9th International Conference "Distributed Computing and Grid Technologies in Science and Education" (GRID'2021), July 5-9, 2021, Dubna, Russia

Development of the Event Metadata System for the NICA experiments

Peter Klimai [1,2], Evgeny Alexandrov [3], Igor Alexandrov [3], Artyom Degtyarev [2, 4], Irina Filozova [3], Konstantin Gertsenberger [3], Alexander Yakovlev [3]

[1] Institute for Nuclear Research of RAS, Moscow, Russia

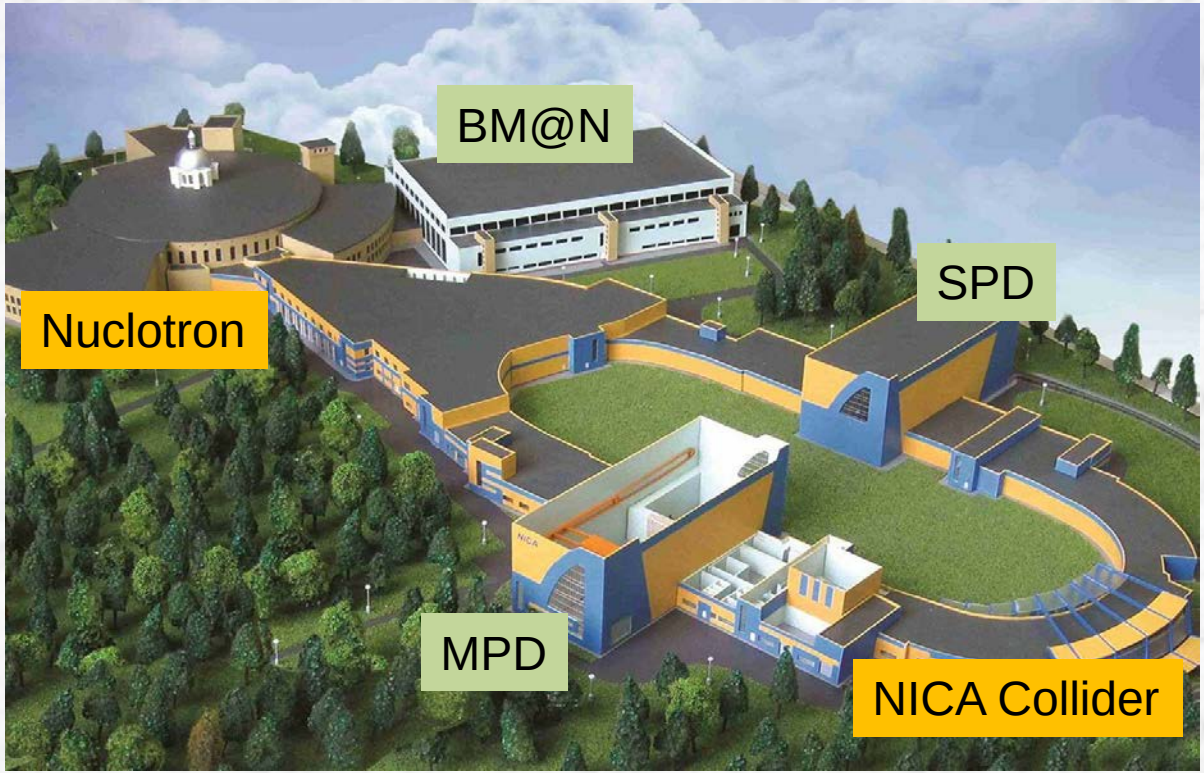
[2] JetBrains Research

[3] Joint Institute for Nuclear Research, Dubna, Russia

[4] Moscow Institute of Physics and Technology, Nuclear Physics Methods Lab

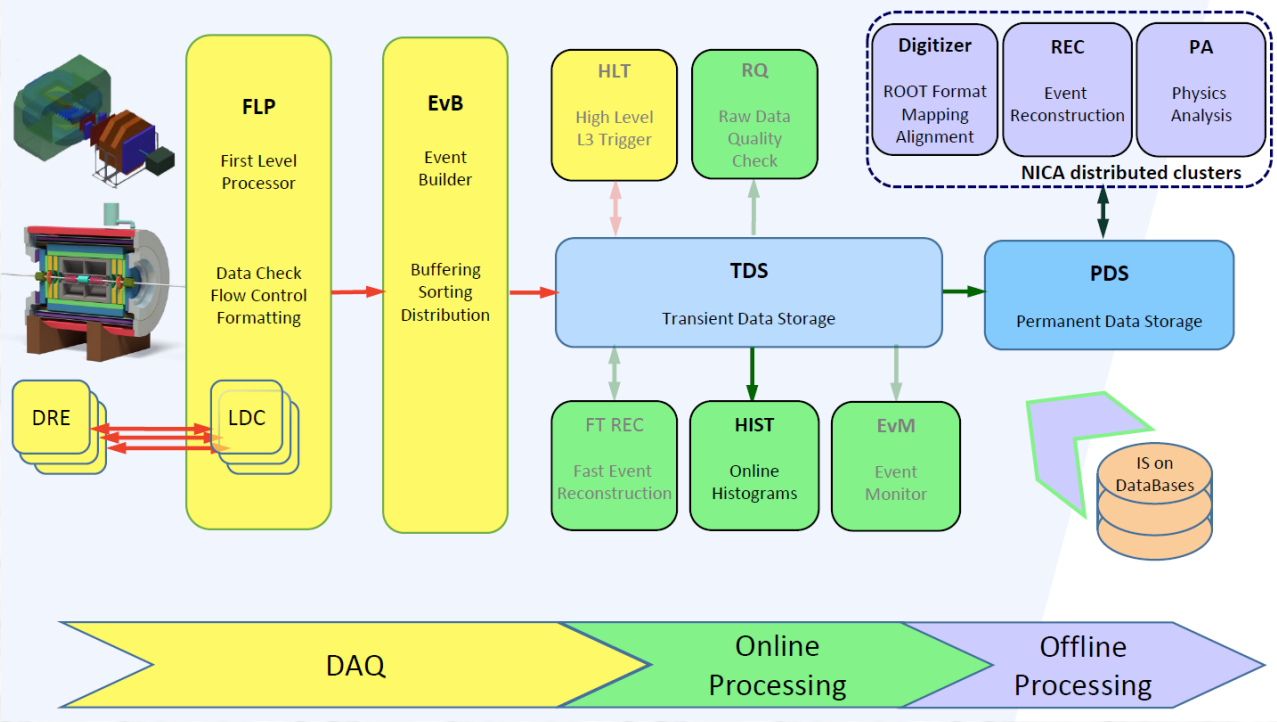
Nuclotron-based Ion Collider fAcility (NICA)

- NICA aim is to study hot and dense strongly interacting matter in heavy ion collisions at center-of-mass energies up to 11GeV / nucleon
- Fixed target (BM@N, since 2015)
- Collider (MPD, SPD)



P. Klimai et al, “Development of the Event Metadata System for the NICA experiments”, GRID-2021

Data Storage and Processing

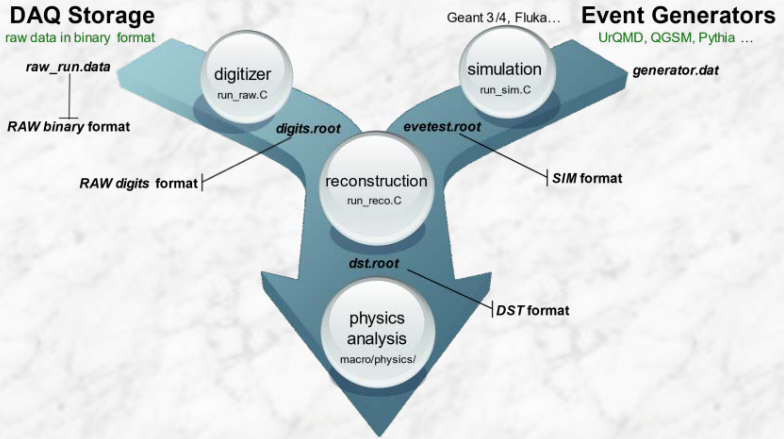


- BM@N data storage and processing pipeline is shown
- Physics analysis is typically performed for a subset of events satisfying given criteria
- To avoid full scan of all collected data, an Event Metadata System is required

P. Klimai et al, “Development of the Event Metadata System for the NICA experiments”, GRID-2021

Event Metadata System Goals

- Indexing of all reconstructed events stored in ROOT DST files
- Storing necessary event metadata, such as:
  - Number of primary and all reconstructed tracks
  - Track number of +/- charged particles
  - Primary and secondary particles found
  - Number of hits by detectors
  - Total input and output charge in the event
  - Software version
  - Reference to the storage location

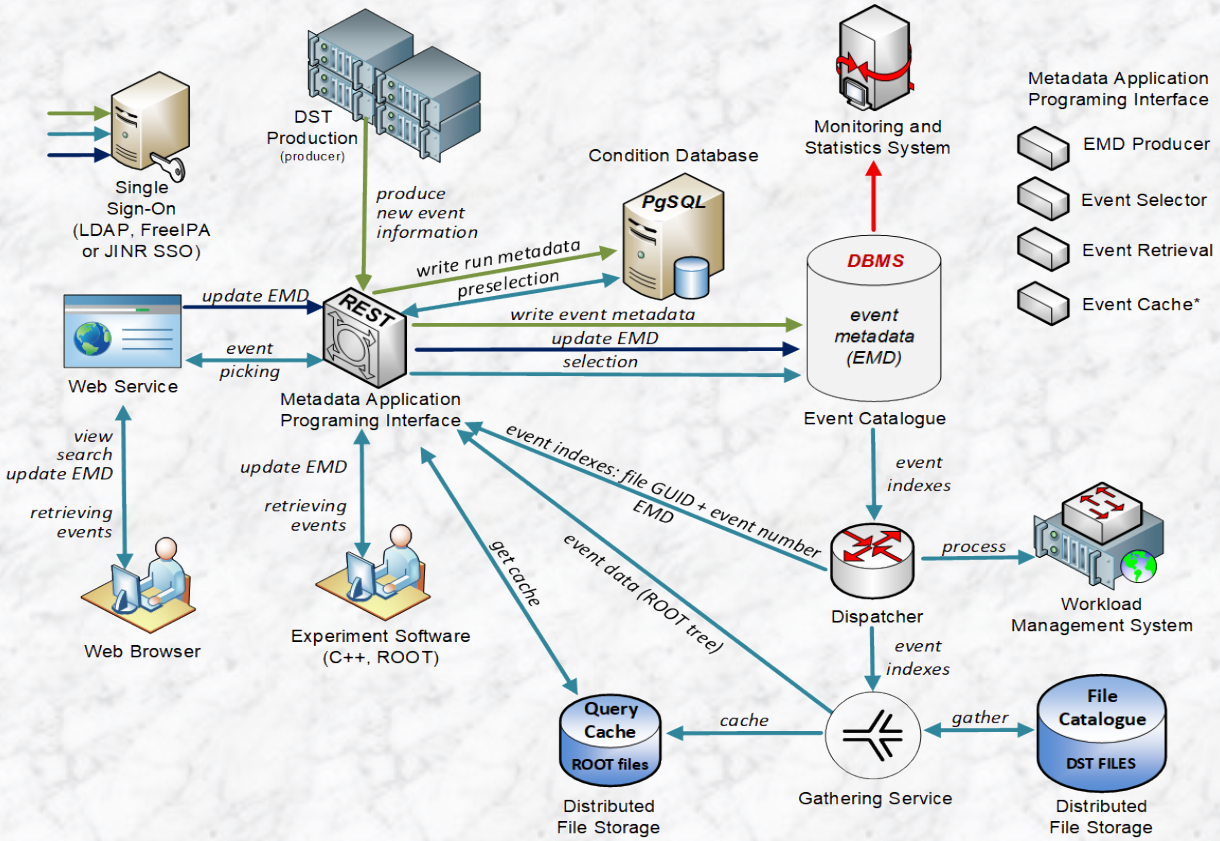


- Flexibly tune per experiment (BM@N, BM@N SRC, MPD, SPD)
- Convenient access to metadata (Web, REST API, C++)
- Search for required set of events
- Provide statistics and check the quality of the catalogue of physics events

Event Metadata System Requirements

- Scalability
  - Today for BM@N: overall ~500M events
  - Future (all NICA experiments): several Billion events per year
- Performance
  - Not too many RPS, but heavy ones
- Availability and fail safety
- Role-based access
  - Event Consumer, Index Writer, Index Administrator
- Interaction with other systems
  - Run metadata is stored in Condition database
  - FairRoot-based frameworks (BmnRoot, etc.)

EMD System Architecture

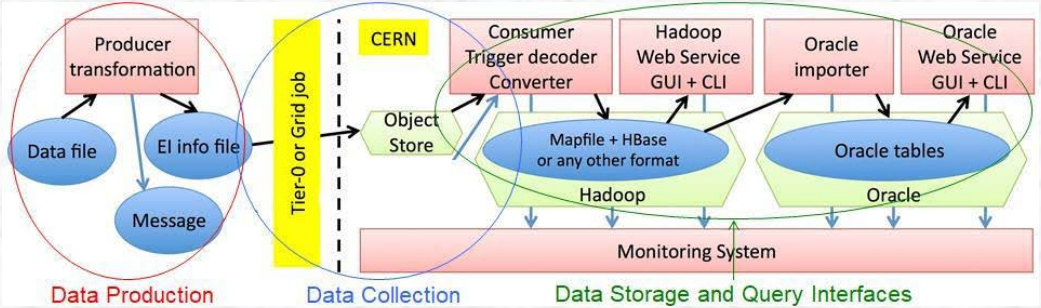


P. Klimai et al, “Development of the Event Metadata System for the NICA experiments”, GRID-2021

EMD Database Choice

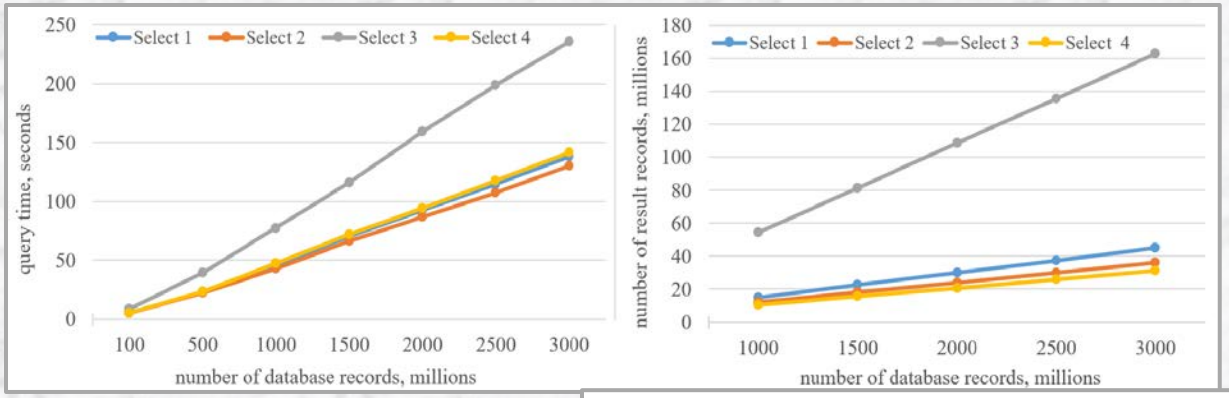
- Existing SQL vs. Existing NoSQL vs. Custom
- Horizontal vs. Vertical scaling
- Ease of maintenance
- Hardware cost
- Other experiment's experience
  - No single "best" solution

ATLAS Event Index architecture example (from: E. Cherepanova et al, "The ATLAS EventIndex using the HBase/Phoenix storage solution", talk at this conference)

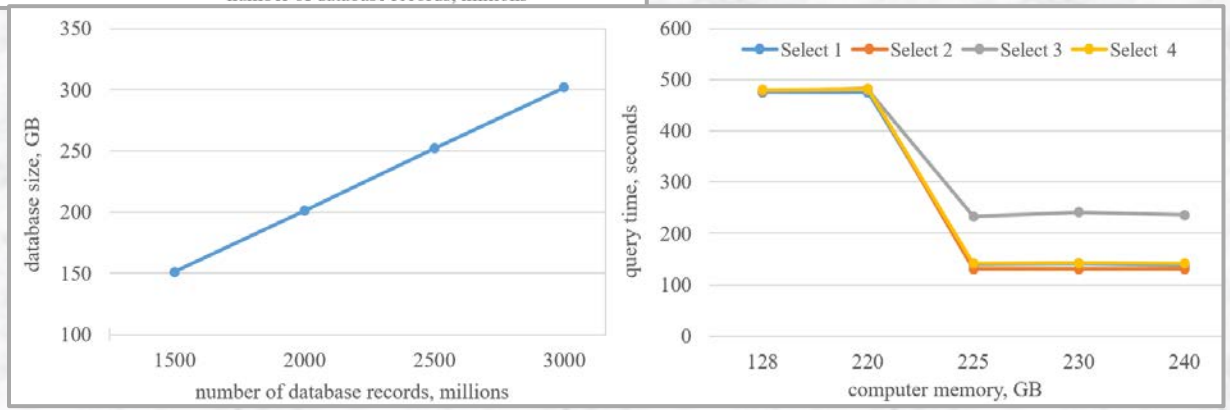


P. Klimai et al, "Development of the Event Metadata System for the NICA experiments", GRID-2021

DBMS Choice – PostgreSQL test

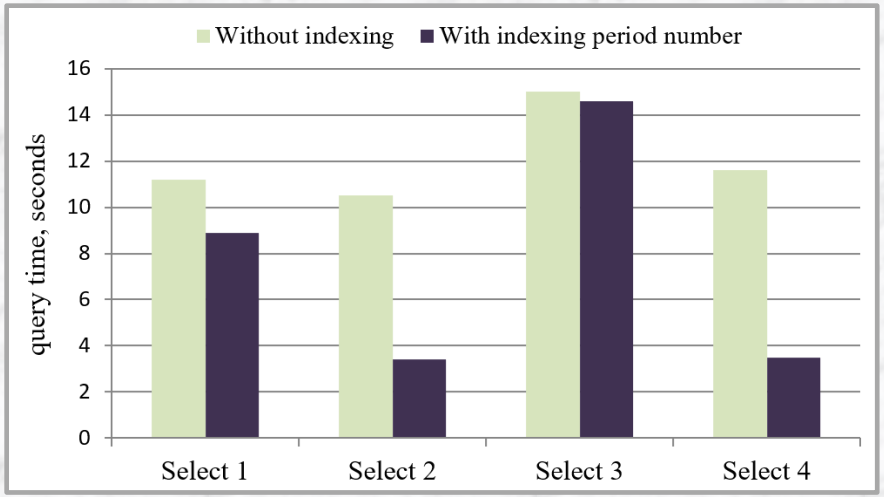


HW Config 1:
2 x Intel Xeon E5-2680, 256 GB RAM, DDR4 2133 MHz, 2xIntel SSD SC1BG400G4R, XenServer, Scientific Linux 7.9, PostgreSQL 12.5



P. Klimai et al, "Development of the Event Metadata System for the NICA experiments", GRID-2021

DBMS Choice – PostgreSQL test (contd.)



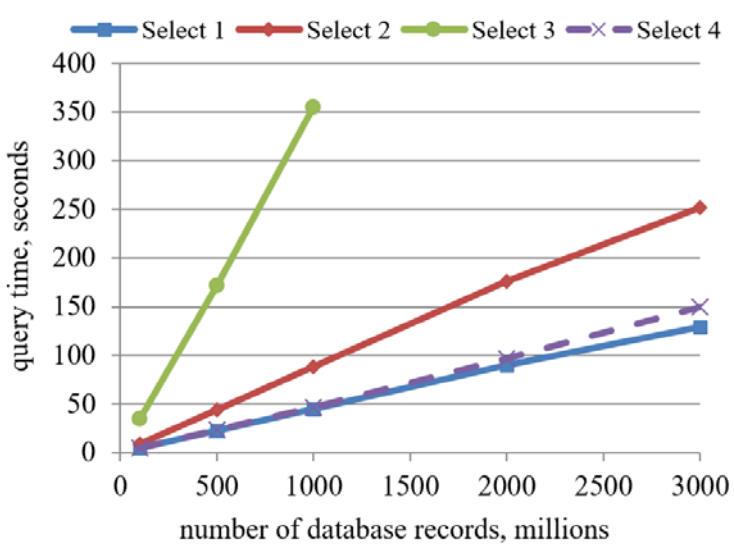
HW Config 2: Intel Core i9-10900F 2.80GHz (20 CPU cores), 64 GB RAM, 1TB NVMe SSD disk, CentOS Linux 8.2, PostgreSQL 12.5

<- Example test result for 500M events

SELECT 1: *PERIOD_NUMBER*=6 AND *SOFTWARE_ID*=0 AND *PRIMARY_TRACKS* > 5;
SELECT 2: *PERIOD_NUMBER*=6 AND *SOFTWARE_ID*=2 AND *PRIMARY_VERTEX* = FALSE AND *DETECTOR_HIT*[BITS:0-5];
SELECT 3: *PERIOD_NUMBER*=5 AND *RUN_NUMBER* > 100 AND *PRIMARY_VERTEX* = TRUE AND *PARTICLES*[BITS:5-9] > 4;
SELECT 4: *PERIOD_NUMBER*=4 AND *SOFTWARE_ID*=1 AND *PRIMARY_TRACKS*>6 AND *ALL_TRACKS*>40 AND *PARTICLES*[BITS:10-14].

DBMS Choice – NoSQL

- Cassandra test
  - Response time same order of magnitude was achieved (same hw)
  - Tables must be built per query to efficiently process different requests
- Other options under consideration
  - HBASE (with Phoenix)
  - DAOS



^-- Results using HW Config 2 with primary key:

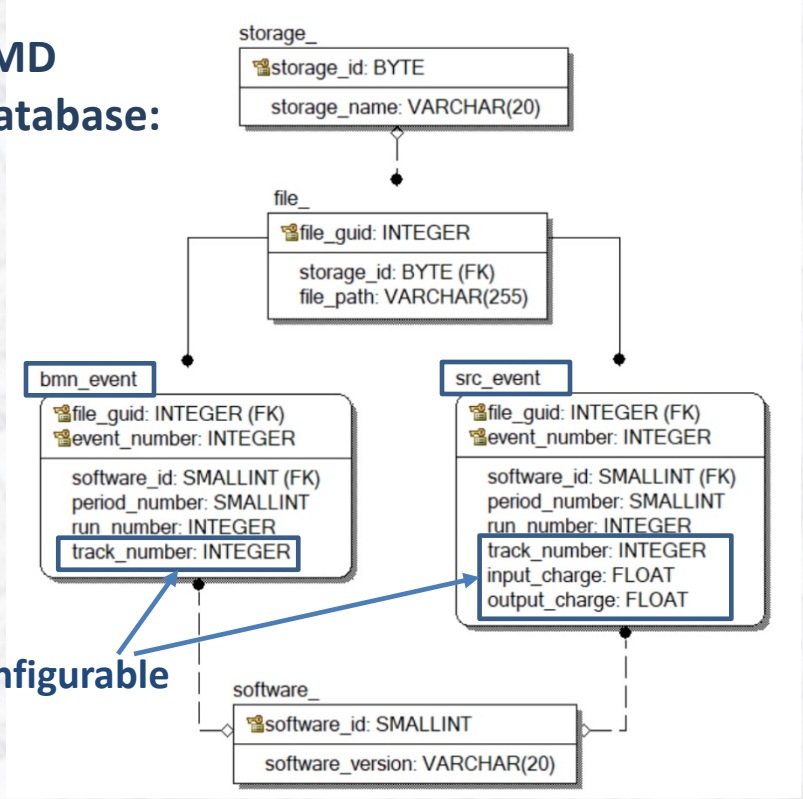
((*PERIOD_NUMBER*, *SOFTWARE_ID*), *PRIMARY_TRACKS*, *EVENT_NUMBER*)

((partition key), clustering columns)

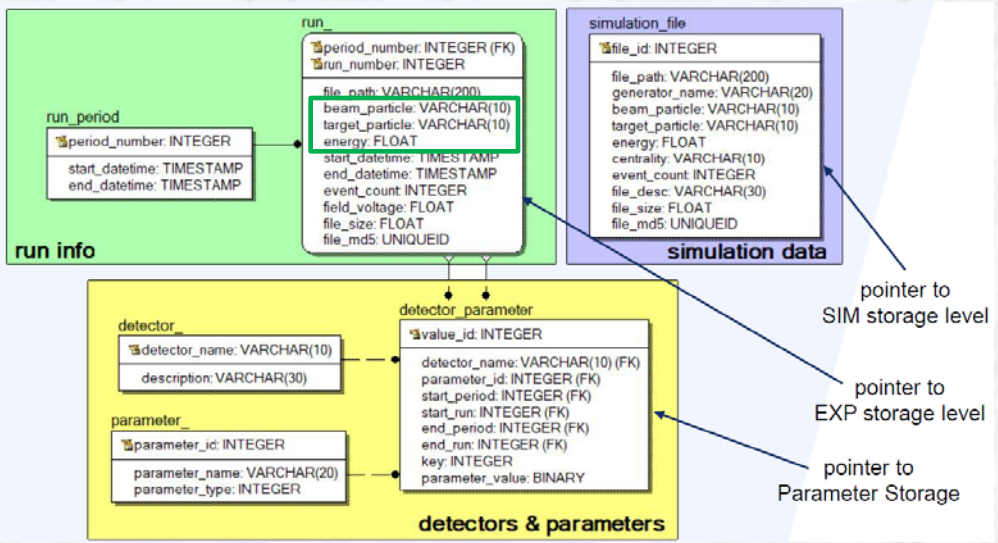
P. Klimai et al, "Development of the Event Metadata System for the NICA experiments", GRID-2021

Database Schema Examples

EMD Database:

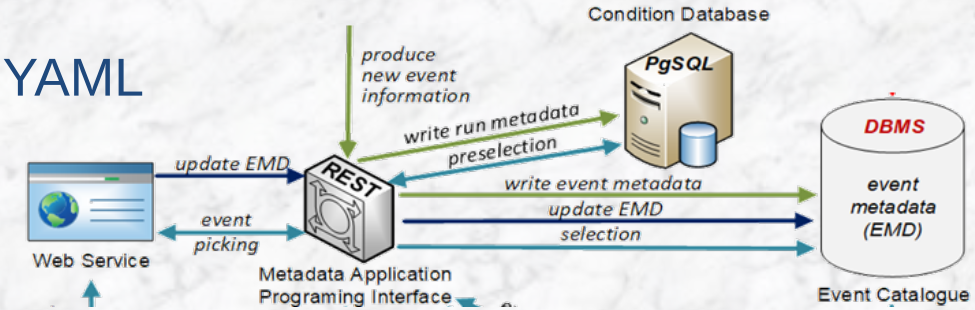


Condition Database:



REST API and Web UI Implementation

- Using Kotlin programming language
  - JVM runtime
  - ktor framework for Web UI and API
  - JDBC for Database connectivity
  - Jackson for (de)serialization
  - Kotlinx.html
- Packed in Docker
- Configuration file provided in YAML
- Auto provisioning (WIP)

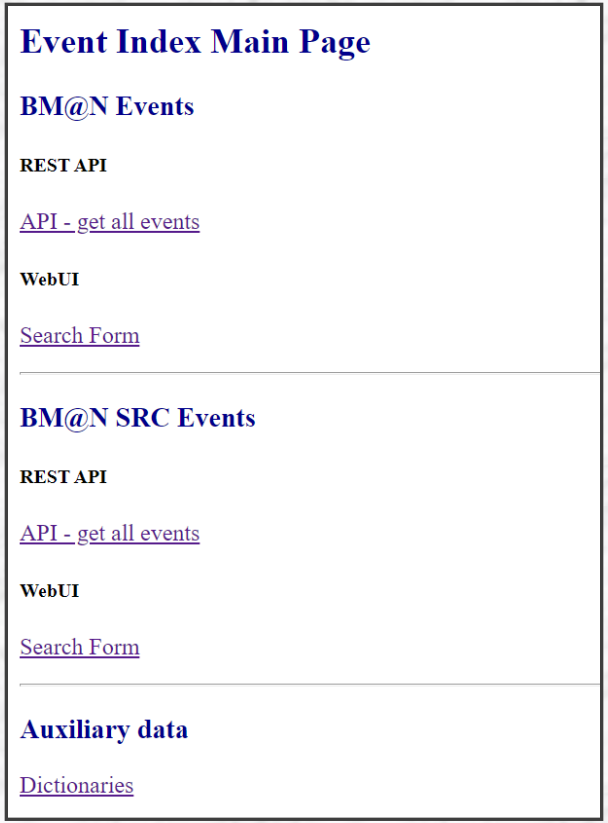


Configuration File Example

```
event_db:     # condition_db - similar
  host: ***
  port: ***
  db_name: ***
  user: ***
  password: ***

title: "Event Index Main Page"

pages:
  - name: "BM@N Events"
    api_url: "/event_api/v1/bmn"
    web_url: "/event_web/bmn"
    db_table_name: "bmn_event"
    parameters:
      - name: track_number
        type: int
        intervals: true
        web_name: "Total track number"
[…]
```

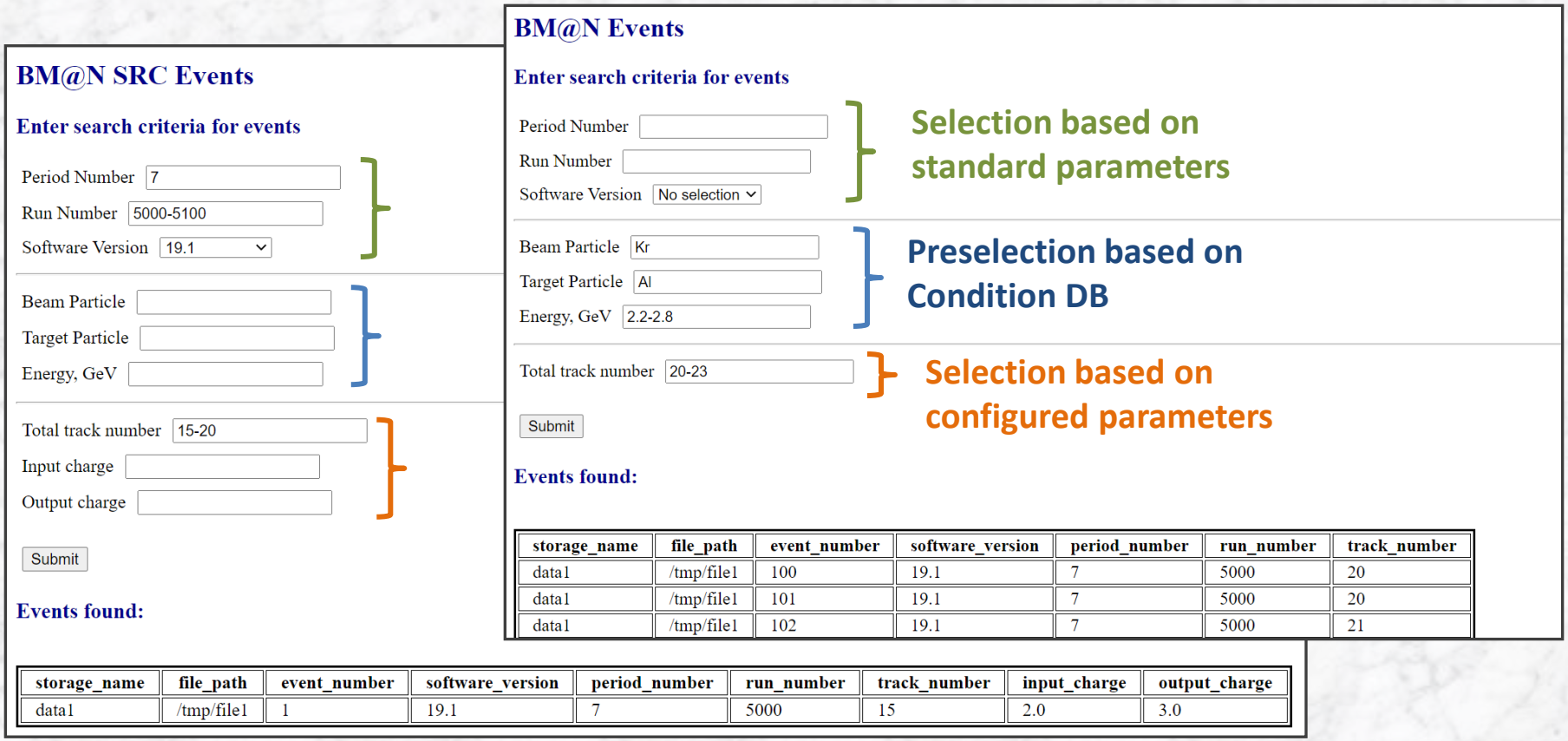
```
[…]

- name: "BM@N SRC Events"
  api_url: "/event_api/v1/src"
  web_url: "/event_web/src"
  db_table_name: "src_event"
  parameters:
    - name: track_number
      type: int
      intervals: true
      web_name: "Total track number"
    - name: input_charge
      type: float
      intervals: true
      web_name: "Input charge"
    - name: output_charge
      type: float
      intervals: true
      web_name: "Output charge"
```

Web UI Main Page (Test Prototype)



P. Klimai et al, “Development of the Event Metadata System for the NICA experiments”, GRID-2021

Web UI View (Test Prototype)



API Details

- HTTP API using JSON formatting
- HTTP POST to create events in the catalog
- HTTP GET to obtain event records
  - Same filtering criteria as Web UI, including range support, e.g.

```
/event_api/v1/bmn/events?period_number=7&run_number=5000+&
   software_version=19.1&track_number=10-15
```


```
{
  - events: [
    - {
      - reference: {
            storage_name: "data1",
            file_path: "/tmp/file1",
            event_number: 1
        },
        software_version: "19.1",
        period_number: 7,
        run_number: 5000,
      - parameters: {
            track_number: 20
        }
    },
    - {
      - reference: {
            storage_name: "data1",
            file_path: "/tmp/file1",
            event_number: 2
```

Current Status

- Completed
  - Architecture
  - DBMS Selection
  - Condition Database integration
  - REST API and Web UI Prototype
- WIP (scheduled to complete end of 2021)
  - Experiment framework integration (FairRoot based)
  - Monitoring (Telegraf, InfluxDB, Grafana)
  - HA and Backup
  - Automated Deployment (Ansible, Docker)

The work was funded by the Russian Foundation for Basic Research (RFBR) grant according to the research project №18-02-40125.

THANK YOU!