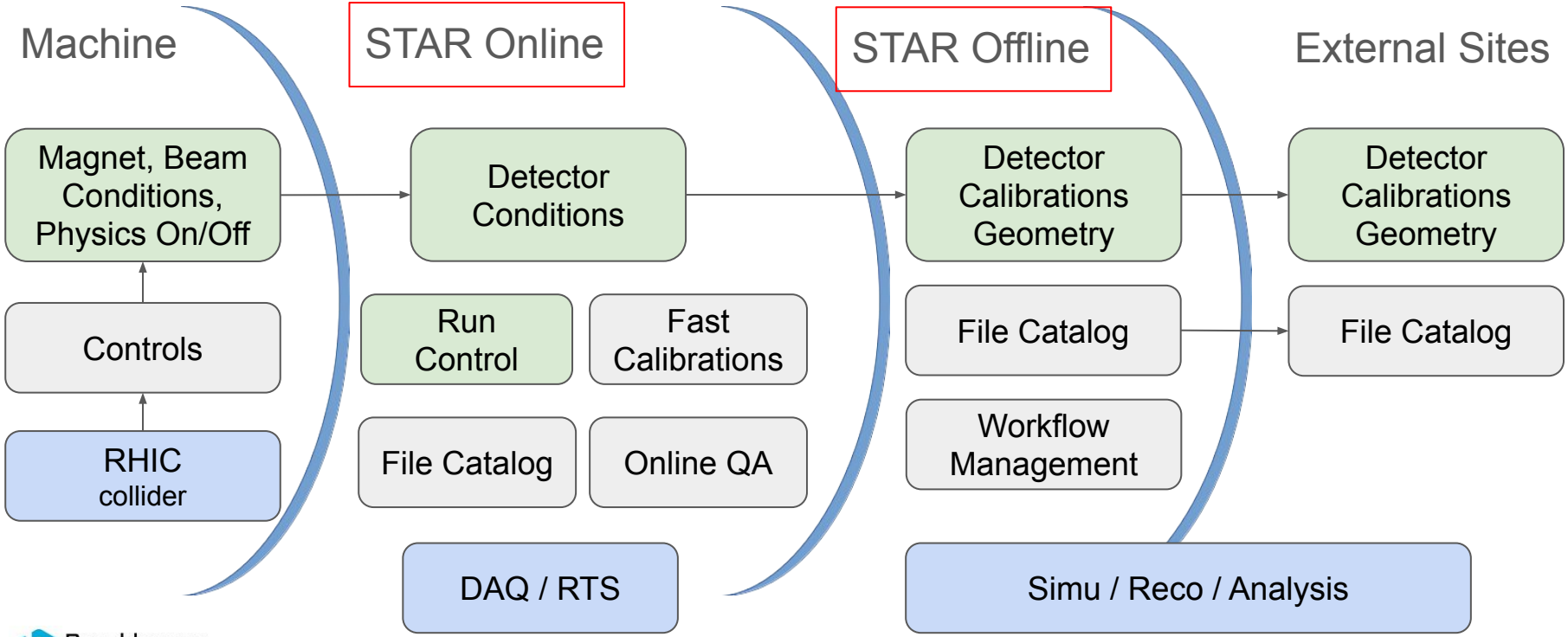

# STAR Databases

## Online to Offline

Dmitry Arkhipkin
SDCC, Tools & Services group

2024-07-25

# STAR Databases: Overview



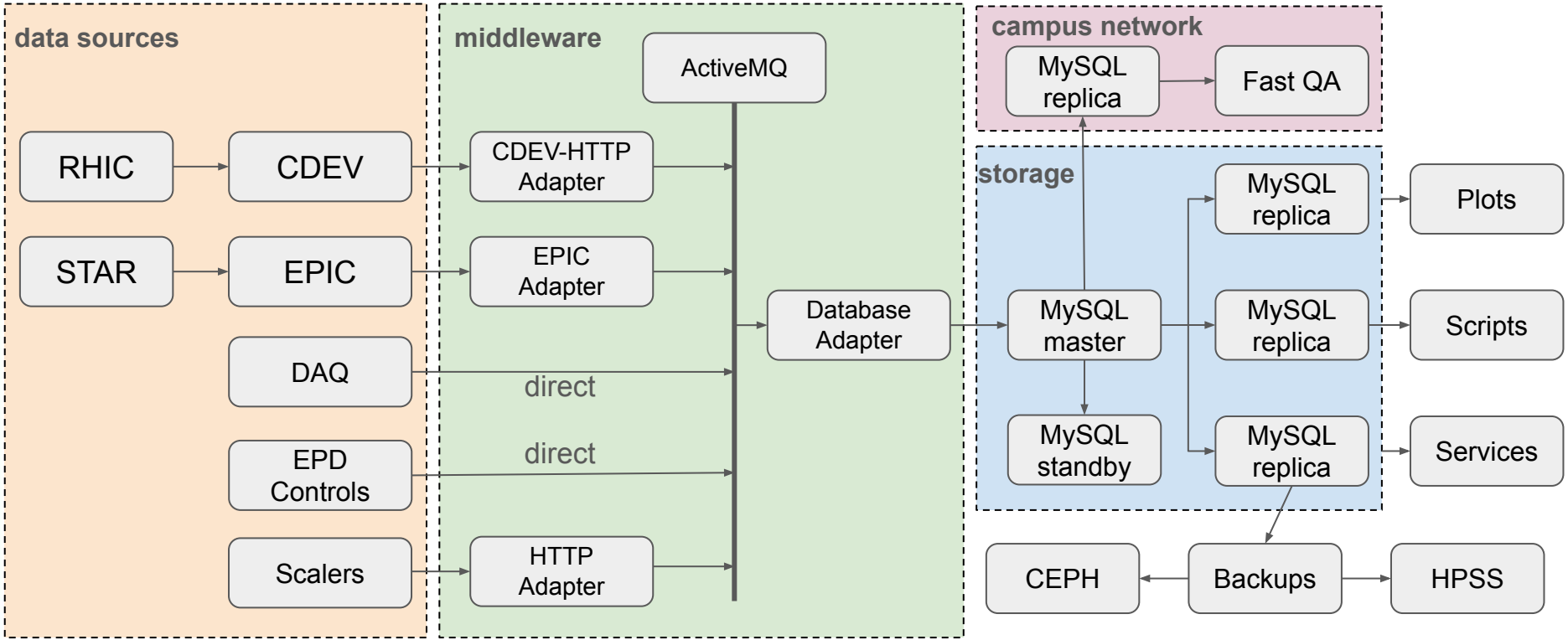

Brookhaven National Laboratory

# STAR: Conditions DB

- Online DB categories:
  - **DAQ + Run Control**
    - per-run configurations: detectors, electronics, thresholds, trigger data, and a lot more
  - **Detector Conditions**
    - EPICS: Slow Controls / SCADA
    - ~40 subdetectors, ~60k PVs
  - **RHIC Conditions**
    - CDEV: Beam Conditions + Magnet
    - per-channel or grouped collider and detector conditions, used by fast calibrations services
  - **Software Infrastructure DBs**
    - Run Log
    - Shift Signup
    - Electronic Shift Log
    - transient File Catalog

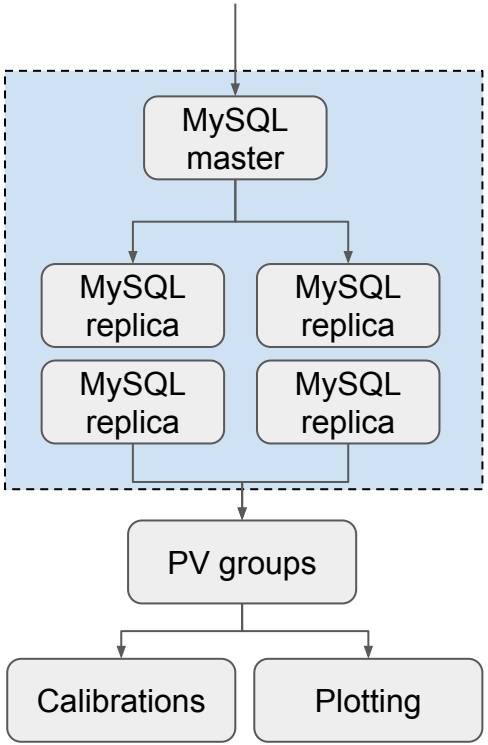
- Implementation and Deployment
  - **MQ Middleware**
    - two ActiveMQ servers (bridged)
    - AMQP protocol: reliable queues
    - MQTT protocol: fast, payload-agnostic
  - **Adapters:**
    - Any-to-MQ
      - JSON-encoded messages, contains hints for the DB-to-MQ
    - MQ-to-DB
      - receives messages from MQ, stores data to the DB, simple schema evolution
  - **Storage / Database**
    - replicated MySQL setup
      - master + N replicas
      - hot standby replica
      - dedicated backup replica
    - MongoDB cluster

# STAR: Conditions DBs deployment



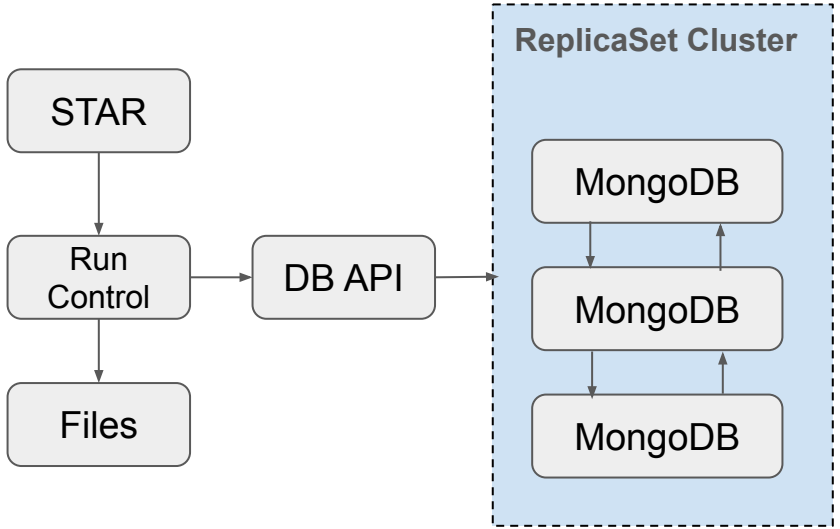
# STAR: Conditions DB

- Database Setup and Stats
  - 1 Master DB + 6 Replicas deployed
    - horizontally scalable, reliable
    - continuously streaming replication: minimal delays between master and replicas (sub-second latency)
    - time-series database structure
    - request caching: Query Cache (MySQL 5.7)
    - ~3k INSERTs per second (one PV group per insert)
    - DB size per Run: O(1TB)
  - Data: PV groups
    - one db per subsystem, one table per PV group, column = PV
    - easy to use by online to offline migration scripts and client services
    - easy to consume by monitoring tools
  - Plotting:
    - custom plotting package (dbPlots)
    - an option to plug the Conditions DB into Grafana exists

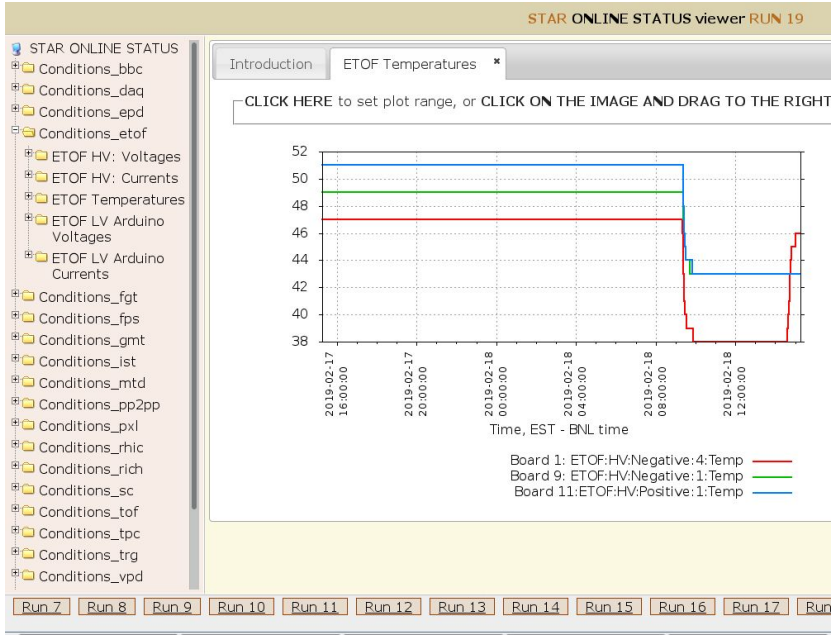


# STAR: Run Control DB

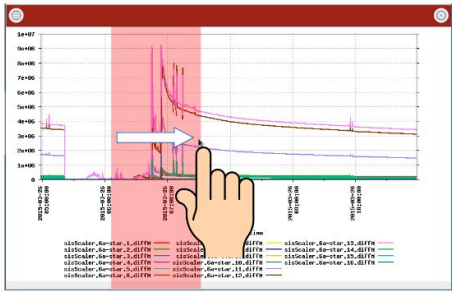
- Database Setup and Stats
  - **Document-based NoSQL DB**
  - One document per run
  - JSON-based storage
  - Data size: O(700MB) ~= 10 yrs of data
- MongoDB cluster
  - ReplicaSet cluster (three nodes) with an **automatic failover and load balancing**
  - clients connect to all nodes at once, automatic handling of Master/Replicas via driver
  - automatic indexing (on-demand)
  - schema-free documents
- Features:
  - Easy to export data, fast backups
  - access from the online web server for RunLog and plotting purposes

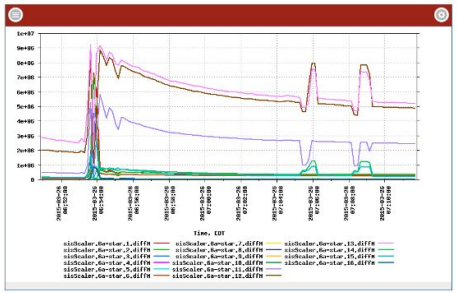


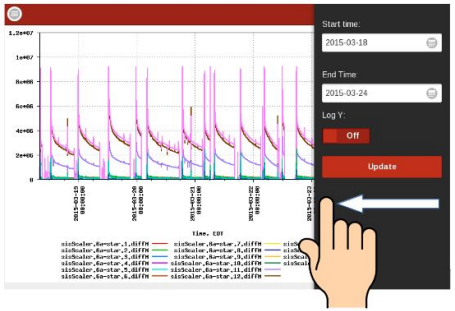
# STAR: Conditions Plotting Package

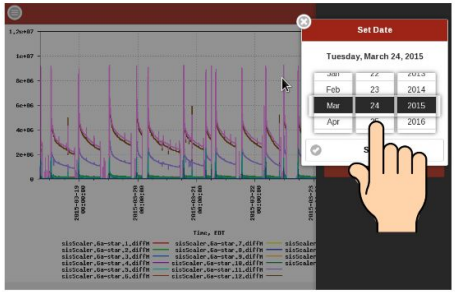



- in-house developed plotting package
- supports hierarchies, channel selection, common features, mobile, etc.







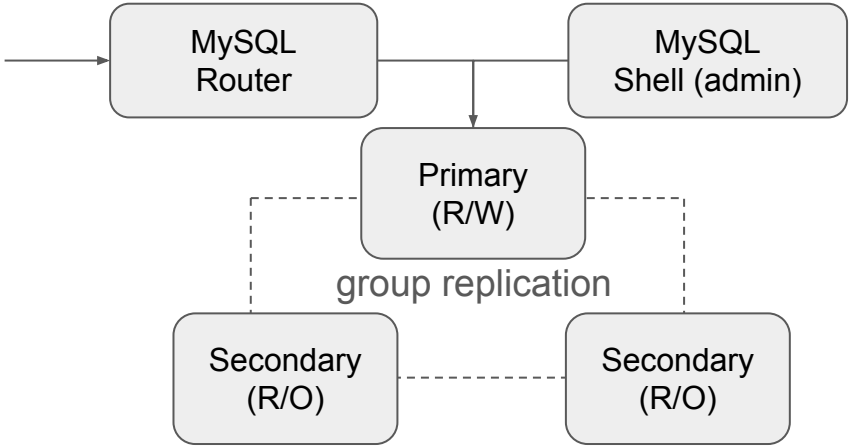




**Modern alternative: Prometheus + Grafana**

Brookhaven National Laboratory

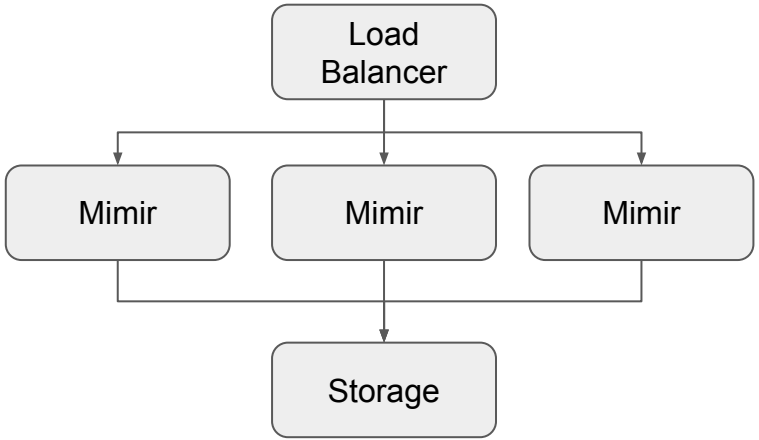
# High-Availability: MySQL and Prometheus

- **MySQL High-Availability Cluster**
  - nearly identical to the replication setup, but with the built-in automatic failover
  - caching via MySQL Router
  - similar solutions exist for Postgres



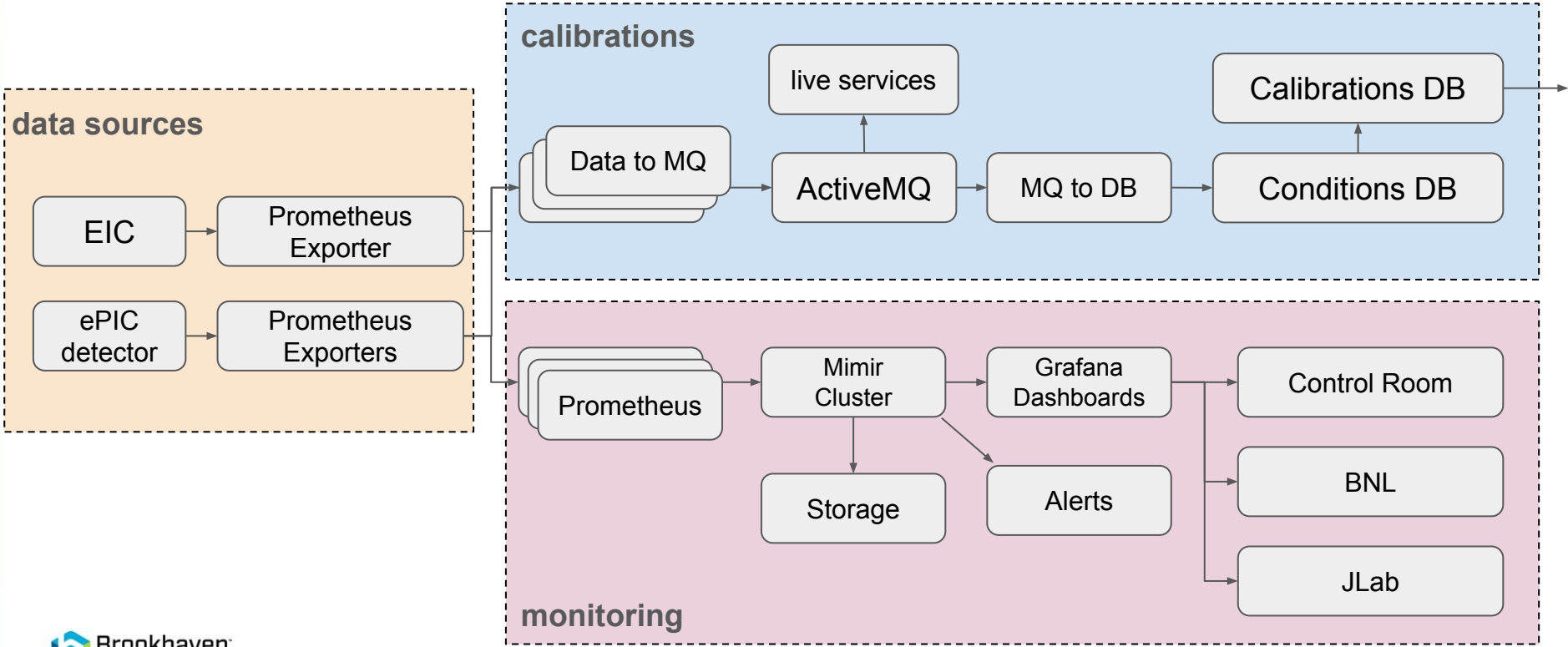

InnoDB High-Availability Cluster

- **Mimir: clustered Prometheus**
  - scalable, load-balancing support
  - automatic failover
  - reliable data storage
  - centralized alerts



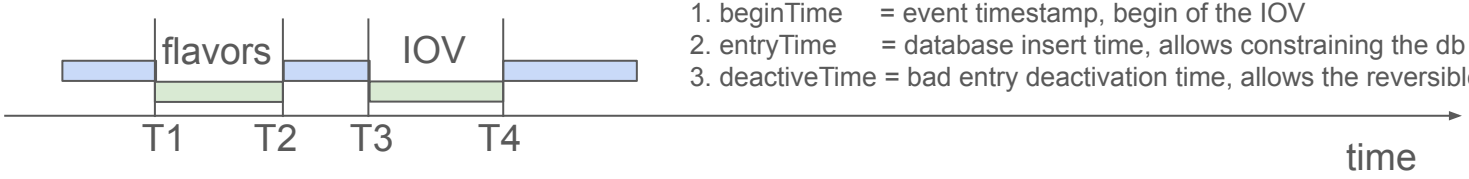

Brookhaven National Laboratory

# Solution for ePIC: DB + Prometheus?



# Offline DBs: Calibrations and Geometry

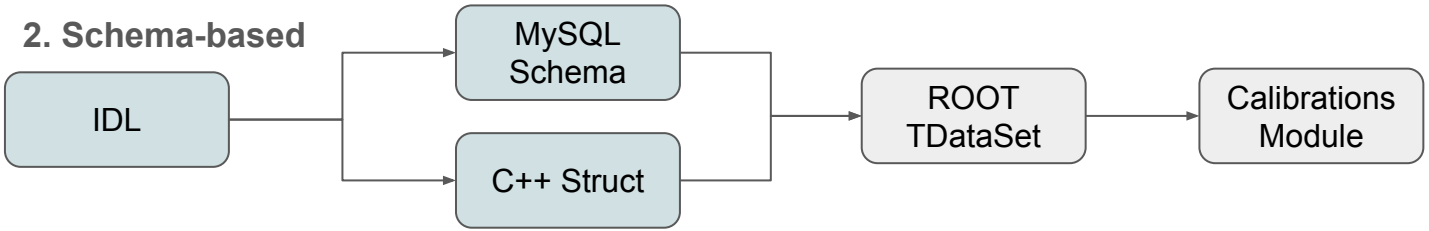
**1. Time-series data, structured**



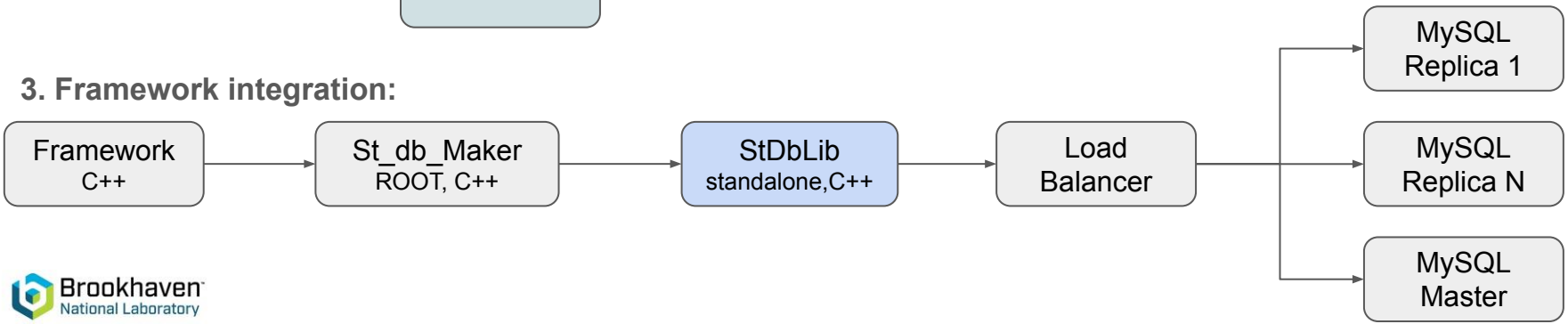

**Three timestamps per entry:**

1. beginTime = event timestamp, begin of the IOV
2. entryTime = database insert time, allows constraining the db by time
3. deactiveTime = bad entry deactivation time, allows the reversible removal of data entries

**2. Schema-based**



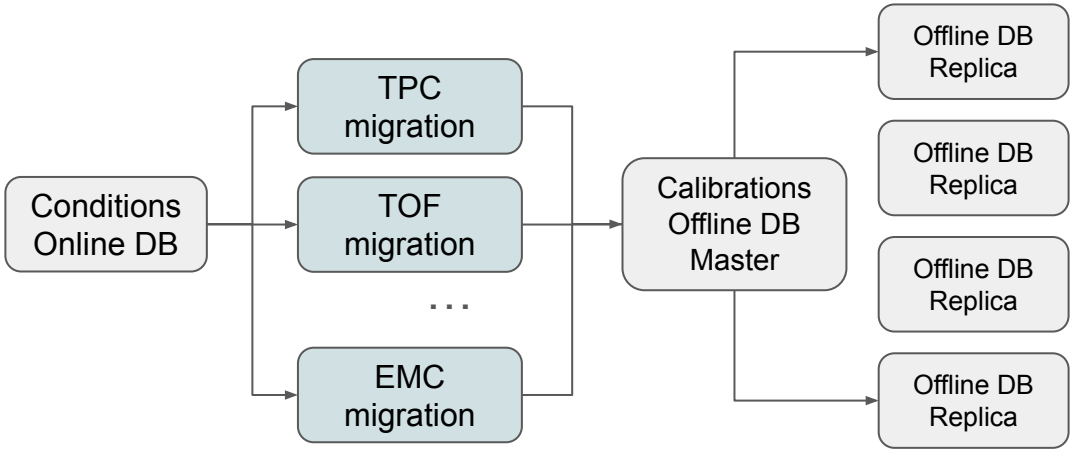

**3. Framework integration:**




Brookhaven National Laboratory

# Online to Offline migration

- Online Databases: Conditions
  - raw detector conditions data: V, I, state
  - minimal grouping/structuring
  - “high” granularity, O(1s)
  - large data volume, O(1TB) / yr
- Offline Databases: Calibrations
  - processed data: ready to be applied
  - highly structured, schema-based
  - multiple conditions combined
  - “low” granularity, O(5m…1hr)
  - small data volume, O(10GB) / yr
  - optimized for fast data downloads
- Migration Codes:
  - dedicated set of scripts processing per-subsystem data
  - data filtering, smoothing, transformation
  - handle data gaps and issues graciously




ETL: extract, transform, load
ROOT macros, cron-based

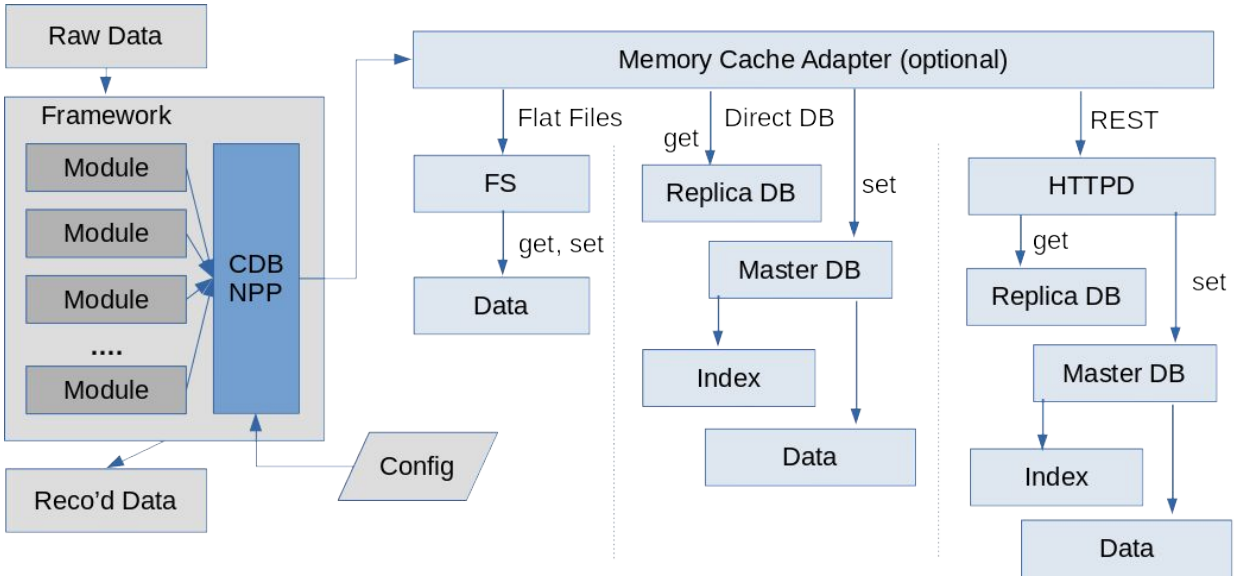
+monitoring

# CDBNPP: improved Offline DB API prototype

https://www.star.bnl.gov/~dmitry/talks/cdbnpp-arkhipkin-2022-02-10.pdf

An attempt to account for all possible use-cases, bottlenecks, and feature requests based on multi-experiment experience

- DB ACCESS API
  - Language: C++
  - DB: Postgres/MySQL/sqlite
- Supported Data Formats:
  - Agnostic/BLOB
  - JSON with schema validation
- Data Storage
  - DB-embedded
  - external URI
- Access methods
  - Direct DB
  - DB-over-HTTP
  - Flat files
- IOV: Time or Run + Seq
- Tags: Hierarchical




https://github.com/dmarkh/cdbnpp

Brookhaven National Laboratory

# Summary and Outlook

- STAR databases
  - structured data, replicated DB setup, horizontally scalable and fault-resistant
  - supports multiple input sources
    - CAD, DAQ/RTS, Slow Controls, custom sources
  - uses MQ middleware for integration purposes
  - current setup is field-tested by ~15 years of running
  - integrates well with many STAR online services
- Proposed solution for Conditions DB @ ePIC
  - combines the best of both worlds:
    - reliable clustered SQL DB setup for calibrations purposes
    - hardened Prometheus + Grafana setup for monitoring purposes
  - flexible enough to accommodate for online services integration
  - will be backed up by the infrastructure of SDCC @ BNL