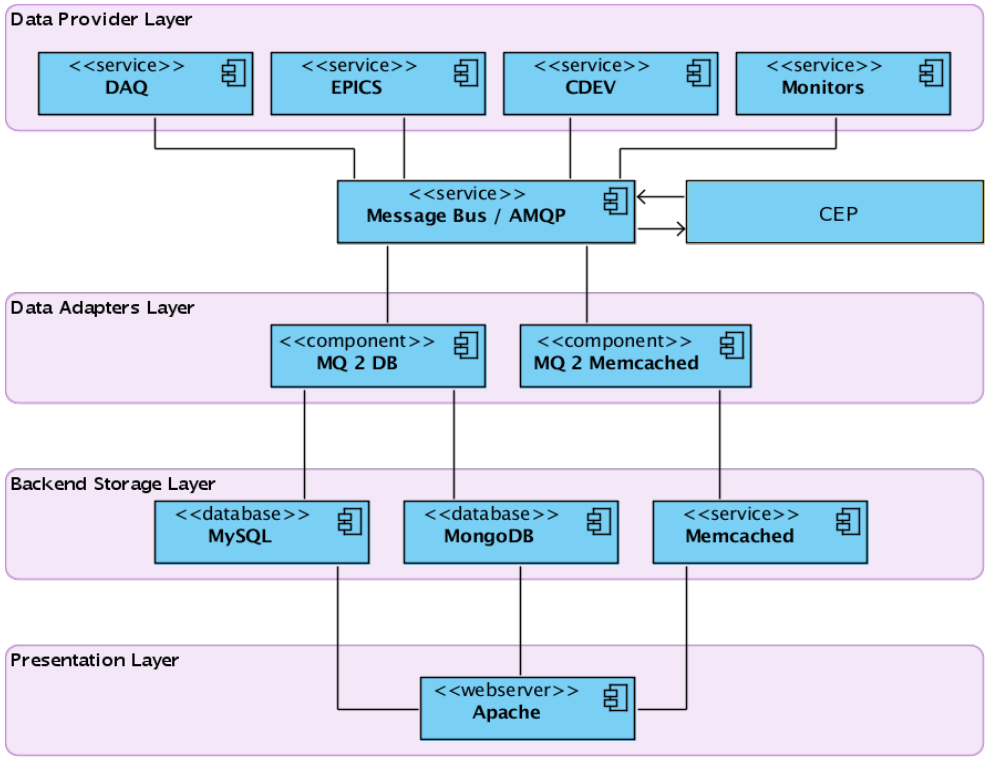
- 18 subsystems, 3 more being added – continual upgrade process
- Slow Control System is EPICS-based, having over 40k process variables at start, ~80k now
- Online Ecosystem includes DAQ, RTS, CS, ALH, Monitoring, subsystem-developed services – all talking to each other
- Since ~2009, obvious need for a modern service integration bus was clear
- MIRA framework was created to answer that need:

D Arkhipkin et al 2012 J. Phys.: Conf. Ser. 396 012002
D Arkhipkin et al 2011 J. Phys.: Conf. Ser. 331 022003

MIRA Framework Overview

Messaging Interface and Reliable Architecture

- Features:
  - Scalable architecture
  - Inter-operable, low-overhead protocol
  - Payload-agnostic messaging
  - Quality of Service regulation
- Originally designed to implement better meta-data collection (archiver) and provide basic service messaging bus
- Implemented using Message-Queuing service bus - AMQP, later MQTT
- Supports Complex Event Processing (CEP)
- With time, expanded to the Control System realm and Alarm Handling



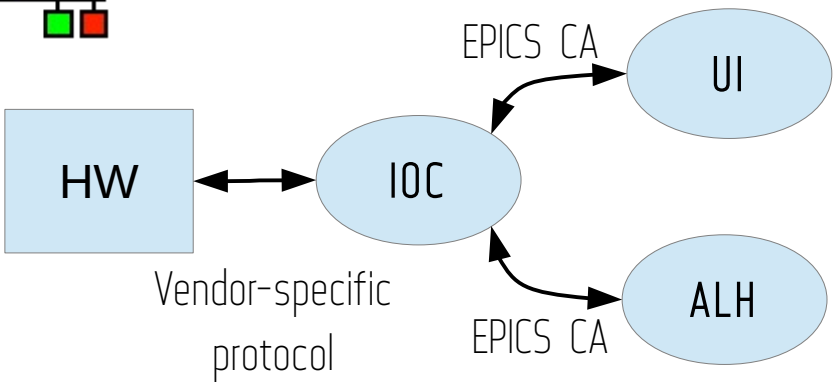
MIRA: basic components overview

D Arkhipkin and J Lauret 2015 J. Phys.: Conf. Ser. 608 012036

Slow Controls: Upgrade

Experimental Physics and Industrial Control System

Basic CS setup: EPICS core



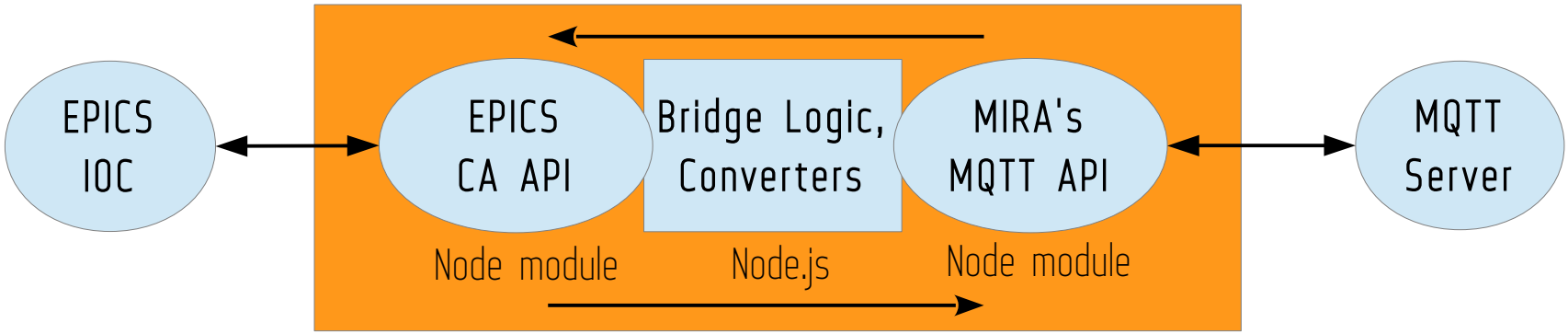
Multiply it by 18 STAR Subsystems
Add in the need to communicate with DAQ, RTS and Archiver
Divide by lacking manpower and subsystem expertise...

- Current Conditions:
  - Current Detector Control System is mostly EPICS-based
  - New hardware is sometimes shipped with EPICS drivers
  - Old hardware is being gradually replaced with new hardware
  - Manpower is sparse, can't upgrade everything at once, so...
- Staged Control System Upgrade:
  - avoid drastic changes in well-defined experimental procedures
  - gradually replace components with newer technologies
  - provide extended functionality and better maintenance for years to come

EPICS to MQTT and back again

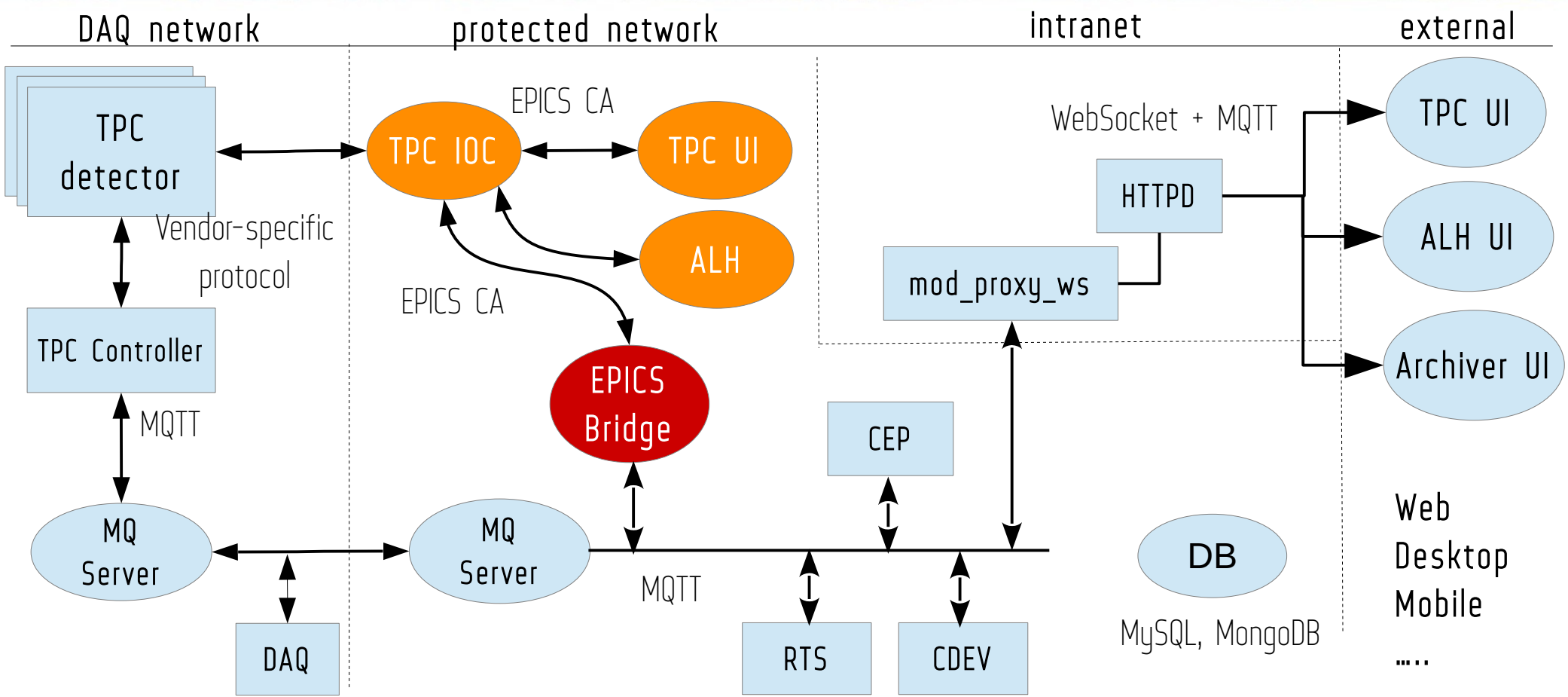
- MQ Bridge = Microservice architecture
  - scalable approach, deploy as many independent instances as you need
  - uses light-weight, technology-agnostic protocol - MQTT
  - enhances cohesion, decreases coupling
  - allows to evolve incrementally, eases maintenance

- EPICS<->MQ Bridge upgrade steps:
  a) keep IOC, keep UI, add EPICS<->MQTT converter
  b) keep IOC, replace EPICS UI with MIRA's web interface
  c) use MIRA's UI, replace EPICS IOC with MQ-enabled IOC emulator



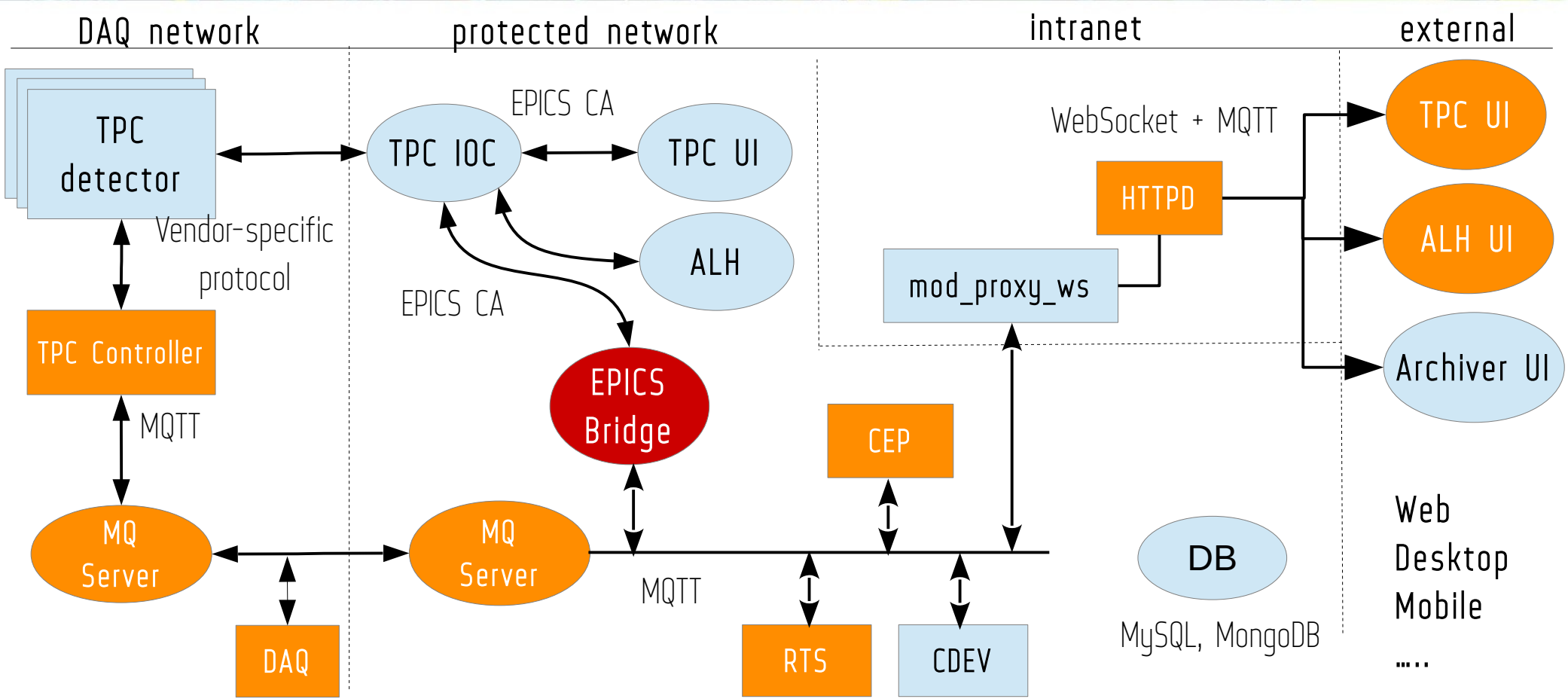
Bridge: step 1

Expose EPICS to MIRA's service bus



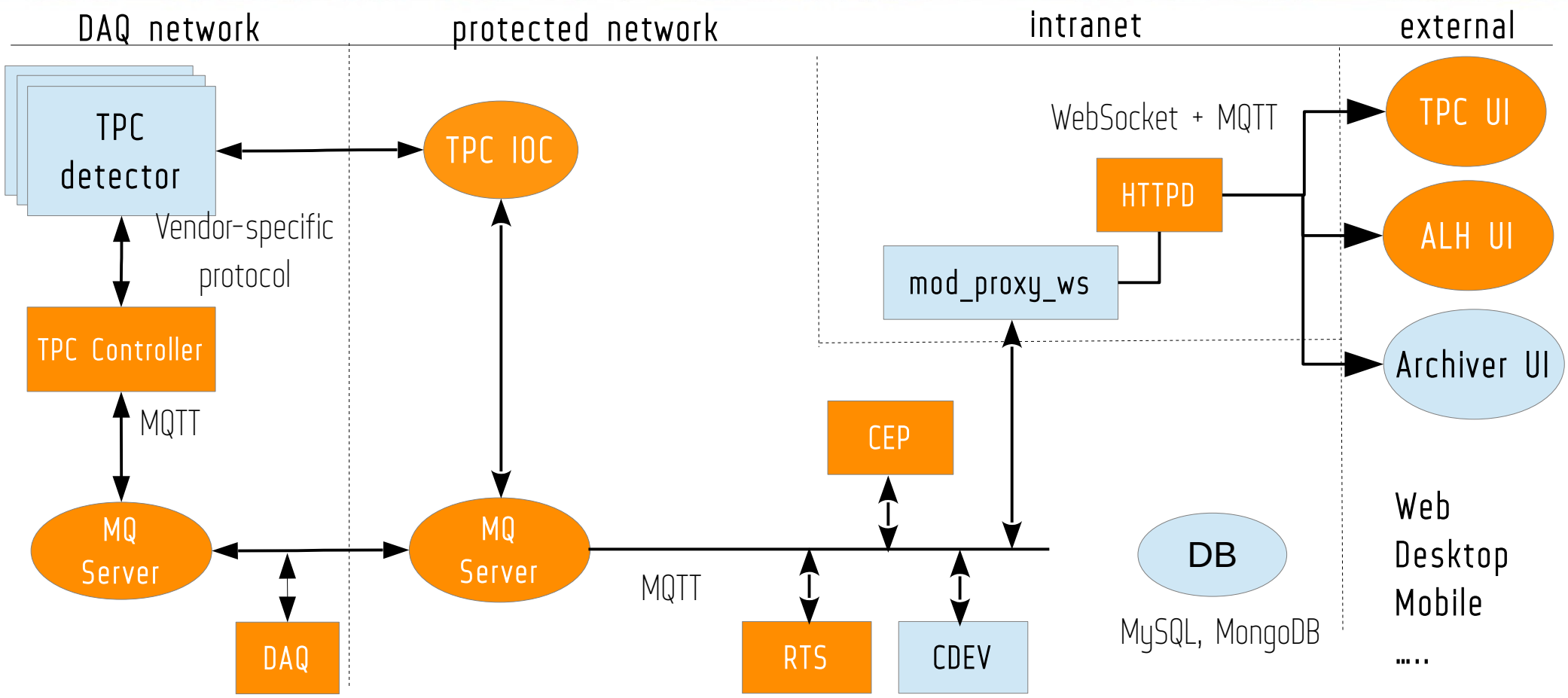
Bridge: step 2

Integrate EPICS Alarms Into MQ bus



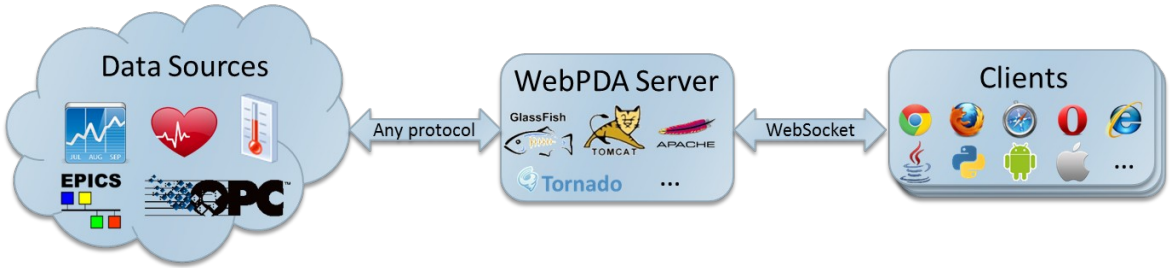
Bridge: step 3

Emulate EPICS IOC with
MQ-enabled Service



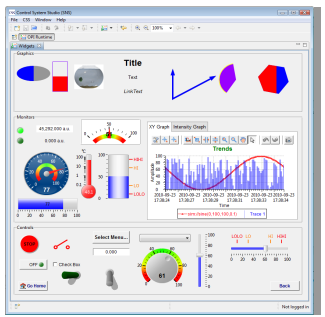
What about other options?

WebPDA is a protocol for accessing real-time data using WebSocket.

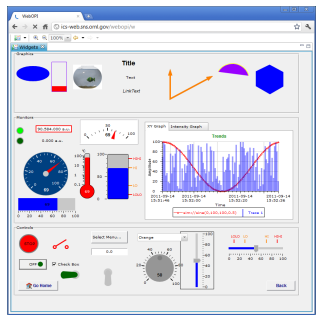


EPICS BOY, WebOPI

EPICS



Java



HTML5

There is a hint for MQTT Support for CS Studio:
https://conference.sns.gov/event/11/session/1/contribution/51

Available tools are mostly focused on "EPICS for Web", rarely on service integration which is very important for STAR.

WebPDA

Great idea, but it requires intermediate java application server, not clear how to allow bi-directional talk of non-EPICS services.

EPICS BOY, WebAPI

EPICS-only service, scalability is questionable, as it creates a separate instance of BOY (potentially, spanning to thousands of channels) per connected client.

Use-Case: Alarm Handler

Now, all alarms are available via MQ bus

STAR Alarm Handler - EPICS:

- MEDM: Motif Editor and Display Manager
- single instance - local application running at STAR Control Room
- configured by plain text file residing at specific node

Upgrade features:

- web-based interface + local app + phone app
- multiple instances – web and local apps attached to the MQ bus
- configured from database back-end, managed by web front-end

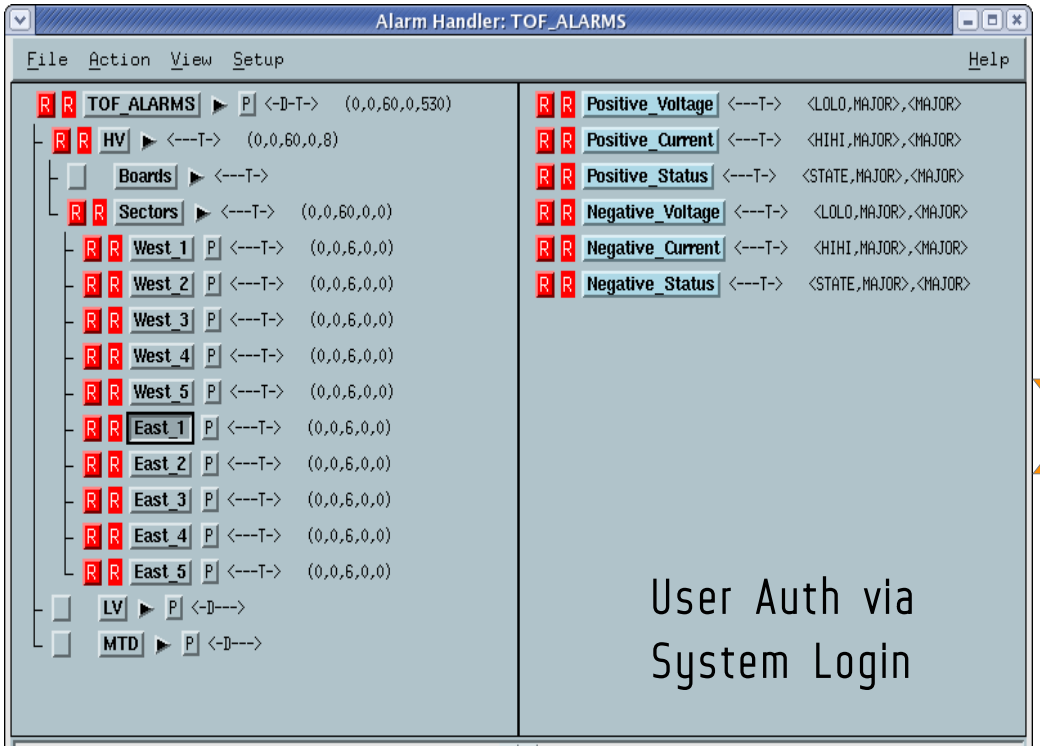
EPICS<->MQTT Bridge Setup:

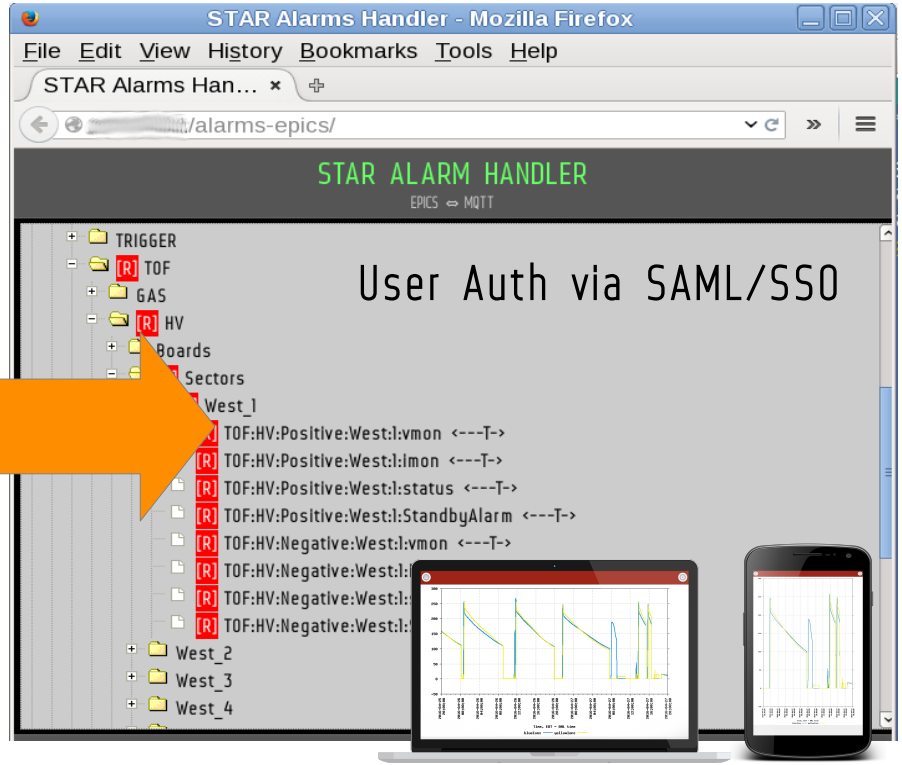
- propagate Alarms only
- Hierachy URI: System/Subsystem/module
- bridging mode: EPICS->MQTT and MQTT->EPICS
- message output throttled (per PV) to ~1 msg/sec
- nuissance alarms suppressed at UI level

Alarm Handler UI Conversion

HTML5 stack provides us with the cross-platform UI

Old UI is being replaced by web-based interface, powered by direct MQTT feed over WebSockets. HTML5-based UI could be deployed as Web, Win/Lin/MacOS/Android/iOS. WebGL could be employed to enable fast widget updates.

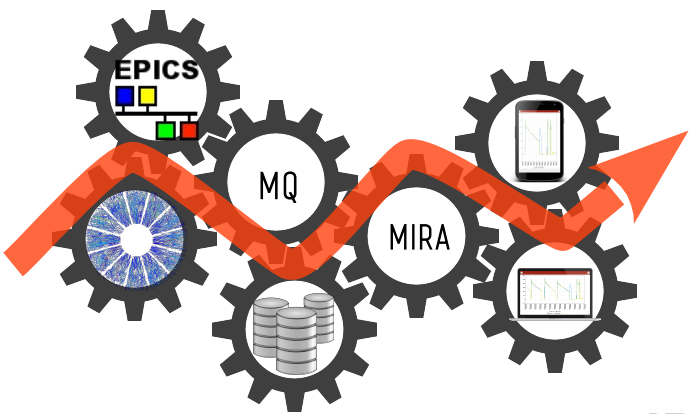




Summary & Outlook

- STAR Meta-data Collection Framework update:
  - STAR started with MIRA (Messaging Interface and Reliable Architecture) to archive/redirect/filter and act (CEP) upon streaming meta-data collected online. MQ-based service messaging bus provided STAR with stable and scalable solution, and enabled near-real-time view of the experimental status for all collaborators.
  - Proof of principle: EPICS Alarm Handler UI was replaced by a web interface relying on our recently developed EPICS<->MQTT bridge approach.
  - STAR Online service integration is now greatly improved for DAQ and RTS subsystems, as our Integrated processing of the alarms produced by various STAR subsystems is completed.

- Future Plans:
  - Bridge and IOC emulator codes are largely experiment-specific for now, but we are working on a generic implementation to make it fully reusable by the HEP/NP/CA community
  - Bridging and service integration got an interest from collider-accelerator developers, discussions are ongoing



Thank You!

Questions? Comments?