BM@N Software for Run 9 data processing

Konstantin Gertsenberger
BM@N Software Coordinator

28 February 2026

BM@N in Nuclotron Runs (2015 – 2026)

- **Nuclotron Run 51** (d,C) — *Feb. 22 – Mar. 15, **2015***
- **Nuclotron Run 52** (d) — *June 29 – June 30, **2016***
- **Nuclotron Run 53** (d, d$^{\uparrow}$) — *Dec. 09 – Dec. 23, **2016***
- **Nuclotron Run 54** (C) — *Mar. 07 – Mar. 18, **2017***
- **Nuclotron Run 55** (Ar,Kr) — *Mar. 03 – Apr. 05, **2018***
- **Nuclotron Run 57** (Xe) — *Dec. 12 – Feb. 02, **2023***
- **NICA Run 1** (Xe) — *Feb. **2026** – April. **2026***

Technical
interaction rate: 5 k/spill
event count: ~300 Mevents

Physics
interaction rate: 8 k/spill
event count: ~650 Mevents

interaction rate: 20 k/spill
exp. event count: 3000 MEvents



- Beam: *Xe* (3.8, 3.0 → 2.2 → 1.6 → 3.0 AGeV), previous runs: *Kr* (2.3, 2.6, 3.0 AGeV), *Ar* (3.2 AGeV), C^{12} (3.5–4.5 AGeV), *d* (4, 4.6 AGeV)
- Target: *Sn* or *CsI*

previous runs: *Pb*, *Sn*, *Cu*, *Al*, C_2H_4, *C*

- Integrated DAQ, T_0 and Trigger systems
- Detectors: VSP, FSD, GEM, CSC, ToF-400&700, DCH1&2, FHCal, ECal, profilometers…
- Detect min bias beam-target interactions to reconstruct hyperons, strange particles, identify charged particles and nucleus fragments…

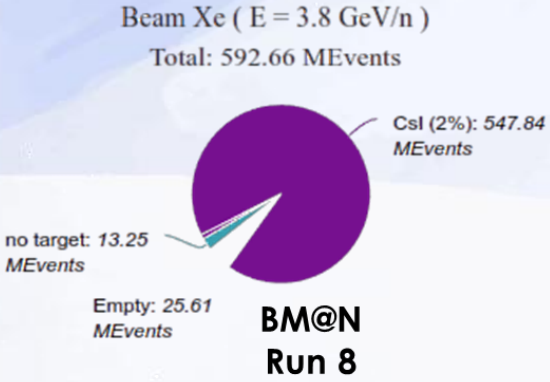
Data Production in BM@N Run 8

1st Physics BM@N Run (2022/2023y)

Two beam energy available for ***Xe***-beam

CsI target is used as similar to *Xe*

More than 600M events were collected



Parameter	Value (approx.)
Data acquisition time	**720 hours**
Average run duration	20 minutes
Average run time break	2.5 minutes
Beam intensity (3.8 AGeV)	up to 900k/2.2 Xe^+/sec up to 900k/12 Xe^+/sec
Trigger rate	**8 000/2.2 event/sec**
Average event size	**0,6 МБ**
Data rate	**up to 2 GB/sec**
Raw file size	**15 GB**
Event count per file (total)	25 000
Total event count (+test, calibration, pedestal)	**645 M**
Total (complete) file count	25 800
Total run count	1 920
Total raw data size	**400 TB**
Total replicated raw data	1.6 PB
Avg digit file size	**1.2 GB**
Avg DST file size	**2.2 GB**

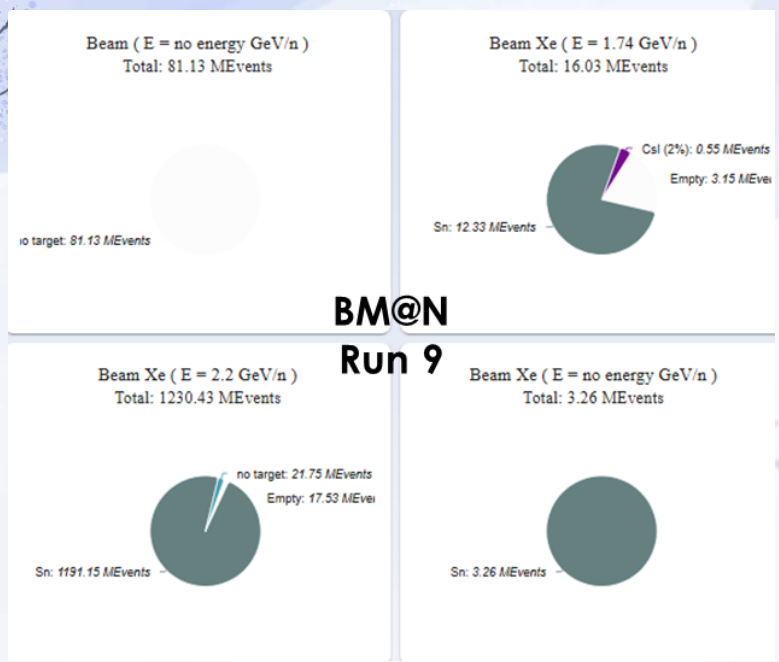
Current (27.02.2026) Production in BM@N Run 9

BM@N Run 9 (2026)

Three beam energy available for ***Xe***-beam

Sn and ***CsI*** targets are used as similar to *Xe*

Up to 3 000 Mevents are expected



Parameter	Value (approx.)
Data acquisition time	**311 hours**
Average run duration	26 minutes
Average run time break	6 minutes – 43 minutes
Beam intensity (2.2 AGeV)	up to 1.6M/6 Xe^{+}/sec D = 6 sec/12 sec
Trigger rate	**20K/6 event/sec**
Average trigger rate (per file)	**1 650 event/sec**
Current event size	**550 КБ**
Data rate	**up to 2 GB/sec**
Current raw file size (run:9537)	**7.8 GB**
Event count per file (total)	**15 000**
Current events (+test, calibration, pedestal)	**1 328 M**
Current file count	88 631
Current run count (incl. bad)	762
Current raw data size	**673 TB**
Avg digit file size	**0.8 GB**
Avg DST file size	**1.3 GB**

Statistics Collection & Success Check Service

- Shows histograms and summary data for:
  - Run – file – event statistics
    & 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/services/statistics

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

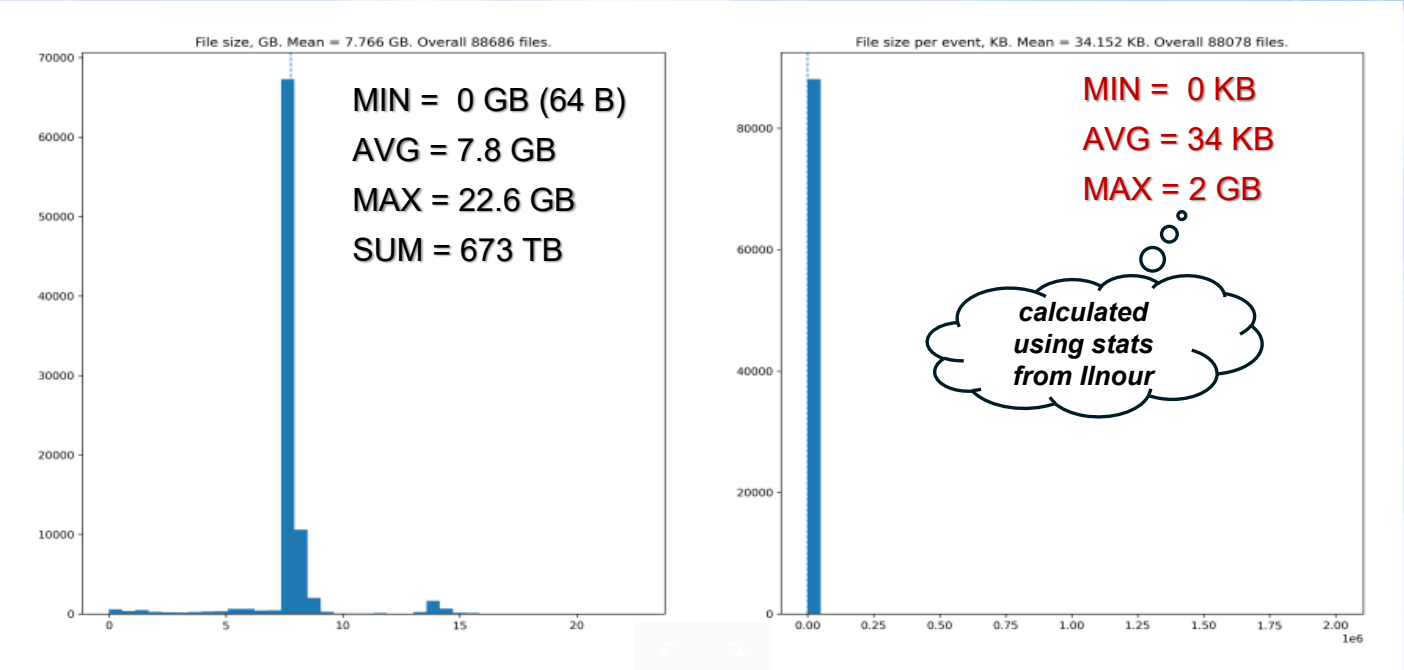
```
# python3 stats.py --time --dir
/eos/nica/bmn/users/logs/ --config
config-time.json
```

*python bmnroot/services/statistics/**stats.py***
*root bmnroot/database/examples/uni_db/**postrun_update.C***
*root bmnroot/database/examples/uni_db/**postrun_check.C***
*root bmnroot/database/tango/examples/**show_avg_field.C***
*root bmnroot/database/uni_db/examples/run/**show_run_stats.C***
*root bmnroot/database/uni_db/examples/raw/**show_raw_stats.C***
*root bmnroot/database/uni_db/examples/root/**show_root_stats.C***
*python bmnroot/services/inspection/raw_inspection **/...***



Run 9 Statistics on Raw Files (27.02.2026)

- "/ceph/bmn/run/run9/beam/": **88 631** files of **673 TB** size → **692 runs** (UniConDa – **702**)



- should corrupted raw files be transferred to offline storages and mass processed?

bad runs are moved to "beam-bad" directory, but "what is bad run? 5 GB or more?", "when move?", "how sync deletion with Airflow/Databases/EOS storages?"

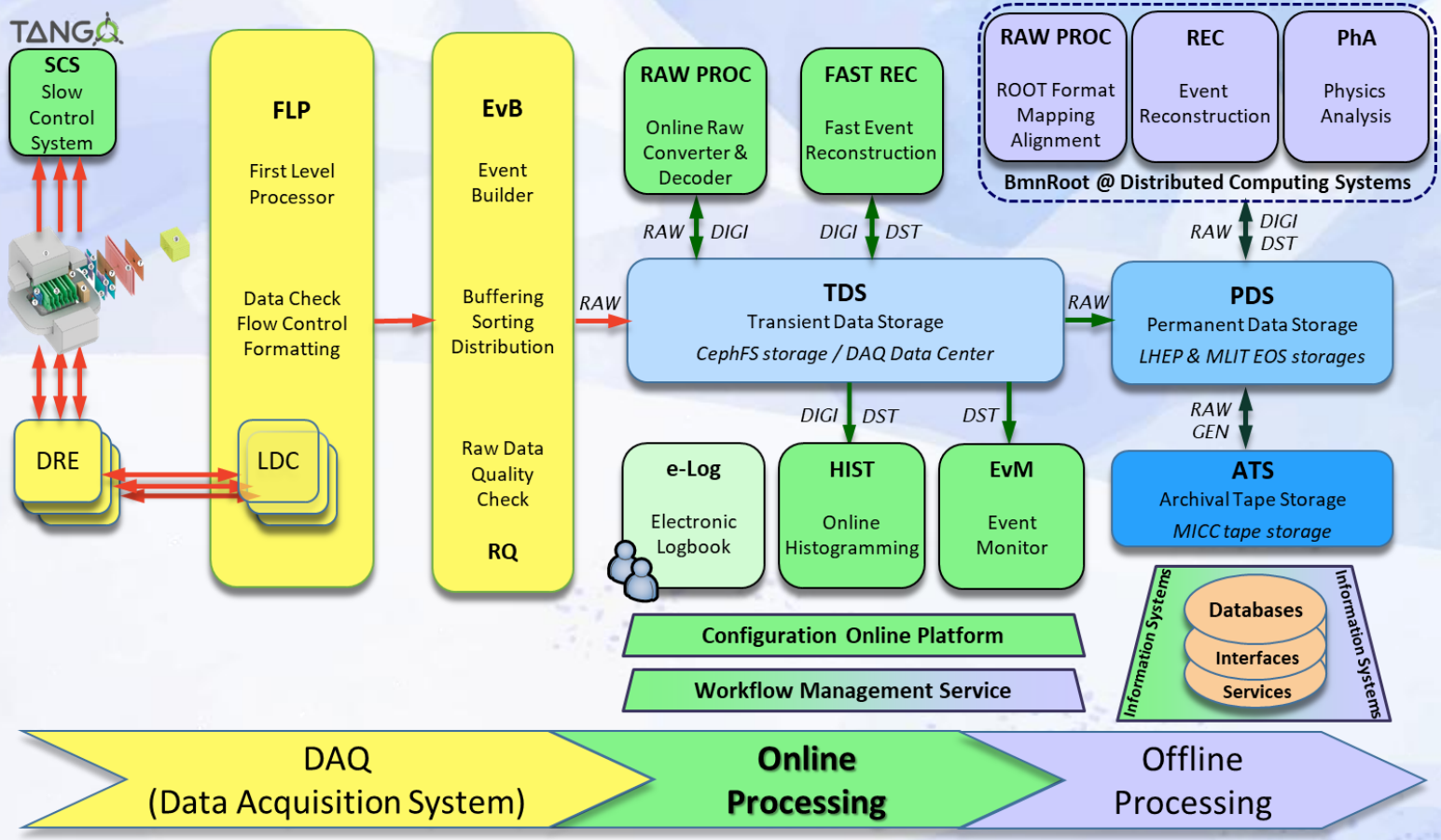
Run 9 Statistics on BM@N runs and events




```
DEBUG nica-scheduler$ Batch command: sbatch -J raw9info  -a 1-5320%60 /ceph/bmn/work/gertsen/bmnroot/b
uild/bin/job_1689179895652662403.sh
sbatch: error: Slurm temporarily unable to accept job, sleeping and retrying
sbatch: error: Batch job submission failed: Resource temporarily unavailable
```

- DDC SLURM parameters have been increased to run raw file metadata collector

BM@N Data Processing Flow

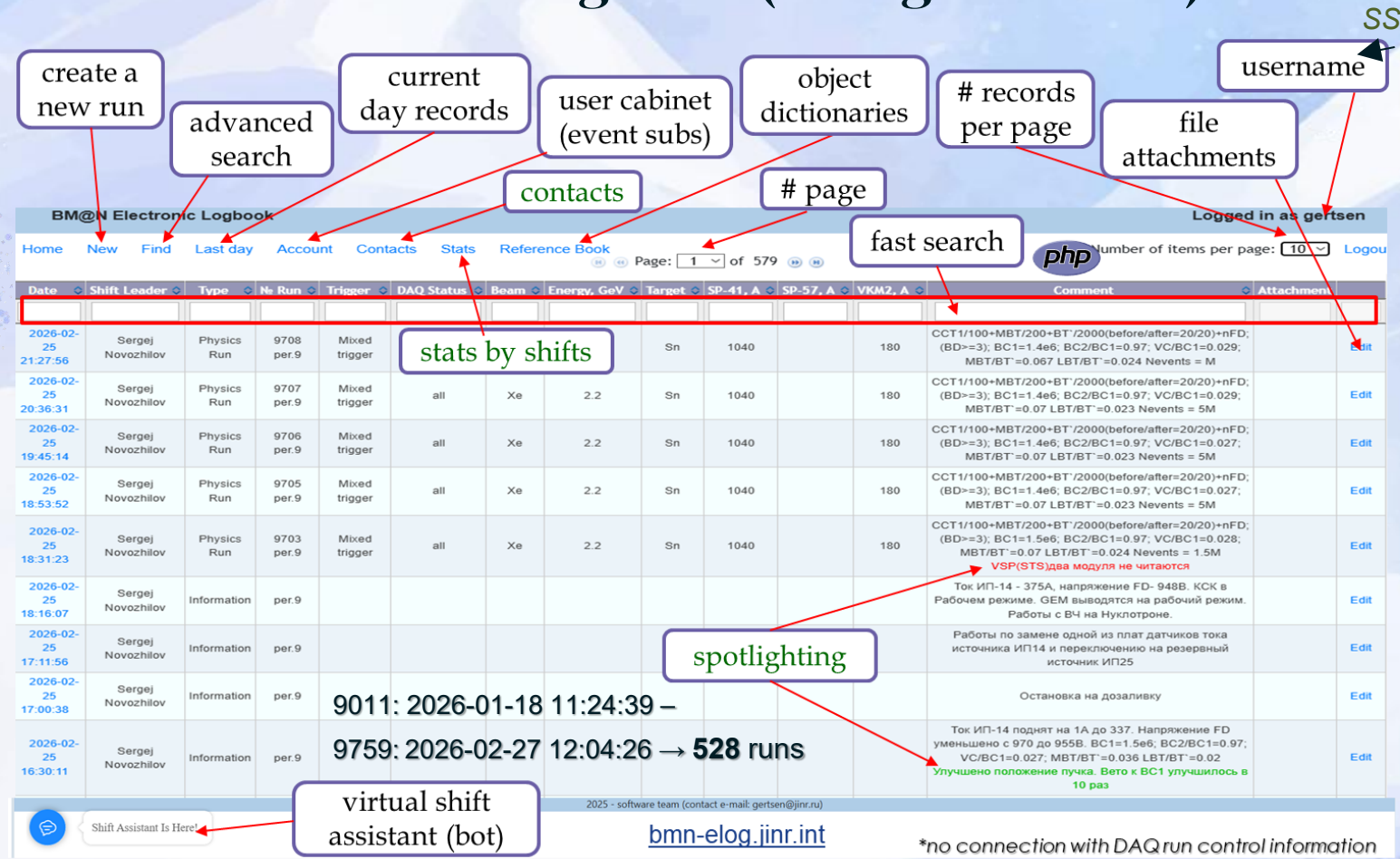


Electronic Logbook Platform

Online Information System



Electronic Logbook (e-Log Platform)



bmn.jinr.int: Software → Run Logbook

e-Log Platform Bugs and Features



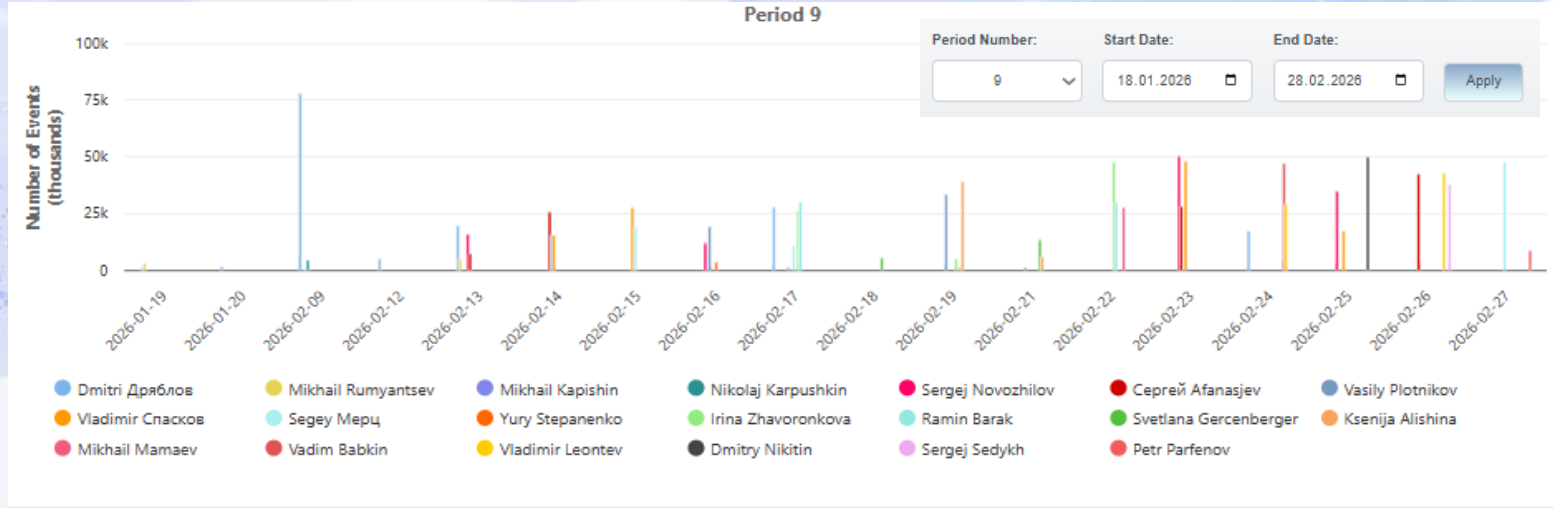
All systems are ok (for statistics)?

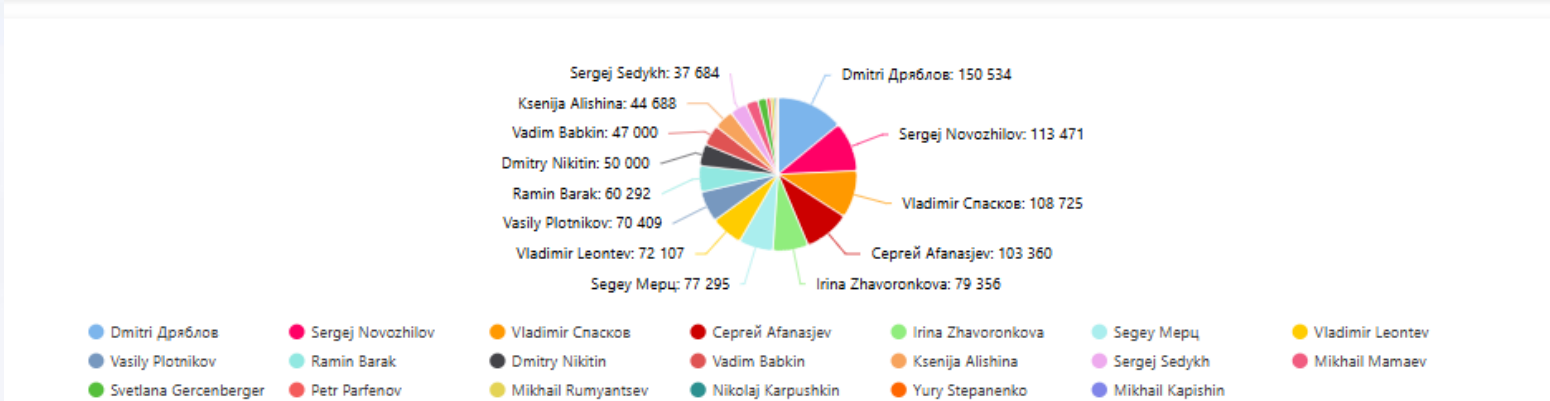


- SSO accounts were used, but *person_name* limit was cancelled because of LDAP issues
- move dictionary items to archive implemented for Triggers, Beams and Targets tables
- ‘previous geometry’ trigger re-enabled

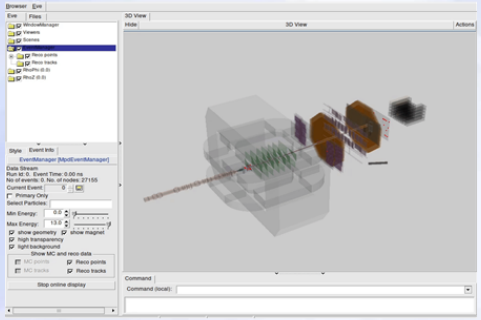
- transfer of magnetic filed to UniConDa corrected
- transfer of run type to UniConDa updated
- run number is optional now
- beam is optional now
- [bold] and [red] options have been added to comments
- changing record type does not clear fields except grayed

e-Log Statistics by Shift Leaders (27.02.2026)





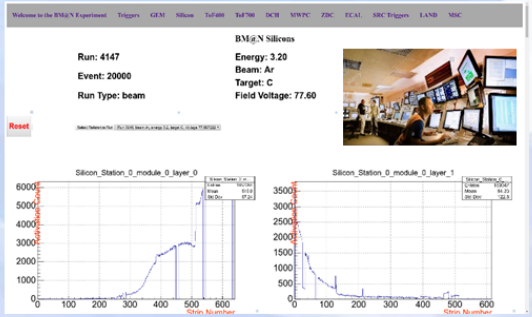
Run and Raw File Metadata for BM@N Data Processing



DAQ system

EvB

Raw Event Builder



Online Raw

ROOT Format Digit Decoder

EvM

Event Monitor

HIST

Online Histogramming

Offline RAW

ROOT Format Digit Decoder

run metadata

period, run

approximate (!) start & end time, event count

raw file metadata (manual!)

period, run

start & end time, start & end event, event count, file path, size and hash



e-Log Platform

run metadata

period, run

run type, beam, energy, target, magnetic field



Condition Database

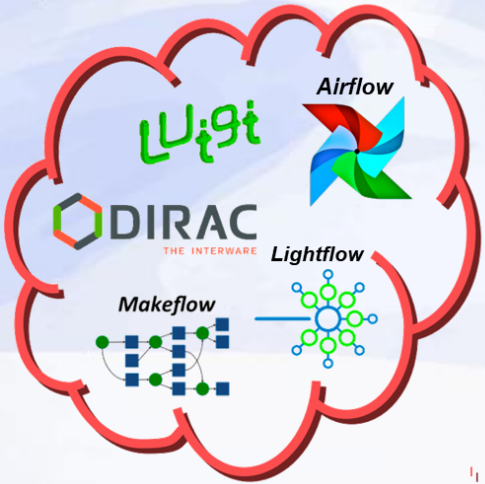
online & offline data processing

Event & Run Statistics

Workflow Management Service



Workflow Management Services



Ability to define dependencies between different tasks and scheduling them is a key

Gaining popularity as businesses increasingly have many complex Extract, Transform, Load (ETL) operations to schedule e.g. search, indexing/ranking, monitoring, statistics creation.

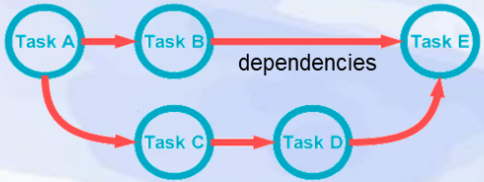
*D. Dossett, M. Sevior. "Automating Calibration at the Belle II Detector

Apache Airflow

https://airflow.apache.org

Platform to author, schedule and monitor workflows (as code)

Python scripting to define Directed Acyclic Graphs (DAGs) as collections of tasks (Operators) with dependencies between them



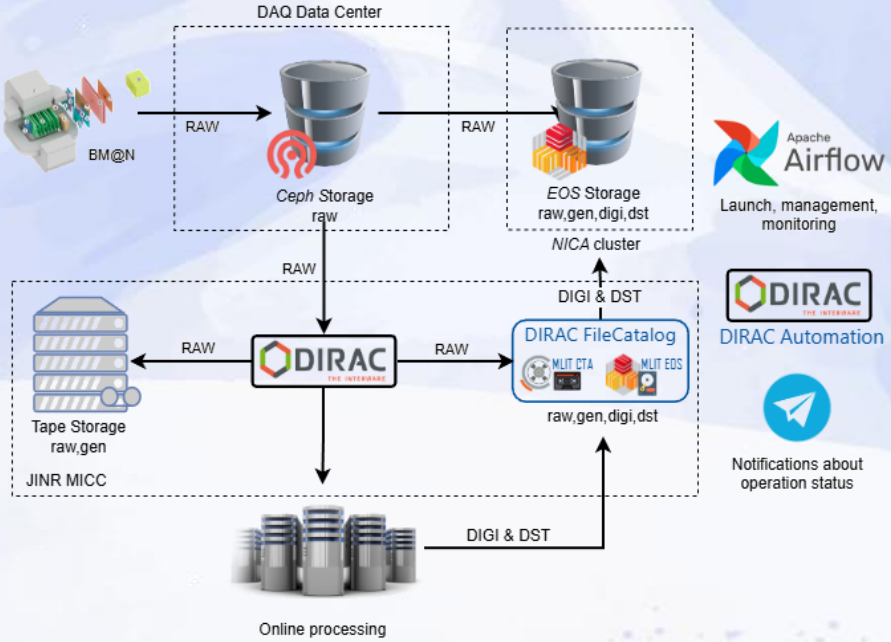
DAGs can be given a regular schedule, triggered manually, or even trigger each other

Includes Flask-based web monitoring to visualize pipelines, monitor progress and troubleshoot issues

Uses Jinja2 templating and SQLAlchemy to write tasks that render templated strings e.g. in database query strings or bash scripts

Operators	Sensors	Hooks
Defines the task in a DAG	Operator that periodically executes a query	Defines the interface to some external system
Different operator types, e.g. Python, Bash, SQL	Won't complete until conditions are met	Retrieves authentication stored in Airflow DB

BM@N Run 9 Orchestration with Workflow Manager



copying of the raw data to the storage systems of the NICA and MICC clusters

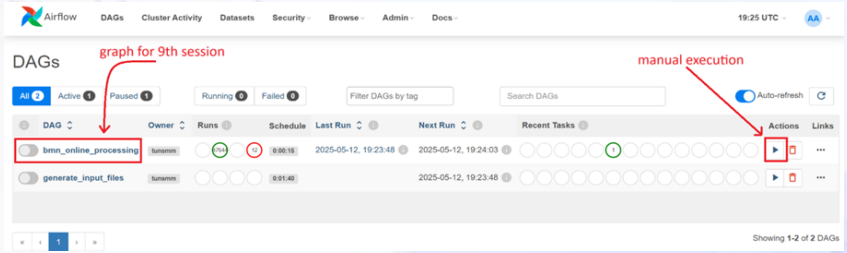
"online" processing of collected data on the provided distributed computing resources

backup to the NICA Tape Storage



Apache Airflow

Online and Offline TaskFlows for mass data processing



BM@N Online Processing DAG in Apache Airflow

DAG bmn_online_processing / Run 2026-02-17, 11:04:29 MSK

Details | Graph | Gantt | Code

Parsed at: 2026-02-21, 22:51:18 MSK

```python
with DAG(
    dag_id="bmn_online_processing",
    default_args=DEFAULT_DAG_ARGS,
    description="DAG to automize online processing",
    schedule_interval=timedelta(seconds=MONITOR_INTERVAL),
    catchup=False,
):
    task_copy = PythonOperator(
        task_id="task_copy",
        python_callable=copy_new_files,
    )

    # task_process = PythonOperator(
    #     task_id="task_process",
    #     python_callable=process_files_via_slurm,
    # )

    # task_archive = PythonOperator(
    #     task_id="task_archive",
    #     python_callable=archive_files_tape,
    # )

    # task_copy >> [task_process, task_archive]
```

automation of moving bad raw runs are proposed
via Apache Airflow DAGs
but synchronization issue is open

Key	Val	Description
INPUT_RAW_DIRECTORY	gertsen@ddc02.jinr.ru:/ceph/bm…	path to the directory with input raw data to be monitored (string)
MIN_RAW_FILE_SIZE	65	minimum size of the raw file in bytes
MONITOR_INTERVAL	301	time interval in seconds, in which the next task for monitoring raw data directo…
OUTPUT_RAW_CICC	gertsen@eos.jinr.ru:/eos/nica/b…	path to the directory on the CICC cluster, where new raw data are copied
OUTPUT_RAW_DIRAC	Invalid	DIRAC LFN
OUTPUT_RAW_NICA	gertsen@ncm.jinr.ru:/eos/nica/b…	path to the directory on the NICA cluster, where new raw data are copied
RETRIES_FOR_FAILED_DAG	1	how many retries are made for failed DAGs
RETRY_DELAY_FOR_FAILED_DAG	30	time interval in seconds, in which the next retry starts in the case of failed DA…
TELEGRAM_CHANNEL_ID	no_id	telegram channel with notifications (string)
TELEGRAM_TOKEN	********	telegram token to send notifications to the official Airflow channel

raw data transferring was stopped after restructuring storing scheme of the raw data

11 767 raw files have been copied to
NCX: **/eos/nica/bmn/exp/raw/run9/beam/**
with matching *Adler32* checksum and
write information to *flow_db* database

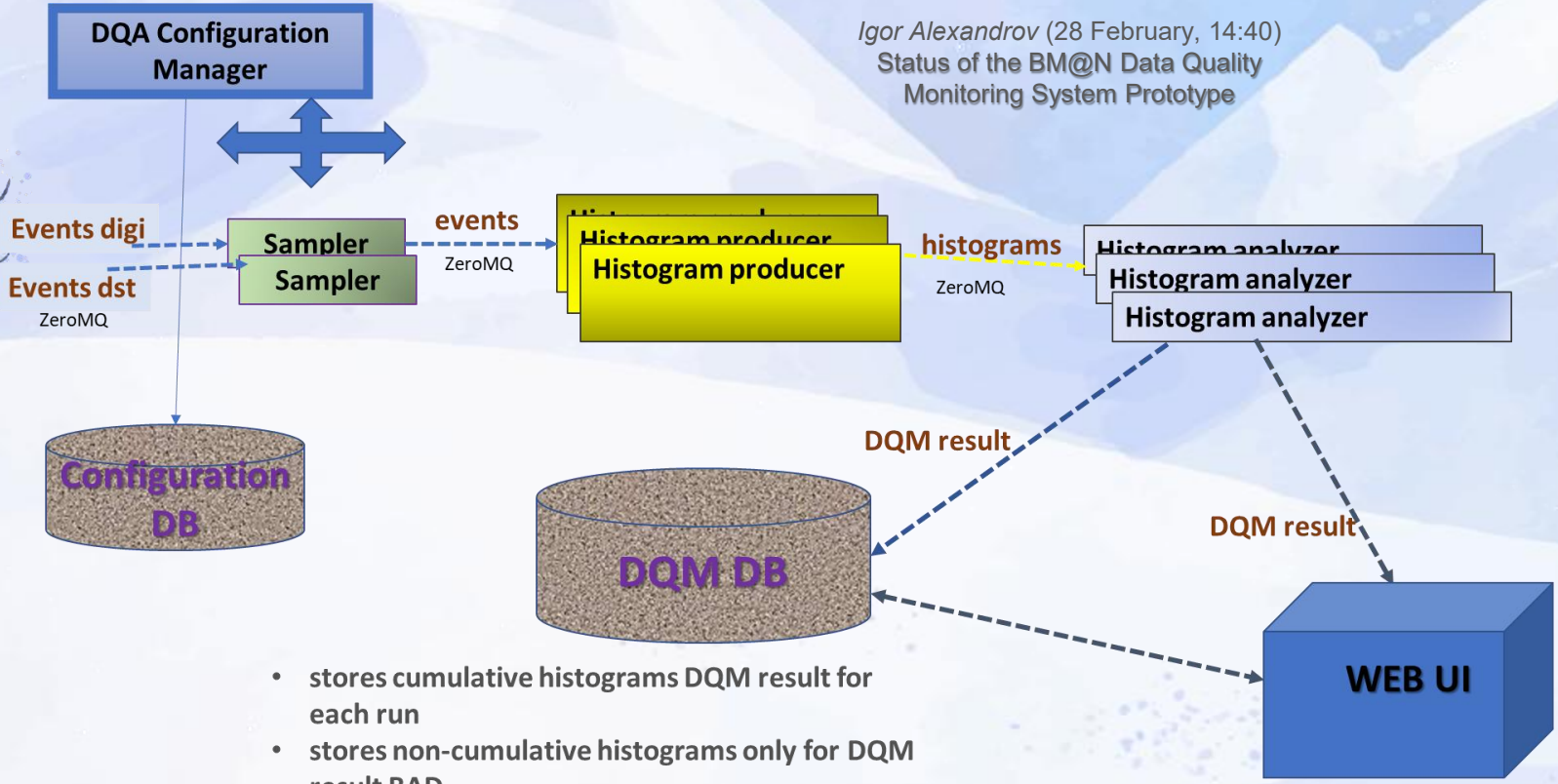
Data Quality Assurance

Online/Offline Software System



Design of the BM@N DQA system

Igor Alexandrov (28 February, 14:40)
Status of the BM@N Data Quality Monitoring System Prototype



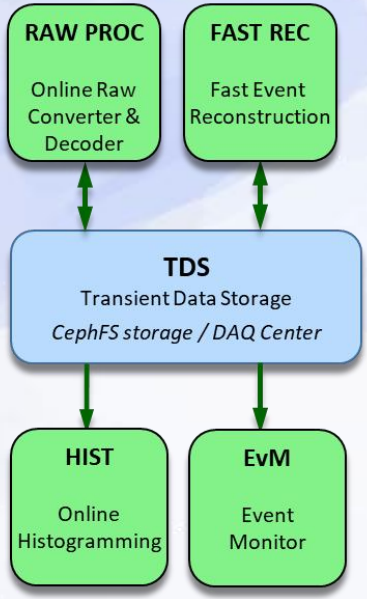
- stores cumulative histograms DQM result for each run
- stores non-cumulative histograms only for DQM result BAD

Configuration Online Platform

Online Data Processing



Online Process Control



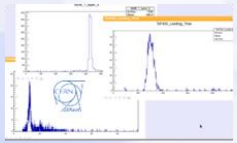
Online Processing

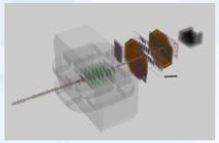




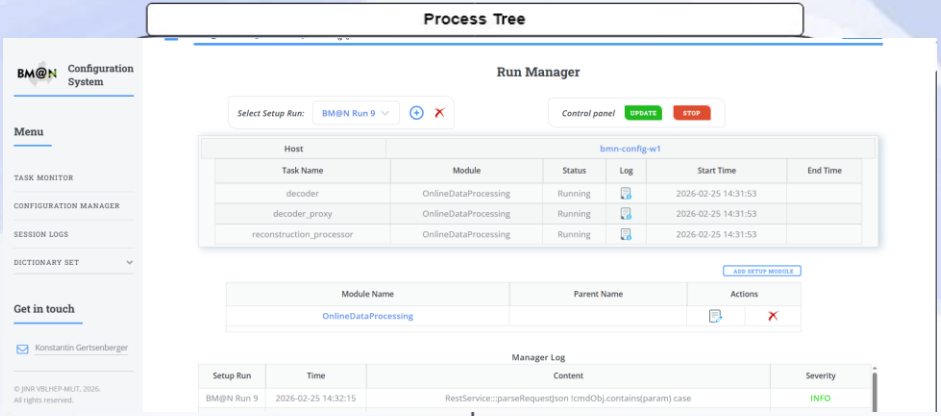
BM@N COnfiguration Online Platform (COOP)

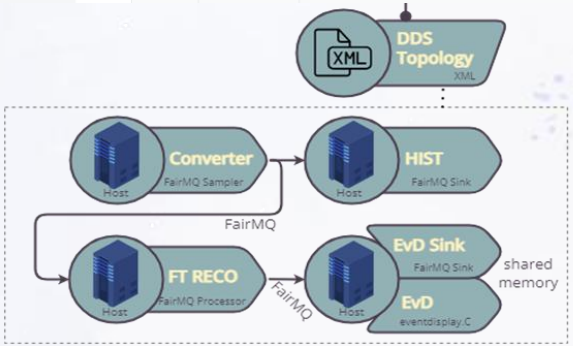
Online Histogramming System





Event Monitor







start/stop
control
keep alive

Dynamic Deployment System (DDS)

process topology

DDS API



Configuration Manager



process management (REST API)



online task topology

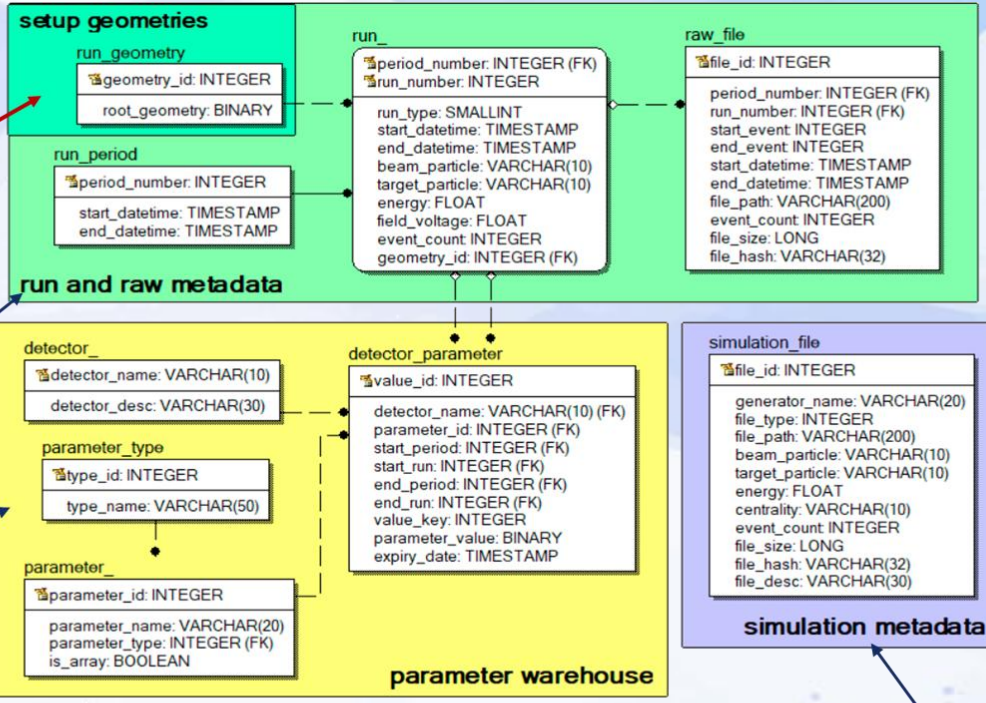
FAIR

Condition Database (UniConDa)

Offline Information System



Unified Condition Database: Scheme



pointer to Geometry Storage *(will be removed after Geometry Database integration)*

pointer to EXP Storage

pointer to Parameter Storage *own serialization in C++*

storing information (metadata) on experiment runs and obtained raw files; on generated simulation files; detectors, parameters and parameter values, and (temporary) setup geometries

pointer to SIM Storage

PostgreSQL 14

UniConDa in BM@N data processing



ROOT-based framework

detector simulation
raw data processing
event reconstruction
physics analysis

C++ interface
(connect, I/O, API)

http

SQL

REST API service

SQL

node.JS



Condition Database

viewing and managing parameter data



users

Keycloak

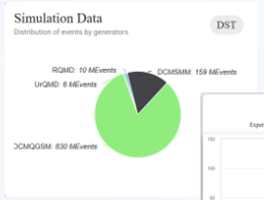
authentication
authorization

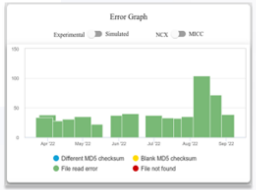


Web Service

service for auto updating of simulation metadata

SimData Crawler





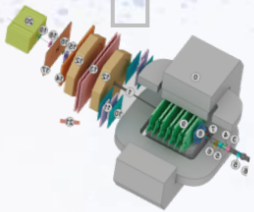
File Inspector

service for automatic regular checking of file integrity

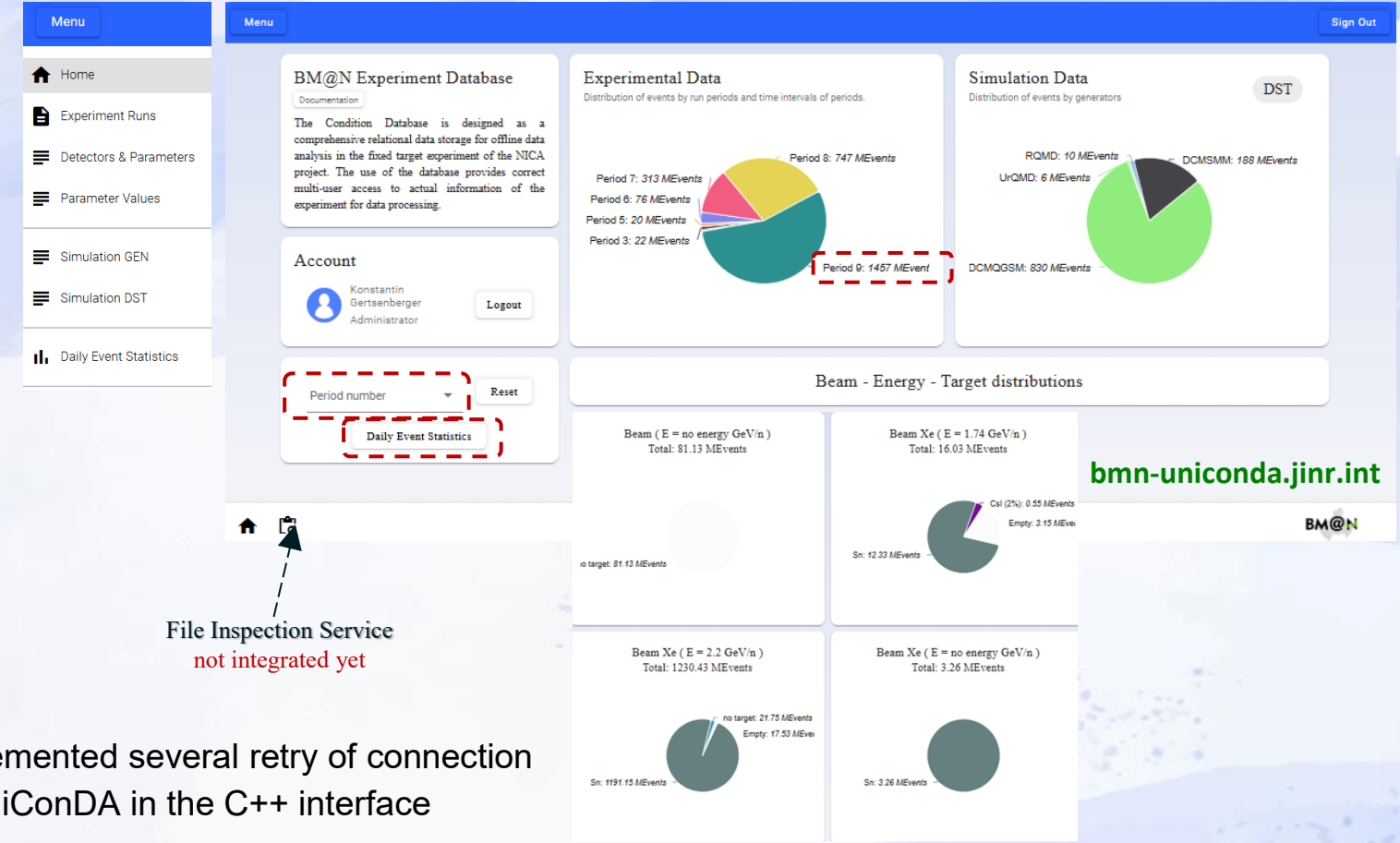
configuration
calibration

data

parameter
algorithm



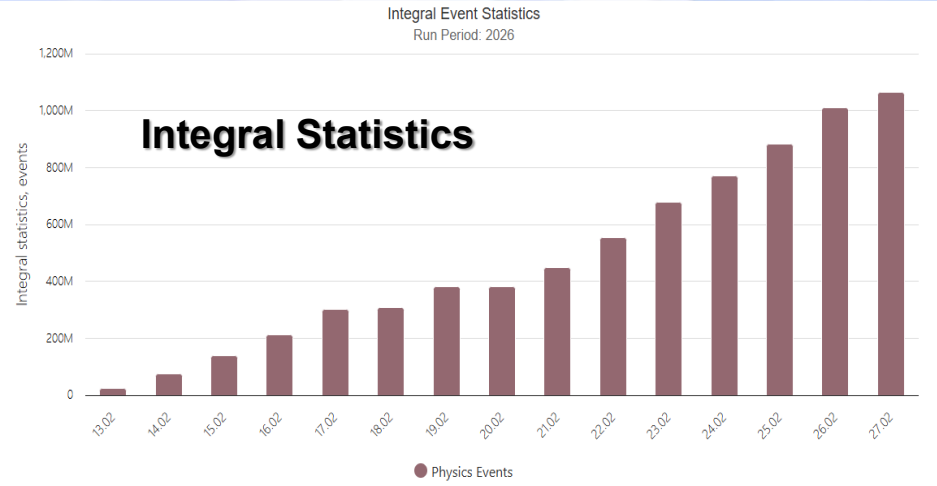
UniConDa. Start Page



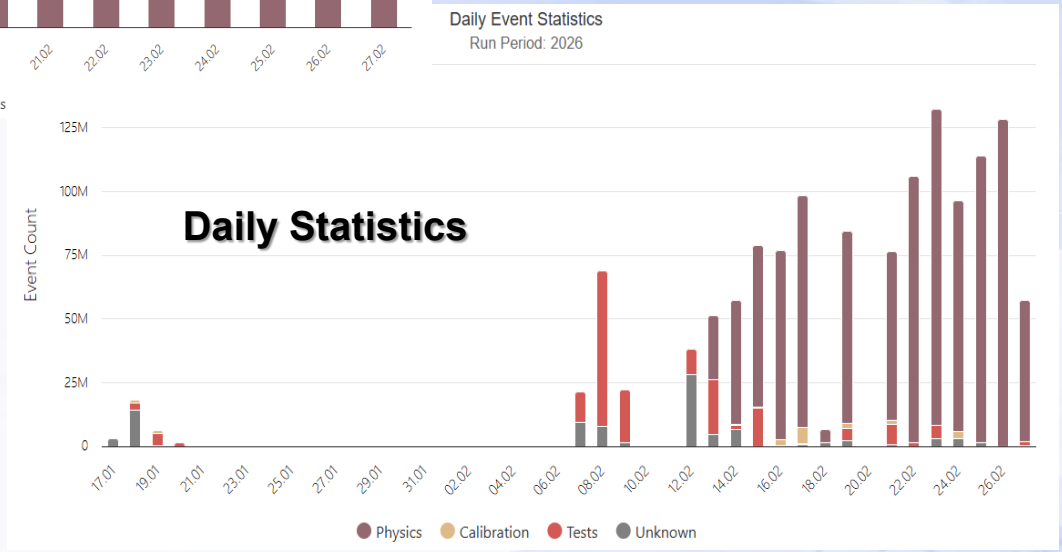
File Inspection Service
not integrated yet

- Implemented several retry of connection to UniConDA in the C++ interface

Daily Statistics in UniConDa (27.02.2026)

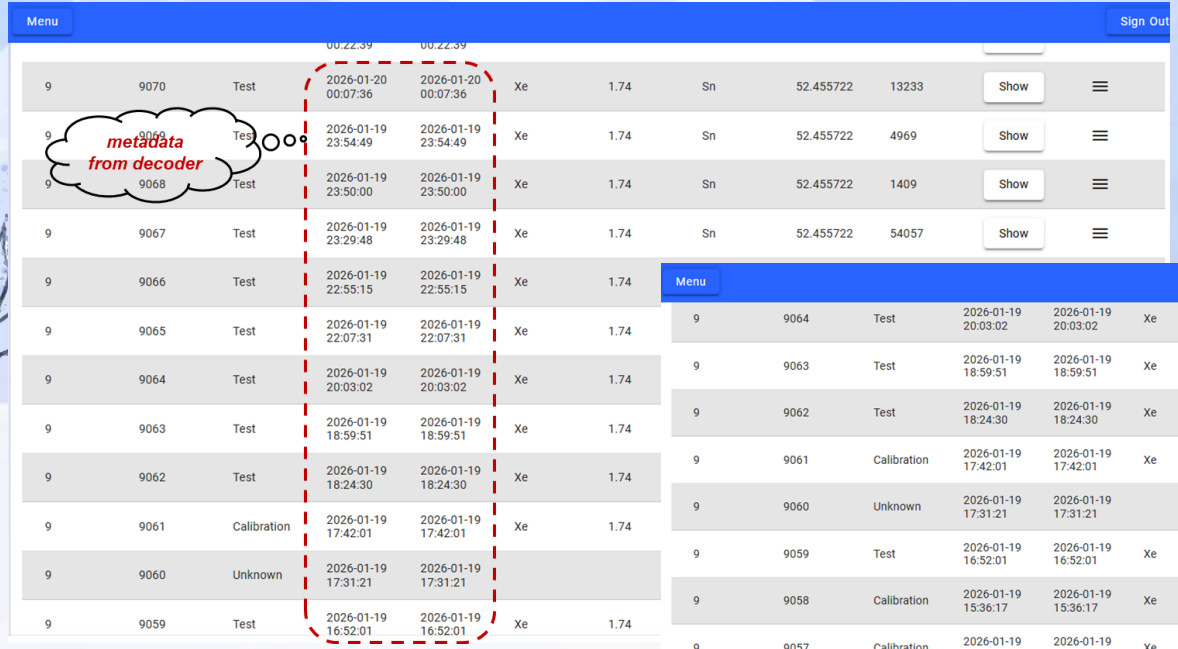






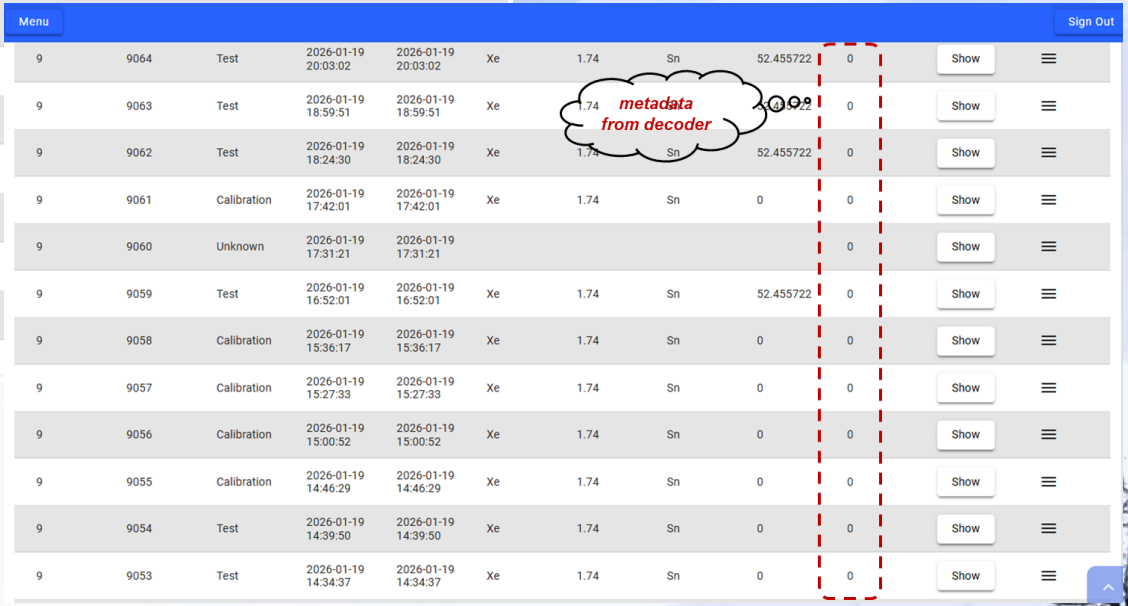
Run statistics from the Condition Database is automatic and accurate but depends on the Online Raw Data Decoder

Incorrect Run Metadata in UniConDa (27.02.2026)



9	9070	Test	2026-01-20 00:07:36	2026-01-20 00:07:36	Xe	1.74	Sn	52.455722	13233	Show
9	9069	Test	2026-01-19 23:54:49	2026-01-19 23:54:49	Xe	1.74	Sn	52.455722	4969	Show
9	9068	Test	2026-01-19 23:50:00	2026-01-19 23:50:00	Xe	1.74	Sn	52.455722	1409	Show
9	9067	Test	2026-01-19 23:29:48	2026-01-19 23:29:48	Xe	1.74	Sn	52.455722	54057	Show
9	9066	Test	2026-01-19 22:55:15	2026-01-19 22:55:15	Xe	1.74				
9	9065	Test	2026-01-19 22:07:31	2026-01-19 22:07:31	Xe	1.74				
9	9064	Test	2026-01-19 20:03:02	2026-01-19 20:03:02	Xe	1.74				
9	9063	Test	2026-01-19 18:59:51	2026-01-19 18:59:51	Xe	1.74				
9	9062	Test	2026-01-19 18:24:30	2026-01-19 18:24:30	Xe	1.74				
9	9061	Calibration	2026-01-19 17:42:01	2026-01-19 17:42:01	Xe	1.74				
9	9060	Unknown	2026-01-19 17:31:21	2026-01-19 17:31:21						
9	9059	Test	2026-01-19 16:52:01	2026-01-19 16:52:01	Xe	1.74				

zero event count from decoder
→ no possibility to show daily statistics during Run



9	9064	Test	2026-01-19 20:03:02	2026-01-19 20:03:02	Xe	1.74	Sn	52.455722	0	Show
9	9063	Test	2026-01-19 18:59:51	2026-01-19 18:59:51	Xe	1.74	Sn	52.455722	0	Show
9	9062	Test	2026-01-19 18:24:30	2026-01-19 18:24:30	Xe	1.74	Sn	52.455722	0	Show
9	9061	Calibration	2026-01-19 17:42:01	2026-01-19 17:42:01	Xe	1.74	Sn	0	0	Show
9	9060	Unknown	2026-01-19 17:31:21	2026-01-19 17:31:21					0	Show
9	9059	Test	2026-01-19 16:52:01	2026-01-19 16:52:01	Xe	1.74	Sn	52.455722	0	Show
9	9058	Calibration	2026-01-19 15:36:17	2026-01-19 15:36:17	Xe	1.74	Sn	0	0	Show
9	9057	Calibration	2026-01-19 15:27:33	2026-01-19 15:27:33	Xe	1.74	Sn	0	0	Show
9	9056	Calibration	2026-01-19 15:00:52	2026-01-19 15:00:52	Xe	1.74	Sn	0	0	Show
9	9055	Calibration	2026-01-19 14:46:29	2026-01-19 14:46:29	Xe	1.74	Sn	0	0	Show
9	9054	Test	2026-01-19 14:39:50	2026-01-19 14:39:50	Xe	1.74	Sn	0	0	Show
9	9053	Test	2026-01-19 14:34:37	2026-01-19 14:34:37	Xe	1.74	Sn	0	0	Show

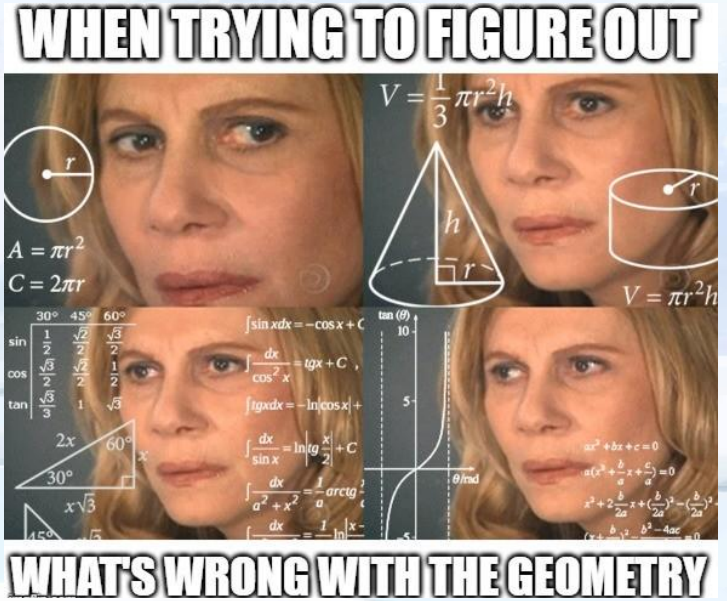
equal timestamps → no possibility to calculate magnetic field from Tango
→ no reconstruction

time: 1995-10-11, 2026-07-27… run: -15432432

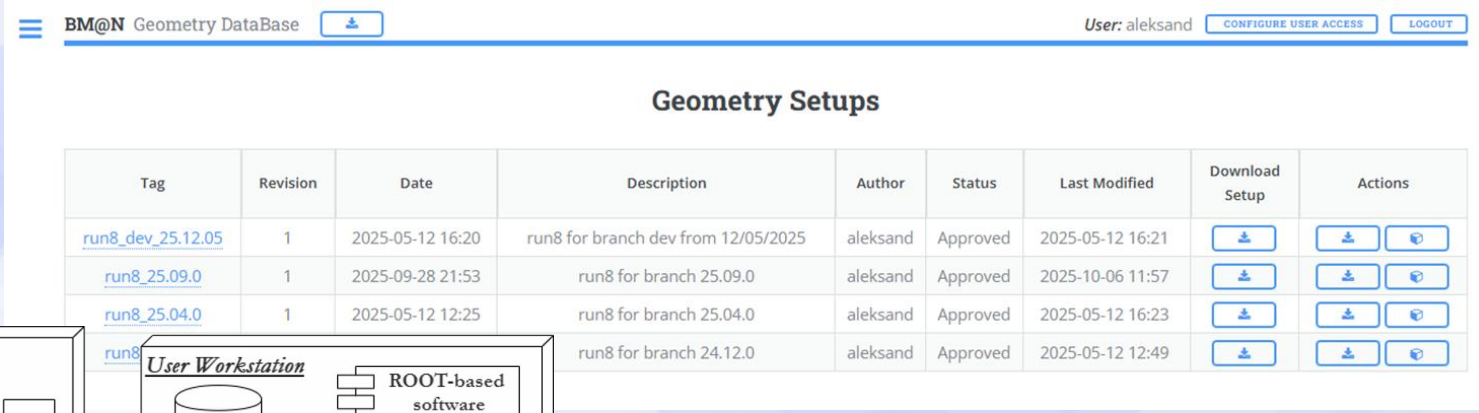
8977 per.9: 2026-01-17 16:44:27 – 9759 per.9: 2026-02-27 12:04:26.000 → 708 runs

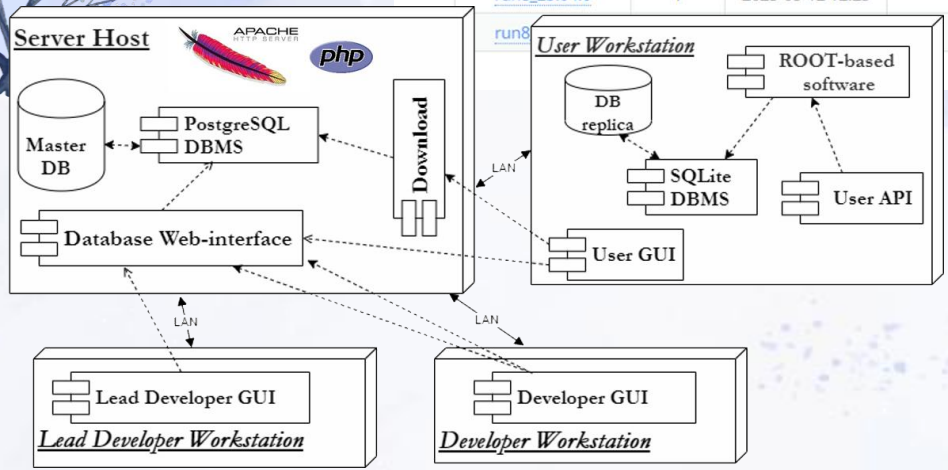
Geometry Database

Offline Information System



BM@N Geometry Information System





The Geometry Database has filled with setup geometries for Run 8 (24.12.0, 25.04.0, 25.09.0 dev_25.12.05, dev_25.11.01, dev_26.02.20) and Run 9 (26.02.0, dev_25.12.10)

Pure BmnRoot takes up 1050 MB of disk space, where “bmnroot/input/” ≈ 700 MB (mostly, magnetic field files) and “bmnroot/parameter/” ≈ 100 MB

Geometry Module → **Setup Module** → **Setup**

Geometry Module
File in the ROOT format containing a detector geometry

Setup Module
Geometry module + link to the parent geometry module + its placement in the parent module

Setup
Combination of setup modules, which represents the full setup geometry

Geometry System. Application Programming Interface

Geometry API provides a set of public functions of the ***GeoSetup*** class intended for selecting and loading a required setup geometry and its components into the ROOT-based software of the experiment to perform simulation, reconstruction and physics analysis tasks

run_sim_bmn.C

```
    // Set the BM@N geometry
#ifdef USE_GEODB
    GeoSetup gSetup;
    int errStatus = gSetup.LoadSetupToFairRunSim("run8_dev_25.11.01");
```

run_reco_bmn.C

```
    // get geometry for run
#ifdef USE_GEODB
    GeoSetup gSetup;
    int errStatus = gSetup.LoadSetupToGeoManager("run9_26.01.0");
```

```
int LoadSetupToGeoManager(string_view setup_tag);  // 0 - loaded, !=0 – error code
int LoadSetupToFairRunSim(string_view setup_tag);     // 0 - loaded, !=0 – error code

FairField* GetMagneticField(double scale = 0);   // nullptr – no field
string_view GetParameterFile(string_view detector_name);  // "" – no parameter file

string_view GetLastApprovedSetupTag (string_view tag_prefix);  // "" – no setup found
```

Distributed Data Processing

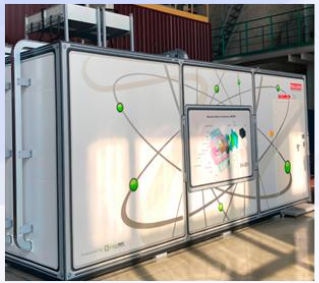
@ Computing Clusters



BM@N Computing Platforms

/bmn-daq-computing-center/

BM@N Online Cluster
(DAQ Data Center)
ddc.jinr.ru
(LHEP, b.205)



bmn.jinr.int/nica-cluster/

NICA Cluster
ncx[101-106].jinr.ru
(LHEP, b.216)



bmn.jinr.int/micc-complex/

GRID Tier1&2 Centres
(CICC complex)
lxui.jinr.ru
(MLIT, b.134)



bmn.jinr.int/hybrilit-govorun/

SC «Govorun»
(HybriLIT platform)
hydra.jinr.ru
(MLIT, b.134)



OS: AlmaLinux 9.7

OS: CentOS / Scientific Linux 7.9 (EOL on June 30, 2024)

Central Software Repository based on ***CVMFS*** for the experiment

CEPH: 2.8 PB (*replica*)
CEPH (hot): 100 TB$_{ssd}$
SLURM: ≈**700** cores

EOS: 1.8 PB (*replica*)
NFS4: **300 TB** (*for NICA*)
SLURM: ≈**1800** cores
(*for all NICA users*)

EOS: 1.9 PB (*replica*)
EOS CTA: **500 TB**$_{tapes}$
SLURM: 2500 cores
(*for all NICA users*)

ZFS: 200 TB
Lustre: **300 TB**$_{ssd}$ (*for NICA*)
SLURM: *bmn* – **192** cores

BM@N software is deployed on *JINR CVMFS* for Centos 7 and AlmaLinux 9
***/cvmfs/bmn.jinr.ru/config/x86_64-alma9/cluster_config.sh* for AlmaLinux 9 (GCC14)**

Data Storages for BM@N (27.02.2026)

ceph

for BM@N online (2.4 PB)

build on HDD with SSD buffer, **900 TB free**

NICA cluster

NFS

fast on NVMe SSD

MLIT CICC

nope

HybriLIT

lustre

fast on NVMe SSD

NICA cluster

EOS

for BM@N offline (1.8 PB)

build on HDD, **470 TB free**

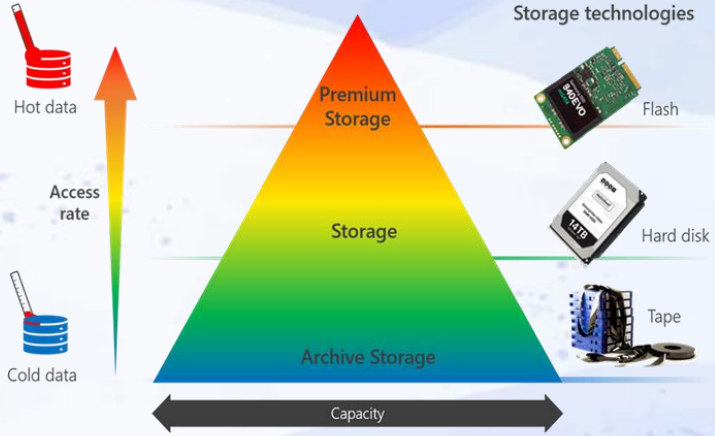
sync | duplicate

MLIT MICC

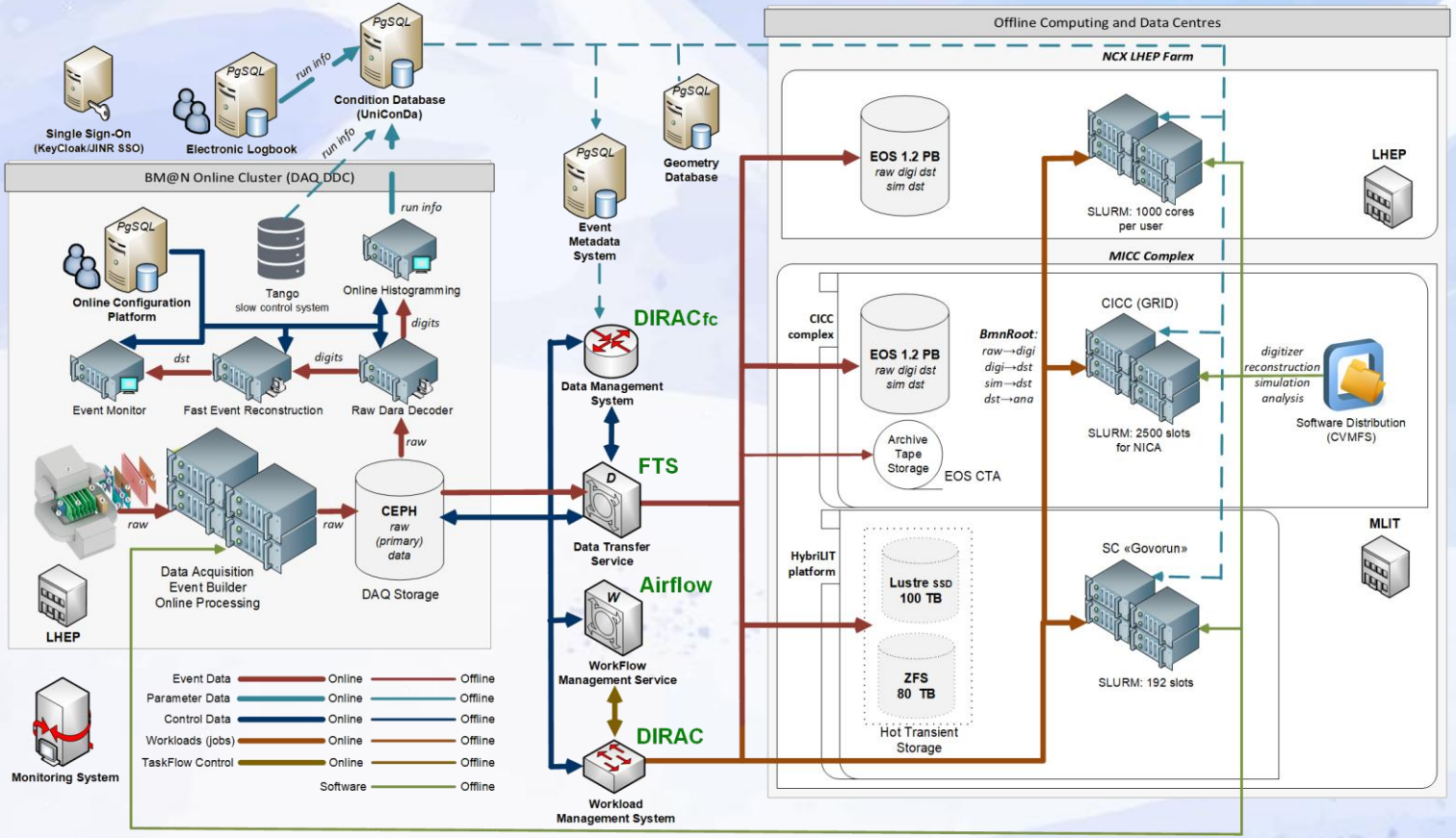
EOS

for BM@N offline (1.9 PB)

build on HDD , **0 TB free**

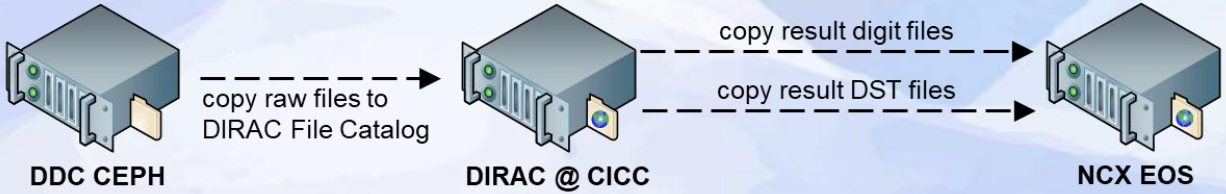


BM@N Computing Software Architecture



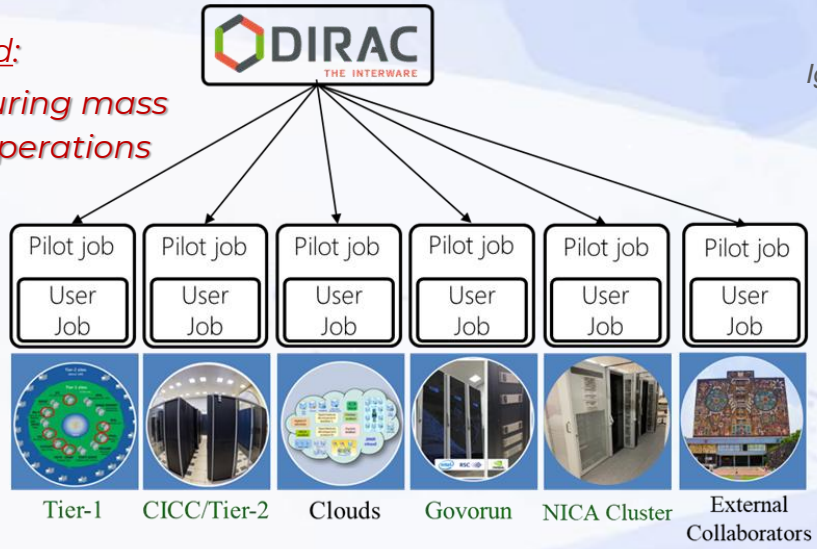
BM@N Offline Processing with DIRAC

- Run mass production of DIGIT and DST files for obtained raw files via DIRAC without writing run and raw file metadata to UniConDa: **BmnRoot tag 26.02.0**



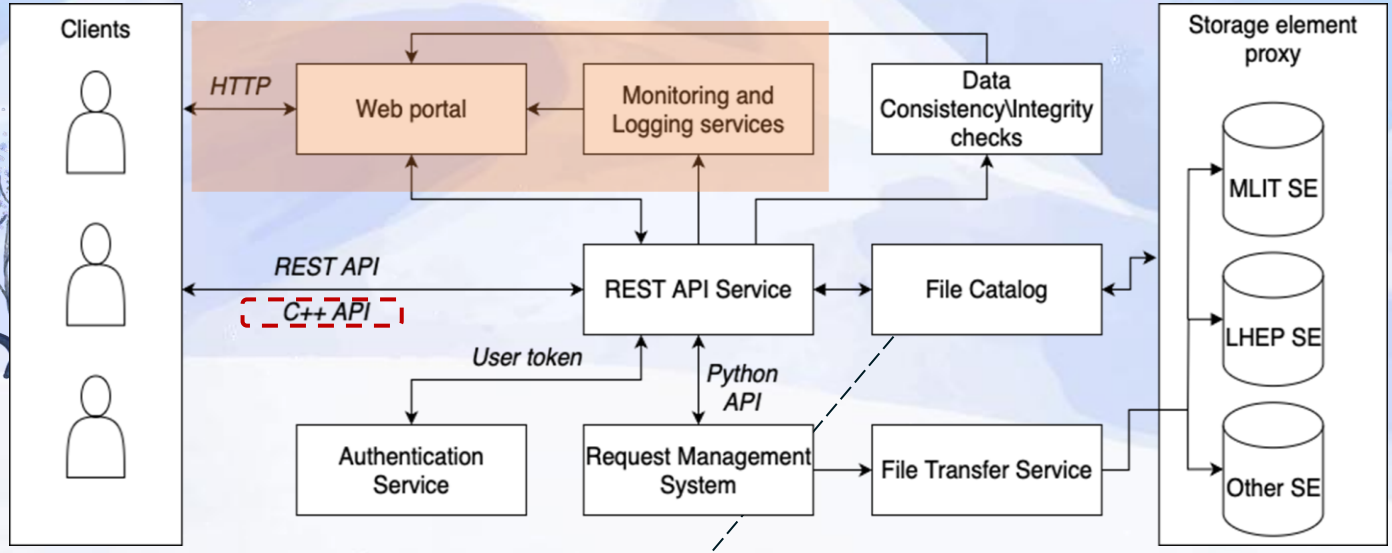
the question to be solved:

collection of raw file metadata during mass production to exclude manual operations



Igor Pelevanyuk (28 February, 14:00)
BM@N Distributed computing readiness for Run 9

BM@N DMS on DIRAC File Catalog



Igor Zhironkin (28 February, 14:20)
Development of the Data Management System for the BM@N experiment

GET
- get file list matching specific metadata
- get specific file metadata
- get all metadata fields
- is file exist
- get file catalog stats. (Number of files/directories/replicas etc.)
- get status of the last consistency check
- get result of the last consistency check
- download file from File Catalog

POST
- add new metadata field
- run consistency check. Return check ID, status can be verified through get
- upload and register file in the File Catalog

PUT
- update file metadata

DELETE
- delete specific file metadata
- remove specific metadata field
- delete file from SE and File Catalog

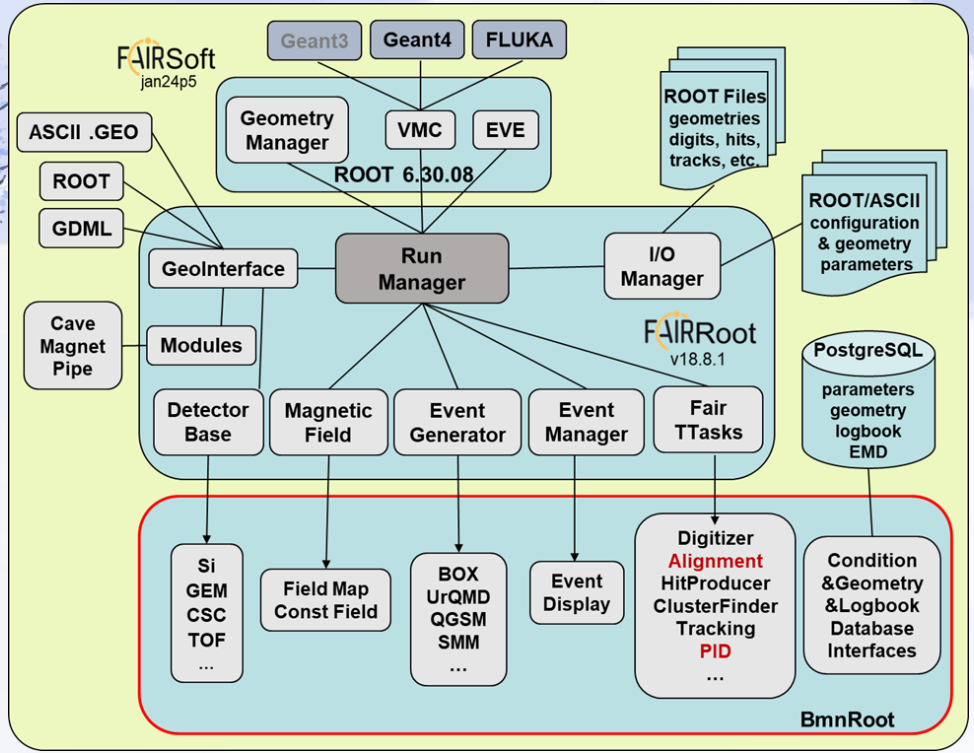
BmnRoot and Collaboration Services



BmnRoot. Production Tag 26.02.0

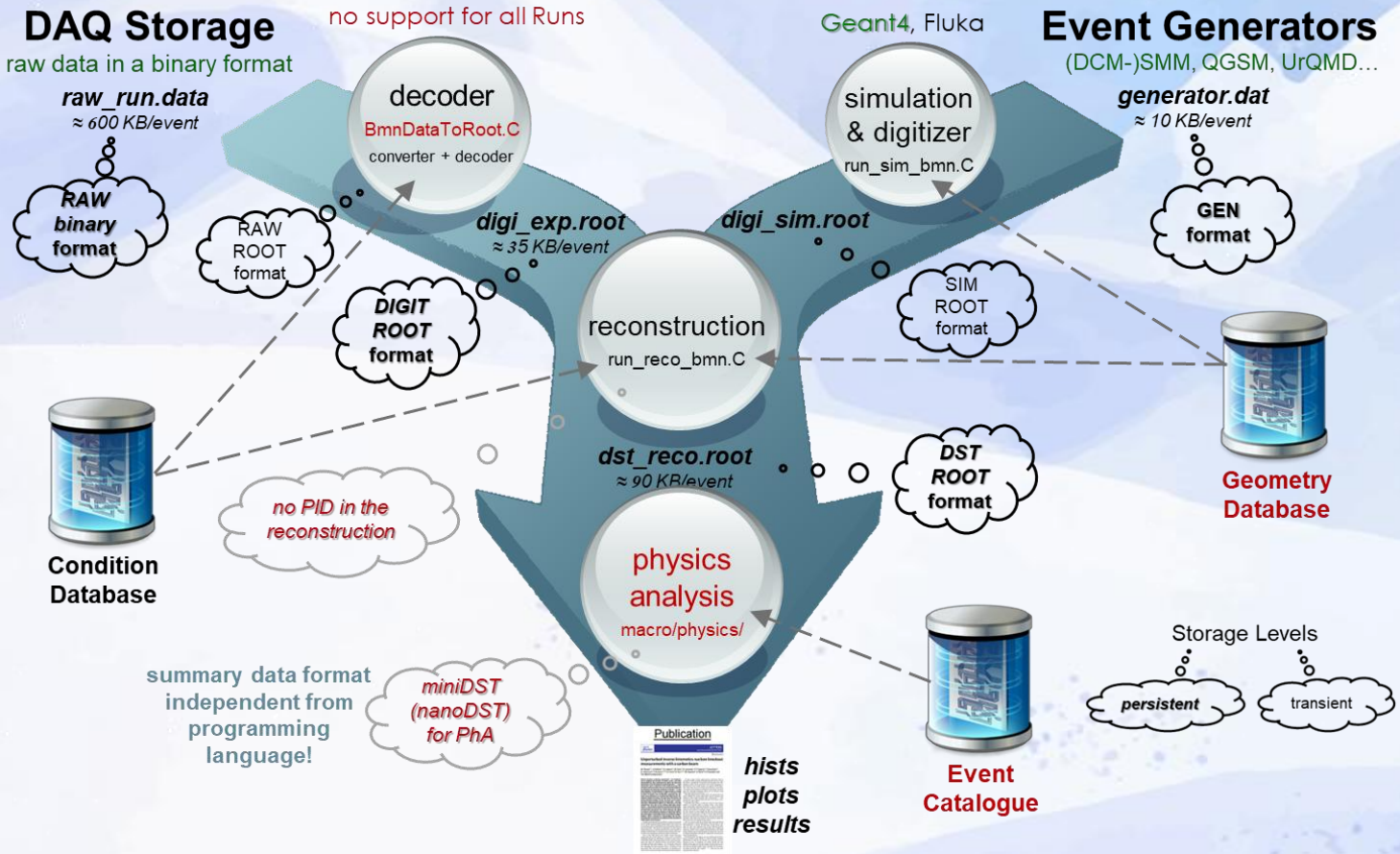
FairSoft jan24p5 FairRoot v18.8.2

- FairSoft jan24p5 is used (with ROOT 6.30.08)
- FairRoot v18.8.2, but 19.0.0 is available (new compilation approach)

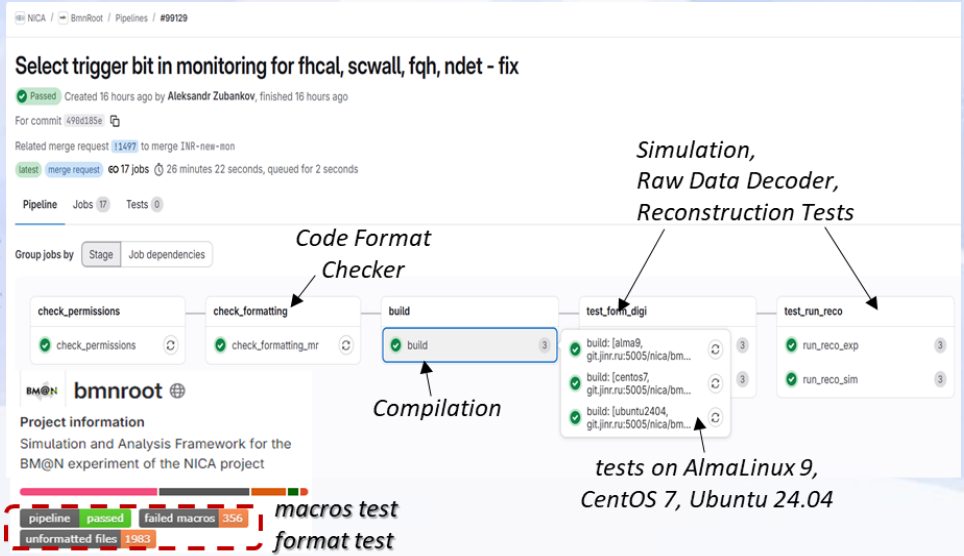


- BmnRoot tag 26.02.0 has been issued and DST production is continuously performing using DIRAC
- The BmnRoot GitLab pipeline has been modified to test the BmnRoot installation version
- Removing deprecated PROOF should be discussed
- It might be worth making one modification to the entire BmnRoot regarding code formatting
- The current BM@N geometry is not passing standard ROOT tests because of several overlaps
- BmnRoot manual should be updated, documentation on data file structure is necessary

BmnRoot. Event Data Model



Software Management & Distribution System

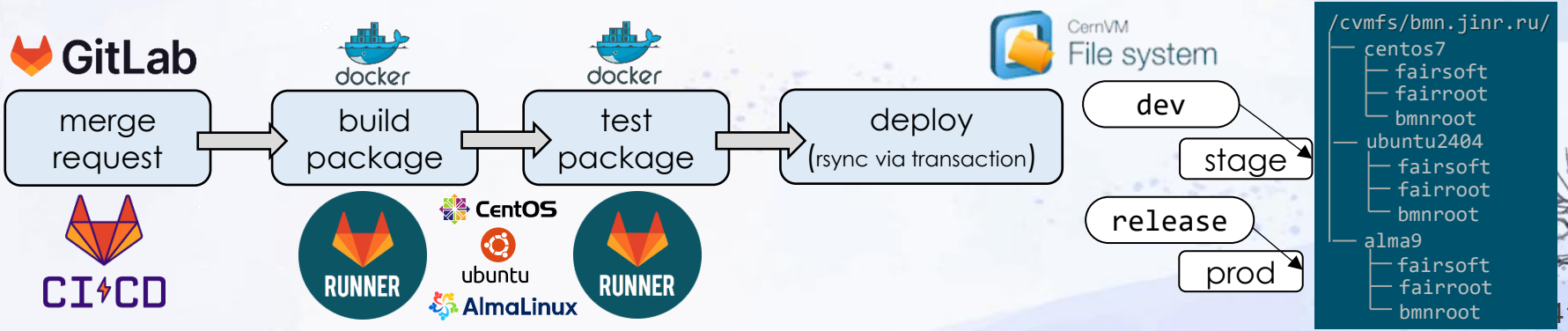


GIT Pipelines on Merge Requests

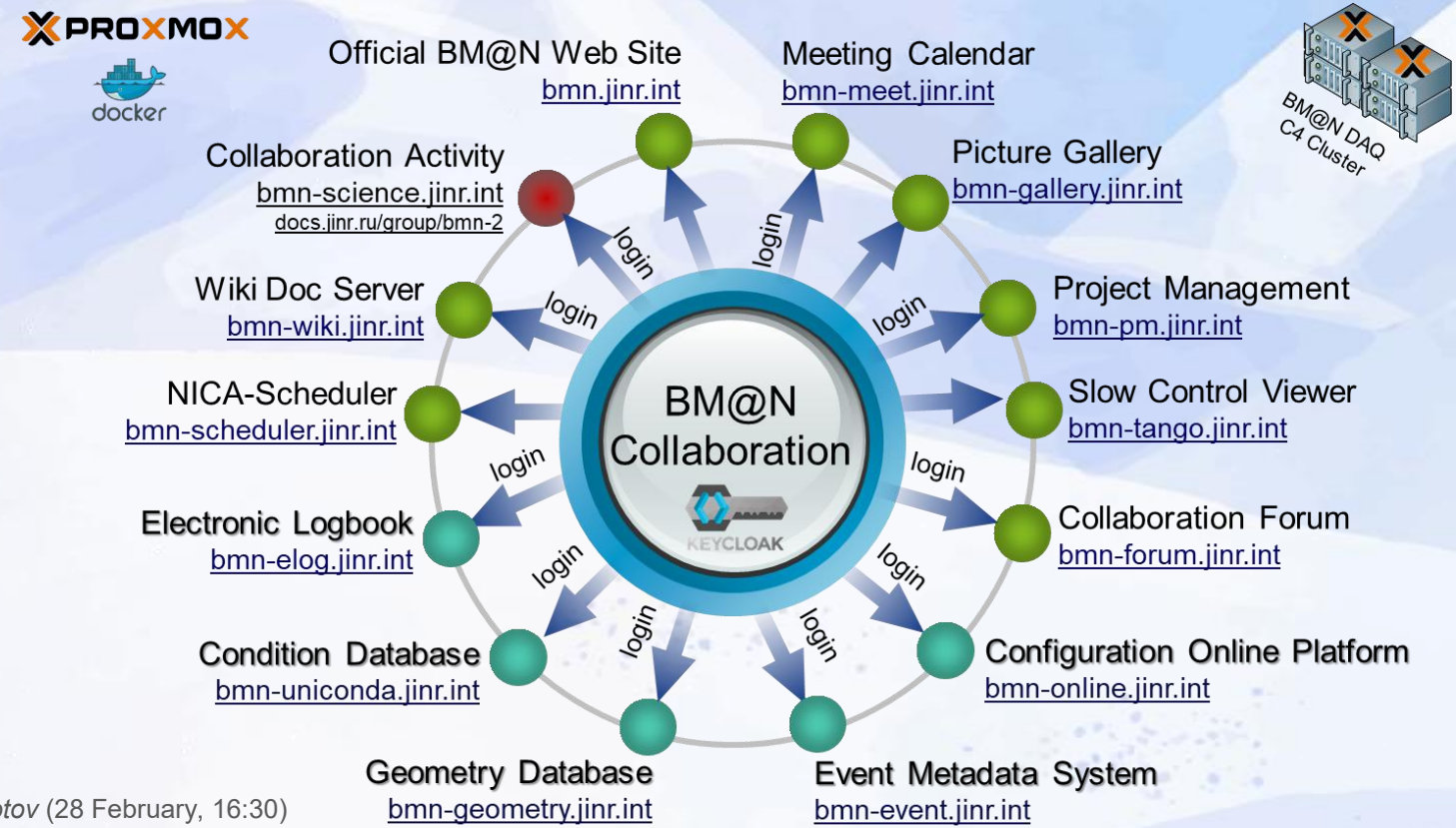
- checking compilation and main macros **after BmnRoot installation**
  → stable dev & pro branches, all tags
- deploy BmnRoot to the CVMFS storage
- deploy BmnRoot containers to the registry
- checking Clang code formatting

GIT Night Tests

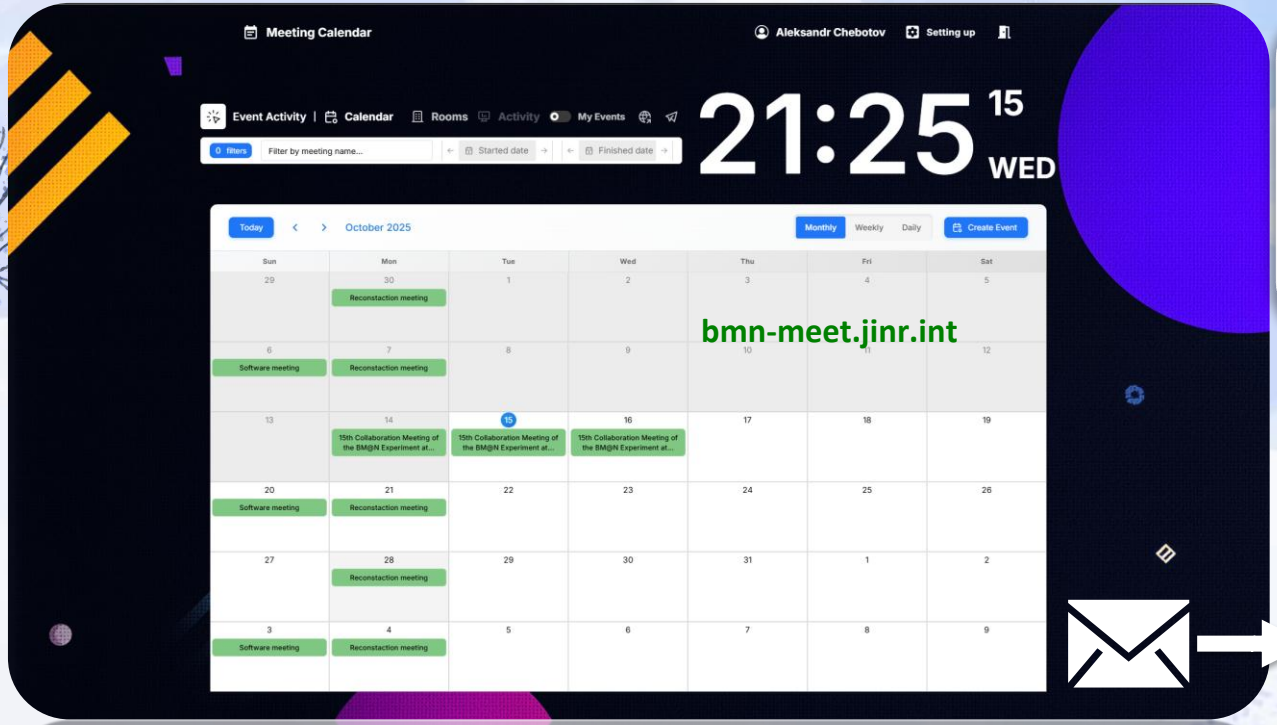
- test all BmnRoot macros
- test all C++ files for correct code format
- generate Doxygen documentation

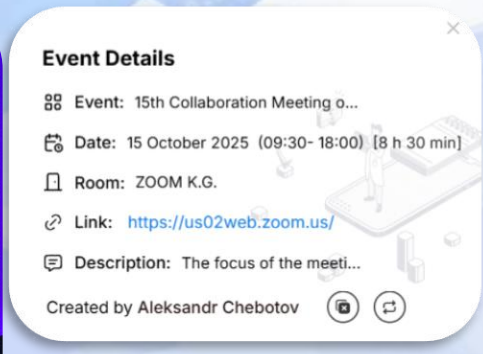


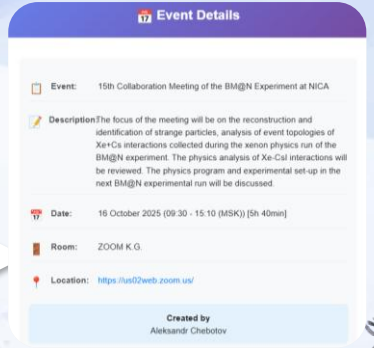
BM@N Software Ecosystem



BM@N Official Meeting Calendar

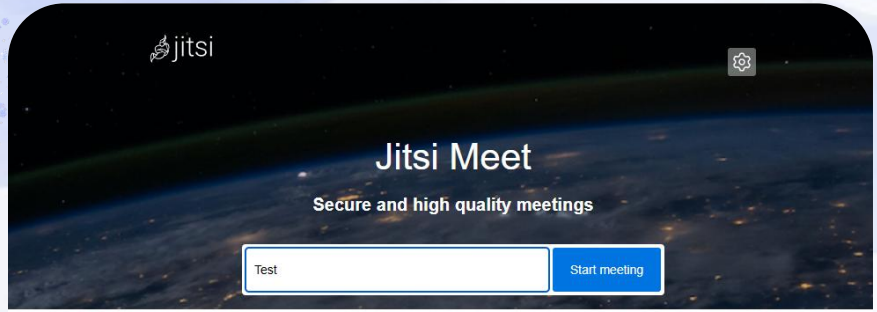




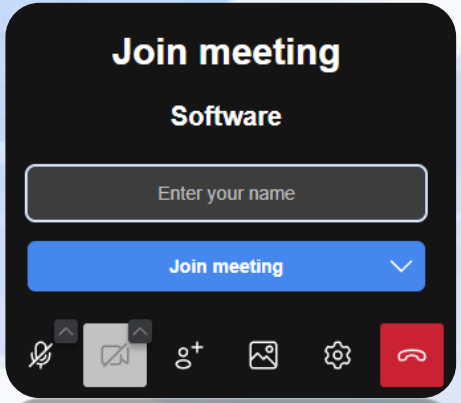


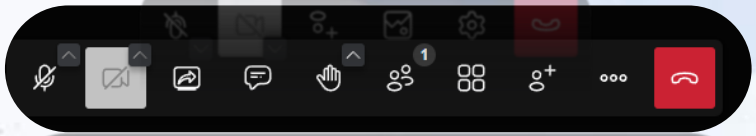
BM@N Jitsi Meet

Open-source, self-hosted video conferencing platform deployed within BM@N infrastructure





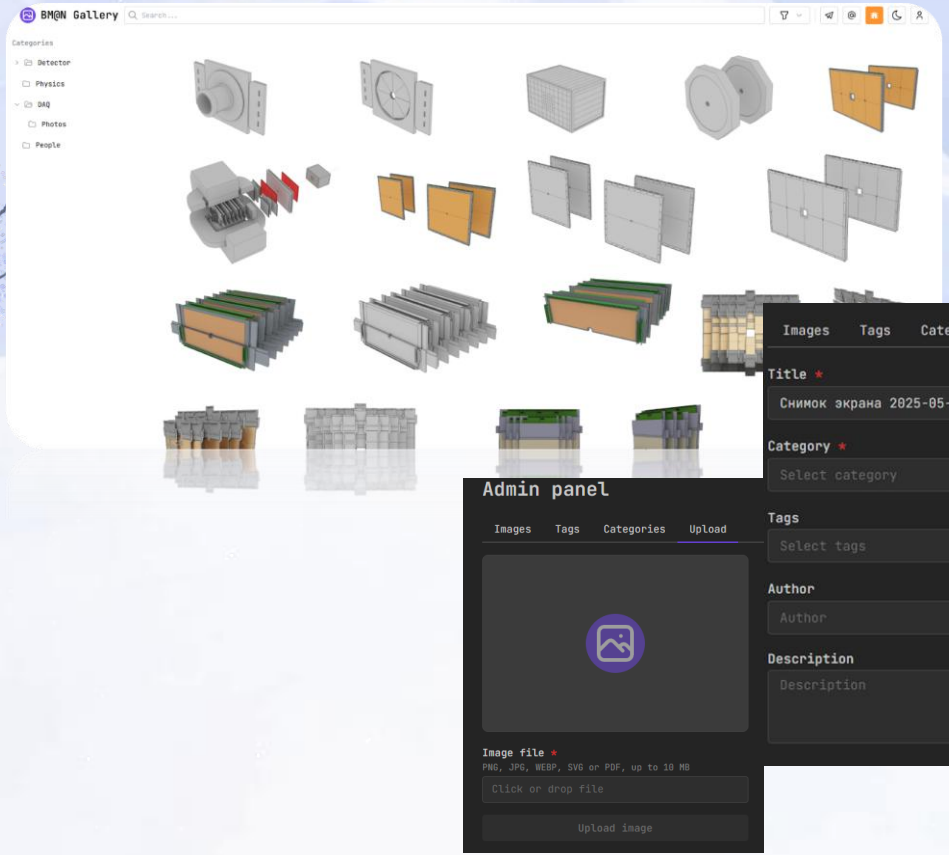




Meeting recordin **Unlimited rooms**

Breakout rooms **Screen sharing**

BM@N Gallery



Gallery for officially approved photo and figures


THANK YOU
FOR YOUR
ATTENTION


TikTok
@ellynjune
What snowboard mates are
good for


What snowboard mates are
good for

BACKUP

raw statistics

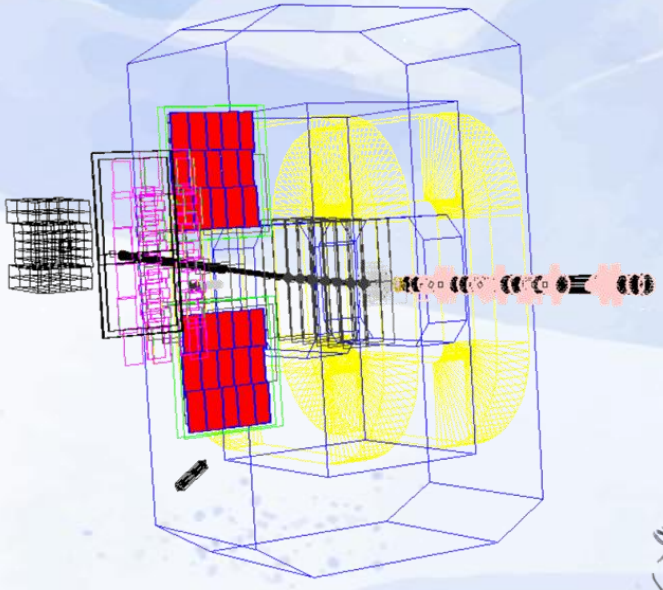
Post-Run Procedure. Actualizing BM@N geometry

- Write the current (in *run_sim_bmn.C*) setup geometry to UniConDa

*root bmnroot/database/examples/uni_db/geometry/**write_actual_geometry_to_db.C**(9,1,100000)*

macro for writing the currently used full setup geometry
to UniConDa for given runs
Technically it runs run_sim_bmn.C and writes obtained
"full_geometry.root" to the database

```
void write_actual_geometry_to_db(int period = 9,
int start_run = 1, int end_run = 100000)
{
  run_sim_bmn("", simFileName, 0, 1, BOX, kTRUE, kTRUE);
  gSystem->Unlink(simFileName);

  write_geometry_to_db("full_geometry.root", period, start_run,
period, end_run);
}
```



Post-Run Procedure. Write Run and Raw File Metadata

- Create and update raw file metadata in the Unified Condition Database

  *root bmnroot/database/examples/uni_db/**write_raw_file_info.C***

OR Raw Data Converter/Decoder can be used to create raw file metadata:

*root 'bmnroot/macro/raw/**BmnDataToRoot.C**("file.data", "", 0, 9, 2)'*

Run Metadata are automatically calculated and synced while adding raw file metadata according with the trigger of the Unified Condition Database (UniConDa)

P.S. Write permissions are required to write to the UniConDa database

Post-Run Procedure. Fix Run and Parameter Range

- Define real first and last runs to replace old virtual numbers

  Execute changing *run_period time interval* and *parameter ranges*

  (***update_run_period_time*** and ***update_detector_paramater_range*** function)

  *root bmnroot/database/examples/uni_db/run/**postrun_update.C***

First run number to be processed

Is it the first number of the raw file or the first run number in the Electronic Logbook or the first record with 'Physics Run' type? 8977 (UniConDa) or 9011 (e-Log)

Last run number to be processed

The same = 9759

Parameter ranges and run_period time interval should be changed, e.g.

```
.L postrun_update.C
update_run_period_time(9,9011,9759)
update_detector_paramater_range(9,8968,9094,9011,9094)
```

Post-Run Procedure. Write and Check Magnet Field

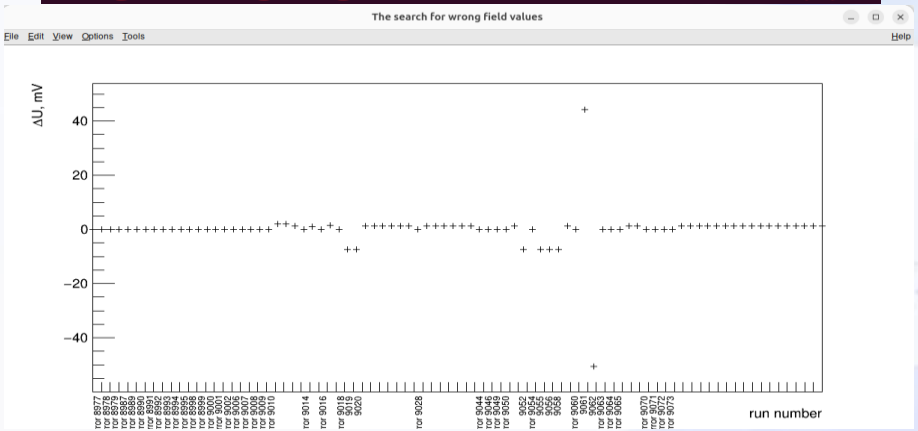
- Write average magnetic field (mV) stored in Tango to UniConDa (***write_avg_field_db****(9, 8977, 9759, /*is_rewrite*/ true, /*tolerance*/ 10)* function) and check possible issues:

a) Tango <> e-Log (***show_avg_field*** function)

b) UniConDa <> e-Log (***compare_avg_field*** function)

*root bmnroot/database/tango/examples/**show_avg_field.C***

```
There is no field value (e-Log) in 38 run(s)
e-Log and Tango magnetic fields differ in 2 run(s)
```



Post-Run Procedure. Fix paths, metadata

- Update run metadata in the Unified Condition Database (UniConDa)

  *root bmnroot/database/examples/uni_db/**postrun_update.C***

  a) Correct raw file paths (***update_raw_file_path*** function or using the SQL query)

  b) Write absent run metadata from e-Log to UniConDa, such as beam, target, energy, field_voltage, run_type (***update_run_metadata_from_elog*** function)

  c) Compare run metadata in e-Log & UniConDa (compare_uniconda_elog functon)

a) Raw file paths has been corrected: **/eos/nica/bmn/exp/raw/run9/beam/**

```
update raw_file
set file_path = '/eos/nica/bmn/exp/raw/run9/beam/' || regexp_replace(file_path, '^.+[/\\]', '')
where period_number = 9 and run_number >= 8977 and run_number <= 9094
```

b) Absent run metadata have been transferred from e-Log to UniConDa (beam, target, energy, field_voltage) using *update_run_metadata_from_elog* function.

c) Check run metadata are not different in e-Log and UniConDa (beam, target, energy, field_voltage, run_type) using *compare_uniconda_elog* function.

Post-Run Procedure. Clean and sync raw data files

- Find by abnormally size and move corrupted raw files to a trash directory.

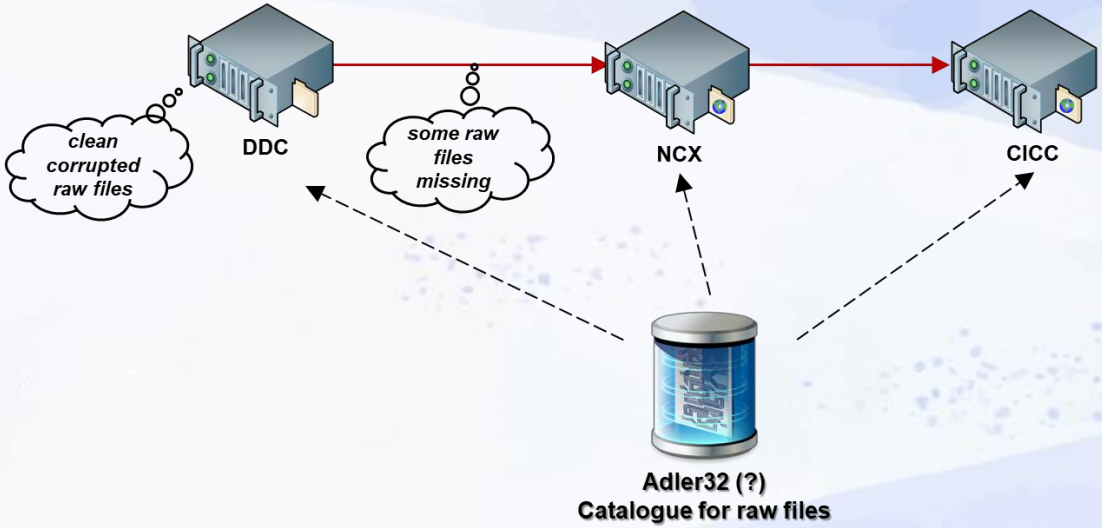
Match (Adler32) and copy missed raw files from the DDC cluster → NCX → CICC

python bmnroot/services/statistics/stats .py

python bmnroot/services/inspection/raw_inspection/filter_raw_data.py –dir /ceph/bmn/...

python bmnroot/services/inspection/raw_inspection/catalog_raw_data.py --run 8 --storage ddc

python bmnroot/services/inspection/raw_inspection/transfer_raw_data.py --run 8 --source ddc --destination ncx



Post-Run Procedure. Test Run and Raw File Metadata

- Match event count and time in UniConDa records between runs and raw files. Possible errors: starting not from 1, more than 1 event between raw files, the sum of the event count of the raw files <> the event count of the corresponding run, the times of the first and last raw files <> the start and end time of the run.

*root bmnroot/database/examples/uni_db/**postrun_check.C***

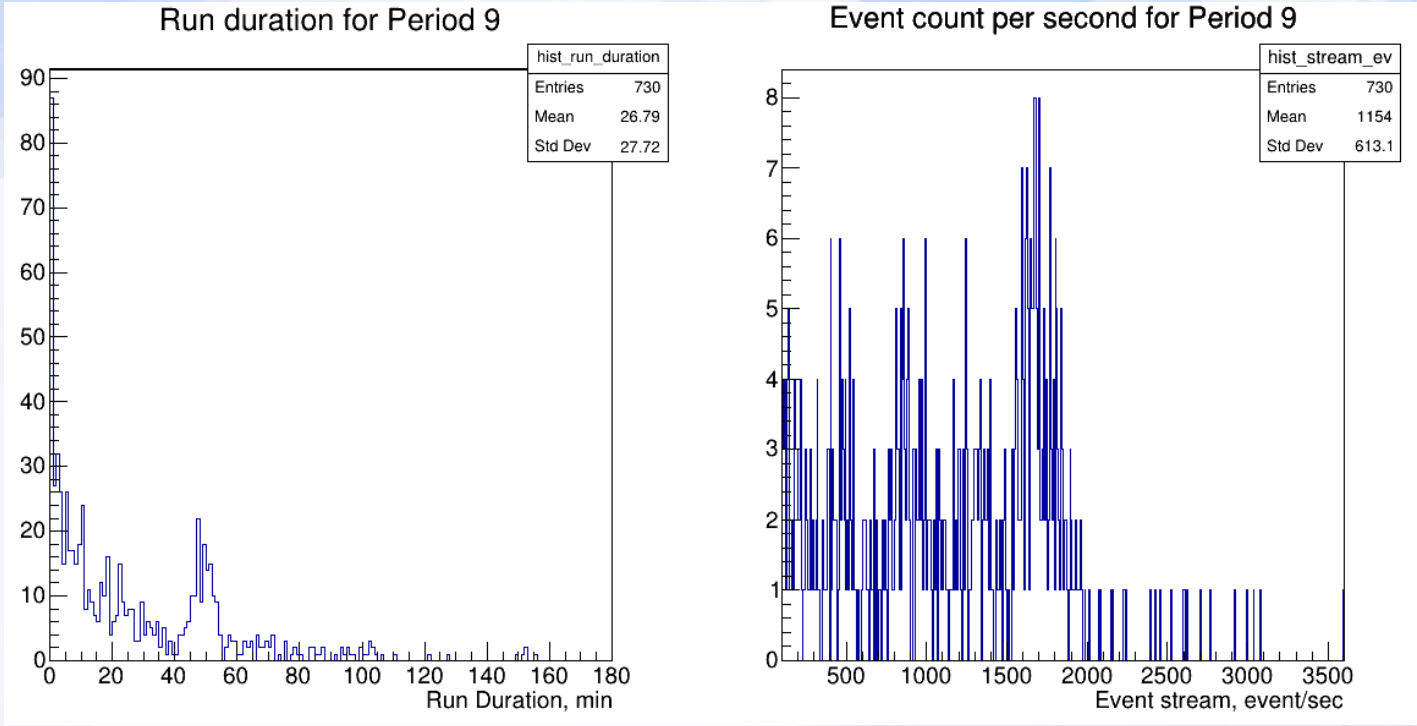
```
No event count in 12 run(s)
The start time of the run is out of range in 112 run(s)
No end time in 2 run(s)
The end time of the run is out of range in 123 run(s)
The start time of the raw file is out of range in 177 run(s)
The end time of the raw file is out of range in 170 run(s)
First event > 1 in 980 run(s)
Parity event gap > 1 in 7102 run(s)
Raw event sum is different from run event count in 1175 run(s)
Run time interval is different from raw file time in 167 run(s)
```

```
Period:Run   8:6693
ERROR: the start time of the run is out of range: 2000-12-12 09:25:48
ERROR: the end time of the run is out of range: 2049-11-28 13:43:48
ERROR: the start time of the raw file is out of range: 2049-11-28 13:42:48
ERROR: the end time of the raw file is out of range: 2049-11-28 13:43:48
ERROR: first event > 1 (first_event = 0)
ERROR: the start time of the raw file is out of range: 2028-12-24 09:44:48
ERROR: the end time of the raw file is out of range: 2028-12-24 09:45:48
ERROR: parity event gap > 1 (current_event = 0, parity start event = 0
ERROR: run time interval (2000-12-12 09:25:48 - 2049-11-28 13:43:48) is different from raw file time (2028-12-24 09:44:48 - 2049-11-28 13:43:48)
```

Post-Run Procedure. Stats on Run Duration and Event Rate

- Get statistics data for **runs** and raw files to check the correctness of the metadata and tune BM@N computing model

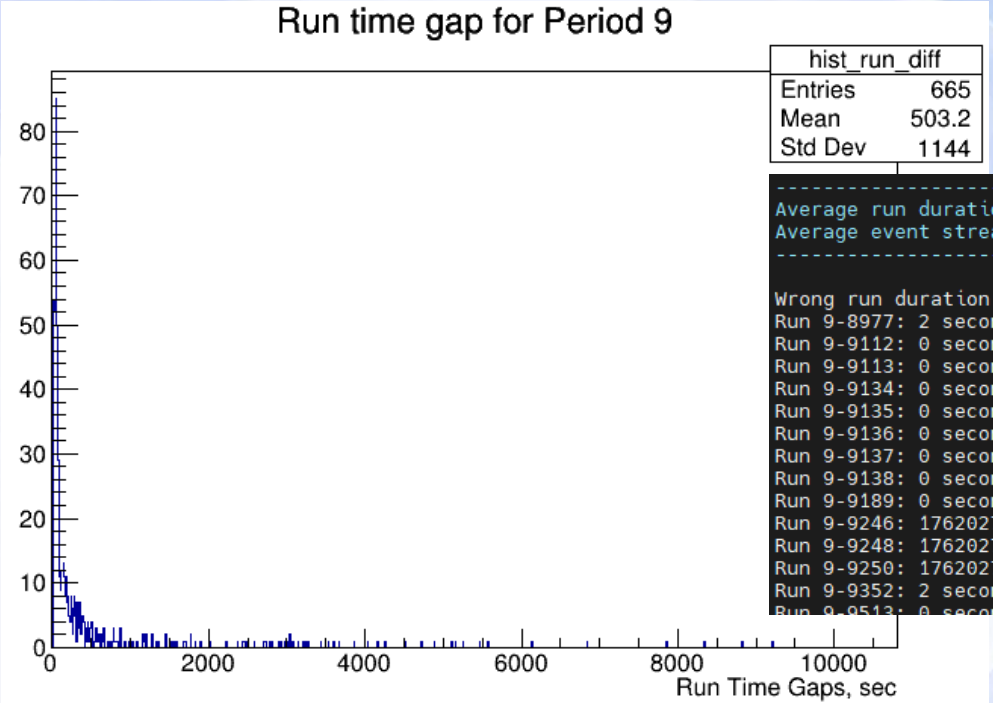
  *root bmnroot/database/uni_db/examples/run/**show_run_stats.C**(9,8977)*



Post-Run Procedure. Stats and checks on Run Duration

- Get statistics data for **runs** and raw files to check the correctness of the metadata and tune BM@N computing model

  *root bmnroot/database/uni_db/examples/run/**show_run_stats.C**(9,8977)*

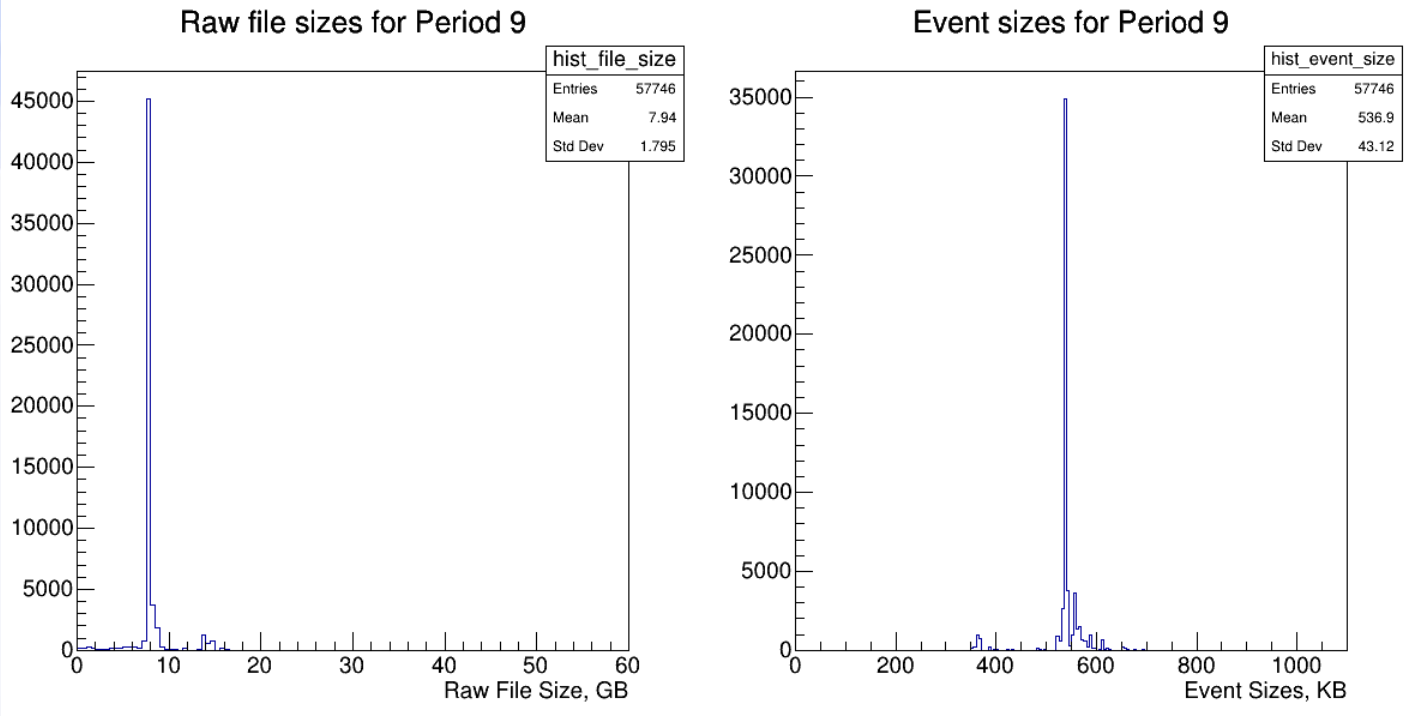



```
------------------------------------------------------------
Average run duration for period 9: 1738 sec (28.96 min)
Average event stream for period 9: 1145 event/sec
------------------------------------------------------------

Wrong run duration:
Run 9-8977: 2 seconds
Run 9-9112: 0 seconds
Run 9-9113: 0 seconds
Run 9-9134: 0 seconds
Run 9-9135: 0 seconds
Run 9-9136: 0 seconds
Run 9-9137: 0 seconds
Run 9-9138: 0 seconds
Run 9-9189: 0 seconds
Run 9-9246: 176202768 seconds
Run 9-9248: 176202768 seconds
Run 9-9250: 176202768 seconds
Run 9-9352: 2 seconds
Run 9-9513: 0 seconds
```

Post-Run Procedure. Stats on Raw File and Event Size

- Get statistics data for runs and **raw files** to check the correctness of the metadata and tune BM@N computing model

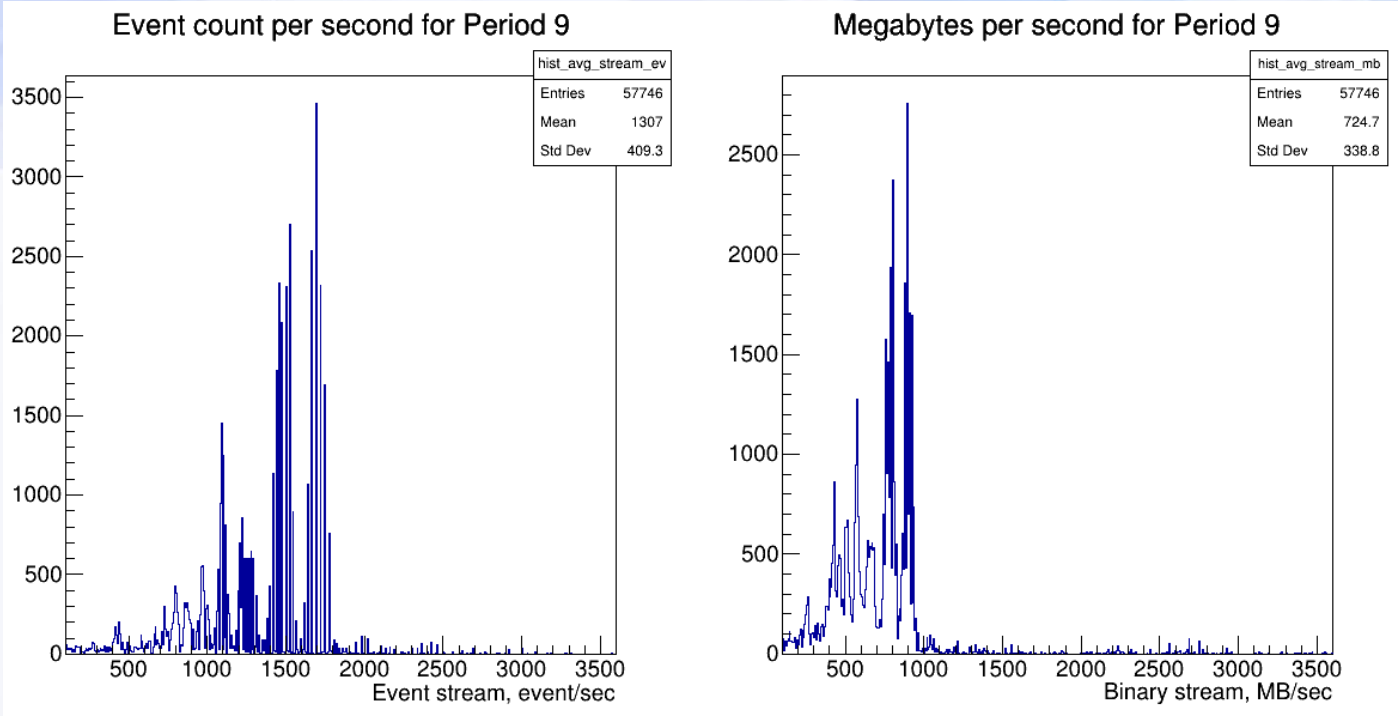
  *root bmnroot/database/uni_db/examples/raw/**show_raw_stats.C**(9,8977)*



Post-Run Procedure. Stats on Event Rate

- Get statistics data for runs and **raw files** to check the correctness of the metadata and tune BM@N computing model

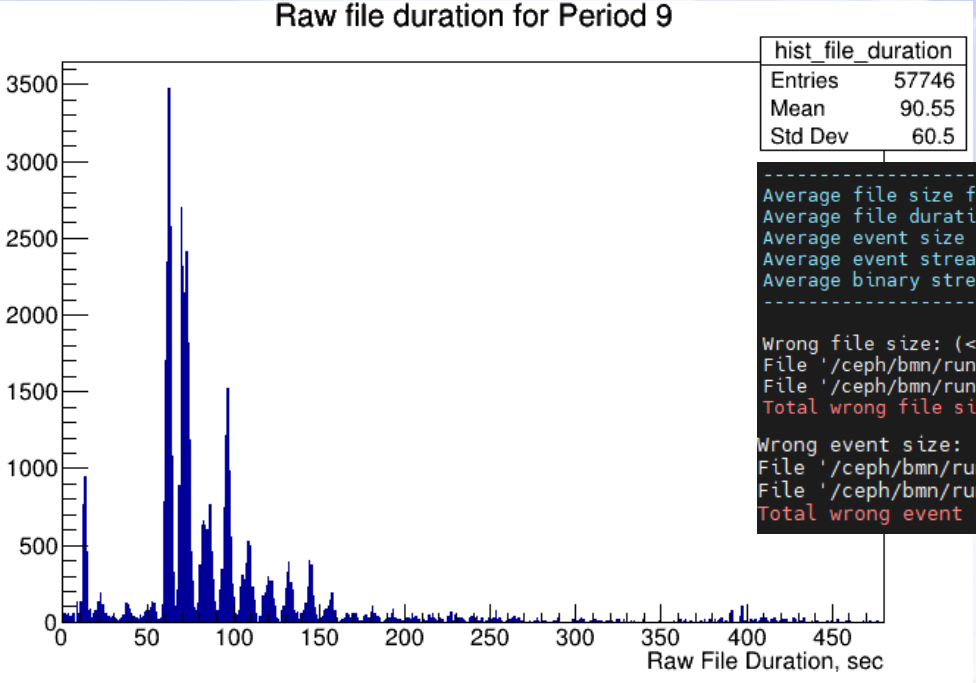
  *root bmnroot/database/uni_db/examples/raw/**show_raw_stats.C**(9,8977)*



Post-Run Procedure. Stats and checks on Raw Metadata

- Get statistics data for runs and **raw files** to check the correctness of the metadata and tune BM@N computing model

  *root bmnroot/database/uni_db/examples/raw/**show_raw_stats.C**(9,8977)*



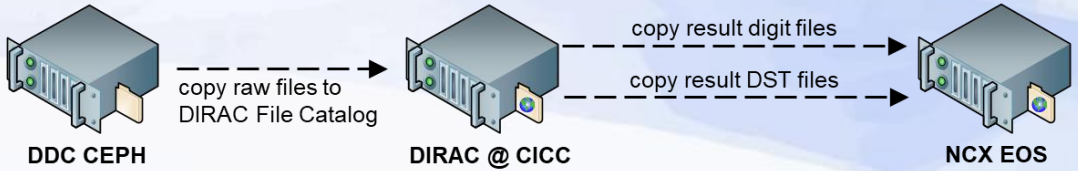

```
------------------------------------------------------
Average file size for period 9: 8123 MB
Average file duration for period 9: 91 sec
Average event size for period 9: 536 KB
Average event stream for period 9: 171 event/sec
Average binary stream for period 9: 89 MB/sec
------------------------------------------------------

Wrong file size: (< 1 MB or > 60 GB):
File '/ceph/bmn/run/run9/beam/9565/run_Top_9565_ev6_p0.data': 6.10352e-05 MB
File '/ceph/bmn/run/run9/beam/9607/run_Top_9607_ev3_p0.data': 6.10352e-05 MB
Total wrong file size: 2

Wrong event size: (< 30 KB or > 1100 KB):
File '/ceph/bmn/run/run9/beam/9274/run_Top_9274_ev0_p0.data: 1465.72 KB
File '/ceph/bmn/run/run9/beam/9467/run_Top_9467_ev0_p0.data: 1531.55 KB
Total wrong event size: 2
```

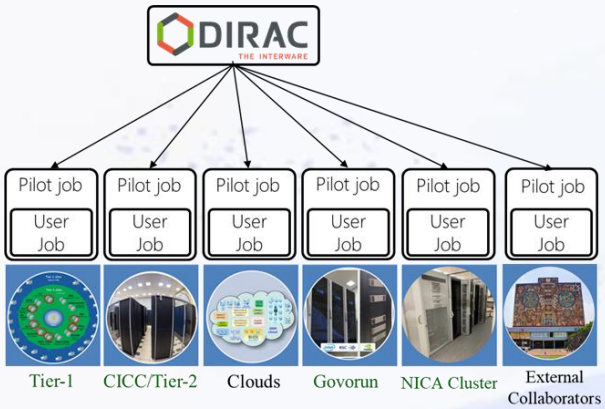
Post-Run Procedure. Mass Production

- Create BmnRoot tag or release and run mass production of digit and DST files for obtained raw files via DIRAC with writing run and file metadata to UniConDa

*python dirac4bmn/**RawToDigi**/submit_jobs.py*

*python dirac4bmn/**DigiToDst**/submit_jobs.py*





Post-Run Procedure. Stats on Digit File and Event Size

- Get statistics data for **digit** and DST files to check the correctness of the metadata and tune BM@N computing model

*root 'bmnroot/database/uni_db/examples/root/**show_root_stats.c**("/eos/nica/bmn/exp/digi/run9/26.02.0/")'*

