Web interface and REST API for BM@N Event Metadata System

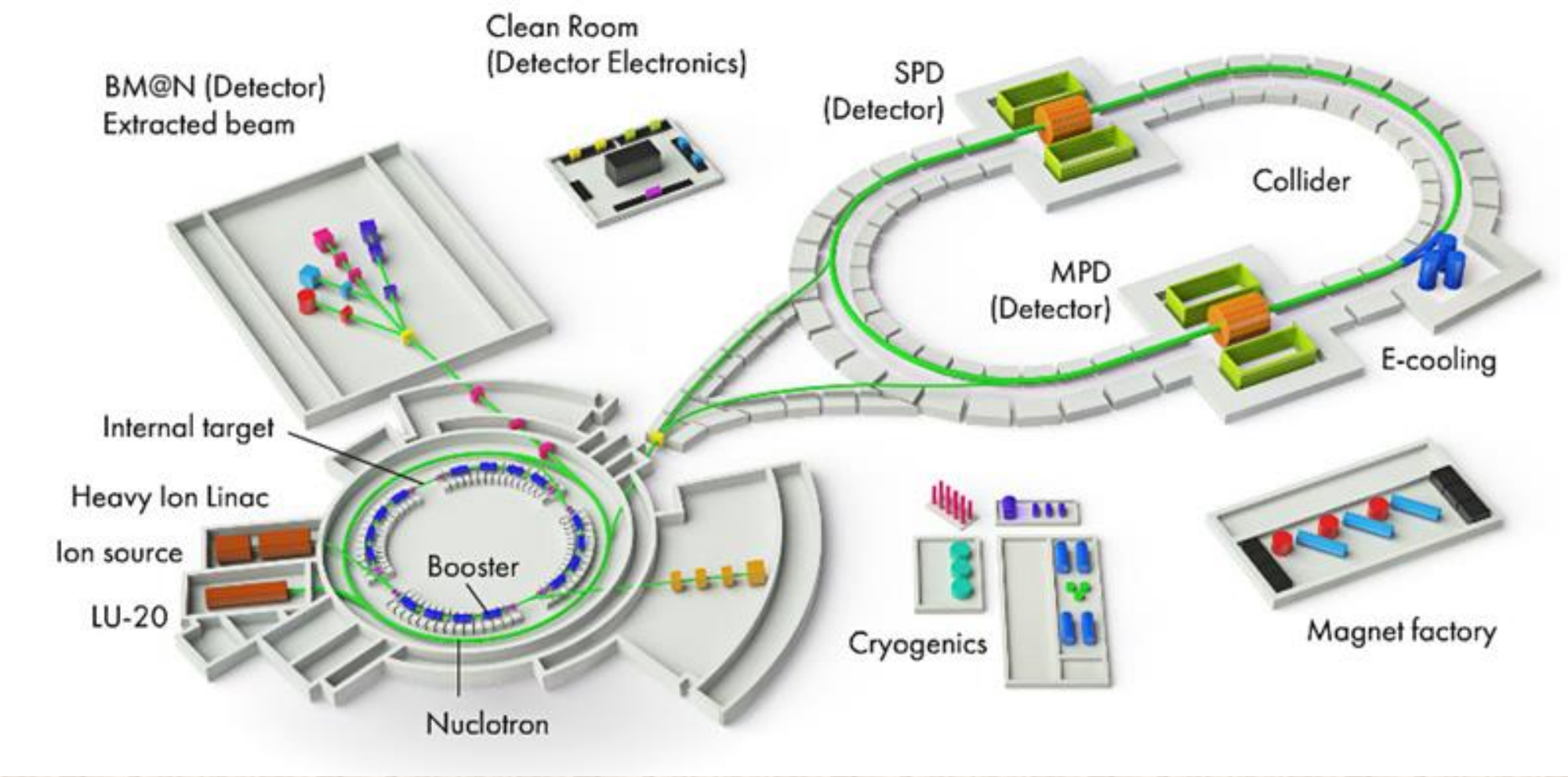
Degtyarev Artyom, MIPT NPM,
Peter Klimai, INR RAS,
Konstantin Gertsenberger, JINR
Alexander Chebotov, JINR

27.10.2022 г.

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


Clean Room
(Detector Electronics)
BM@N (Detector)
Extracted beam
SPD
(Detector)
Collider
MPD
(Detector)
E-cooling
Internal target
Heavy Ion Linac
Ion source
LU-20
Booster
Nuclotron
Cryogenics
Magnet factory

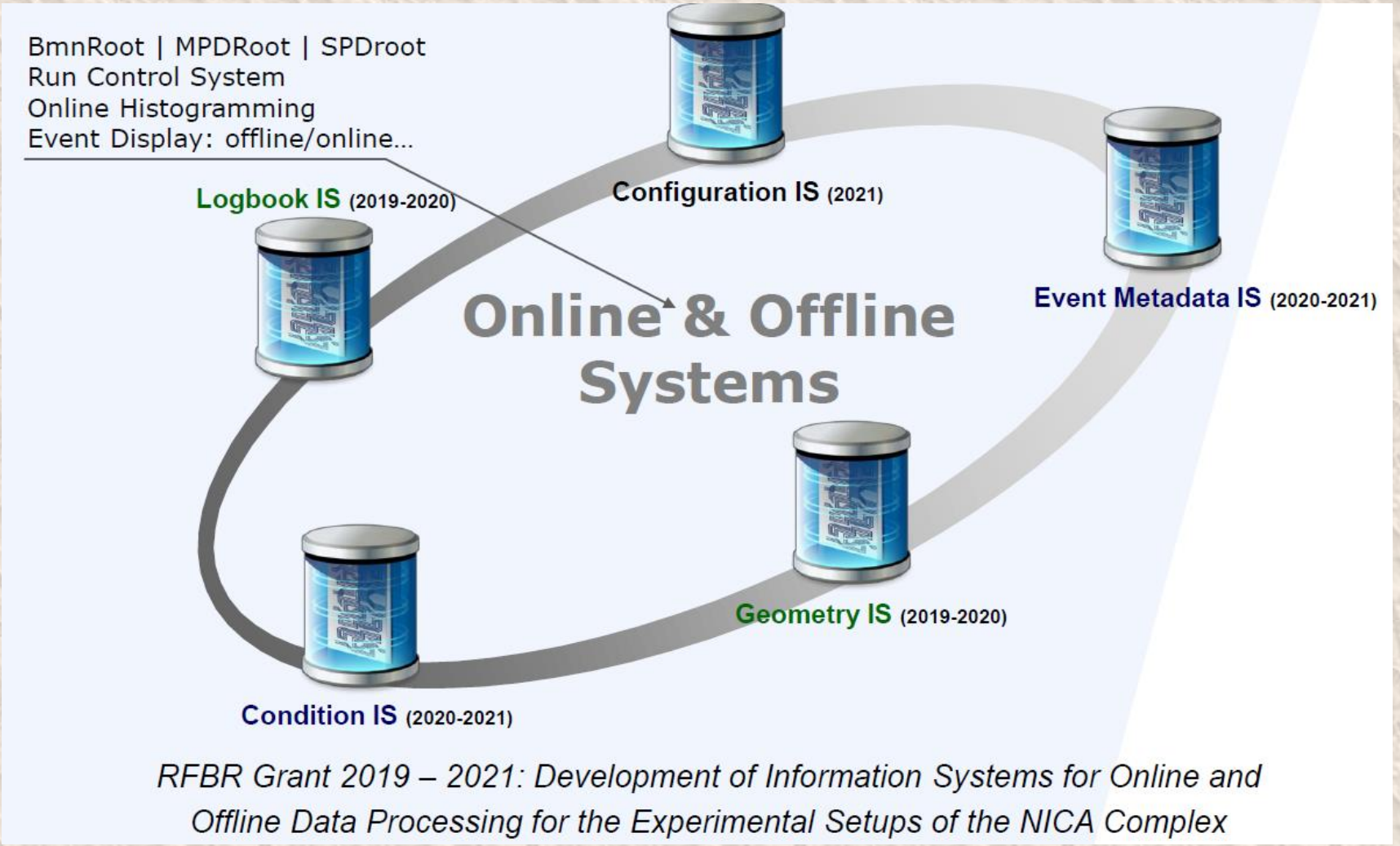
NICA experiment

- 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

Prerequisites of the data management with IS

- High interaction rate, e.g. BM@N up to 50 kHz
- High particle multiplicity, up to 1000 charged particles for central collisions at the NICA energies
- NICA data stream is estimated up to 20 PB of raw data per year
- Review of modern Information Systems in HEP experiments showed that “IS are used in all large physics experiments and have become an important part of the software, but the existing solutions are highly dependent on specifics of conducted experiments and are an inseparable part of them” [E. Alexandrov, I. Alexandrov, K. Gertsenberger, et al., Information Systems for Online and Offline Data Processing in Modern High Energy Physics Experiments, Modern Information Technologies and IT Education, No. 15(3), 654 671 (2019)]

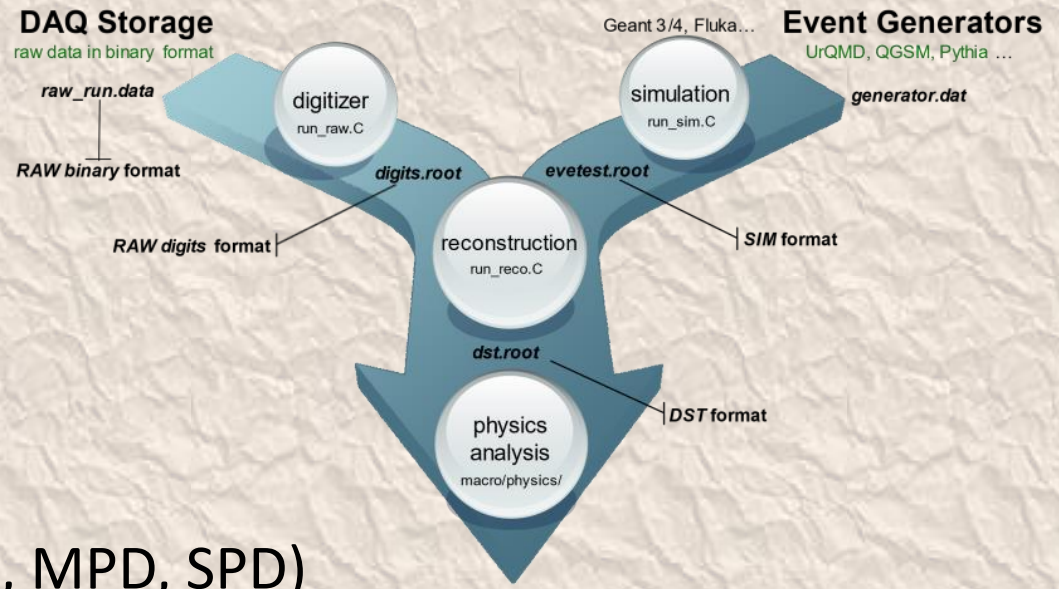
BM@N Information Systems



RFBR Grant 2019 – 2021: Development of Information Systems for Online and Offline Data Processing for the Experimental Setups of the NICA Complex

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

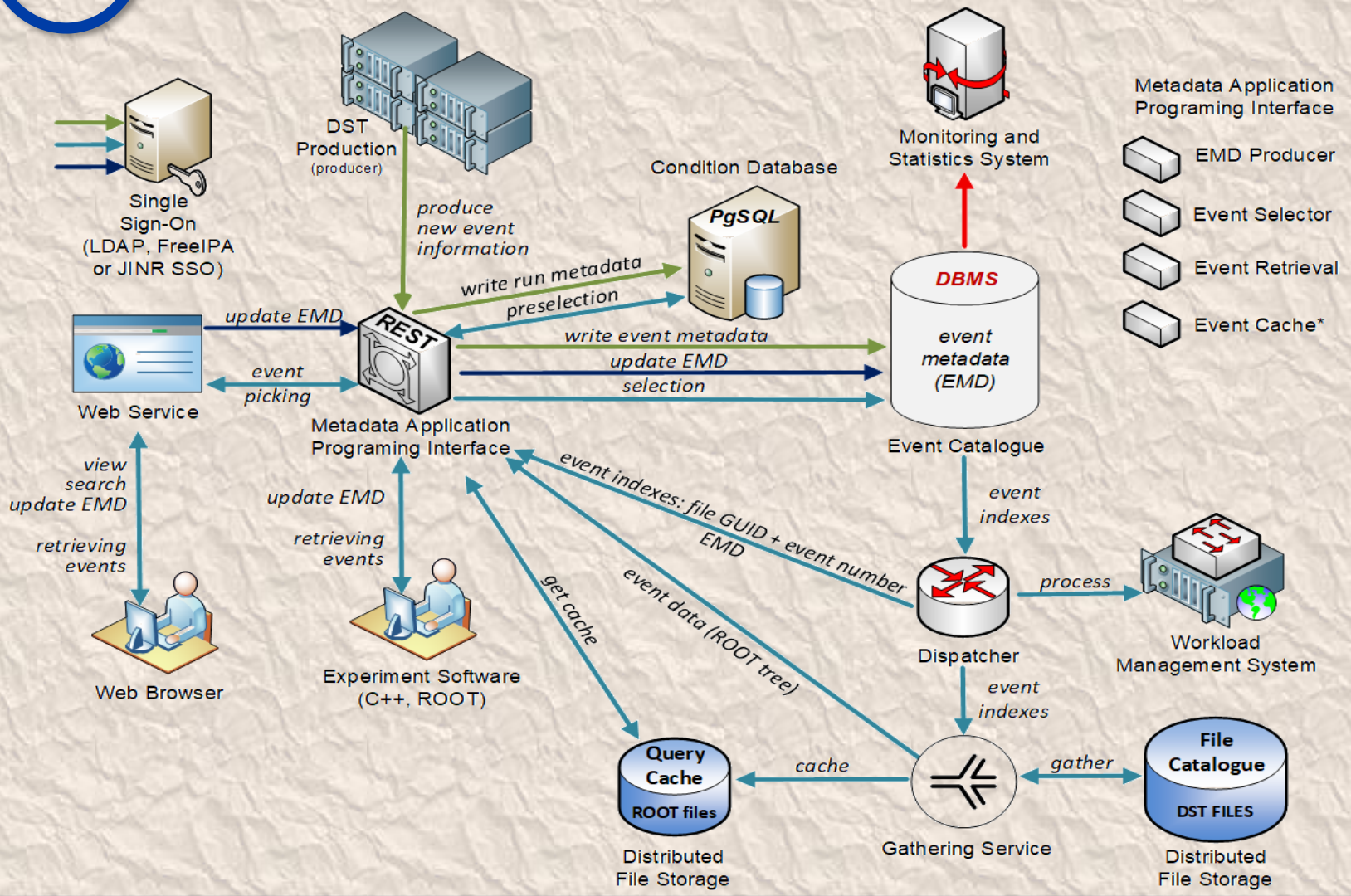


- Flexibly tune per experiment (BM@N, BM@N SRC, MPD, SPD)
- Convenient access to metadata (Web, REST API, C++)
- Search for required set of events
- Provide statistics and check the quality of the catalogue of physics events

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
  - Authentication and authorization via FreeIPA
  - FairRoot-based frameworks (BmnRoot, etc.)

EMS Architecture



For more details:

E. Alexandrov, I. Alexandrov, A.Degtyarev, K. Gertsenberger, I.Filozova, P. Klimai, A. Nozik and A. Yakovlev, “Design of the Event Metadata System for the Experiments at NICA”, Phys. Part. Nuclei Lett. 18, 603–616 (2021).

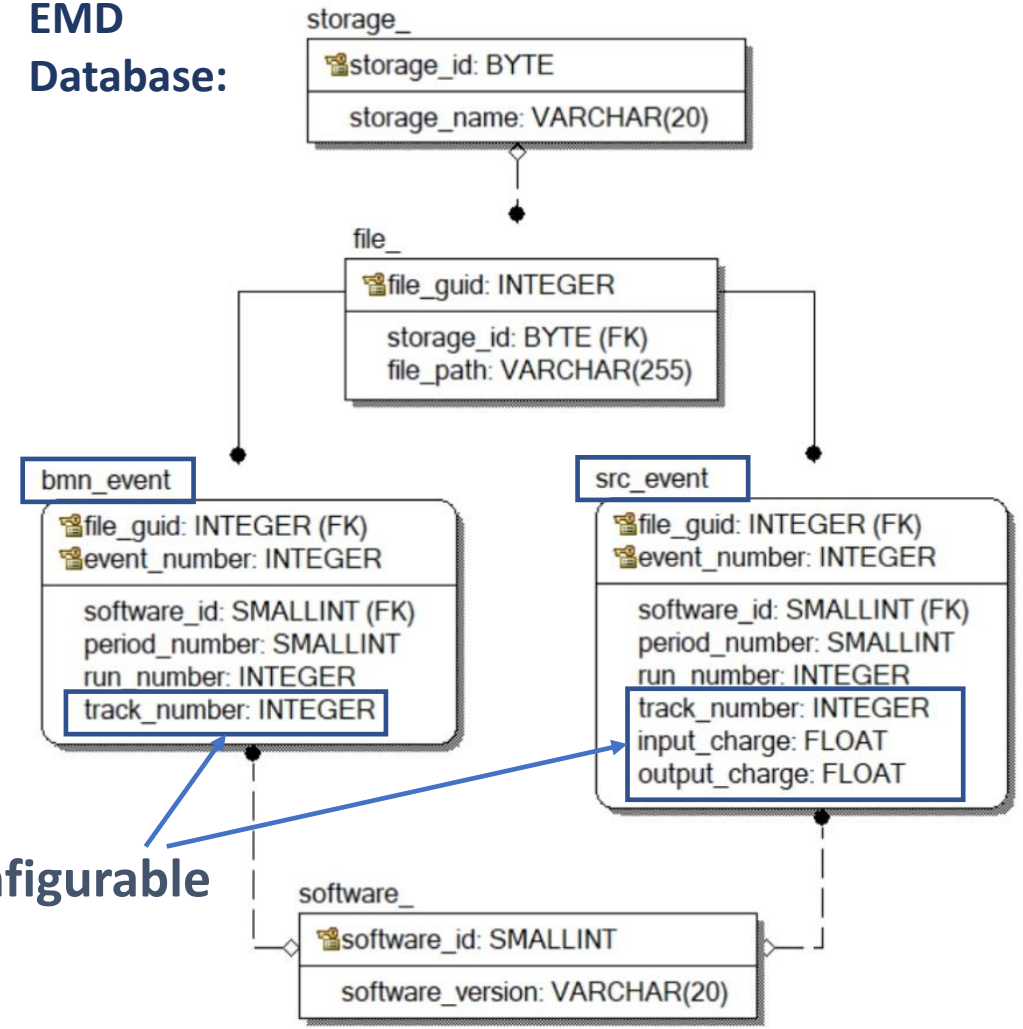
Choice of DBMS

- Comparison of DBMS was conducted
- PostgreSQL was chosen, but another interesting option is Cassandra
- For more details:

1. E. Alexandrov, I. Alexandrov, A.Degtyarev, K. Gertsenberger, I.Filozova, P. Klimai, A. Nozik and A. Yakovlev, “Design of the Event Metadata System for the Experiments at NICA”, Phys. Part. Nuclei Lett. 18, 603–616 (2021).
2. A. Degtyarev, K. Gertsenberger, and P. Klimai, “Usage of Apache Cassandra for Prototyping the Event Metadata System of the NICA Experiments”, Physics of Particles and Nuclei Letters, 2022, Vol. 19, No. 5, pp. 562–565.

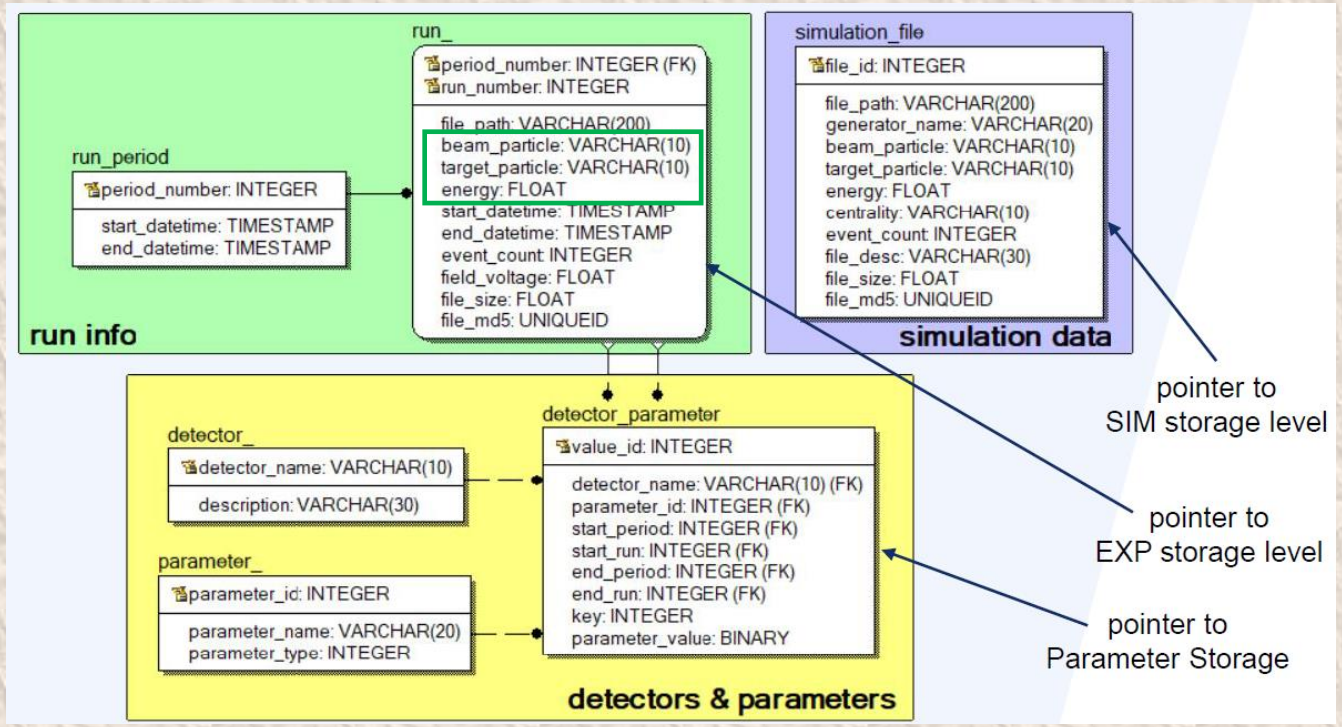
Current BM@N Database Schemas

EMD Database:



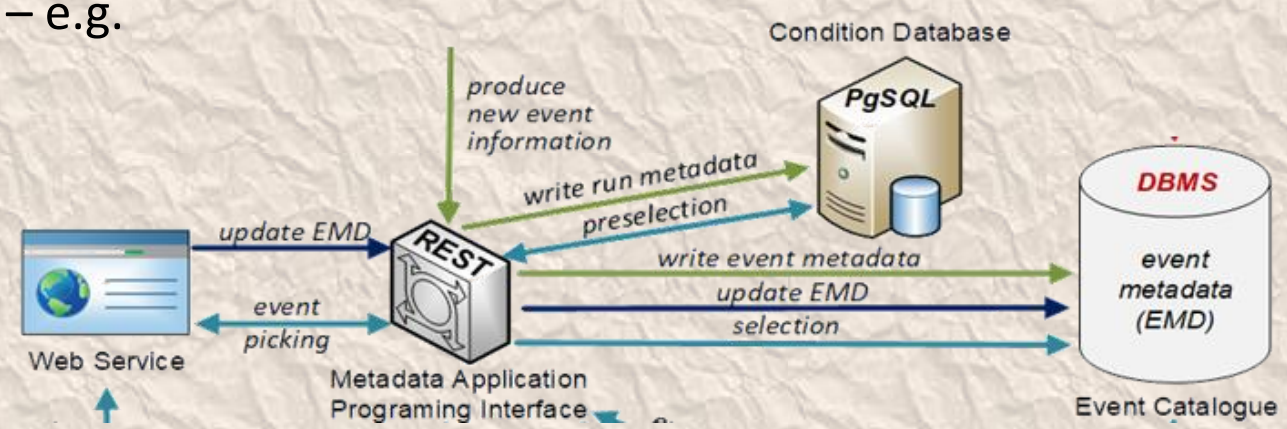
Configurable

Condition Database:



REST API and Web UI Implementation

- Using Kotlin programming language
  - Multiplatform
  - JVM runtime back end
  - ktor framework for REST API
  - React-based front end
  - Kotlin-wrappers (MUI)
    - Not all components readily available – e.g. DataGrid
- Packed in Docker
- Configuration file in YAML

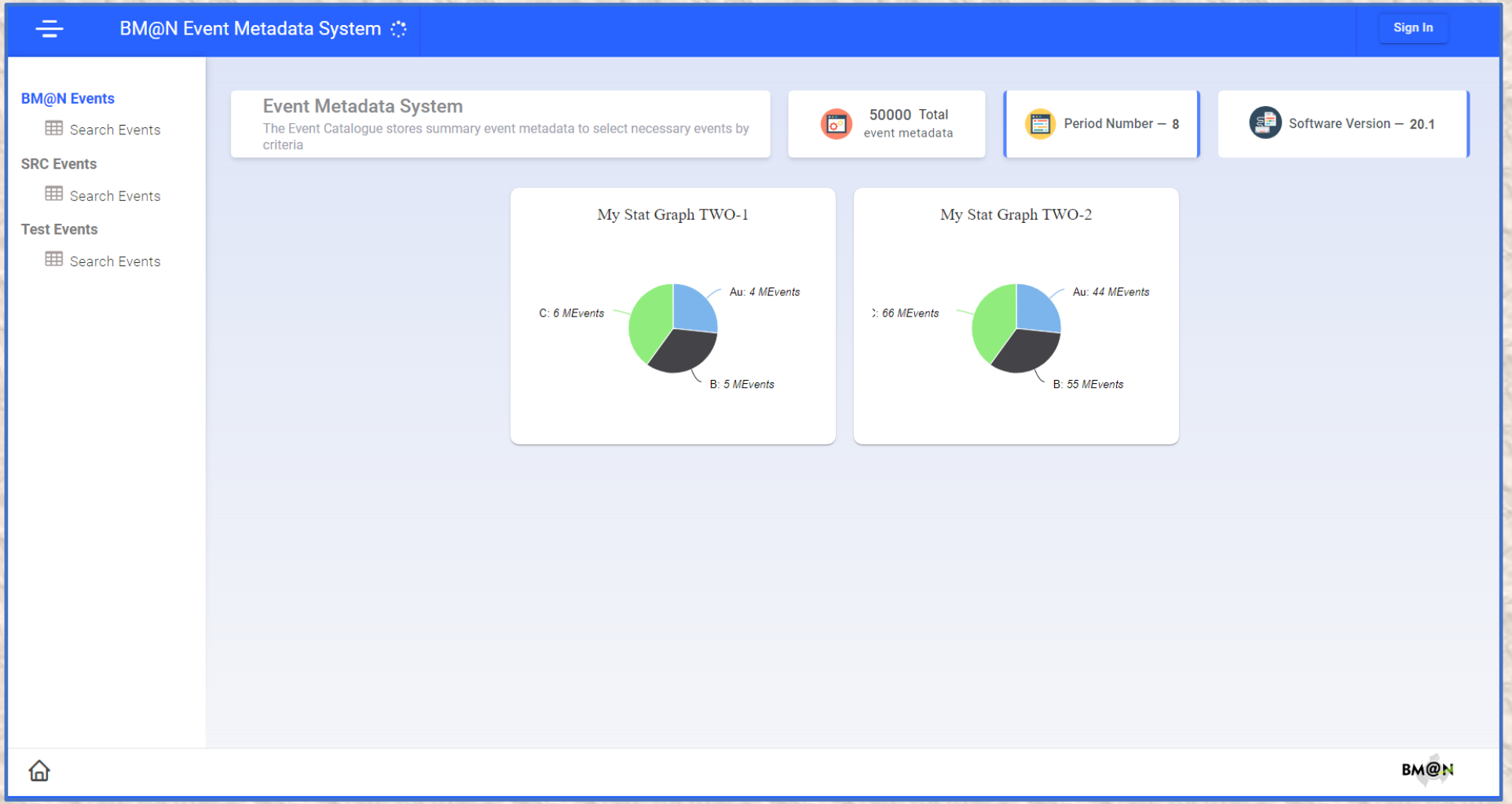


Configuration File Example

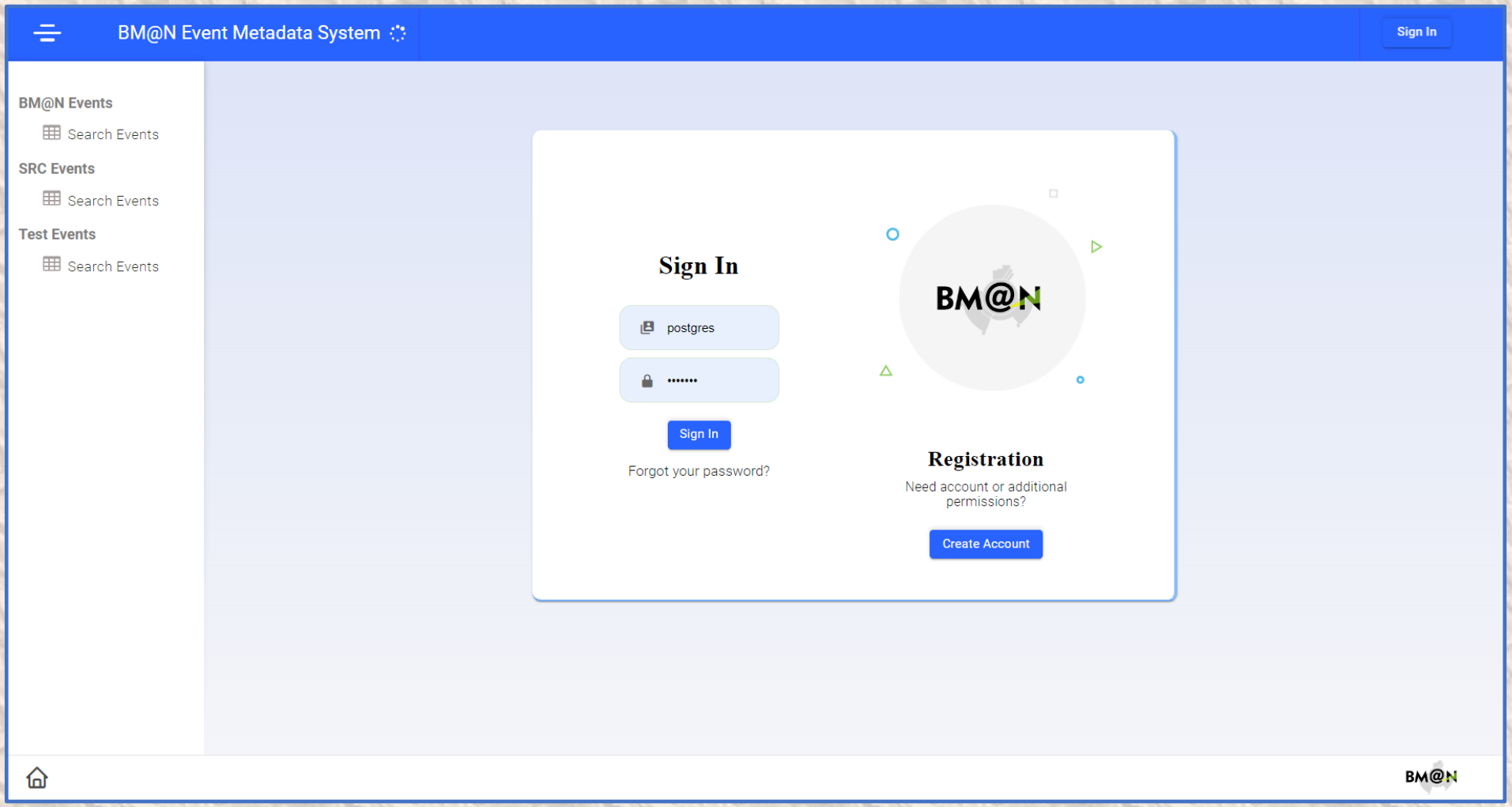
```
event_db:      # condition_db - similar
  host: ***
  port: ***
  db_name: ***
  user: ***
  password: ***

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"
```

New Web UI Main Page



Sign-In


BM@N Event Metadata System
Sign In
BM@N Events
Search Events
SRC Events
Search Events
Test Events
Search Events
Sign In
postgres
Sign In
Forgot your password?
BM@N
Registration
Need account or additional permissions?
Create Account
BM@N

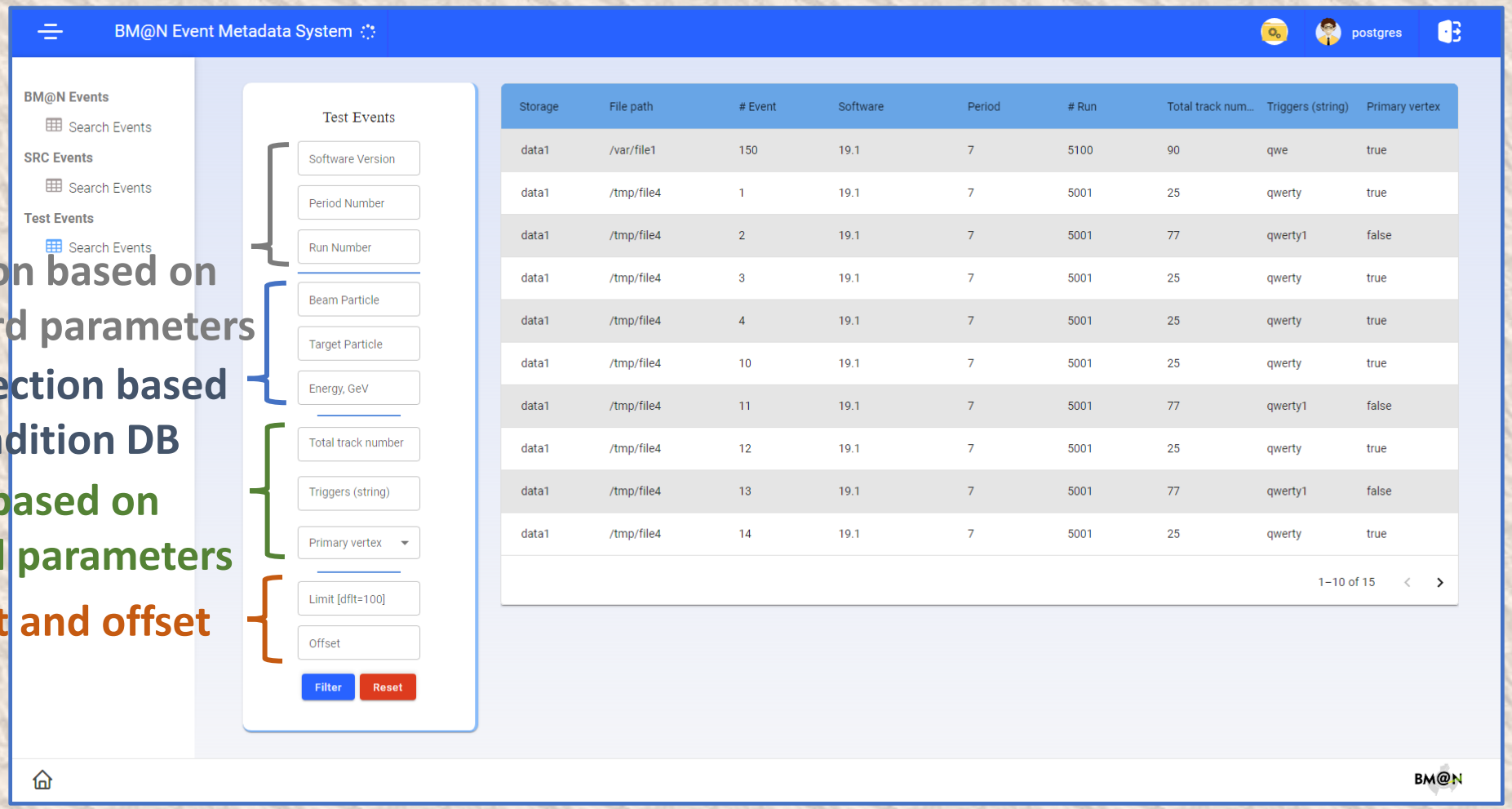
Main search page

Selection based on standard parameters

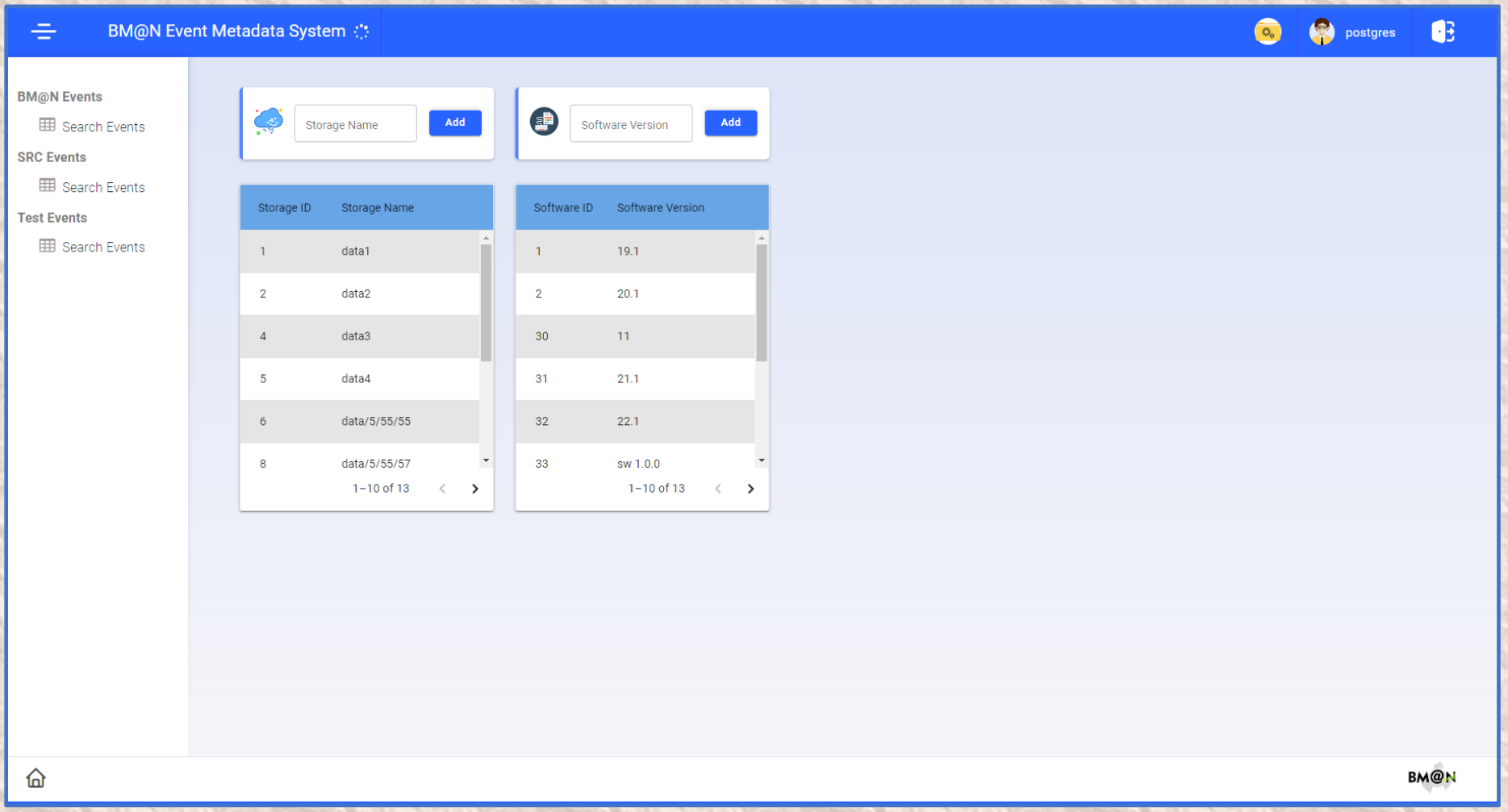
Preselection based on Condition DB

Selection based on configured parameters

Limit and offset



Dictionaries



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.

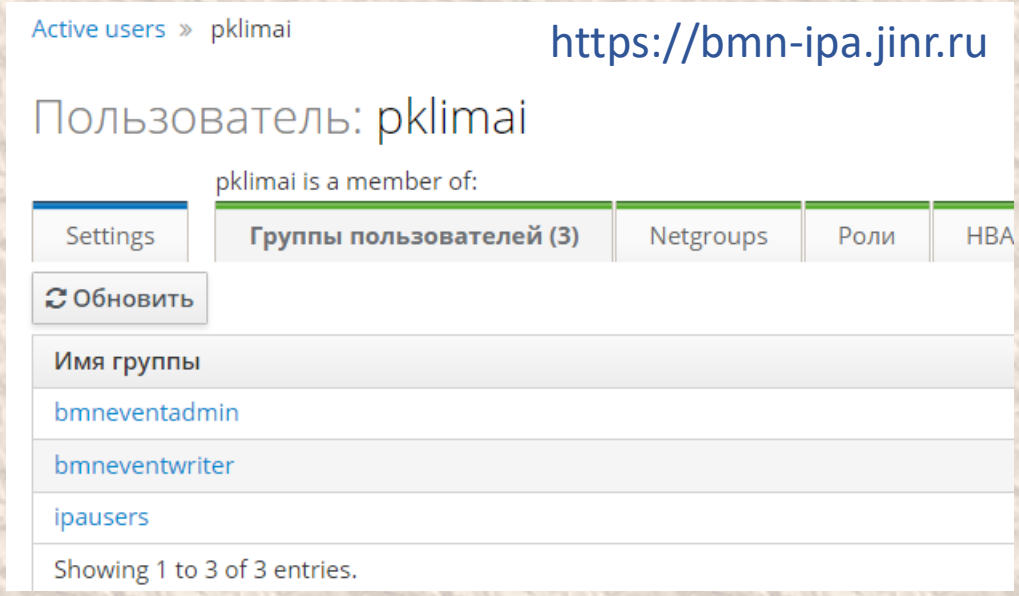
```
/event_api/v1/bmn/events?period_number=7&run_number=
5000+&software_version=19.1&track_number=10-15
```


```
{
  - 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
```

EMS FreeIPA Integration

```
user_auth:
  ldap_server: bmn-ipa.jinr.ru
  ldap_port: 389
  user_dn_format: "uid=%s,cn=users,cn=accounts,dc=jinr,dc=ru"
  ldap_username: *****
  ldap_password: *****
  writer_group_dn: "cn=bmneventwriter,cn=groups,cn=accounts,dc=jinr,dc=ru"
  admin_group_dn: "cn=bmneventadmin,cn=groups,cn=accounts,dc=jinr,dc=ru"
```

- Writer or admin role – can create events in the catalogue
- Admin role – can delete event records
- Everyone else – read only access



EMS Current Status

- Event Metadata System current status
  - EMS Database based on PostgreSQL is deployed
  - Integrates with BM@N Condition database
  - REST API and Web UI
  - Macro to write BM@N events in the catalogue
  - Role-based access control implemented
  - Monitoring
- Work in progress
  - Fill the event catalogue with actual BM@N events
  - HA and Backup
  - Automated Deployment

Thank You!