



# Design and implementation of the event catalogue for physics analysis in the NICA experiments

*Artyom Degtyarev, Evgeny Alexandrov, Igor  
Alexandrov, Alexander Chebotov, Konstantin  
Gertsenberger, Peter Klimai, Alexander Yakovlev  
13.10.2021*

---



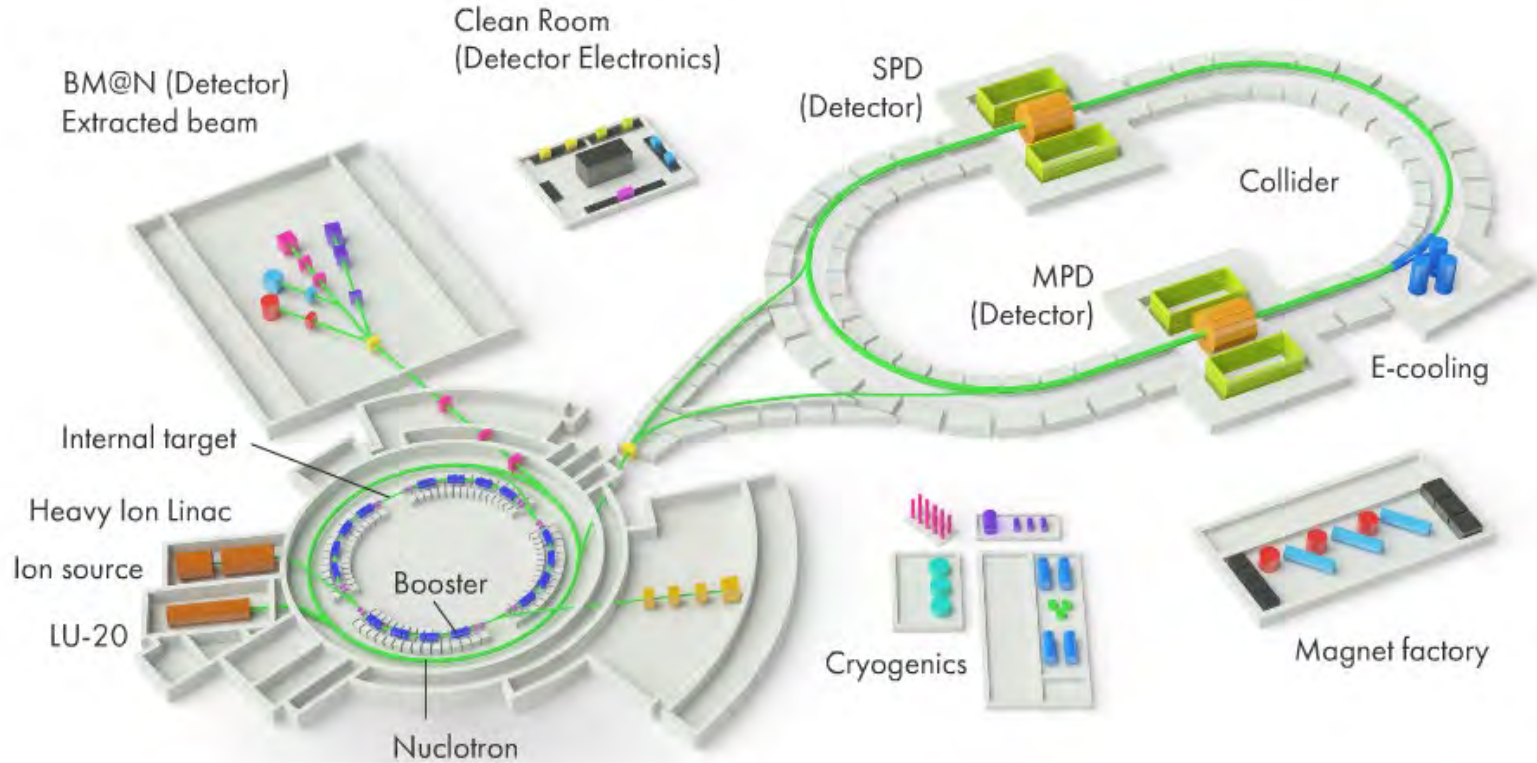
## IT systems in particle collision experiments

---

- Allows to automate data processing, storage and analysis
- Important type – event metadata systems
- Implemented in the ATLAS LHC, CMS, BES III and other experiments



# NICA experiment



<https://bmn.jinr.ru>



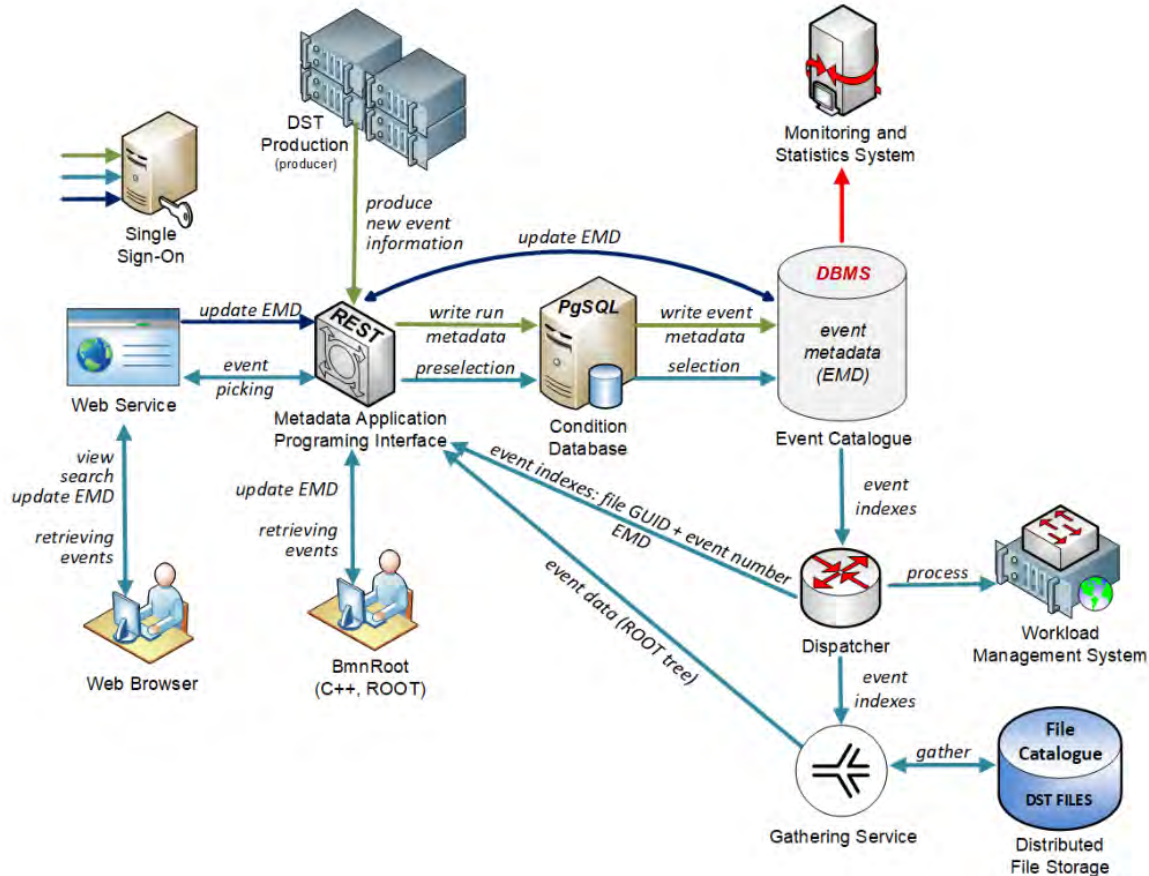
# NICA experiment

---

- Purpose – to study different features of strongly interacting matter:
  - Equation-of-state, microscopic structure of strongly interacting matter, in-medium modifications of hadrons
  - Theoretical models suggest different scenarios, so new data is needed
  - BM@N – first experiment
-



# Event Metadata System - construction





## Event Metadata System – DST files

---

DST files of the BM@N contain the following data:

- file header with some service information
- run header with the corresponding information
- hierarchical ROOT tree (with name 'bmndata'), which includes reconstructed event data



# Event Metadata System – selection criteria

---

- period number and run number
- software version
- event time (a start point in time is sufficient to not store a corresponding time interval)
- number of primary and all reconstructed tracks
- track number of positively and negatively charged particles from the primary vertex
- primary and secondary particles found
- number of hits by detectors
- total input and output charge in the event and other



## Event Metadata System – main functions

---

- Summary description of particle collision events and their identifiers
- Recording and storing event metadata in the Event Catalog
- Providing convenient and quick access to the metadata
- Online and offline interfaces for selecting events of interest





## Event catalog – choice of DBMS

---

- Several billions events and more are planned to be collected for the BM@N experiment
- System scalability and query performance are important
- Testing on a dataset with from 500 million to several billion events



## Event catalog – choice of DBMS, SQL vs. NoSQL

---

- Relational – highly structured, flexible in queries, easy to administer
- Non-relational –horizontal scalability, decentralization and fault tolerance, it's easier to change data structure working with the database
- PostgreSQL, HBase and Cassandra chosen for testing



# Data storage in Cassandra

PRIMARY KEY



Partition key

Clustering key

**Column**

Name



Value

**Row 1**

Clustering key: val1

Clustering key: val2

Partition key:  
val1

Column 1

Column 2

Column 1

Column 2

**Row 2**

Clustering key: val3

Clustering key: val1

Partition key:  
val2

Column 1

Column 2

Column 1



# PostgreSQL vs. Cassandra

---

- In PostgreSQL indexes can be used to improve performance
- Cassandra has a more flexible way of storing data
- PostgreSQL has more flexibility in queries
- In 3 of 4 tests Cassandra is 3-5 times faster, but in another test about 3 times slower
- PostgreSQL was chosen as being easier to administer and resilient to more queries



# REST API

---

- Kotlin, Ktor framework
- JDBC for database connectivity
- Jackson for serialization and deserialization
- Configuration files are provided in YAML
- JSON formatting
- POST and GET requests



# WEB UI prototype

## Event Index Main Page

### BM@N Events

#### REST API

[API - get all events](#)

#### WebUI

[Search Form](#)

---

### BM@N SRC Events

#### REST API

[API - get all events](#)

#### WebUI

[Search Form](#)

---

### Auxiliary data

[Dictionaries](#)



# Summary

---

- Architecture of the Event Metadata System
- Comparison of different DBMS's
- Approaches to the development of databases
- REST API and the WEB UI



## Future plans

---

- To complete integration with FairRoot-based software frameworks (such as BmnRoot)
- Ensure high availability and data backups
- Add monitoring
- Add script automated deployment





Thanks for your attention!

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