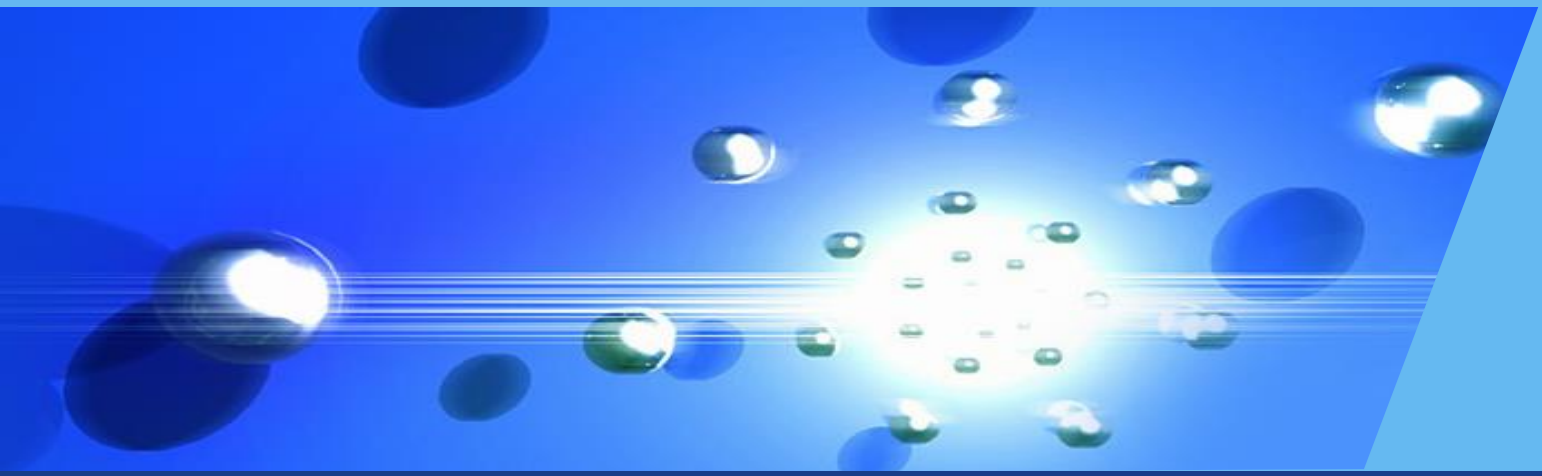


BM@N Software Development
Summary of the Software Session

Konstantin Gertsenberger

V. Veksler and A. Baldin Laboratory of High Energy Physics
Joint Institute for Nuclear Research, Dubna, Russia

JINR, Dubna
19-20 April 2021

BM@N Software Contribution

Research

Peter KLIMAI (19 April 12:05)
Software contribution from MIPT: implementation of systems and services for BM@N

MIPT (NPM) group (Head: Tagir AUSHEV)

Sergei NEMNYUGIN (19 April 11:40)
Software contribution from SPbU: algorithmic and code optimization of the BmnRoot framework

SPbU group (Head: Sergei NEMNYUGIN)

JINR LIT (Director: Vladimir KORENKOV)

Irina FILOZOVA, Igor ALEXANDROV, Evgeniy ALEXANDROV and staff: Development of Information Systems in frame of the RFBR grant

JINR LHEP (Spokesperson: Mikhail KAPISHIN)

Konstantin GERTSENBERGER
Alexander CHEBOTOV

BM@N Software Contribution
(RFBR grant till begin of 2022 + in-kind contribution)

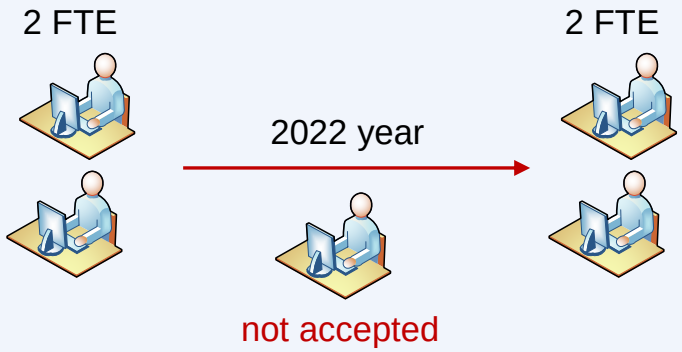
Software Group Status

BM@N Software Group (2 person):

Konstantin GERTSENBERGER: group leader

Alexander CHEBOTOV: software engineer in JINR since 2018

The Software Group has no department connection with the Department of the BM@N experiment → problems with asking the division for resources for the BM@N Software Group



BM@N Computing and Technical Contribution

BM@N Computing & Technical Contribution
(no financial support, own motivation)

JINR LHEP (Computing Leader: Andrey DOLBILOV)

Ivan SLEPOV:
Deployment of the information services for BM@N on the NICA cluster

JINR LIT (Director: Vladimir KORENKOV)

Nikita BALASHOV: ***CVMFS Deployment, GitLab Services, Docker Containers***

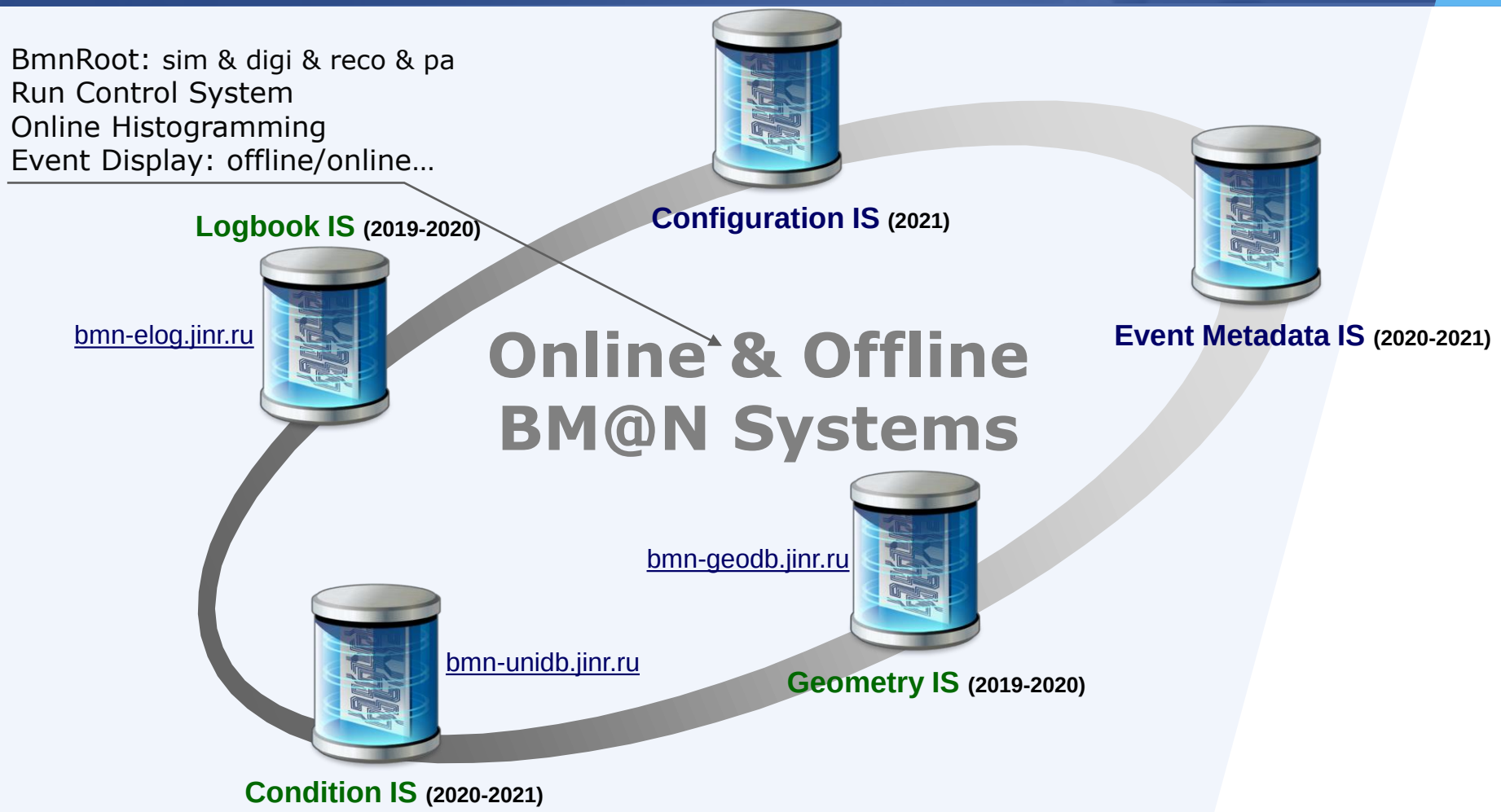
Dmitriy PODGAYNY, Oksana STRELTSOVA, Maksim ZUEV: ***HybriLIT and SC Govorun support***

Igor PELEVANYUK: ***DIRAC workload management system***

Vladimir TROFIMOV, Daria PRIAKHINA, et. al: ***Simulation of BM@N data and processing centers***

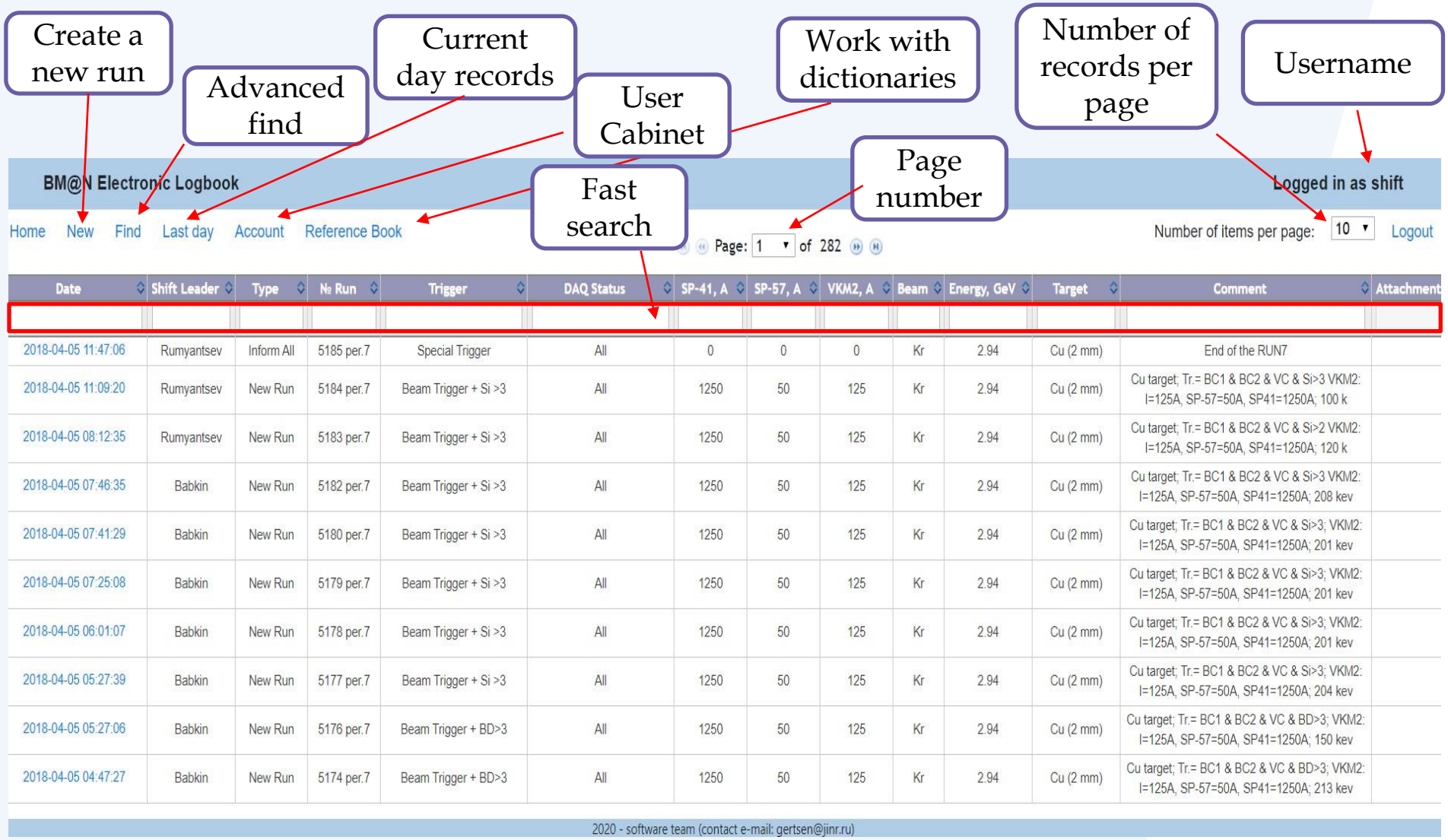
Information Systems:
Databases + Services

Information Systems for online & offline processing



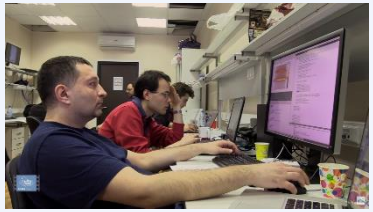
RFBR Grant 2019 – 2021: Development of Information Systems for Online and Offline Data Processing for the Experimental Setups of the NICA Complex

Current version of the e-Log platform

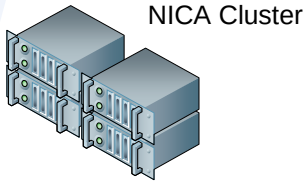


Electronic Logbook Completeness

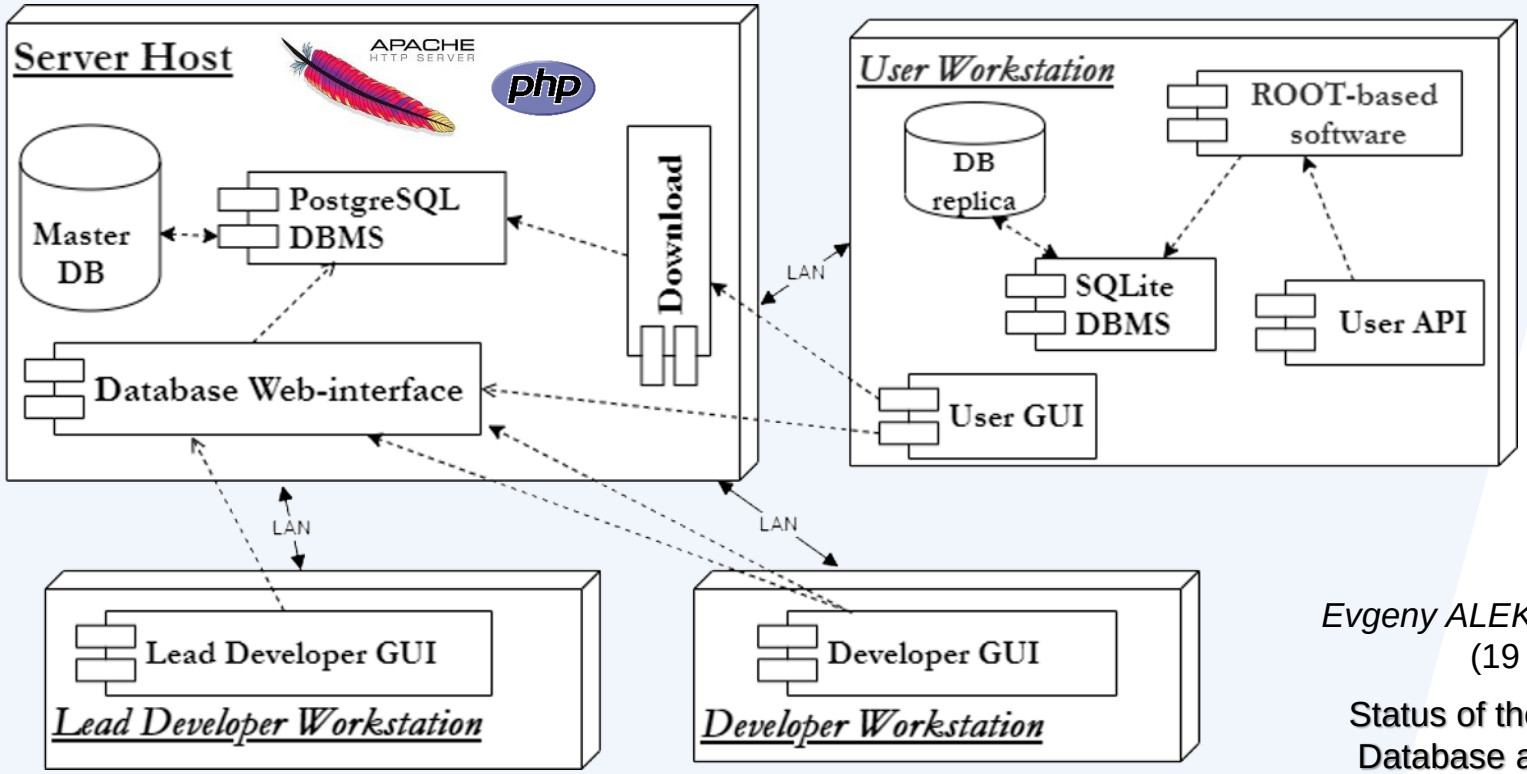
- The Electronic Logbook was employed in the last session of the BM@N experiment, and, at present, it contains records for conducted BM@N runs.
- The Electronic Logbook System has been deployed on the NICA cluster and is ready to use it for the next runs.
- The Common Deployment System can be used to install the Online Logbook System for all the experiments of the NICA project taking into account some specifics of the experiments. db-nica.jinr.ru/mpd_elog/
- The Electronic Logbook System has been implemented (now tests of the deployment system are performed) and are actively used by collaboration members.



BM@N Electronic Logbook
Runs 1 - 7
records ~ 3 000



Geometry Information System Architecture



Evgeny ALEKSANDROV
(19 April 10:00)

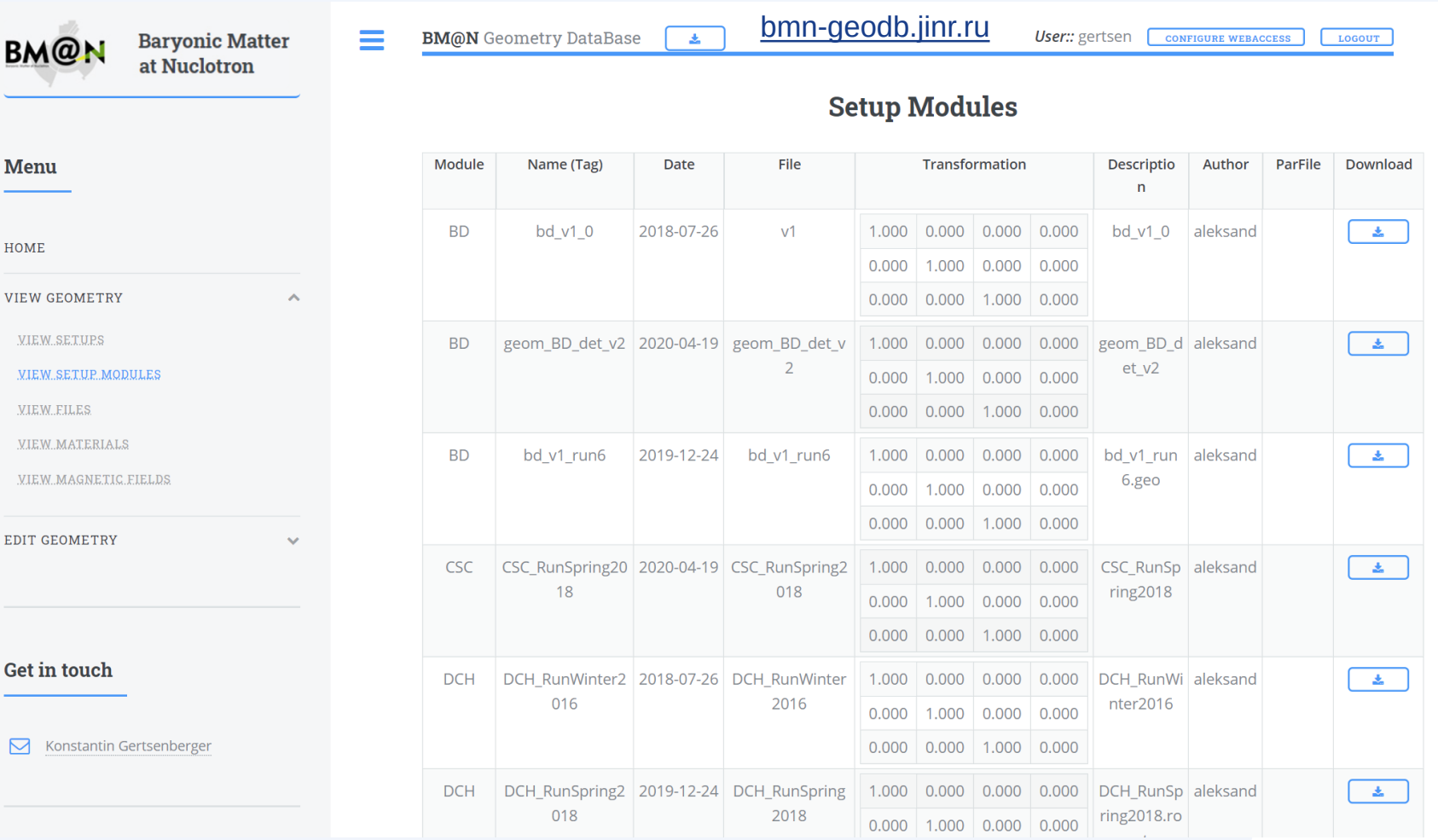
Status of the Geometry Database and steps to integrate into BM@N

Three user roles:
Lead Developer | Developer | User (Reader)

PostgreSQL

SQLite

Status of the Geometry Web Platform



Module	Name (Tag)	Date	File	Transformation	Description	Author	ParFile	Download
BD	bd_v1_0	2018-07-26	v1	1.000 0.000 0.000 0.000 0.000 1.000 0.000 0.000 0.000 0.000 1.000 0.000	bd_v1_0	aleksand		
BD	geom_BD_det_v2	2020-04-19	geom_BD_det_v2	1.000 0.000 0.000 0.000 0.000 1.000 0.000 0.000 0.000 0.000 1.000 0.000	geom_BD_det_v2	aleksand		
BD	bd_v1_run6	2019-12-24	bd_v1_run6	1.000 0.000 0.000 0.000 0.000 1.000 0.000 0.000 0.000 0.000 1.000 0.000	bd_v1_run6.geo	aleksand		
CSC	CSC_RunSpring2018	2020-04-19	CSC_RunSpring2018	1.000 0.000 0.000 0.000 0.000 1.000 0.000 0.000 0.000 0.000 1.000 0.000	CSC_RunSpring2018	aleksand		
DCH	DCH_RunWinter2016	2018-07-26	DCH_RunWinter2016	1.000 0.000 0.000 0.000 0.000 1.000 0.000 0.000 0.000 0.000 1.000 0.000	DCH_RunWinter2016	aleksand		
DCH	DCH_RunSpring2018	2019-12-24	DCH_RunSpring2018	1.000 0.000 0.000 0.000 0.000 1.000 0.000 0.000	DCH_RunSpring2018.ro	aleksand		

BM@N Geometry Database has filled with setup geometries for Run 7 and 6 (all releases + dev)

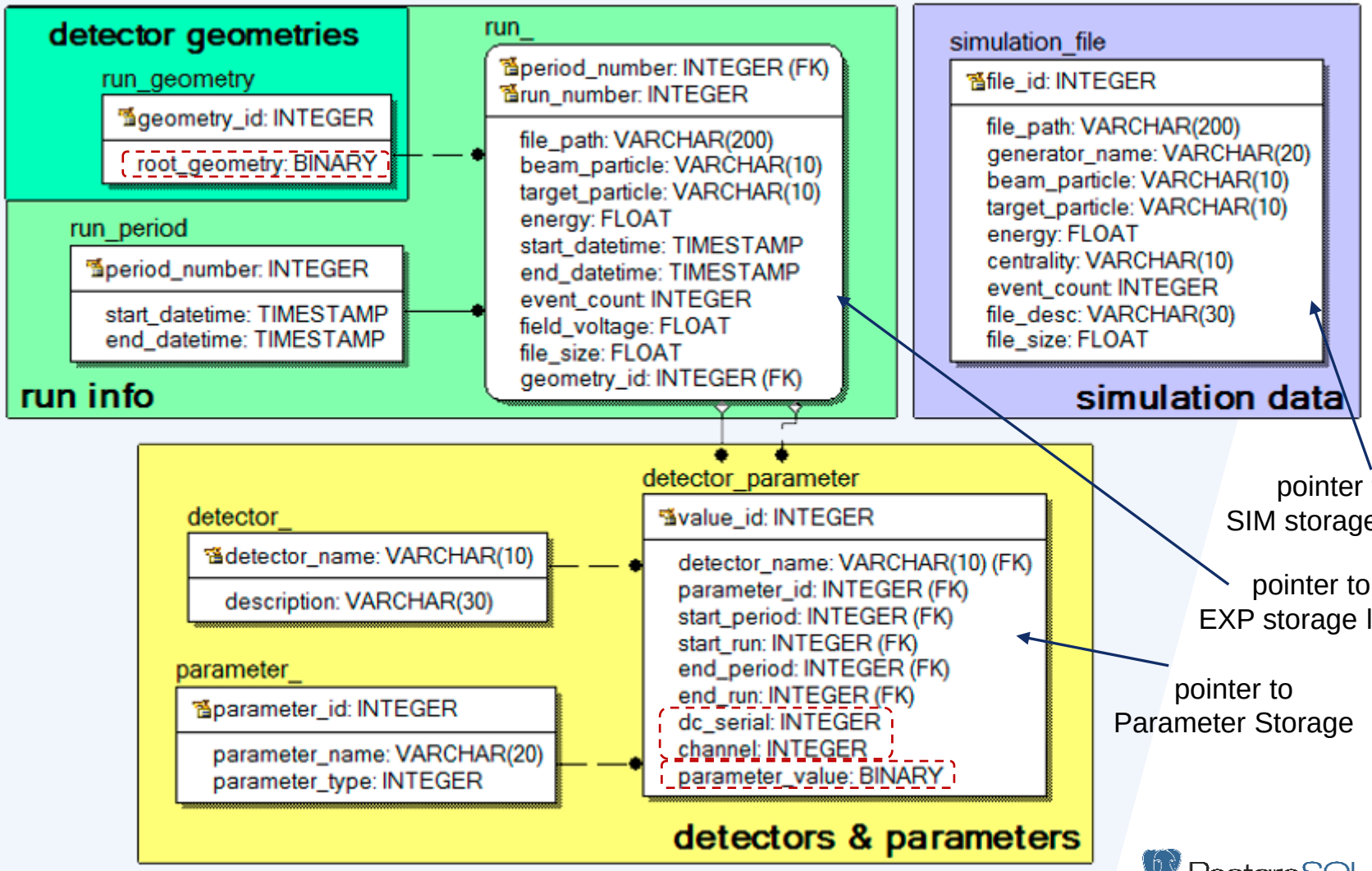
Graphical User Interface Functions:

View Edit Download

Geometry Information System Completeness

- The developed Geometry Information System provides a central storage of the detector geometries and a set of convenient tools to manage geometry modules and software assemblies of various setup versions in the form of a combination of these modules and auxiliary files containing a description of magnetic field, detector materials and media.
- The geometries of the experiments stored in the database can be further used to process event data of the particle collisions.
- Collaboration members can search and select a required version of the geometry setup to work with.
- The Geometry Information System has filled with setup geometries for the last BM@N Run 7 an 6 and is ready to apply for the BM@N experiment.
- It can be employed in the other NICA experiments using developed unified installation script and customized interfaces.

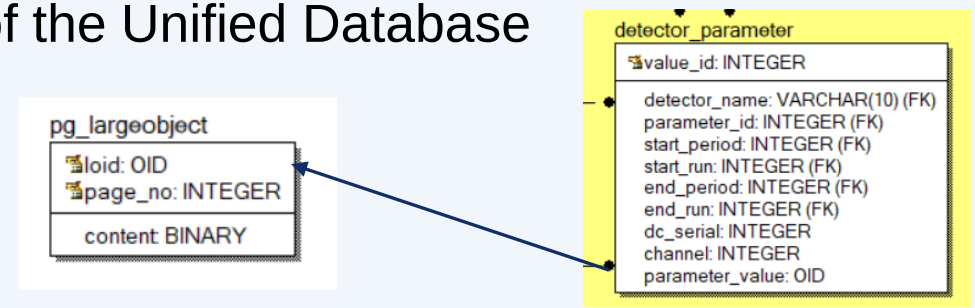
Unified Database Diagram (before)



PostgreSQL 8.4

BM@N Database: Encountered Problems

1. PostgreSQL 9 (and subsequent) limits access to the *pg_largeobject* table with binary data of the Unified Database



2. Detector parameters were stored as packed structures, it seriously constrained possible parameter types

struct IIStructure { int int_1; int int_2; }; **__attribute__((packed)) ?**

3. Separate functions for each parameter type sufficiently increased the size and complexity of the code.

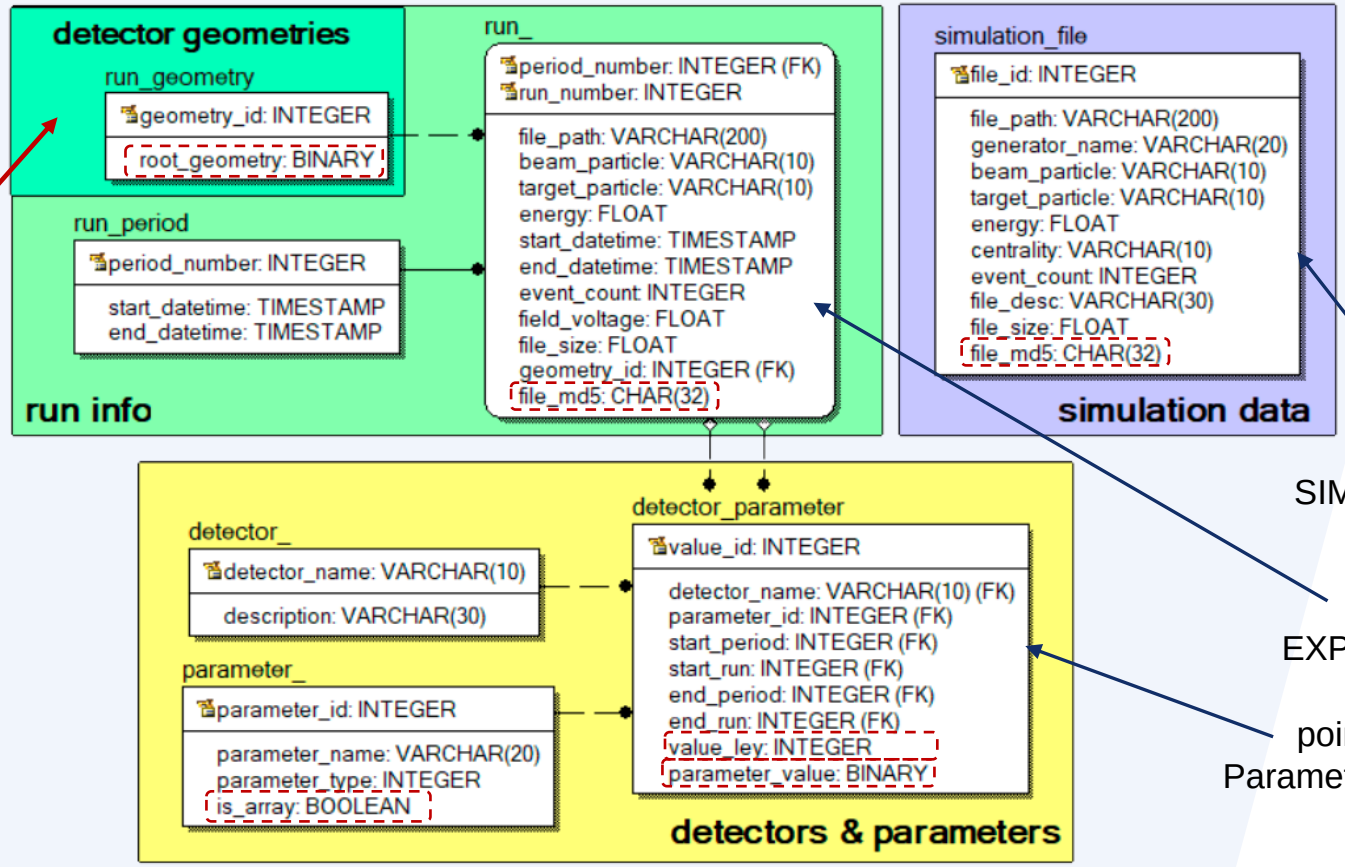
4. '*dc_serial*' and '*channel*' attributes of the detector parameters (t. 15 571) seriously increased the number of database calls and duplicated all functions to work with, e.g.:

CreateDetectorParameter(detector, parameter, start_per, start_run, end_pe, end_run, GemPedestalStructure* value, count)
CreateDetectorParameter(detector, parameter, start_per, start_run, end_per, end_run, dc_serial, channel, GemPedestalStructure* value…)

Moving to the New Version of the Database

1. Binary data were moved from *pg_largeobject* table directly to the *parameter_value* (*detector_parameter* table) and *root_geometry* (*run_geometry* table). A new PostgreSQL 12 instance has been deployed.
2. The following solutions were considered to replace packed structures: ZeroMQ, MessagePack, BOOST, Protobuf, FlatBuffers, ROOT/TStreamer, C++ manual serialization. C++ manual serialization was implemented with *UniValue* base class.
3. Separate functions for each parameter type were unified into **GetValue**(vector<UniValue*>& parameter_value) and **GetValue**(UniValue*& parameter_value), and corresponding setters.
4. '*dc_serial*' and '*channel*' attributes of the detector parameters (t. 971) were moved inside the parameter types. Integer '*value_key*' attribute was added to the table.
5. MD5 hash was added to the tables with experimental and simulated file paths to ensure integrity of the stored files with BM@N events.

{ Unified → Condition } Database

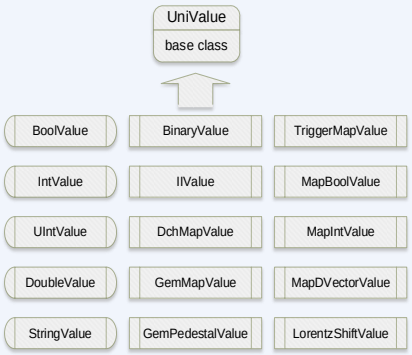


pointer to Geometry Storage

pointer to SIM storage level

pointer to EXP storage level

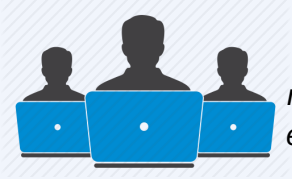
pointer to Parameter Storage



storing information on experiment sessions and runs, setup geometries, detectors, parameters and parameter values, and generated simulation files

PostgreSQL 12

BM@N Database for offline processing

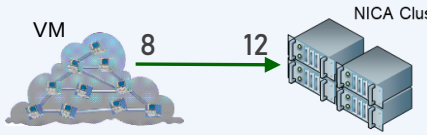


detector simulation
raw data processing
event reconstruction
physics analysis

BmnRoot

C++ database interface w/o SQL (connect, SQL I/O)

Serialization of BM@N parameters was restructured



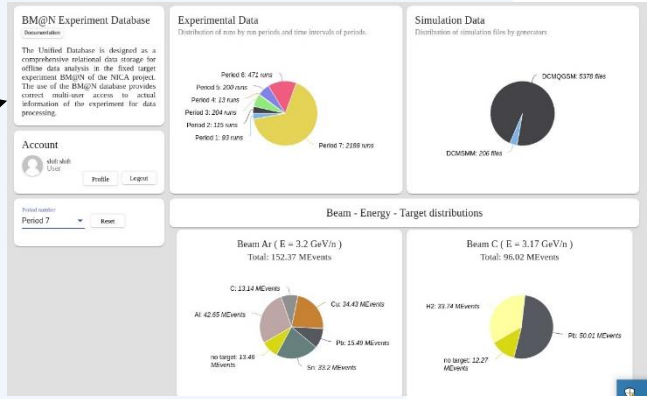
ROOT

Node.JS

Condition Database

users

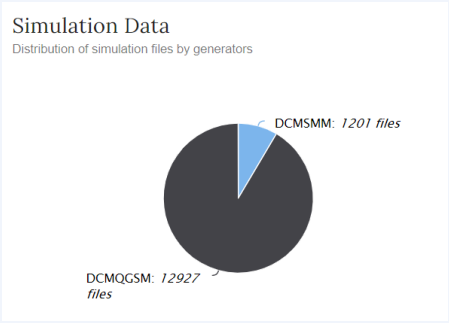
FreeIPA | authentication



Web interface (bmn-unidb.jinr.ru)

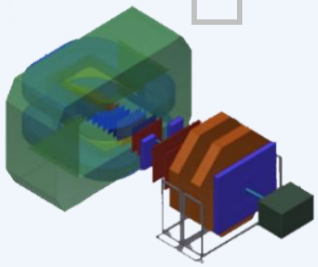
- Some improvements were added to the Web service
- Added viewing parameters of all the types
- Visualization of the stored BM@N setup geometries was added

python script for auto update of simulation file list

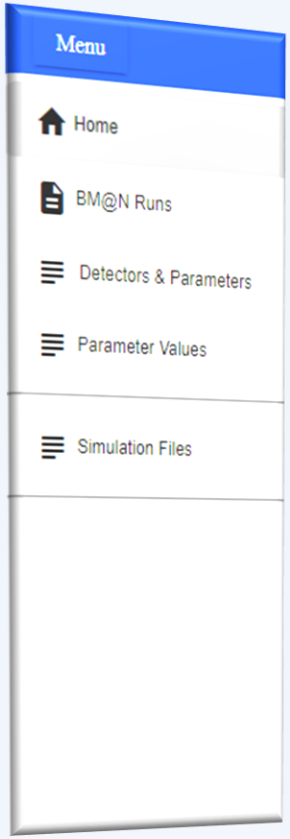


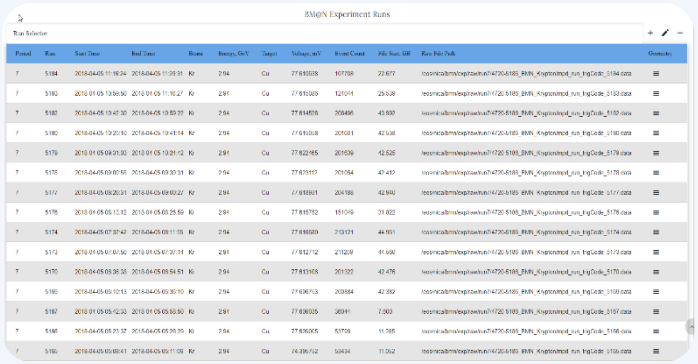
configuration calibration

parameter and algorithm data

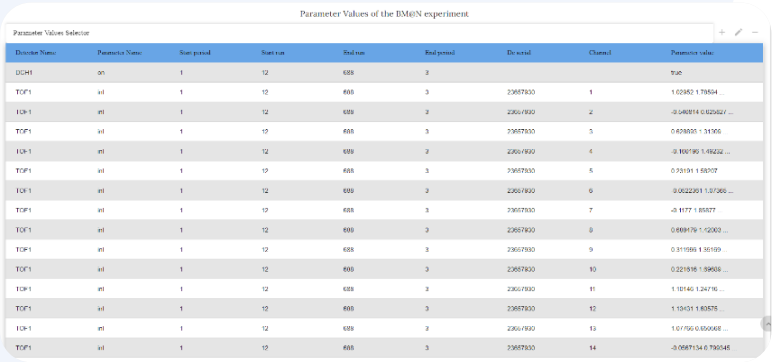


Tabular View of the BM@N stored data

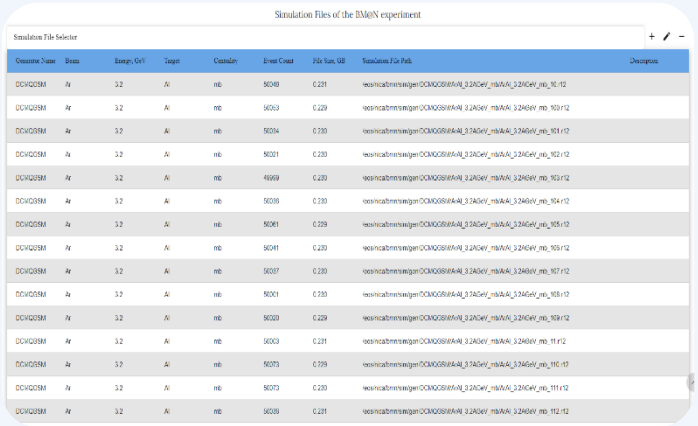




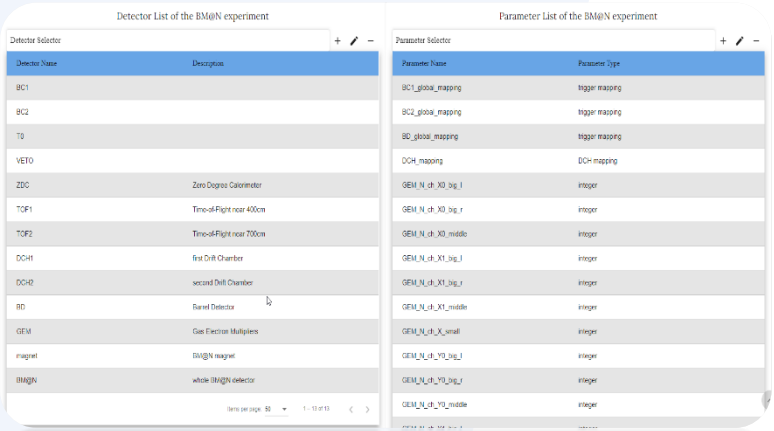
BM@N Runs



Parameter Values

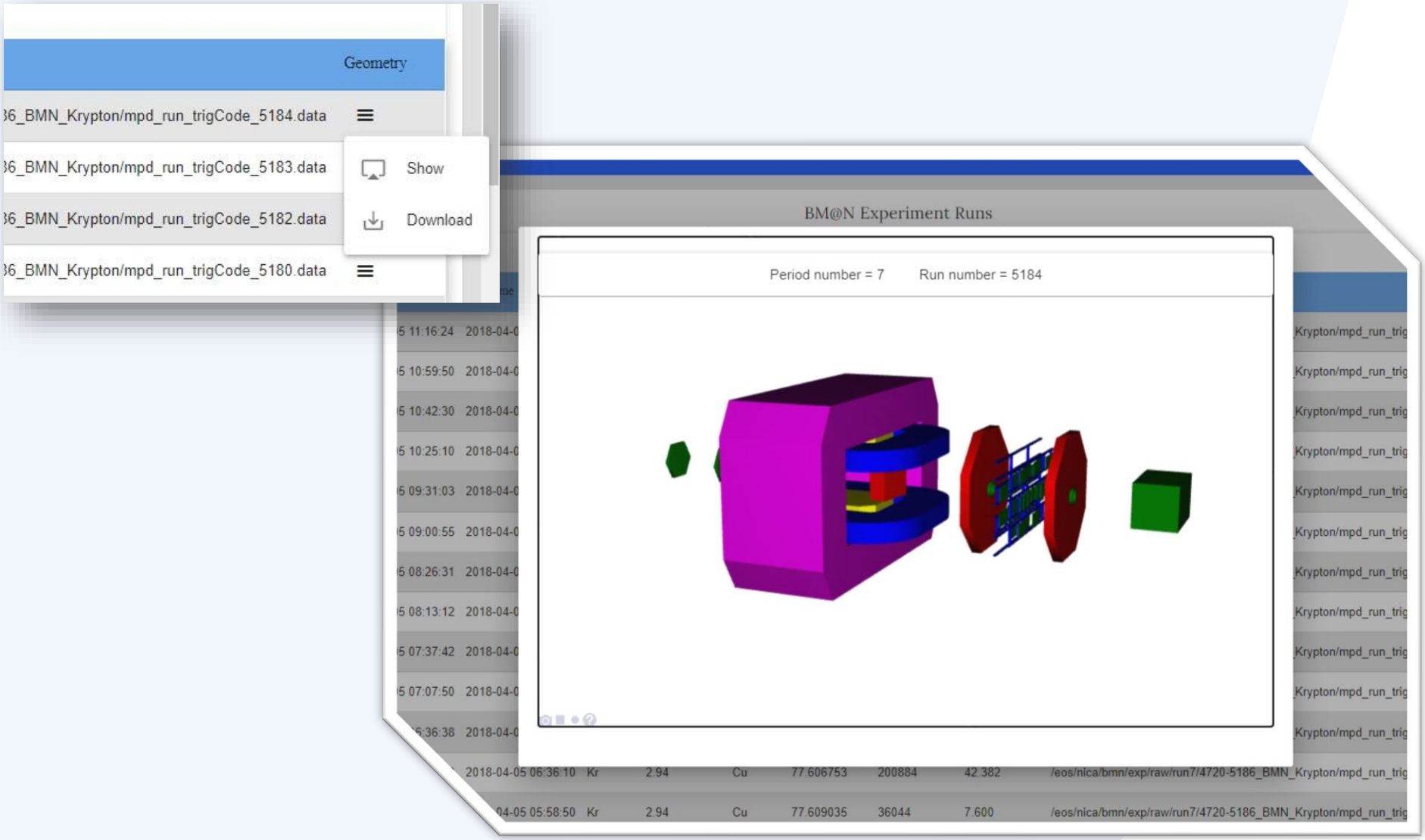


Simulation Files



Detector & Parameters

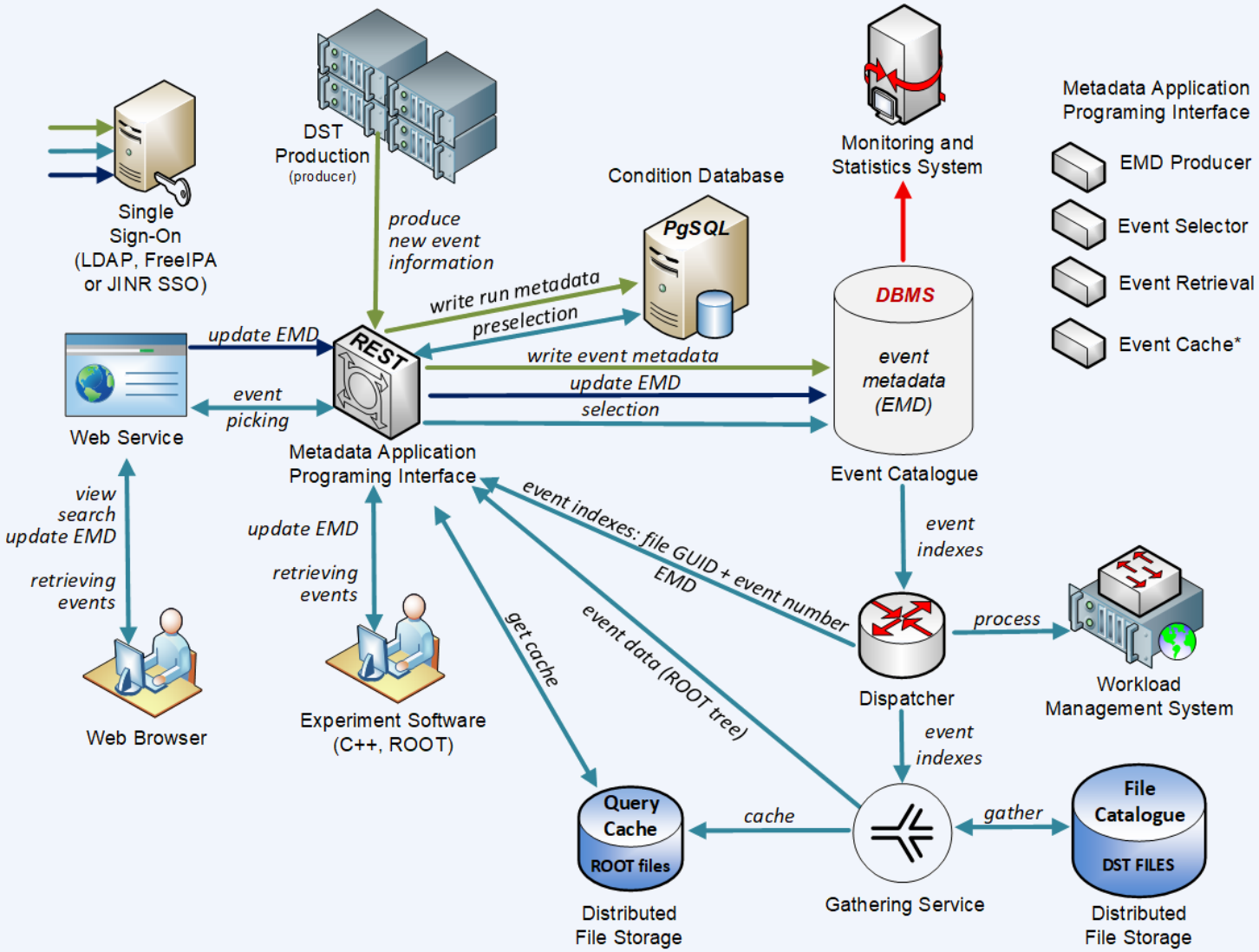
Visualization of the stored BM@N geometry



Event Metadata System (EMS)

- main functions are description of particle collision events, storing of necessary event metadata, their management and convenient access, and organizing online and offline interfaces to the metadata
- is based on the Event Database called Event Catalogue, which contains summary properties of particle collision events and references to their storage location
- allows user to quickly search for a set of events required for a particular physics analysis by various criteria and parameters
- is responsible for creating, maintaining and checking the quality of the catalogue of physics events

Architecture of the Event Metadata System



Web interface

for viewing and searching for event metadata stored in the Event Catalogue and retrieving events which satisfy given user parameters

Metadata API

for writing new metadata to the Event Catalogue while data processing and requesting events selected by criteria for physics analysis in BmnRoot

Event Metadata Structure (prototypes)

- period and run number (4+4 bytes)

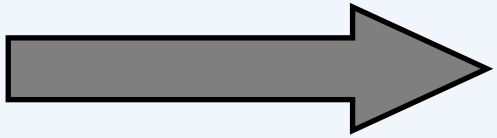
- **file pointer (GUID) (4 byte)**
- **event number (4 byte)**

- software version (2 bytes)
- event time (4 byte)
- flag to determine whether primary vertex was found (1 byte)
- number of primary tracks (4 byte)
- number of all reconstructed tracks (4 byte)
- track number of positively charged particles from primary vertex (4 byte)
- primary & secondary particles (4+4 bytes)
- number of hits by detectors (4 bytes)
- total input charge in the event (4 byte)
- total output charge in the event (4 byte)

EMS provide the following functions: summary description of collision events and their identifiers, which can be used to select events for a desired analysis goal; recording and storing event metadata in the Event Catalogue; management and a convenient access to the metadata; organization of online and offline interfaces for selecting events of interest

DBMS for Event Catalogue

It is assumed that the number of the events will increase from the current value of hundreds of million BM@N events to billions of events per year (terabytes of event metadata)



high performance

correct multi-access

good scalability

easy configuration

universal design

Alexander YAKOVLEV
(19 April 10:20)

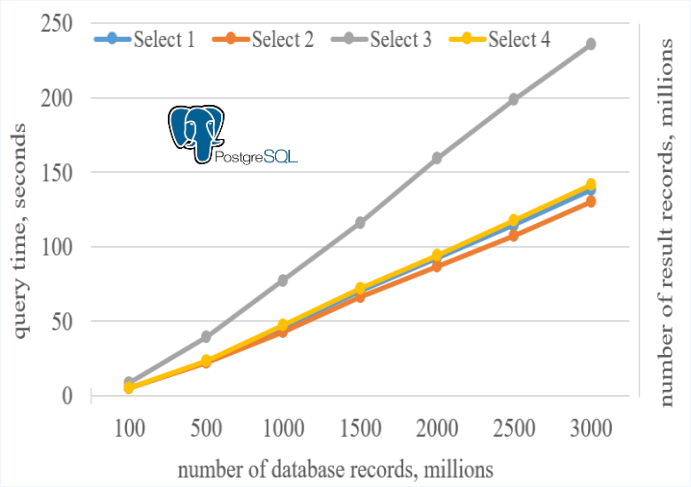
Design of the Event Metadata System for BM@N and testing the first prototypes

Prototypes of the Event Catalogue (single-node)

Configuration VM:

2 x Intel Xeon E5-2680
DDR4 240 GB 2133 MHz
SSD 400 GB Intel

Scientific Linux 7.9
PostgreSQL 12.5
HBase 2.2.3, Hadoop 3.2.1

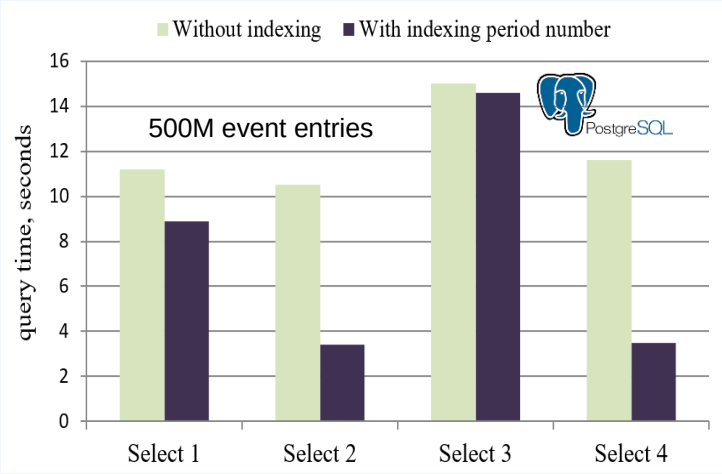


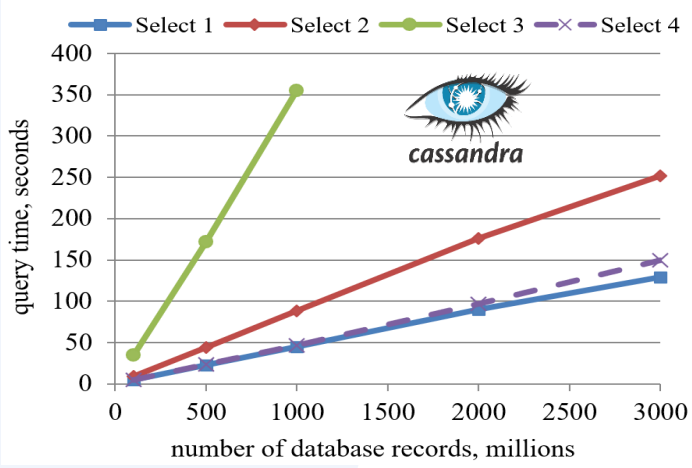
500M	HBase C1	HBase C2	HBase C3
Test 0	56 min	28 min	63 min
Test 1	29 min 55 sec	28 min 02 sec	5 min
Test 2	32 min 4 sec	28 min 47 sec	11 min
Test 3	30 min 20 sec	29 min 52 sec	8 min
Test 4	28 min	not supported by Apache Phoenix	1 min 52 sec
Test 5	29 min	not supported by Apache Phoenix	2 min 12 sec

Configuration:

Intel Core i9-10900F
DDR4 64 Gb 3200MHz
SSD 1TB NVMe Samsung

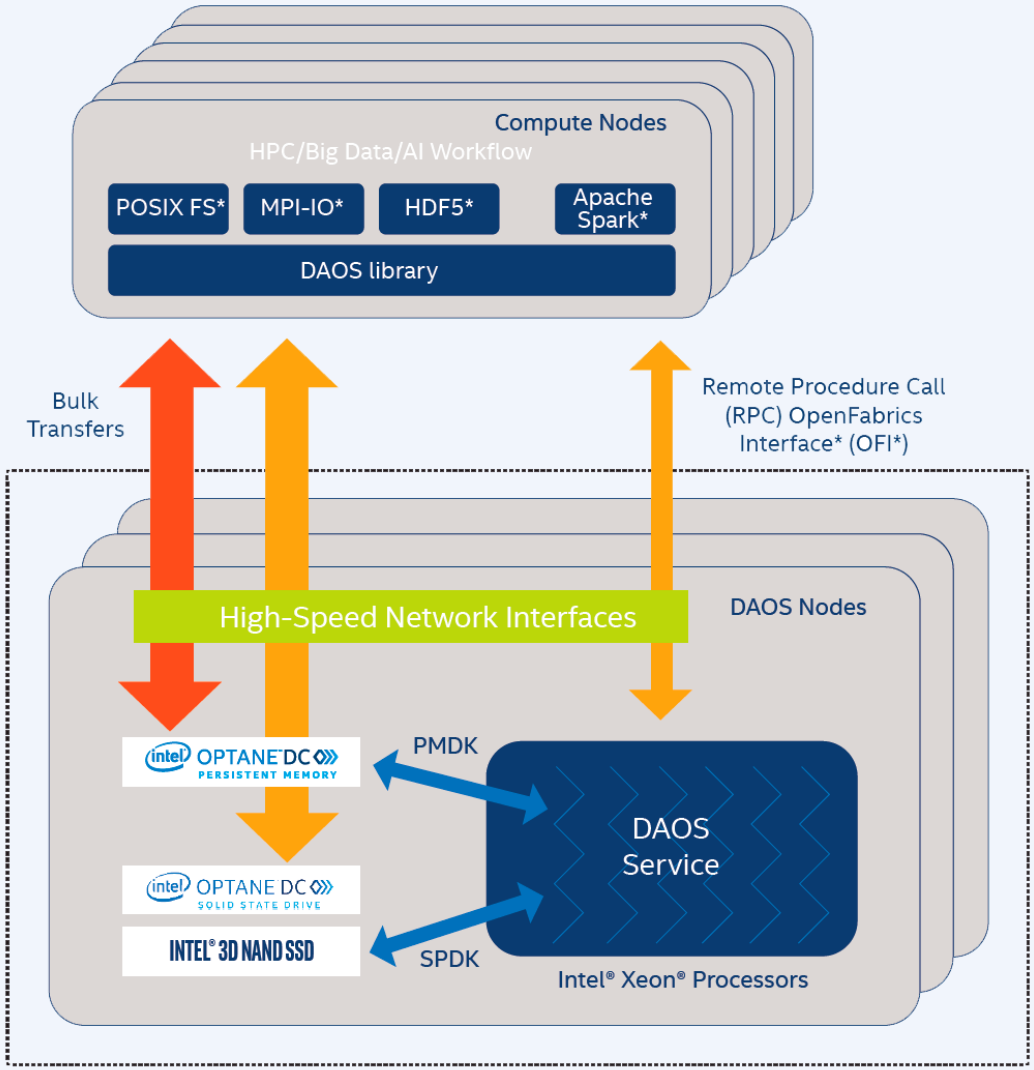
CentOS Linux 8.2
PostgreSQL 12.5
Apache Cassandra 3.11.8





primary key: ((*PERIOD_NUMBER*, *SOFTWARE_ID*), *PRIMARY_TRACKS*, *EVENT_NUMBER*)

Intel DAOS with Optane for HPC Storage



8 servers for storage with
2x Intel Optane 512 Gb
RAM 192 Gb

Intel OPA 100 Gbit/s

Mikhail MATVEEV
(19 April 11:00)

Intel DAOS with Optane Technology for High-Performance Storage

Event Metadata Structure

the first metadata structure was accepted at the software meeting

write event metadata only
if primary vertex was
found in the event

BM@N program

- **file pointer (GUID) (4 byte)**
- **event number (4 byte)**

- period and run number (4+4 bytes)
- software version (2 bytes)
- number of all reconstructed tracks (4 byte)

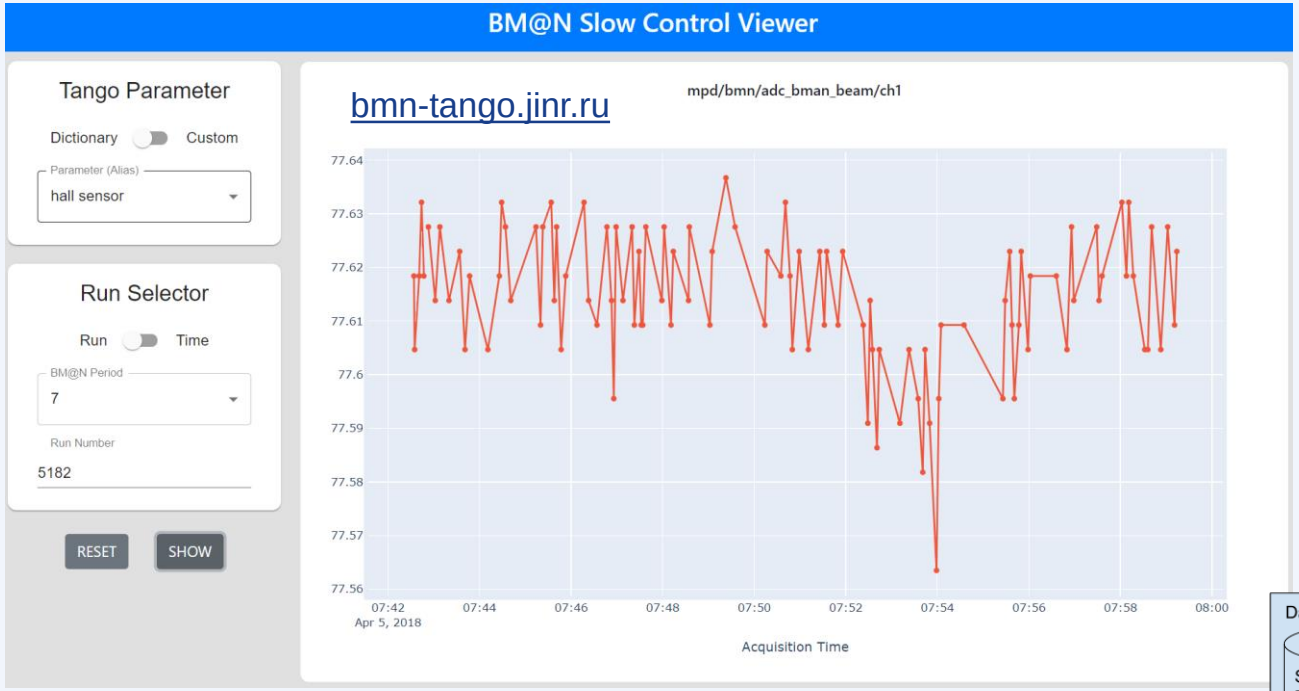
SRC program

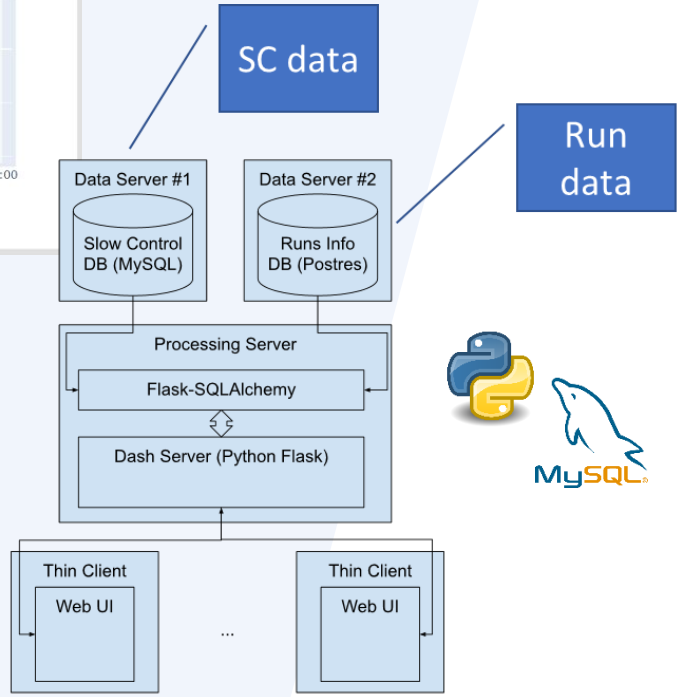
- **file pointer (GUID) (4 byte)**
- **event number (4 byte)**

- period and run number (4+4 bytes)
- software version (2 bytes)
- number of all reconstructed tracks (4 byte)
- total input charge in the event (4 byte)
- total output charge in the event (4 byte)

Additional Services

Tango (Slow Control) Viewer



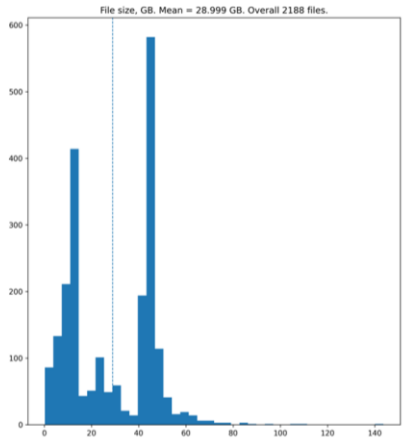


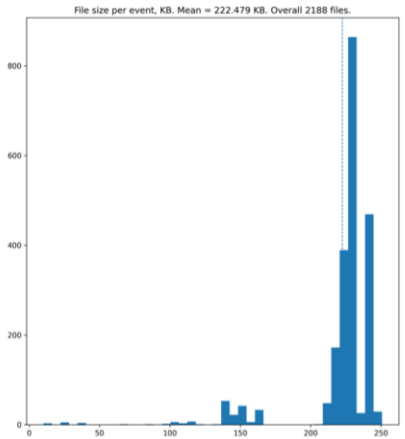
- Web interface for BM@N slow control system
- Shows sensor data graph based on №run or time interval, and parameter name (dictionary or custom)
- If a parameter is 1D array, in this case a multigraph is displayed
- Uses Dash framework and packed in Docker container

Statistics Collection & Reprocessing Service

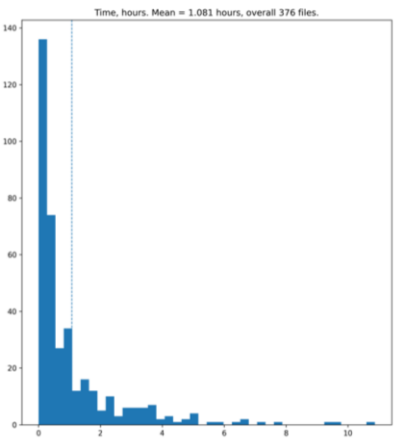
- Shows histograms and summary data for:
  - File (run) sizes and sizes per event for a given directory with data
    & checks integrity of files: event number being equal to a value in the database
  - Processing time for runs and times per event by parsing job logs
    & defines failed jobs and forms a list of data files for reprocessing
- Implemented on Python (stats.py)
- https://git.jinr.ru/nica/bmnroot/uni_db/services/statistics

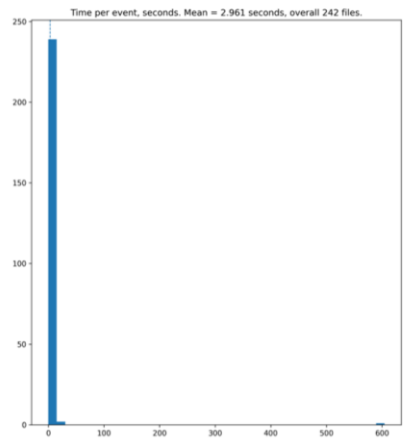
```
# python3 stats.py --size --dir
/eos/nica/bmn/exp/raw/run7/ --config
config-size.json --recursive
```





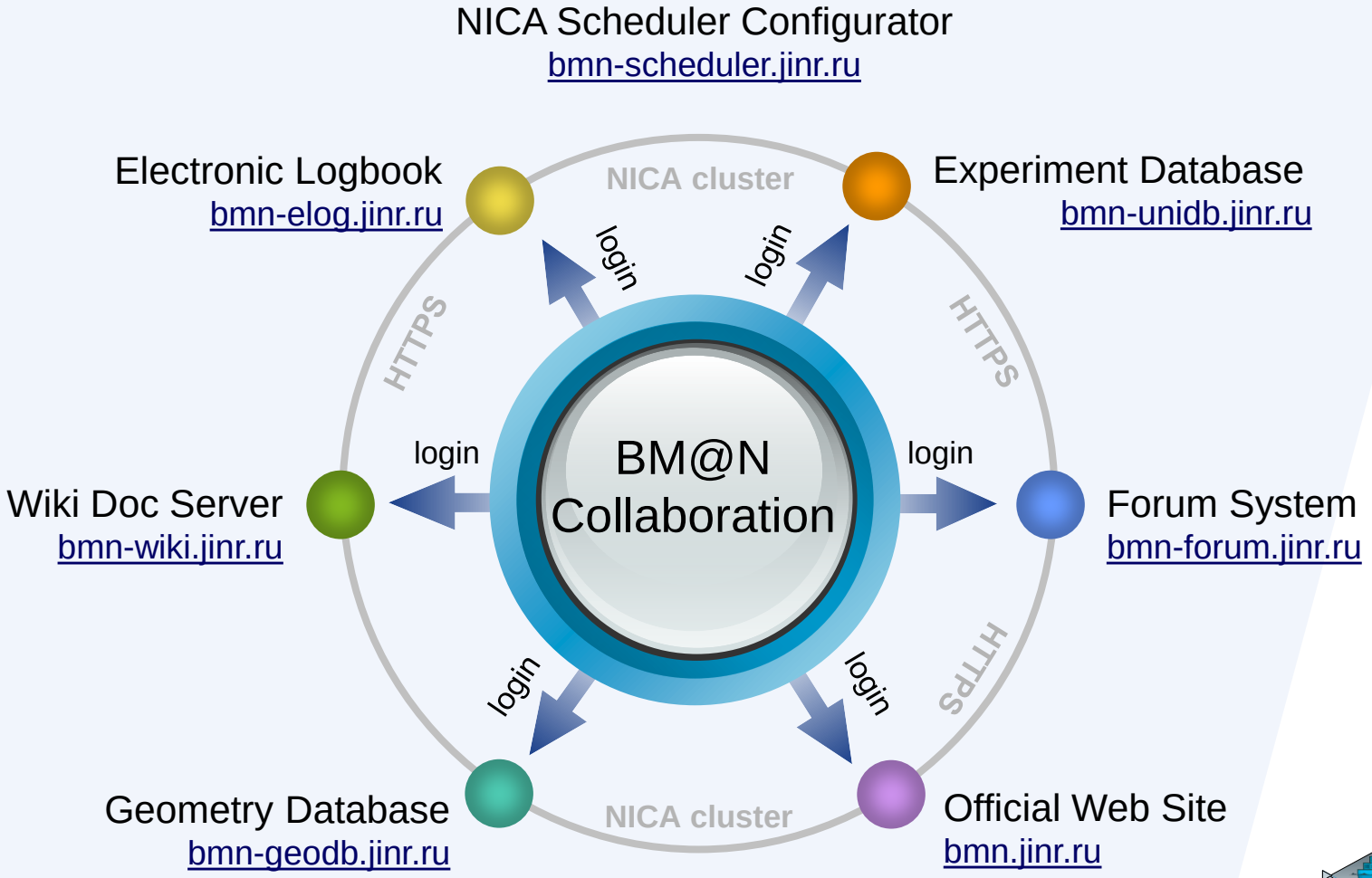

```
# python3 stats.py --time --dir
/eos/nica/bmn/users/logs/ --config
config-time.json
```





Collaboration Services

Evolution of the BM@N Information Services

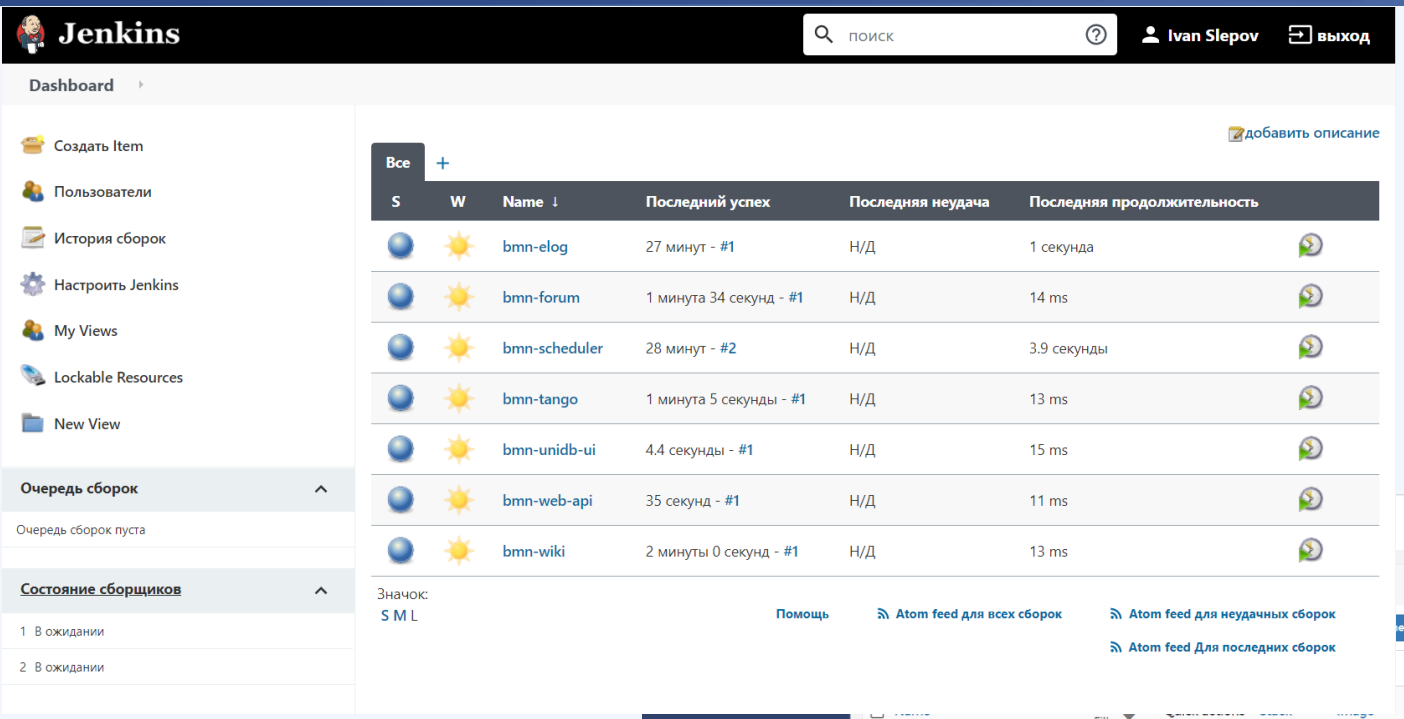


docker

NICA Cluster

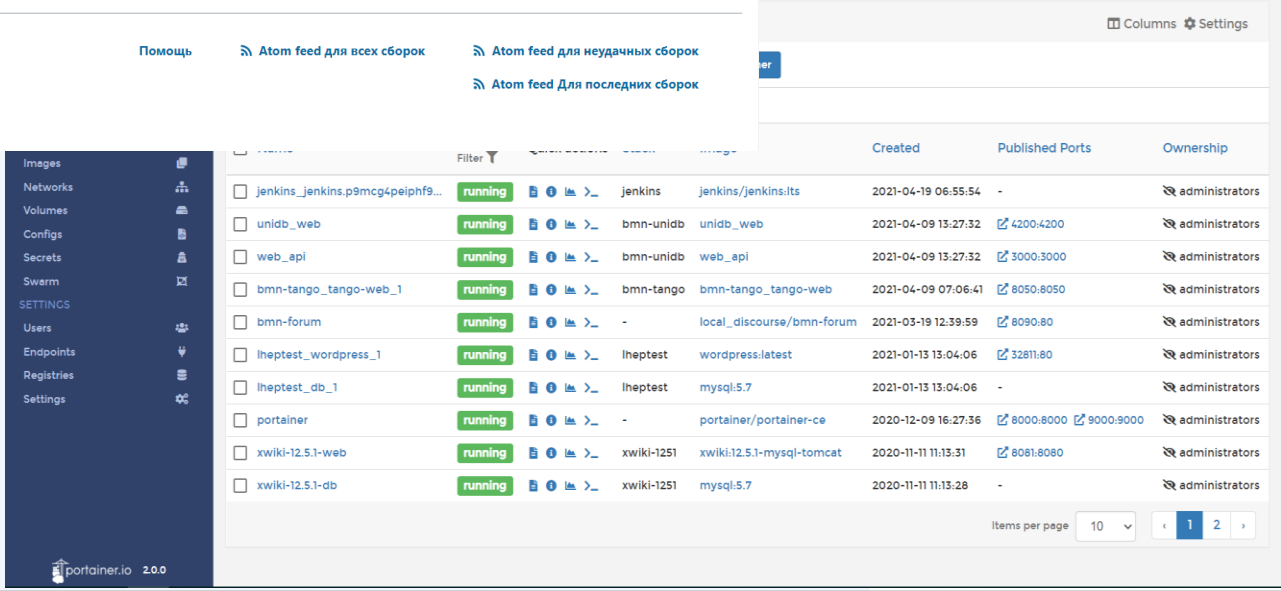
Ivan SLEPOV (19 April 12:45)
Deployment structure of the information services for BM@N on the NICA cluster

Deployment of the BM@N Information Services



Jenkins

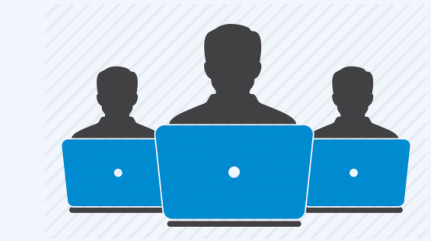
Portainer



3 servers on the NICA cluster:

- Web services/sites on NC9
- FreeIPA on NC3
- Databases on NC13

FreeIPA: Single Authentication & Authorization



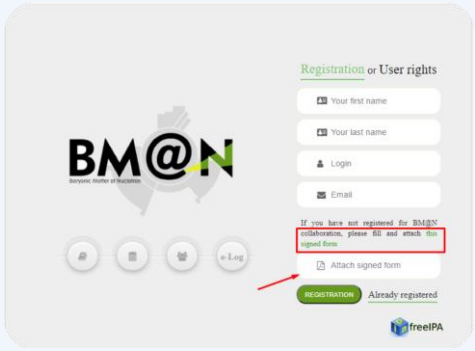
Collaboration members

Single Account

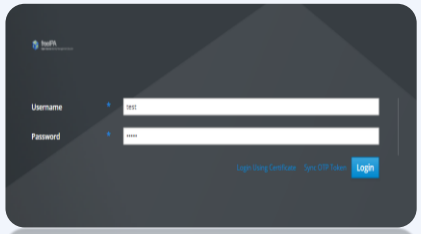
FreeIPA

register

request additional rights



bmn.jinr.ru/registration



bmn-ipa.jinr.ru

admins *writers* *users*

Electronic Logbook

Wiki Document Server

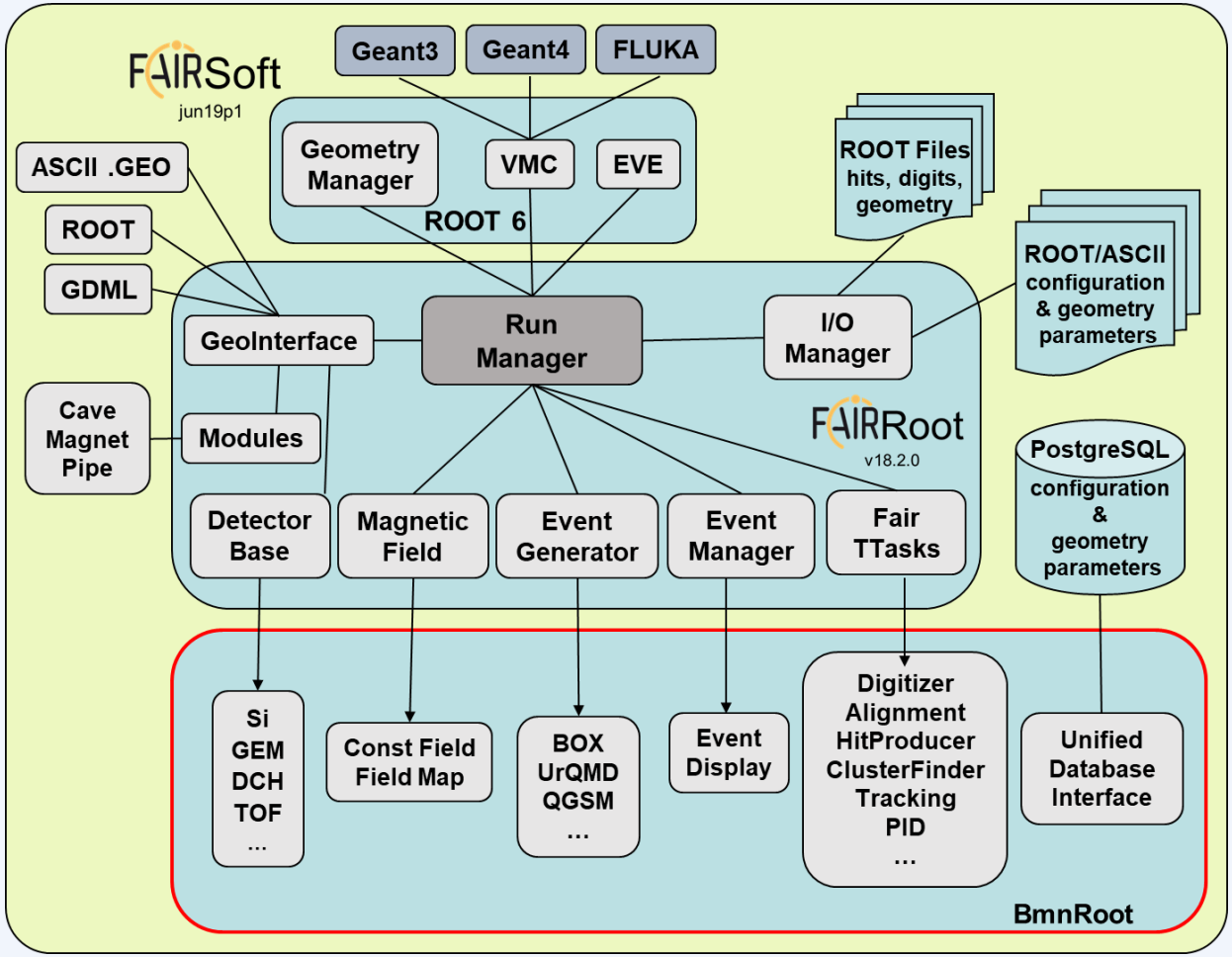
Experiment Database

BM@N Forum

BmnRoot Development

BmnRoot Release preparation: 21.05.0

FAIRSoft apr21 FAIRRoot v18.6.2 cannot be updated because of NCX-cluster (LIT MICC & HybriLIT are ready)



- Huge efforts have been expended to correct all problems before Release
- The pre-release has to be approved by detector groups
- The pre-release is tested at distributed clusters before Release issue
- Simulation and reconstruction results should be checked very carefully
- *decoder/BmnTof1Raw2Digit* should be corrected before the BmnRoot Release

The mass production of the BM@N digits and reco data for Run 7 is used to check pre-release

Elimination of memory leaks and errors

Class with memory leak	Valgrind description of memory leak	Number of function calls with memory leak
BmnInnTrackerAlign.cxx	10,626 (24 direct, 10,602 indirect) bytes in 1 blocks	2
	384 (88 direct, 296 indirect) bytes in 1 blocks	2
	10,818 (24 direct, 10,794 indirect) bytes in 1 blocks	2
	8,000 bytes in 1 blocks are possibly lost	2
UniDbRun.cxx	4 bytes in 1 blocks	2
BmnMwpcHitFinder.cxx	1,008 bytes in 21 blocks	2
	128 bytes in 4 blocks	1
	64 bytes in 2 blocks	1
	32 bytes in 1 blocks	1
BmnFillDstTask.cxx	312 (144 direct, 168 indirect) bytes in 1 blocks	1
	24 bytes in 1 blocks	1
	16 bytes in 1 blocks	1
	8 bytes in 1 blocks	1
BmnFieldMap.cxx	24 bytes in 1 blocks	1
	48 (24 direct, 24 indirect) bytes in 1 blocks	1
BmnMwpcHitFinder.cxx	2,016 bytes in 42 blocks	2

Sergei NEMNYUGIN
(19 April 11:40)

Software contribution from SPbU: algorithmic and code optimization of the BmnRoot framework

Dynamic analysis with Valgrind 3.15.0

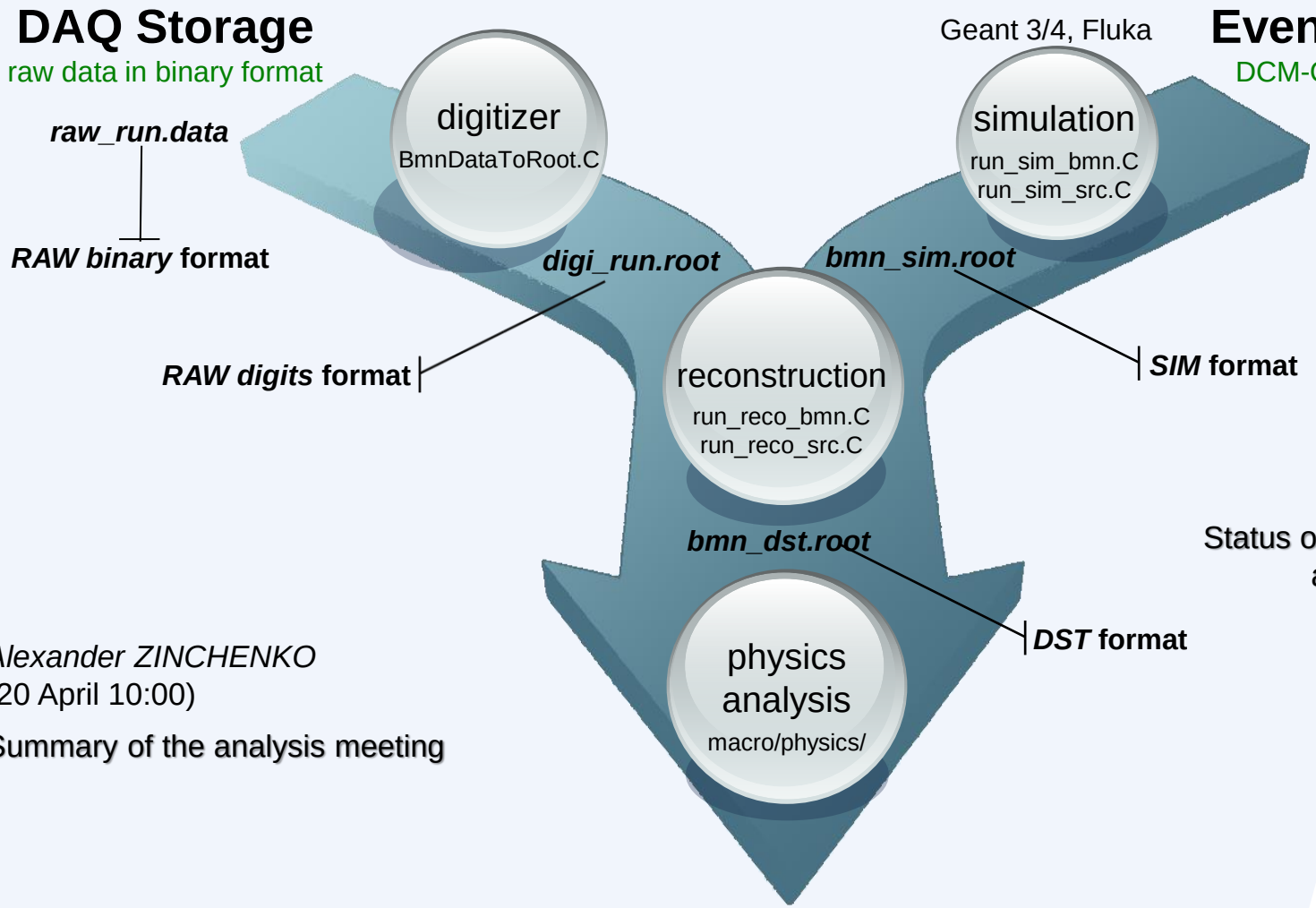
Configuration:

- OS: Ubuntu 20.04.2 LTS x86_64
- compiler GCC 9.3

Reconstruction modules:
run_reco_src.C & run_reco_bmn.C

- Memory leaks have been localized.
- Incorrect access to array elements have been localized
- Memory leaks are consequence of non-release of dynamically allocated memory both explicitly and implicitly.
- Work is in progress

BmnRoot Data Processing



Alexander ZINCHENKO
(20 April 10:00)

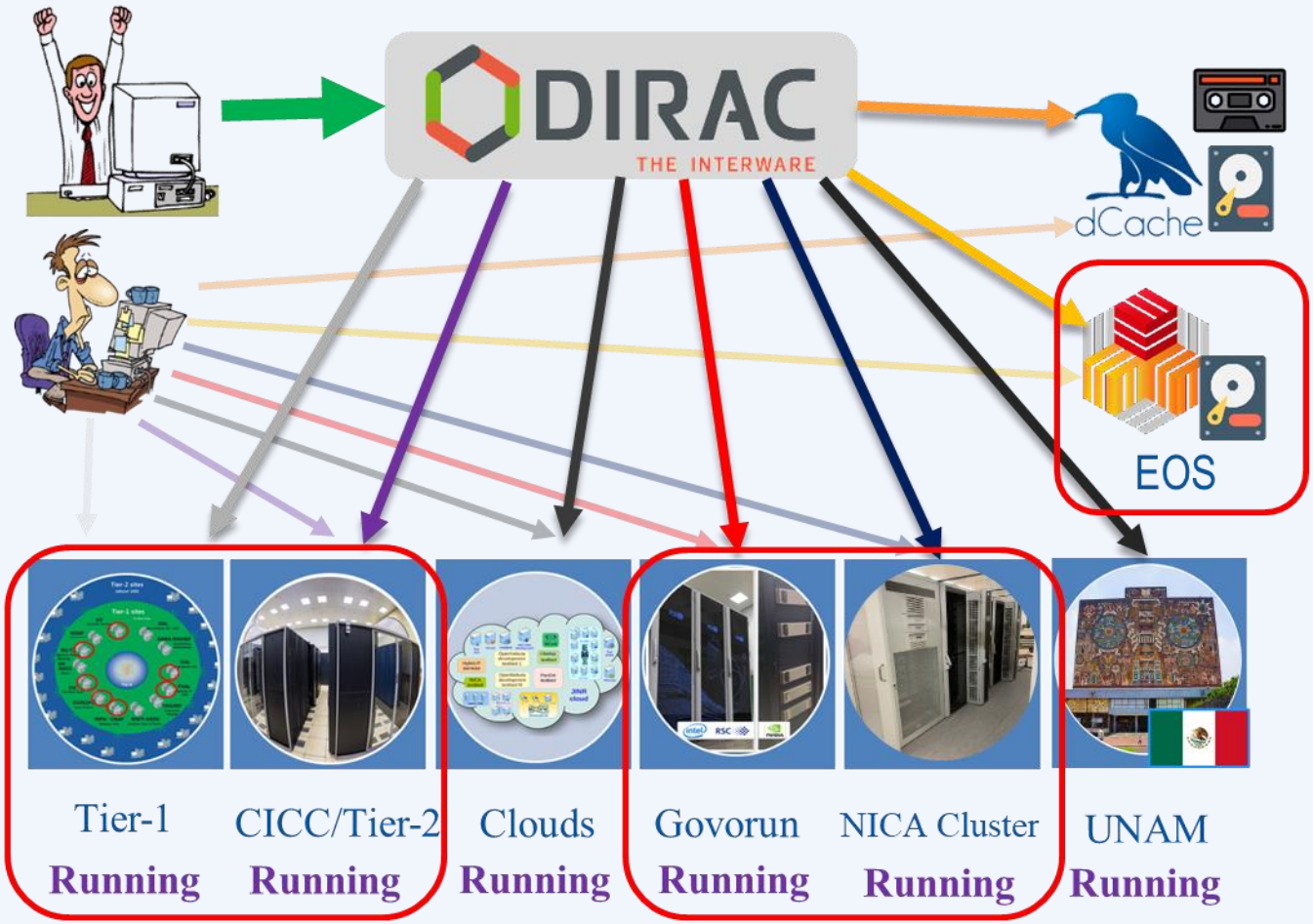
Summary of the analysis meeting

Sergei MERTS
(20 April 11:45)

Status of the BM@N simulation and data reconstruction

Distributed Computing

BM@N WorkFlow Services via DIRAC



- Workload management
- Data management
- Metadata management
- File Catalog
- Accounting
- User Interface
- API

Igor PELEVANYUK
(19 April 10:40)

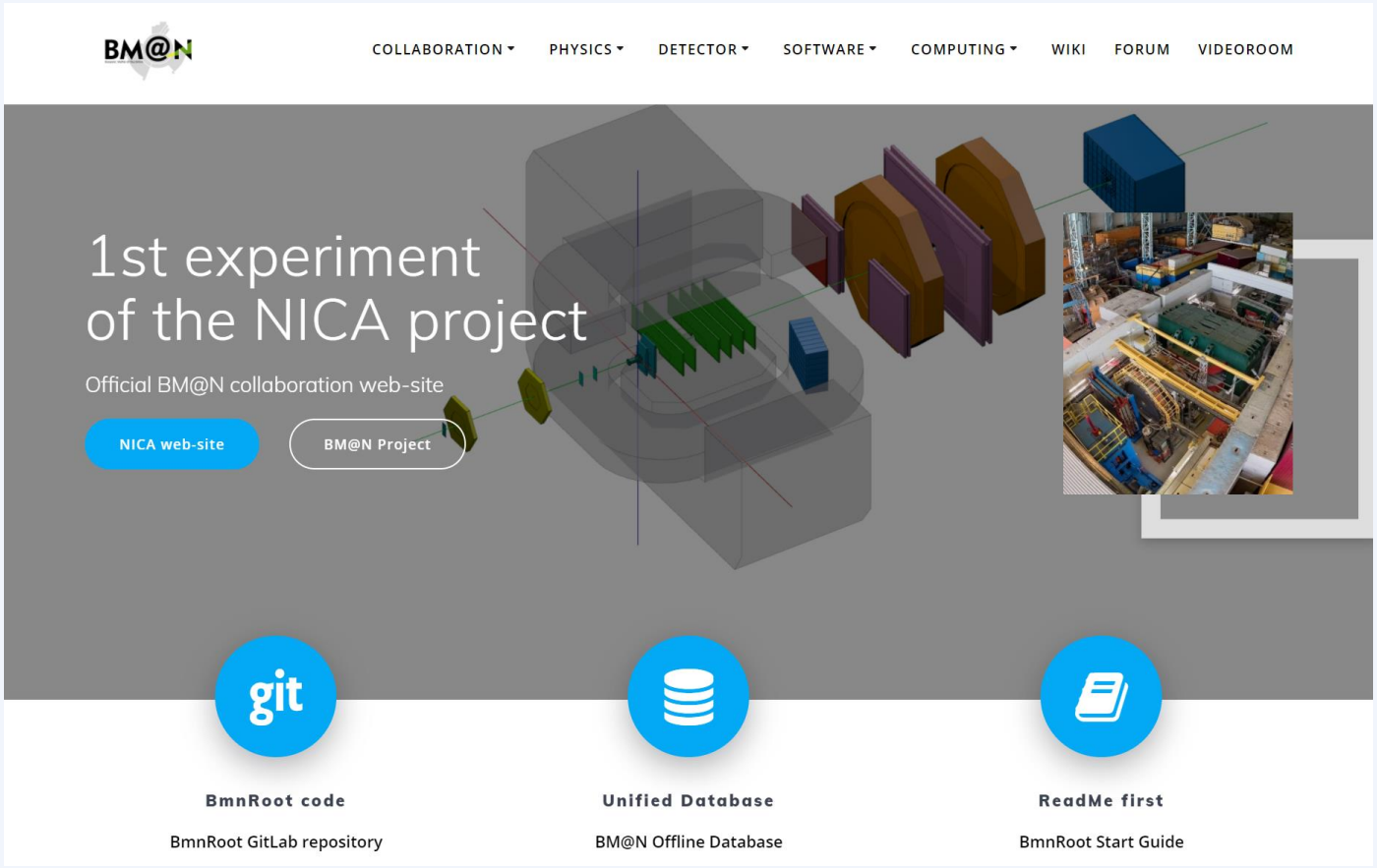
DIRAC workload management system and the very first results for the BM@N experiment

Testing BM@N event processing via DIRAC

	Govorun	NICA cluster	Tier1	Tier2
RawToDigi	Custom – OK CVMFS – OK	-	-	-
DigiToDst	Custom – OK CVMFS – OK	Custom – OK CVMFS – OK	Custom – OK CVMFS – OK	Custom – OK CVMFS – OK
GenToSim	Custom – OK CVMFS – OK	Custom – OK CVMFS – OK	Custom – OK CVMFS – OK	Custom – OK CVMFS – OK
SimToDst	Custom – OK CVMFS – OK	Custom – OK CVMFS – OK	Custom – OK CVMFS – OK	Custom – OK CVMFS – OK

```
dirac-dms-get-file /bmn/digi/run7/1002.root
root -l -q -b run_reco_bmn.C("1002.raw", "out.root",0,0)
dirac-dms-put-file /bmn/dst/run7/1002_dst.root out.root JINR-EOS-BMN
```

Official BM@N Web-site: *bmn.jinr.ru*



- ✓ ***Collaboration***
- ✓ ***Information***
- ✓ ***Documents***
- ✓ ***Software***
- ✓ ***Databases***
- ✓ ***Computing Section (NICA Cluster, MICC Complex, HybriLIT & Govorun)***
- ✓ ***Guides, Manuals***
- ✓ ***Wiki***
- ✓ ***Forum***
- ✓ ***Webex rooms***
- ✓ ***BM@N Mail-lists***
- ✓ ***etc.***

Conclusions

- SPbU Group joined to the Software Collaboration to make good contribution to the BM@N Software.
- The Information Systems (Geometry and Condition Database, Logbook) and related services have sufficiently been improved to simplify data processing by collaboration members. The Event Metadata System and Configuration Database is under development.
- RFBR support with the NICA grant (ending in March, 2021) enables to significantly improve the Information Systems for BM@N data processing.
- BmnRoot Release 21.05.0 is scheduled to be issued after approval procedure with the latest BM@N and SRC simulation, reconstruction, analysis and software improvements.
- The architecture of the BM@N mass data processing is under development. The work with the DIRAC interware has started.
- The lack of manpower for BM@N software development is a problem to be solved, but which has not been endorsed.

Thank you for your attention!

More information: **bmn.jinr.ru**
nica.jinr.ru

Email: *gertsen@jinr.ru*

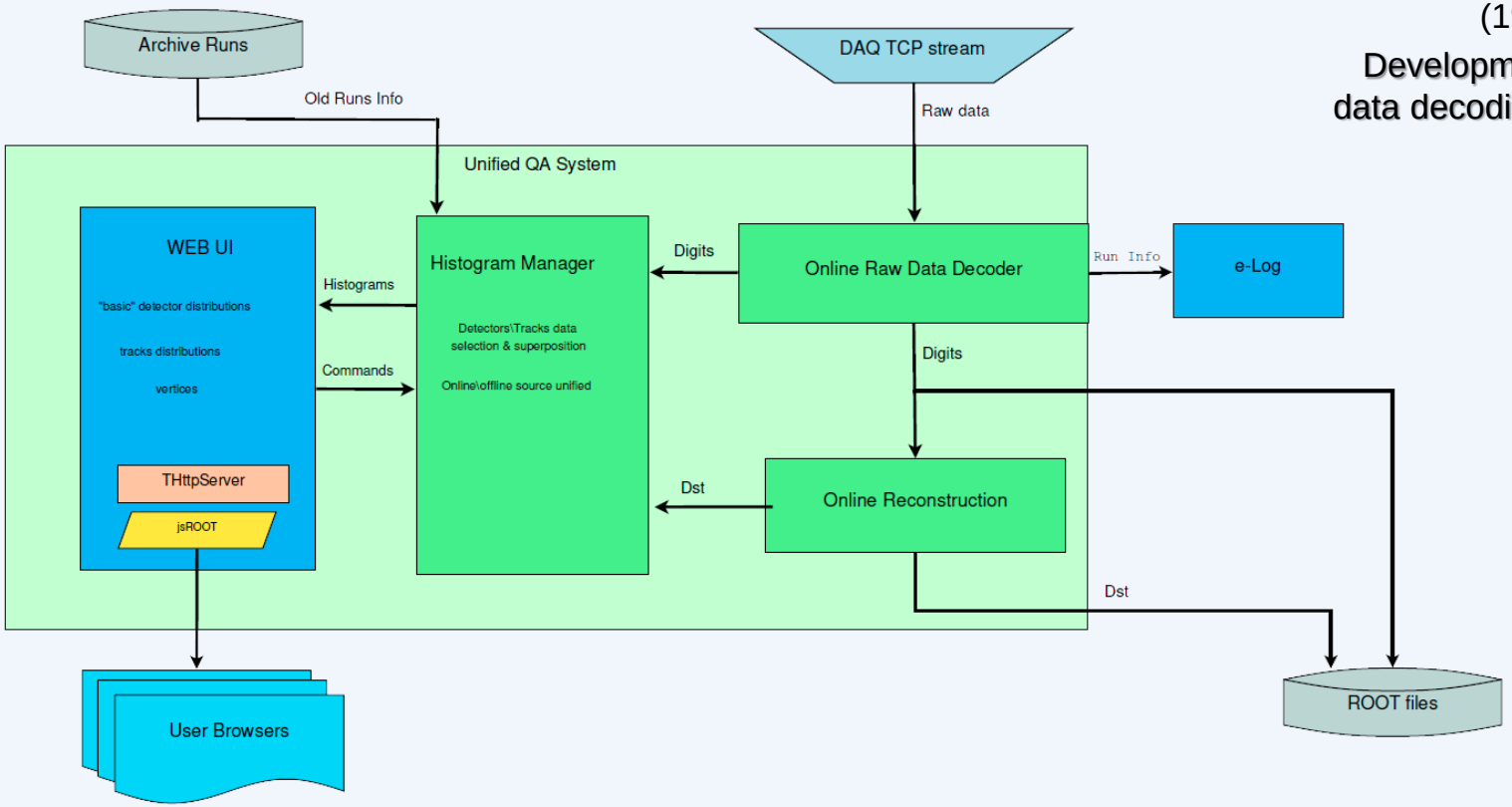
We are open for cooperation!

Backup

Improvement of Online Histogramming System

Ilnur Gabdrakhmanov
(19 April 12:30)

Development of tools for event data decoding and quality analysis




```
{
  "Name": "GEMS",
  "Title": "GEM Canvas",
  "DivX": "2",
  "DivY": "1",
  "Pads": [
    {
      "Class": "TH2I",
      "Name": "GEM-hits0",
      "Title": "GEM hits",
      "Options": "colz",
      "Dimensions": [
        200,
        0,
        400,
        400,
        -200,
        200
      ]
    },
    {
      "Class": "TH1F",
      "Name": "GEM0",
      "Title": "Gem Strip",
      "Dimensions": [
        200,
        0,
        400
      ]
    }
  ]
```

- Make addition of histograms simple and flexible (not require code rebuild)
- Move configuration of online histogramming outside of the code
- Detector groups add histograms as simple configurations in json files