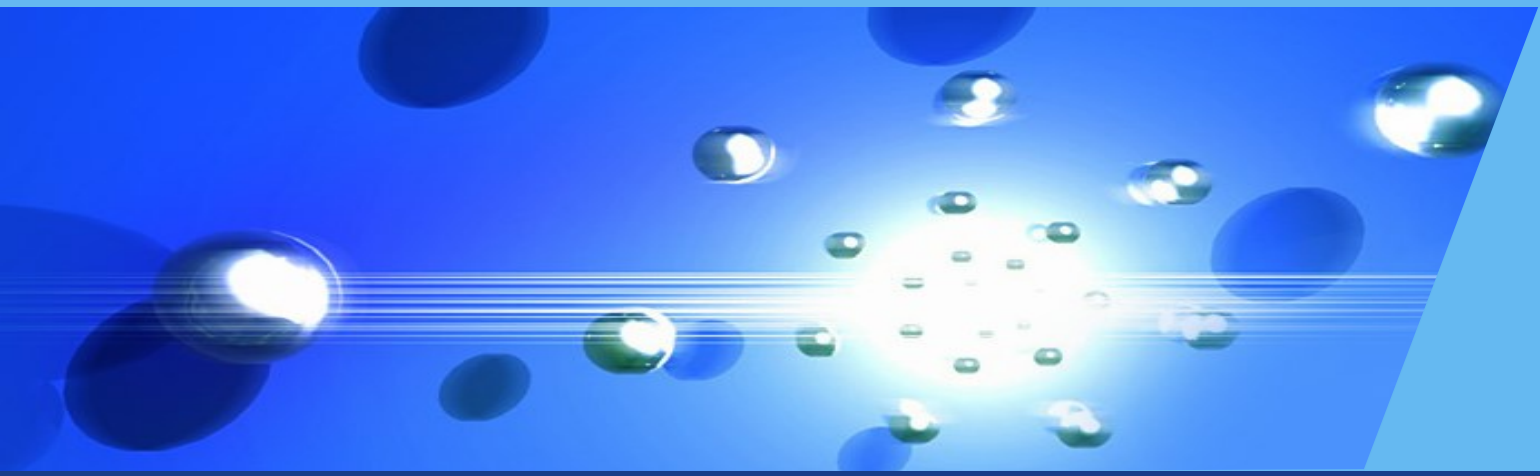
RDIG-M Consortium Meeting
Russian Data Intensive GRID for Megascience



BM@N Computing Model and Resource Prospect

Konstantin Gertsenberger
BM@N Software Coordinator
Joint Institute for Nuclear Research

April 12, 2024

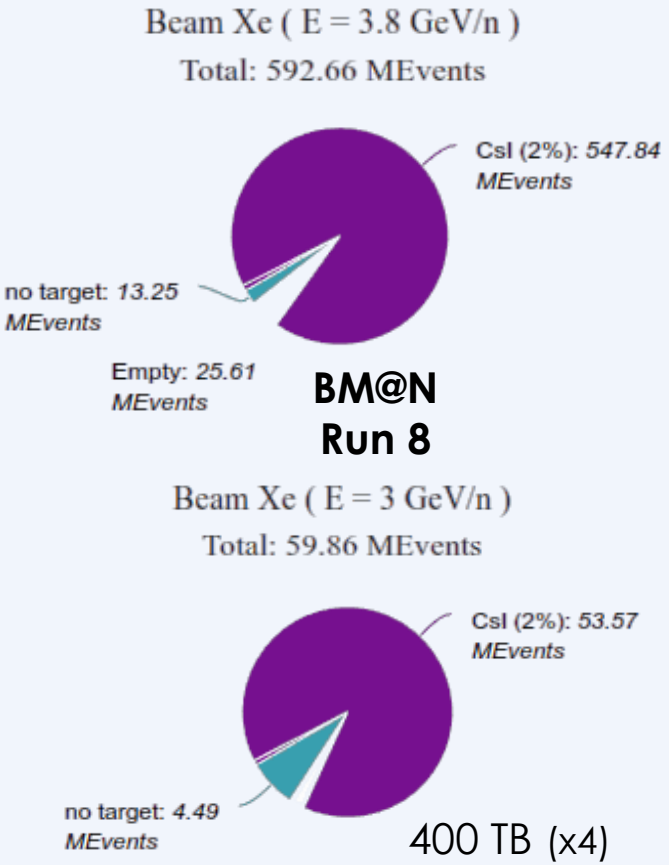
Data Production in BM@N Physics Run

1st Physics BM@N Run

Two beam energy available for *Xe*-beam

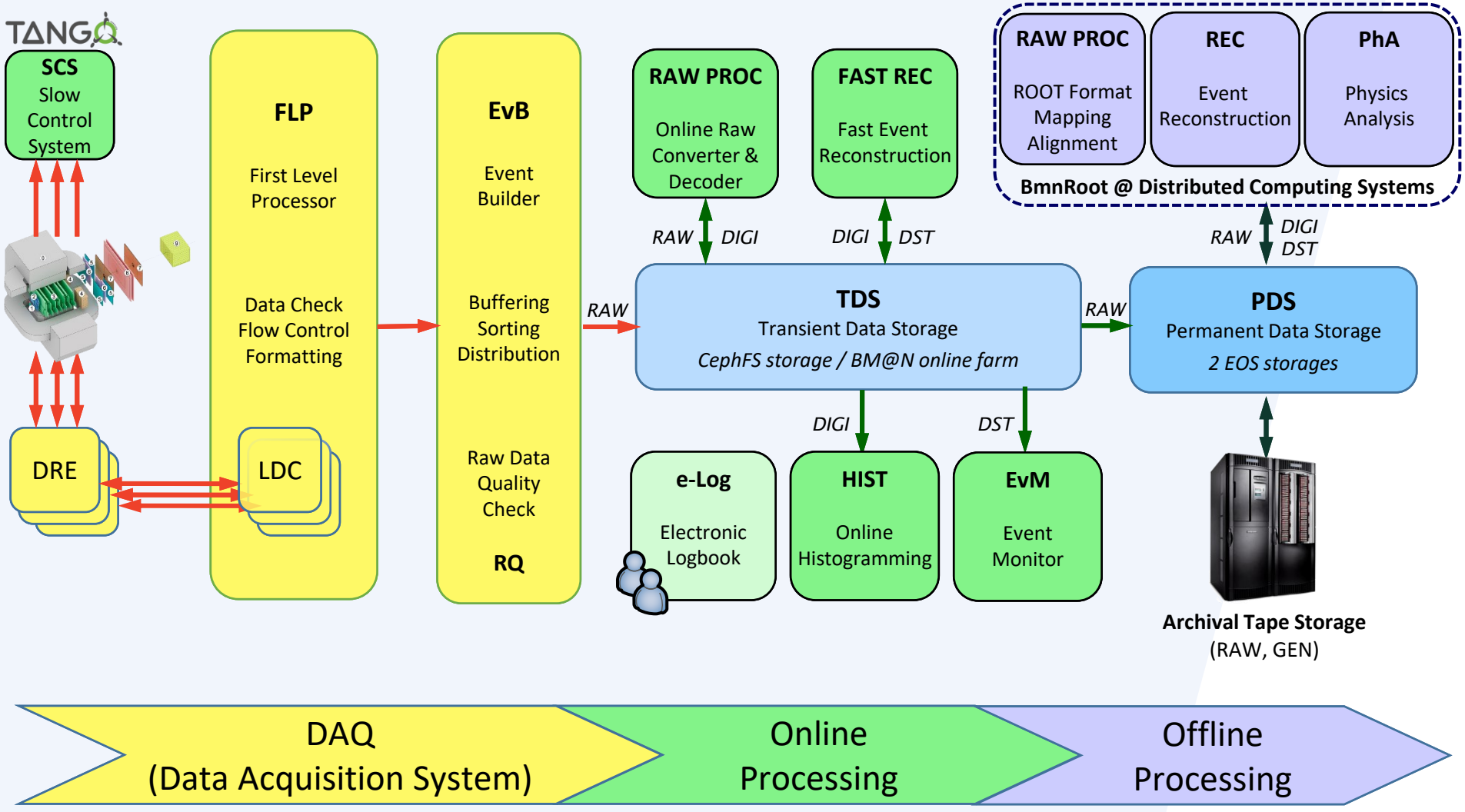
CsI target is used as more similar to *Xe*

More than 600M events were collected



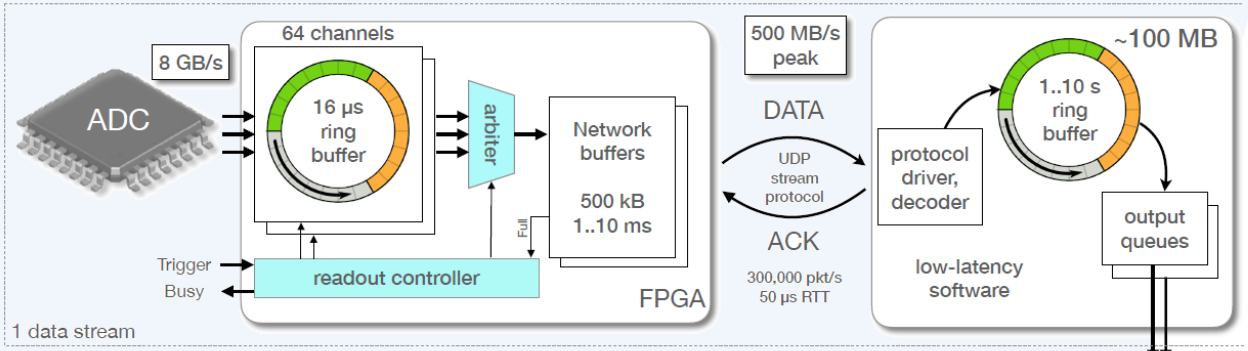
Parameter	Value (approx.)
Data acquisition time	**720 hours**
Average run duration	20 minutes
Average run time break	2.5 minutes
Beam intensity (3.8 AGeV)	up to 900k/2.2 Xe^+/sec up to 900k/12 Xe^+/sec
Trigger rate	**8 000 / 2.2 event/sec**
Average event size	**0,6 МБ**
Data rate	**up to 2 GB/sec**
Raw file size	**15 GB**
Event count per file	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	**870 MB**
Avg DST file size	**2 GB**

BM@N Data Processing Model

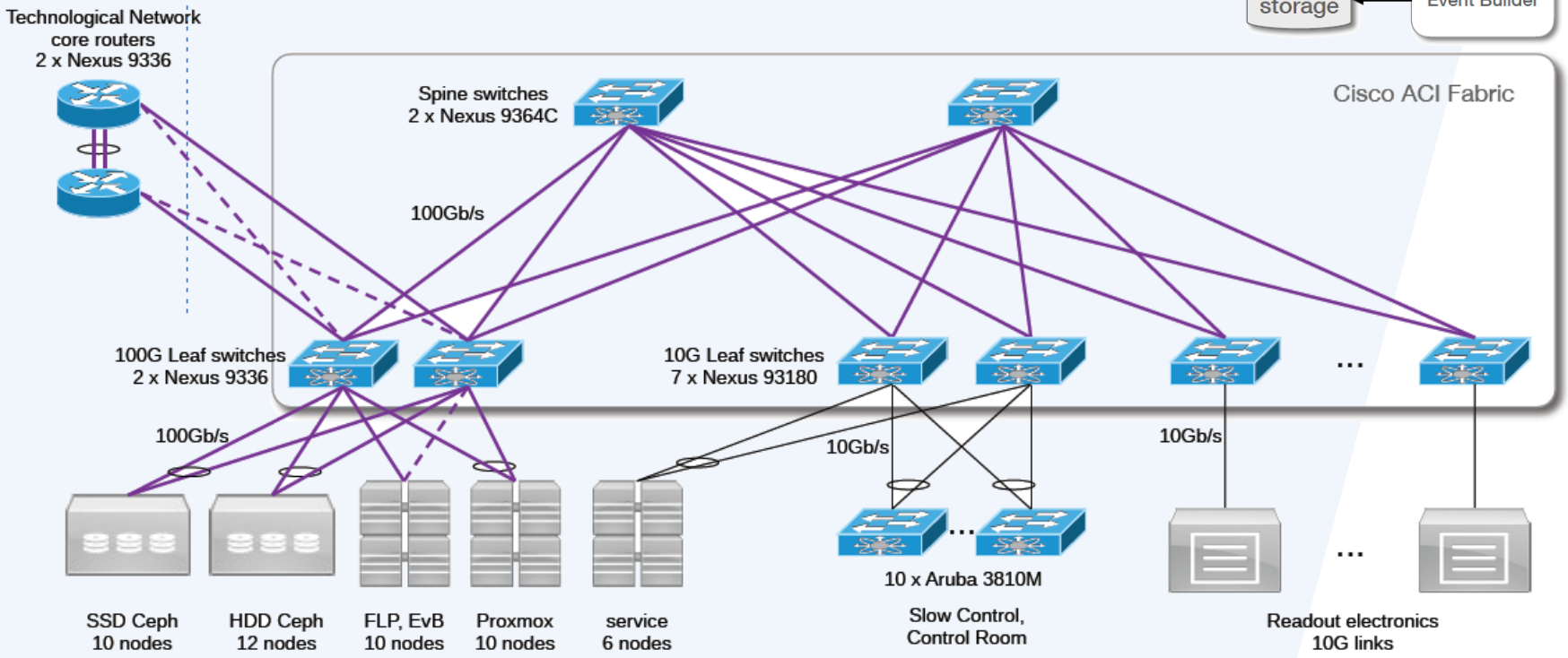


BM@N DAQ Infrastructure

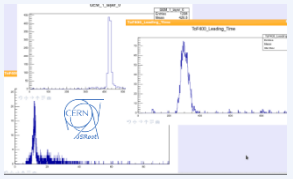
BM@N DAQ
200 data streams
7 GB/s @ 15 kHz



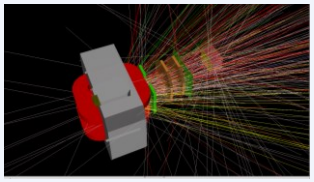
CPU: 1500 cores
HDD: 2.8 PB (EC-replicated)
SSD: 100 TB (triple replicated)
Network external & fabric: 200 Gb/s



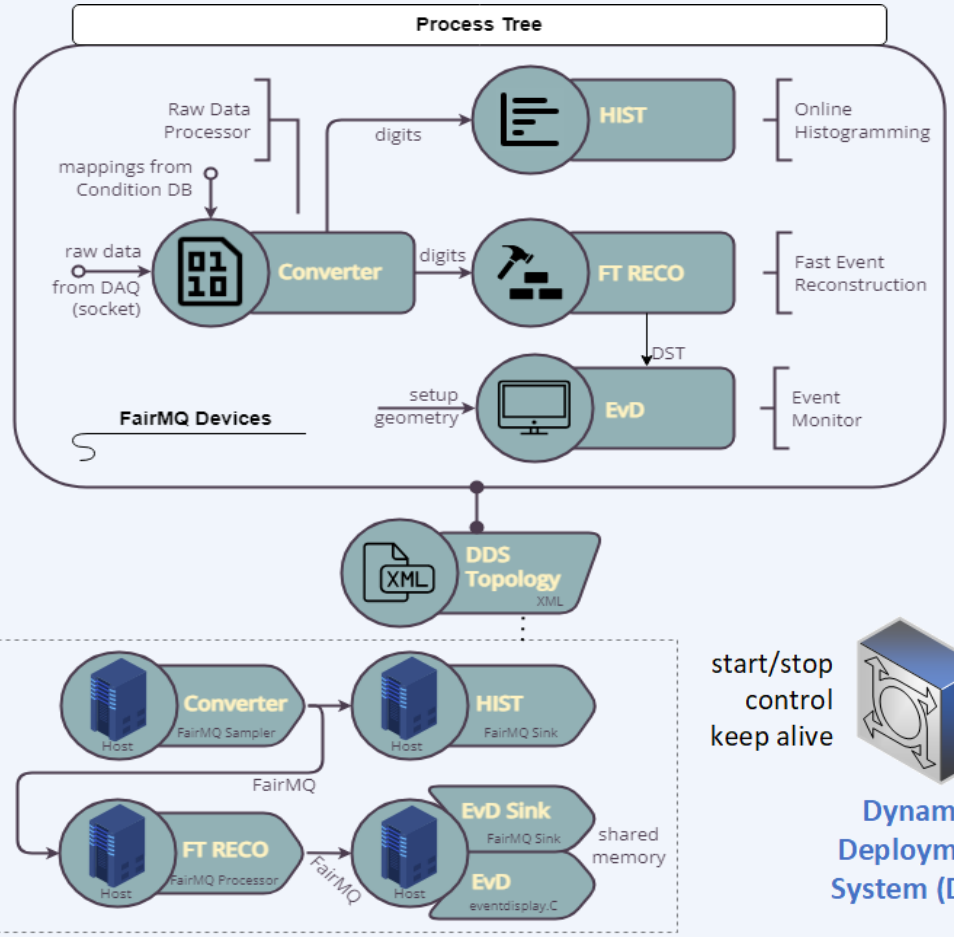
BM@N Online Data Processing

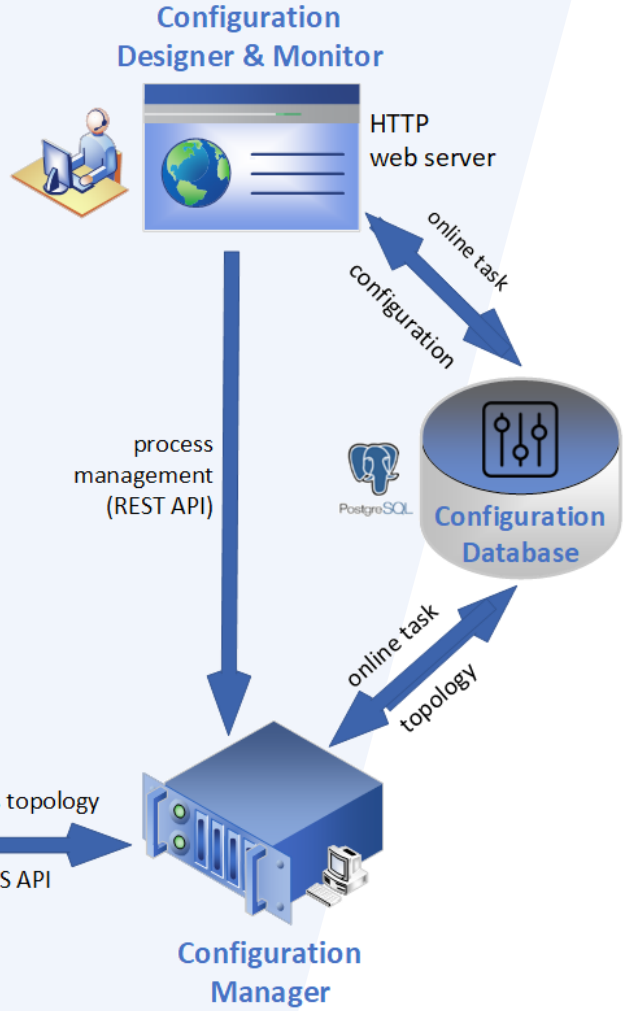


Online Histogramming System

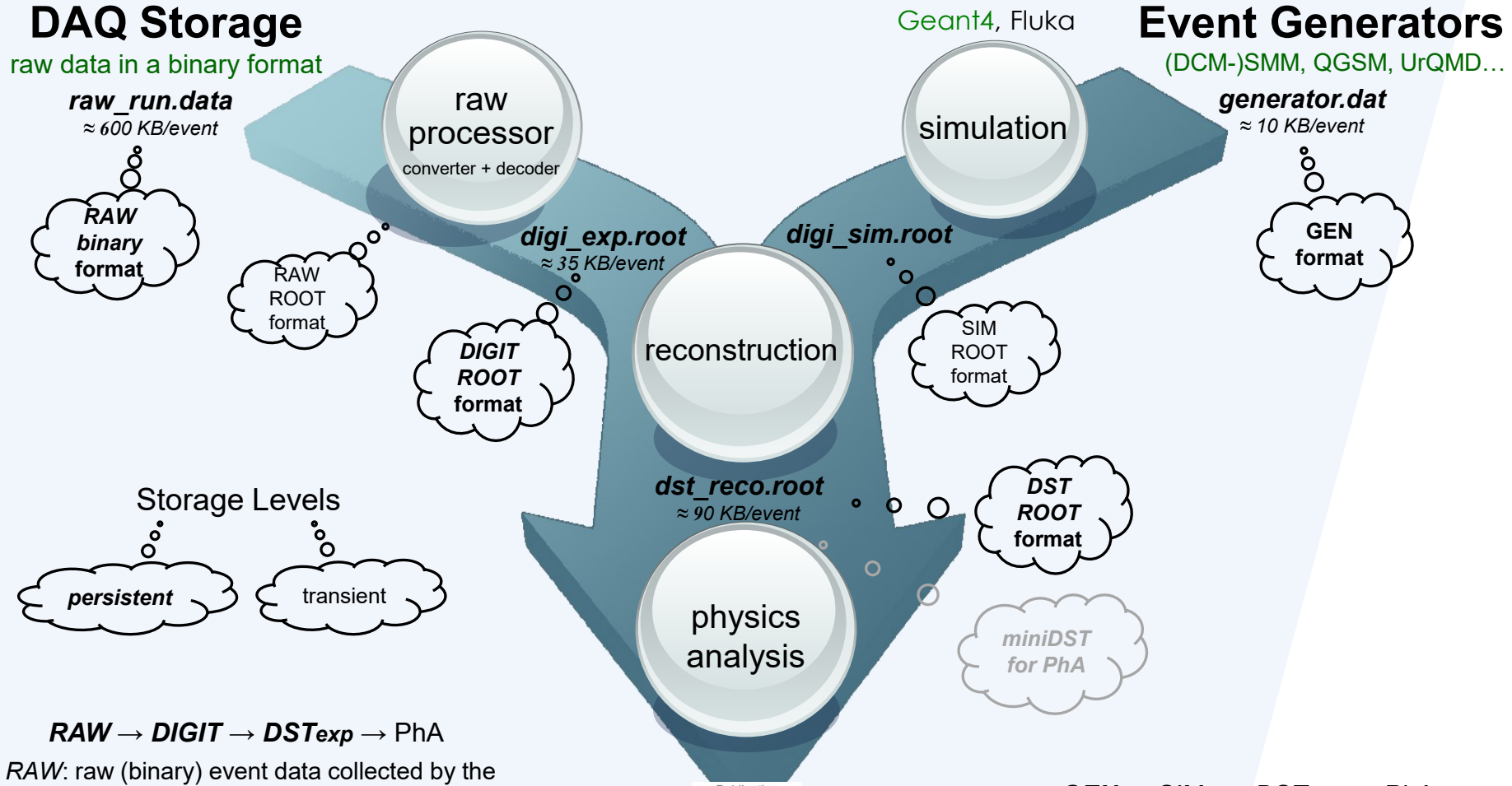


Event Monitor





Event Data Model in *BmnRoot*



RAW → ***DIGIT*** → ***DSTexp*** → PhA

RAW: raw (binary) event data collected by the DAQ system after the Event Builder

DIGIT: detector readings (event digits) after the raw data decoder (ROOT macro)

DSTexp: reconstructed data of experimental events

hists
plots
results

GEN → SIM → ***DSTsim*** → PhA

GEN: particle collisions description received by event generators

DSTsim: reconstructed data of simulated events

Components of BM@N distributed complex

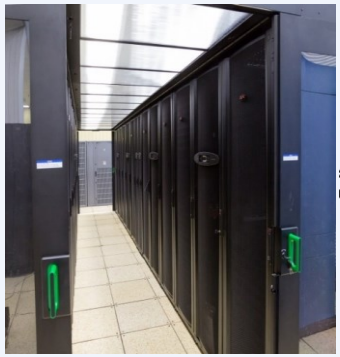
- computing platforms for the BM@N experiment
- software distribution system as a central repository of the experiment software
- data storages on distributed FS for experimental and simulated files
- file and event catalogues organizing smart namespaces with metadata
- workload management system for parallel task/job distribution
- data transfer services enabling the transfer of large amounts of data between users and storages within the federal administration
- workflow management service orchestrating task flows on data processing
- information systems based on databases providing necessary information for offline and online processing
- user interfaces (Web, API, CLI) to manage databases and distributed data processing
- central authentication and authorization system to regulate access rights
- monitoring system to control state of server nodes, databases and interfaces

Computing Platforms for BM@N

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

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

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



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

OS: CentOS / Scientific Linux 7.9

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

CEPH: 2.8 PB (*replica*)
SLURM: 1500 cores after the upgrade

EOS: 1.2 PB (*replica*)
NFS: **300 TB** (*for NICA*)
SLURM: 3000 cores (*for all NICA users*)

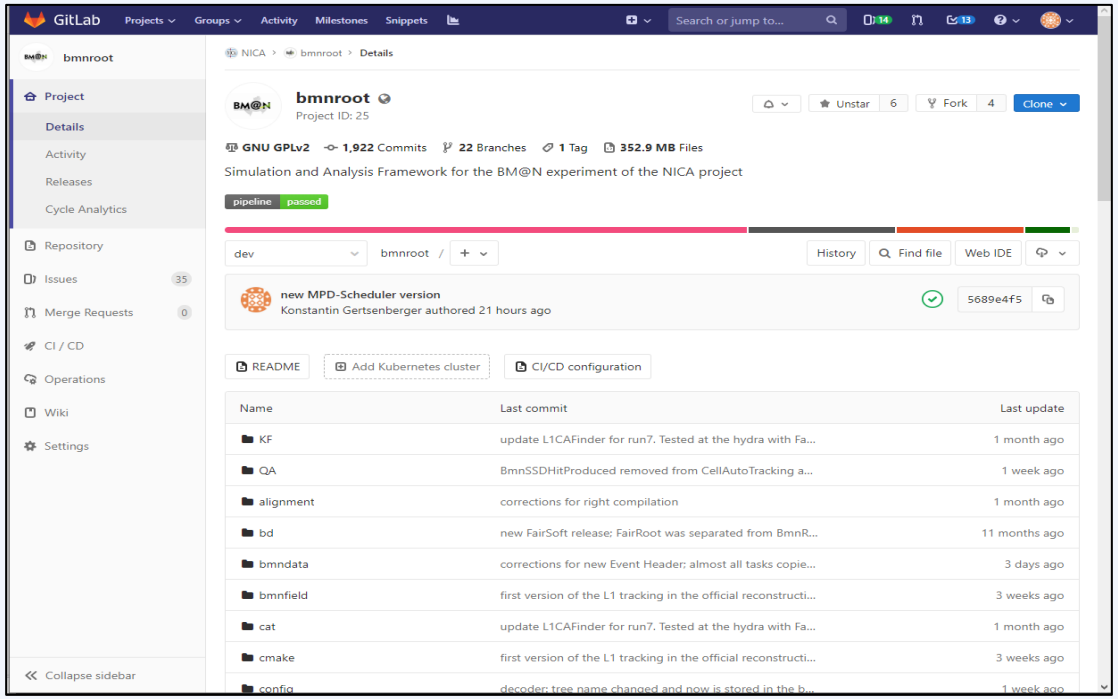
EOS: 1.2 PB (*replica*)
EOS CTA: **500 TB**
SLURM: 2500 cores (*for all NICA users*)

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

BM@N software has been installed & configured on JINR CVMFS (/bmn.jinr.ru/)

Automatic software deployment of the BmnRoot package on CVMFS with GIT CI

Software Distribution System with CVMFS



GIT: Version Control System

Repository branch protection

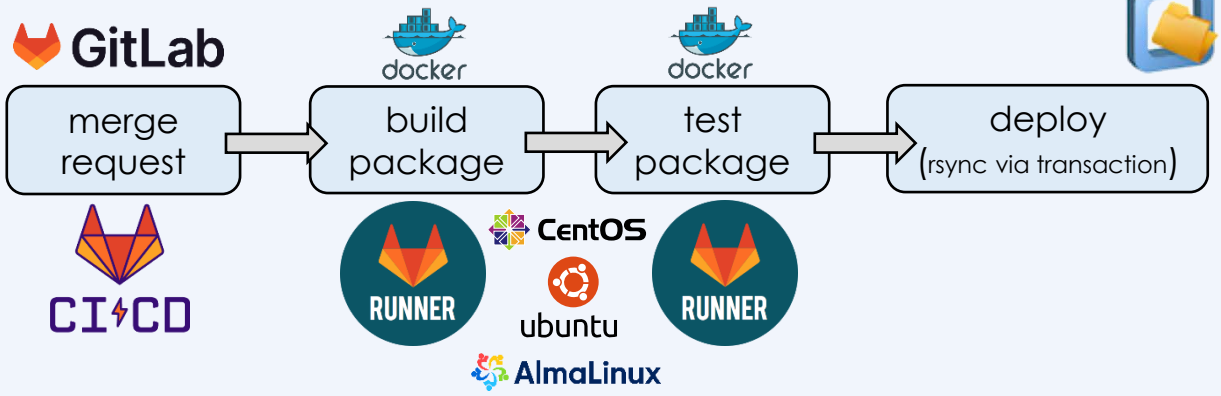
Role-based access control to projects

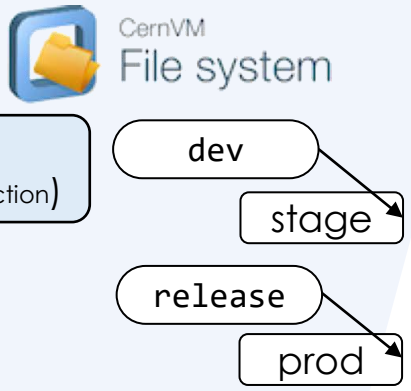
Issue Tracker *(as a Project Management System)*

Automated Tests & Deployment *(GL Runners)*

Software Distribution via CernVM File System

Read-only network file system with aggressive caching, optimized for software distribution via HTTP in a fast, scalable and reliable way






```
/cvmfs/nica.jinr.ru/
├── centos7
│   ├── fairsoft
│   ├── fairroot
│   └── bmnroot
├── ubuntu2004
│   ├── fairsoft
│   ├── fairroot
│   └── bmnroot
└── alma9
    ├── fairsoft
    ├── fairroot
    └── bmnroot
```

Data Storages for BM@N

ceph

for BM@N online

build on HDD with SSD buffer

NICA cluster

NFS

fast on NVMe SSD

MLIT CICC

no, but it is planned

HybriLIT

lustre

fast on NVMe SSD

NICA cluster

EOS

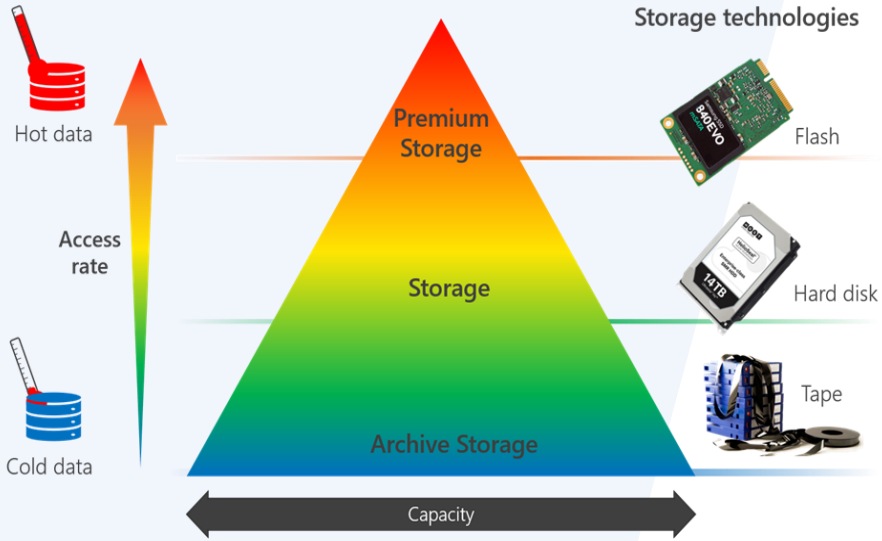
for BM@N offline

sync | duplicate

MLIT CICC

EOS

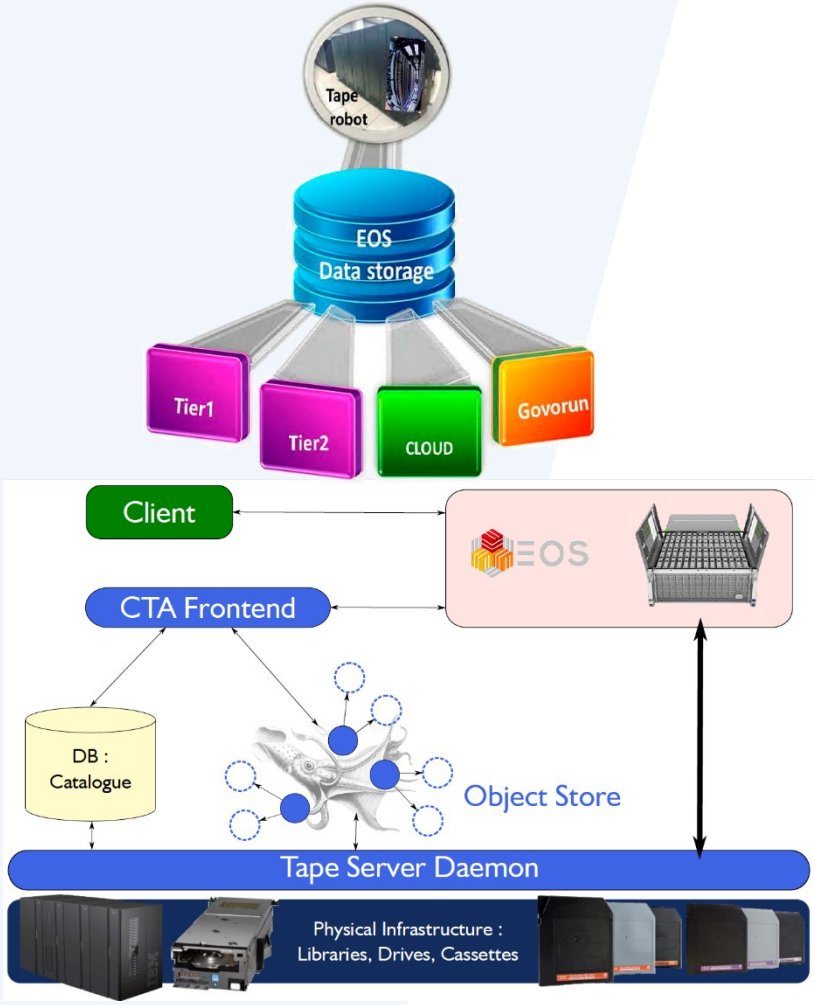
for BM@N offline



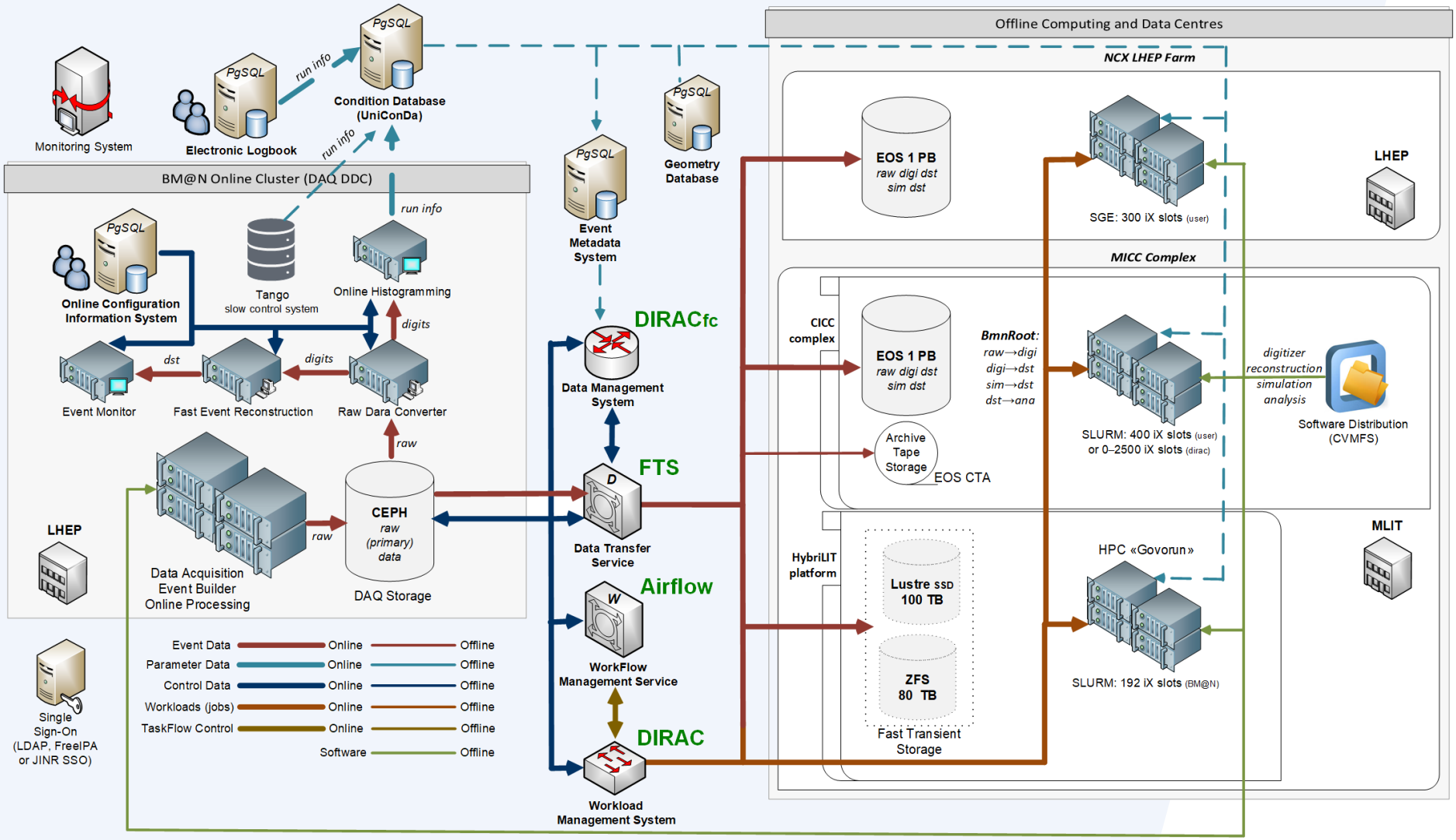
Archive Tape Storage for BM@N

EOS CTA Integration in MLIT

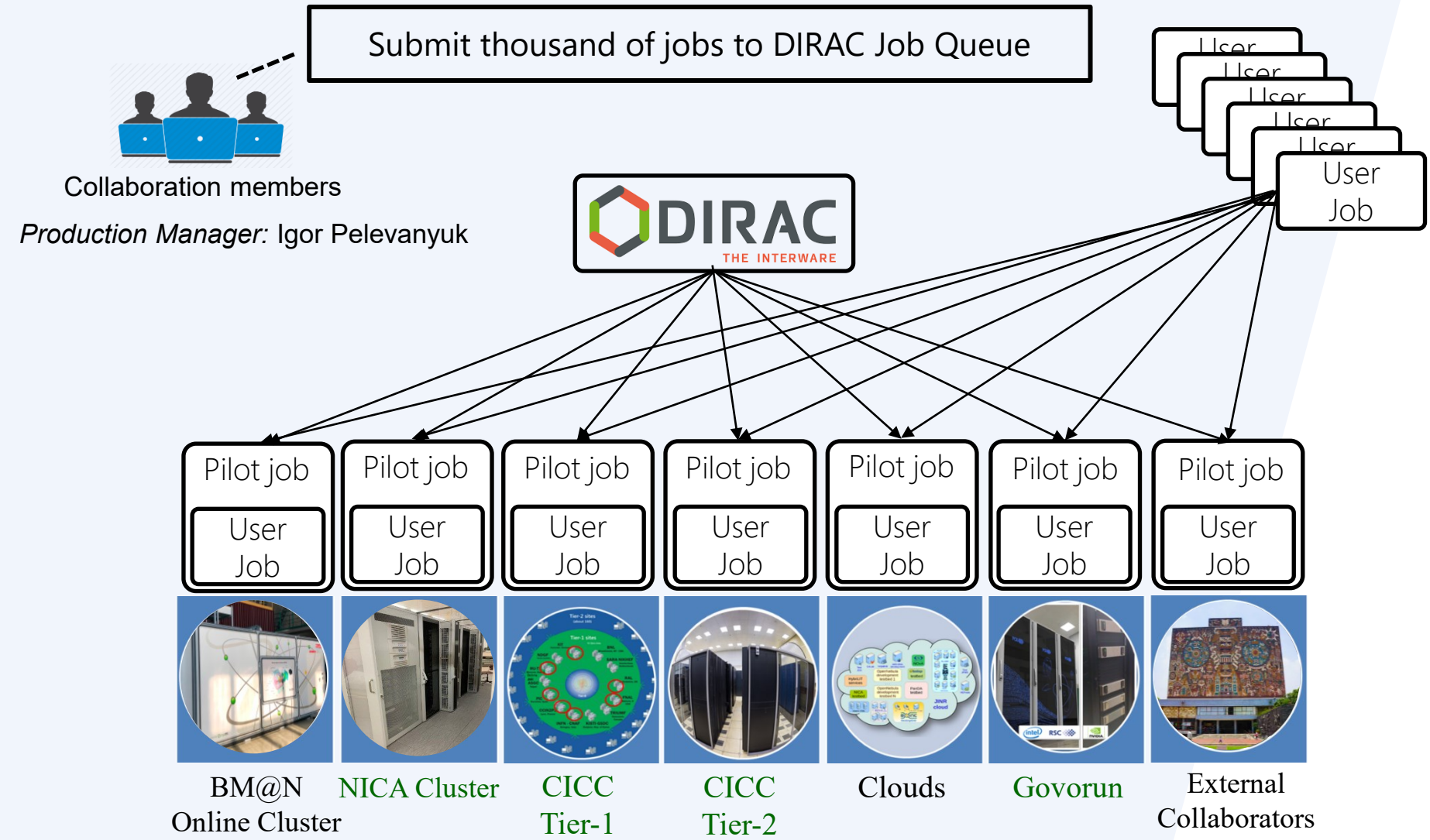
- CTA tape is a new archive solution developed at CERN to replace Castor
- Extends MLIT EOS with tape backend functionality
- Tape “bringonline” exposed via EOS and XRootD protocols
- Gfal2 XRootD plugin
- Can be handled transparently by FTS
- Advantages: long lifespan, cost of use, energy efficiency, security
- Tape robotic systems – a long-term storage for BM@N, stores *raw* and *gen* data, *online raw data backup to tapes*



BM@N Computing Software Architecture

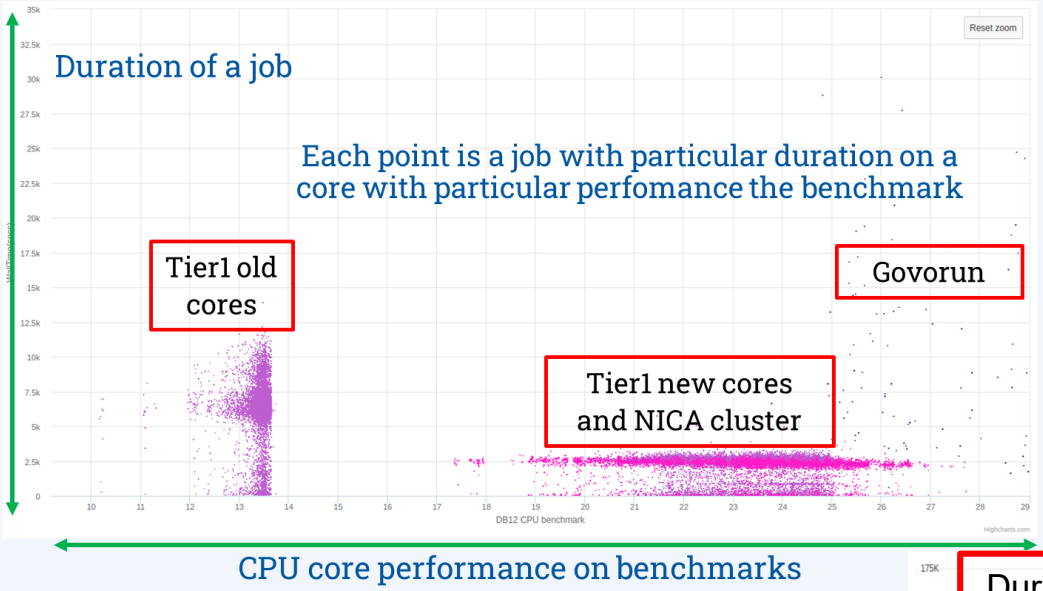


DIRAC Workload Manager for BM@N



BM@N DST Production via DIRAC (Run 8)

Duration of Raw2Digi campaign – 35 hours (0.16 s/ev)



Quotas (cores):

Tier1: 1500 (for NICA)
Tier2: 1000 (for NICA)
Govorun: 192 (BM@N)

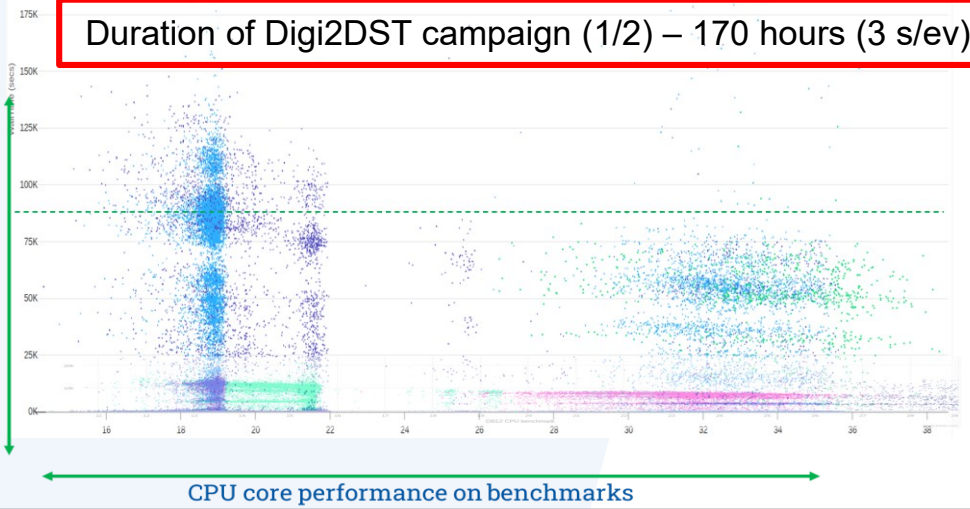
NICA cluster: 1000 (per user)
NICA cluster: max 100 slots (big files)

Total files: **30 741** Total raw size: **393 TB**
Average transfer speed (20 streams): **1.92 GB/s**
Total transfer duration: **2d 15h**
Max transfer speed (R+W) EOS@MLIT: **7.5 GB/s**

Disk usage**:** tmp file: **8 GB** result file: **800 MB**
Total disk usage per job (15 GB): **25 GB**
RAM usage: **2 GB**

Total wall time**: 70 CPU years**

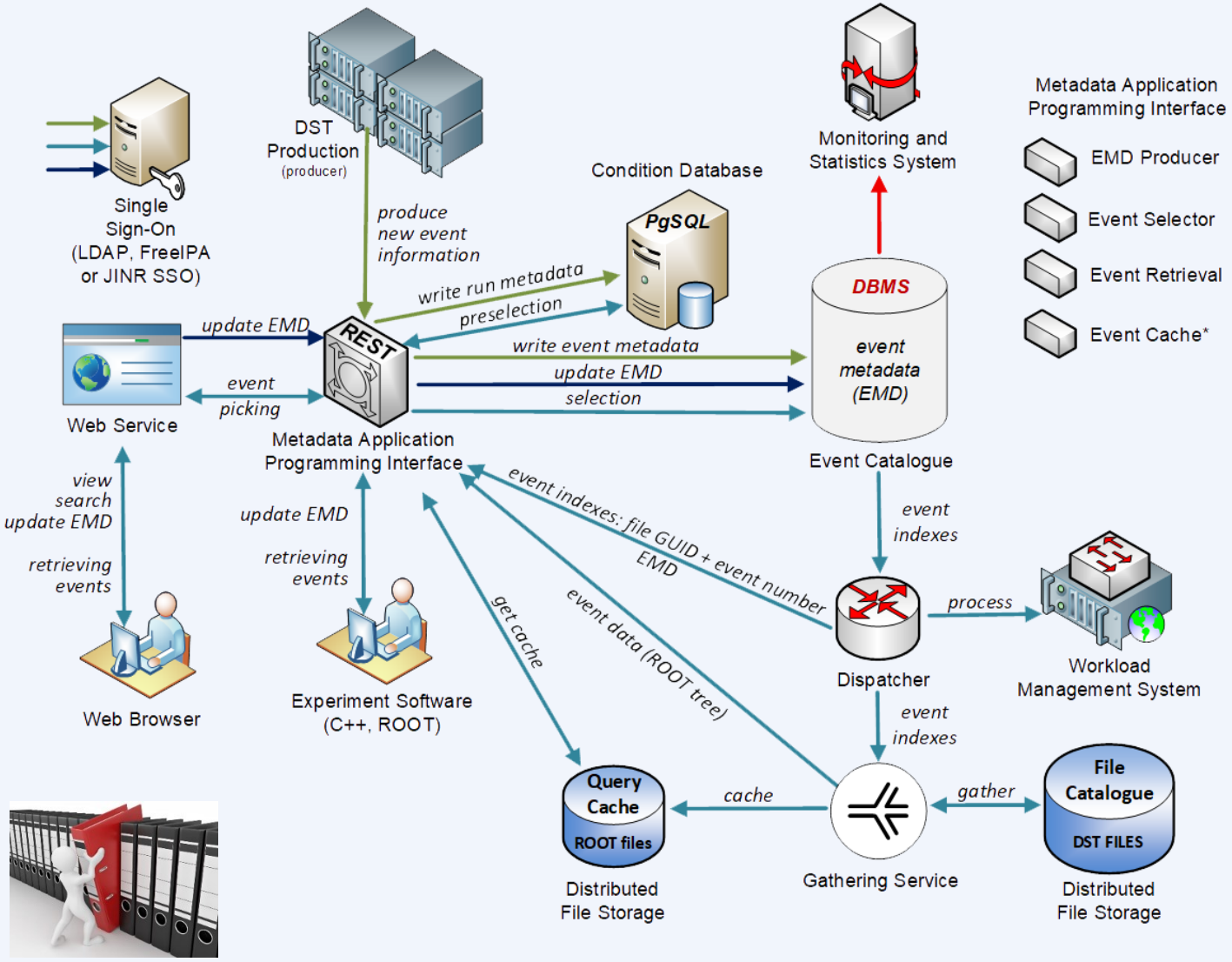
Duration of Digi2DST campaign (1/2) – 170 hours (3 s/ev)



File Catalogue Choice for BM@N

- File Catalogues map a Logical File Name (LFN) to the Physical File Name (PFN) at distributed computing platforms
- The native DIRAC File Catalog (DFC) combines both replica and metadata functionality. In the DFC metadata can be associated with any directory, and subdirectories inherit the metadata of their parents
- RUCIO is a Distributed Data Management System initially developed for the ATLAS experiment in 2014 providing file and dataset catalogue and transfers between sites and staging capabilities, policy engines, caching, bad file identification and recovery, and many other features.

BM@N Event Catalogue



Event Catalogue based on PostgreSQL

Integrated with the Condition Database

REST API and Web UI developed on Kotlin multiplatform

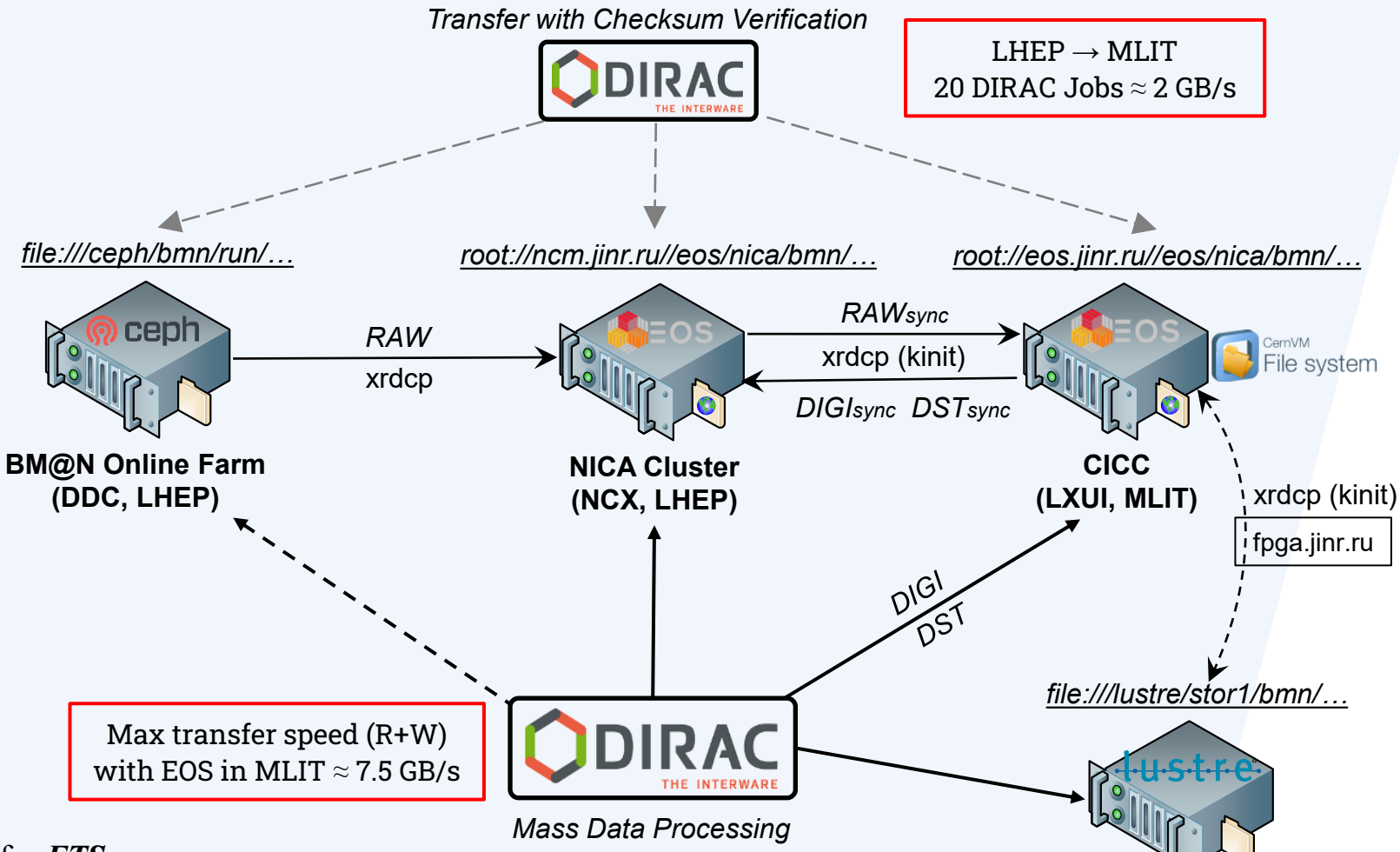
Configurable to support arbitrary metadata

Service for automatic writing new event metadata to the Catalogue

Role-based access control

Monitoring System

BM@N Data Transfer

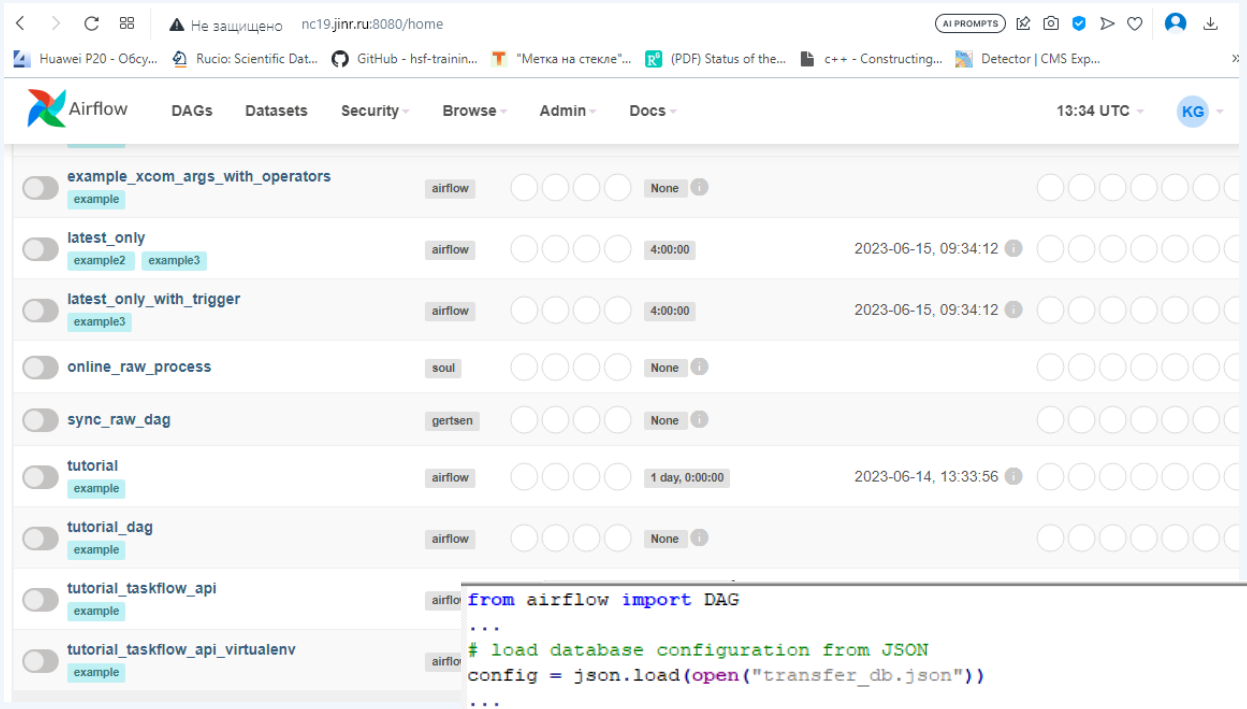


Waiting for ***FTS***

Access protocols: ***XRootD***, *EOS, FUSE*

Authentication protocol: ***Kerberos***

First steps in BM@N Workflow Management



Airflow deployed on the NC-farm

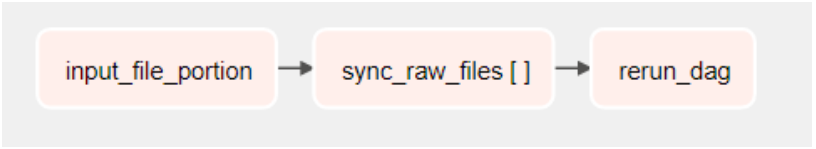
Used for BM@N Run 8 to transfer raw data emerging on the NICA-cluster to the LIT EOS storage and to check the integrity of the source and destination files

To be employed for managing online (for emerging raw data files) and offline data production via DIRAC

Apache Airflow

MC simulation pipeline
event filtering digitizing
reconstruction analysis
…

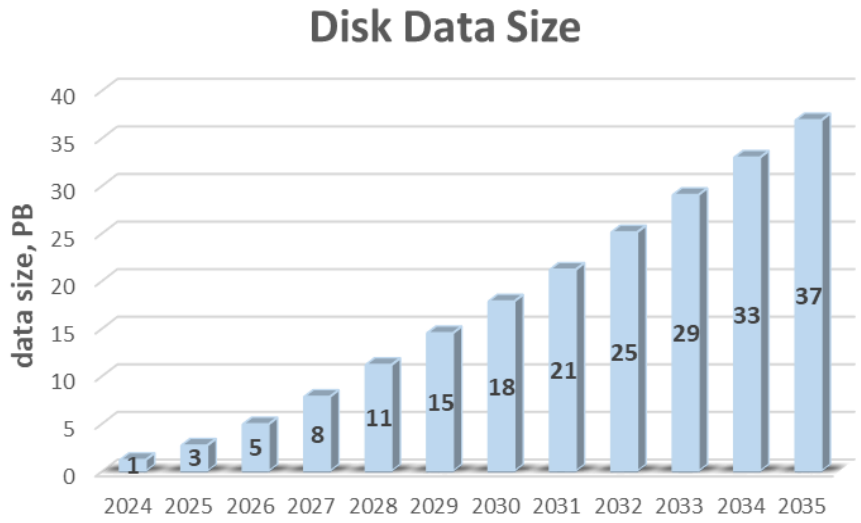
```python
from airflow import DAG
...
# load database configuration from JSON
config = json.load(open("transfer_db.json"))
...
with DAG('sync_raw_dag', description='This DAG is for copying new raw data files from an inpput directory to LIT EOS',
         default_args=default_args, schedule_interval=None, catchup=False, max_active_runs=1) as dag:

    @task
    def input_file_portion():
        ...
        return process_list

    @task(max_active_tis_per_dag=8)
    def sync_raw_files(input_file_path):
        ...

    trigger = TriggerDagRunOperator(task_id='rerun_dag',
                                    trigger_dag_id="sync_raw_dag")

    sync_raw_files.expand(input_file_path=input_file_portion()) >> trigger
```

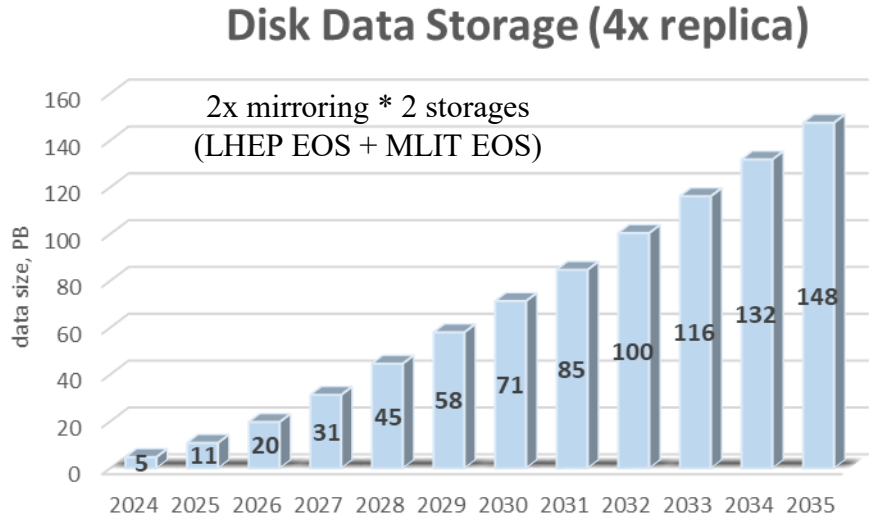


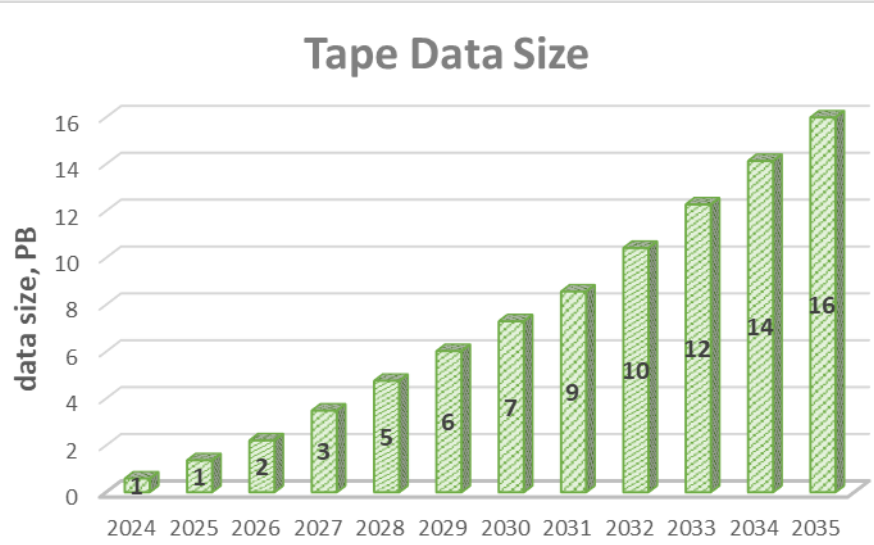
BM@N Resource Prospect for 2024-2035

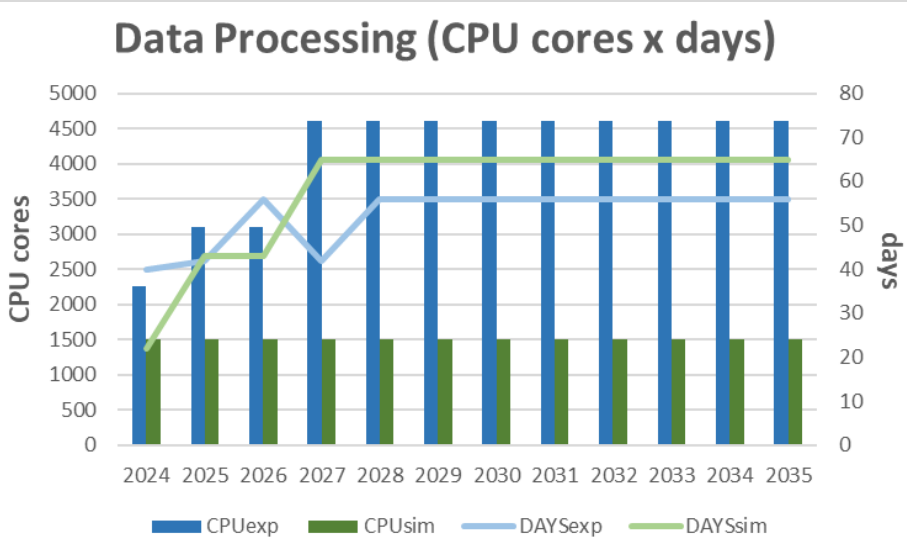
BM@N	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035
Production, days	5	30	30	45	45	45	45	45	45	45	45	45
Event rate, event/sec	3 636	3 636	3 636	3 636	3 636	3 636	3 636	3 636	3 636	3 636	3 636	3 636
Duty factor	18%	25%	25%	25%	25%	25%	25%	25%	25%	25%	25%	25%
RAW event count, M	**140**	**1200**	**1200**	**1800**	**1800**	**1800**	**1800**	**1800**	**1800**	**1800**	**1800**	**1800**
RAW event size, KB	580	640	700	700	700	700	700	700	1024	1024	1024	1024
RAW data size, TB	83	790	860	1290	1290	1290	1290	1290	1890	1890	1890	1890
REC event size, KB	50|100	55|110	90|180	90|180	90|180	90|180	90|180	90|180	90|180	90|180	90|180	90|180
REC production times	3c+1н	3	4	3	4	4	4	4	4	4	4	4
REC data size, TB	284	553	1207	1358	1810	1810	1810	1810	1810	1810	1810	1810
EXP full storage (4x), PB	**4.0**	**9.3**	**17**	**28**	**40**	**52**	**64**	**76**	**91**	**105**	**120**	**134**
SIM event size, KB	20|550	20|550	20|550	20|550	20|550	20|550	20|550	20|550	20|550	20|550	20|550	20|550
SIM event count, M	200	400	400	600	600	600	600	600	600	600	600	600
SIM data size, TB	106	212	212	319	319	319	319	319	319	319	319	319
SIM full storage (4x), PB	**0.7**	**1.5**	**2.4**	**3.6**	**4.9**	**6.1**	**7.4**	**8.6**	**9.8**	**11.1**	**12.3**	**13.6**
Disk Data Size, PB	**1.2**	**2.7**	**4.9**	**7.8**	**11.2**	**14.5**	**17.8**	**21.2**	**25.1**	**29.0**	**33.0**	**36.9**
Disk Storage (4x), PB	4.7	11	20	31	45	58	71	85	100	116	132	148
Tape Data Size, PB	**0,6**	**1,4**	**2,2**	**3,5**	**4,8**	**6,0**	**7,3**	**8,6**	**10,4**	**12,3**	**14,1**	**16,0**
EXP processing, sec	0.3+2.8	0.3+2.8	0.3+2.8	0.3+2.8	0.3+2.8	0.3+2.8	0.3+2.8	0.3+2.8	0.3+2.8	0.3+2.8	0.3+2.8	0.3+2.8
EXP cores*days	**2260*40**	**3100*42**	**3100*56**	**4600*42**	**4600*56**	**4600*56**	**4600*56**	**4600*56**	**4600*56**	**4600*56**	**4600*56**	**4600*56**
SIM processing, sec	9 + 5	9 + 5	9 + 5	9 + 5	9 + 5	9 + 5	9 + 5	9 + 5	9 + 5	9 + 5	9 + 5	9 + 5
SIM cores*days	**1500*22**	**1500*43**	**1500*43**	**1500*65**	**1500*65**	**1500*65**	**1500*65**	**1500*65**	**1500*65**	**1500*65**	**1500*65**	**1500*65**
CPU years (x512GFlops)	**340**	**530**	**650**	**800**	**970**	**970**	**970**	**970**	**970**	**970**	**970**	**970**

BM@N Resource Prospect for 2024-2035









Thank you for your attention!

***Director*: S. V. SHMATOV. *Scientific Leader*: V. V. KORENKOV**

JINR MLIT Contribution to BM@N

Igor ALEXANDROV, Evgeniy ALEXANDROV, Irina FILOZOVA, et alia
Development of the Geometry Database and Online Configuration Systems

Nikita BALASHOV:
CVMFS Deployment, GitLab Services, Docker Containers

Igor PELEVANYUK:
DIRAC workload management system and BM@N mass production

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

Daria PRIAKHINA, Vladimir TROFIMOV
Modelling System for BM@N computing infrastructure

Zarif SHARIPOV, Zafar TUKHLIEV
Automation of BM@N Alignment

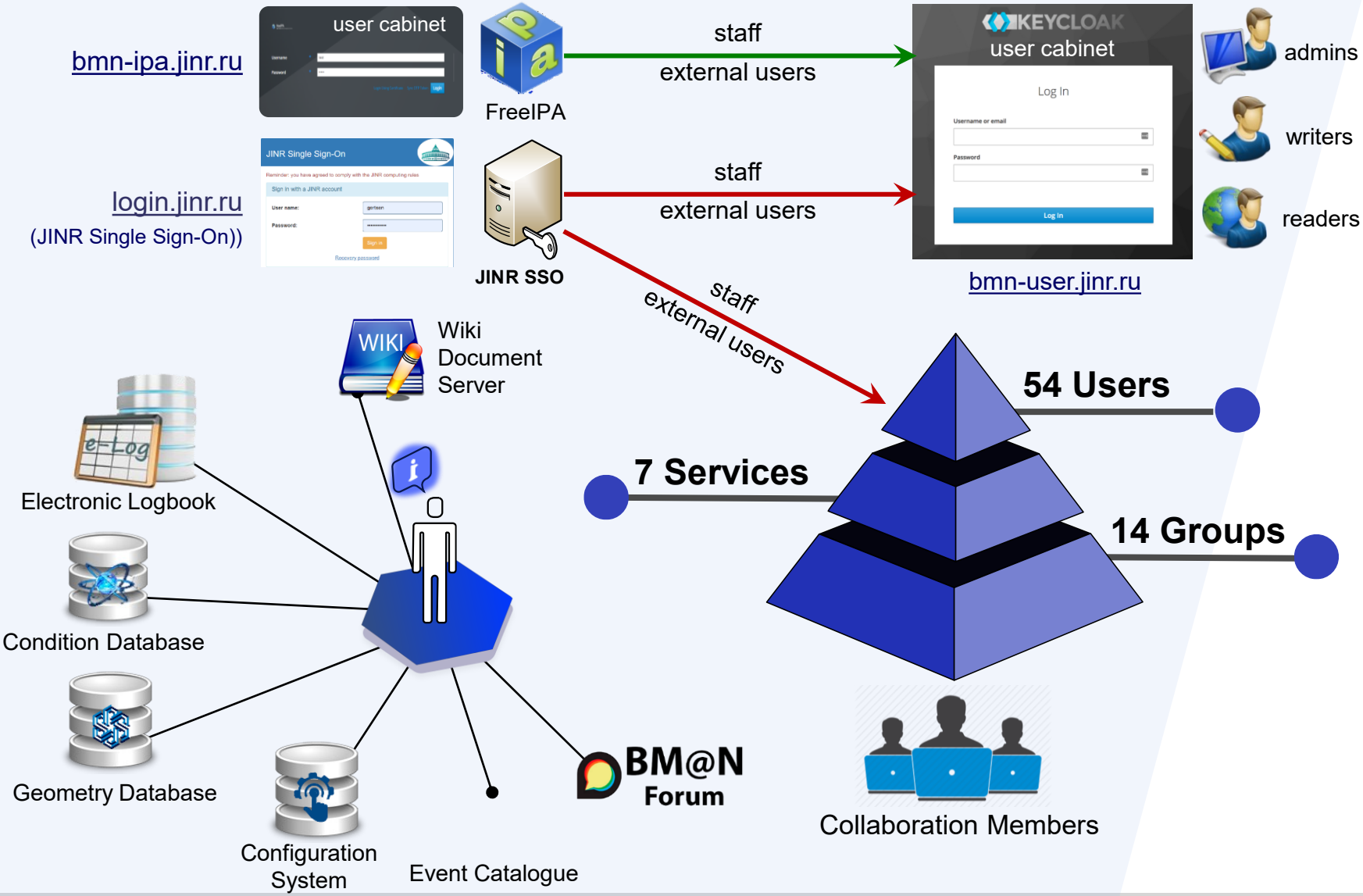
Alexander AYRIYAN, Vladimir PAPOYAN
Implementation of BM@N PID based on ML

thanks to the DDC, CICC, NCX & HybriLIT teams for computing support

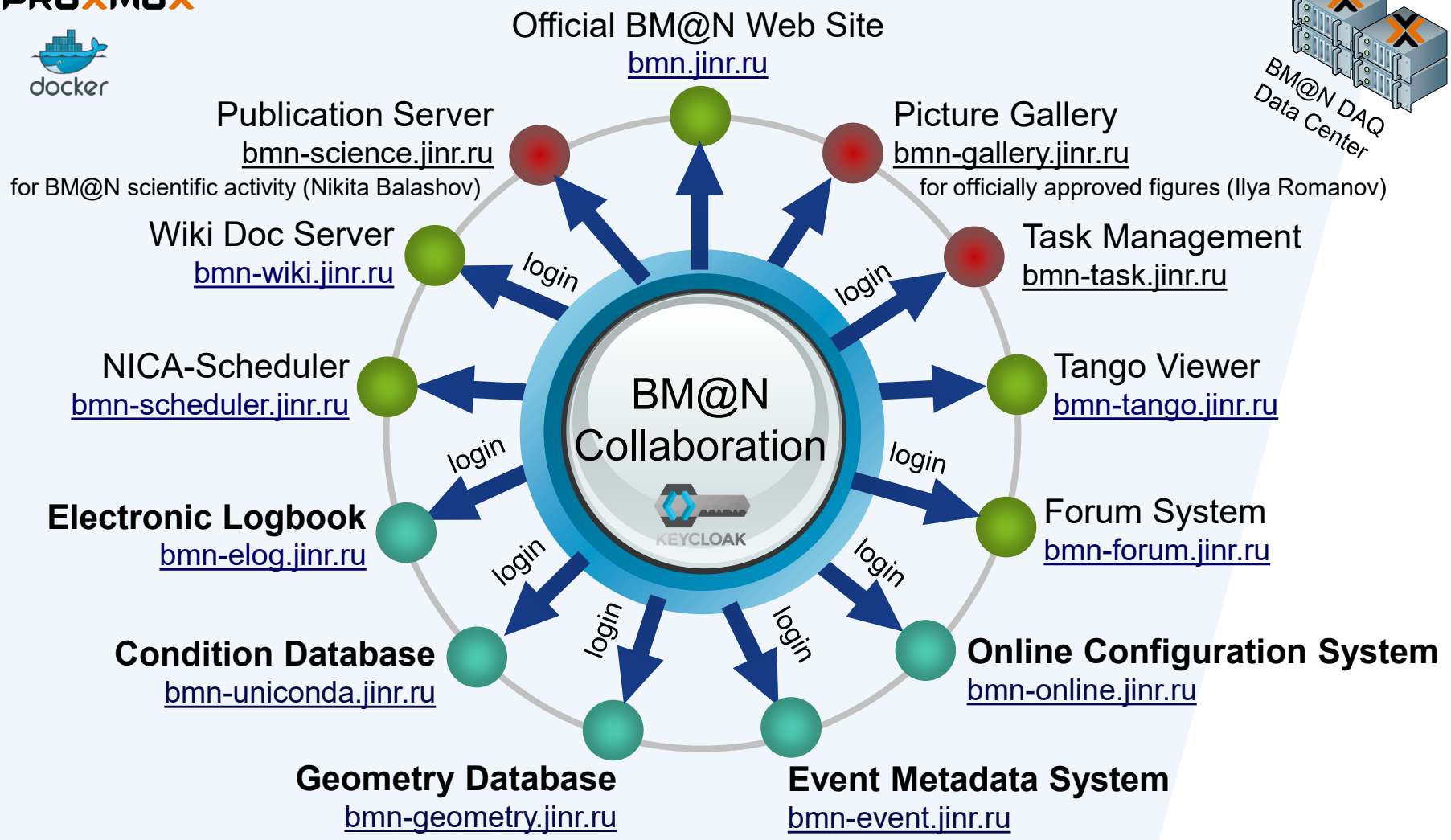
contact email: *gertsen@jinr.ru*

BACKUP

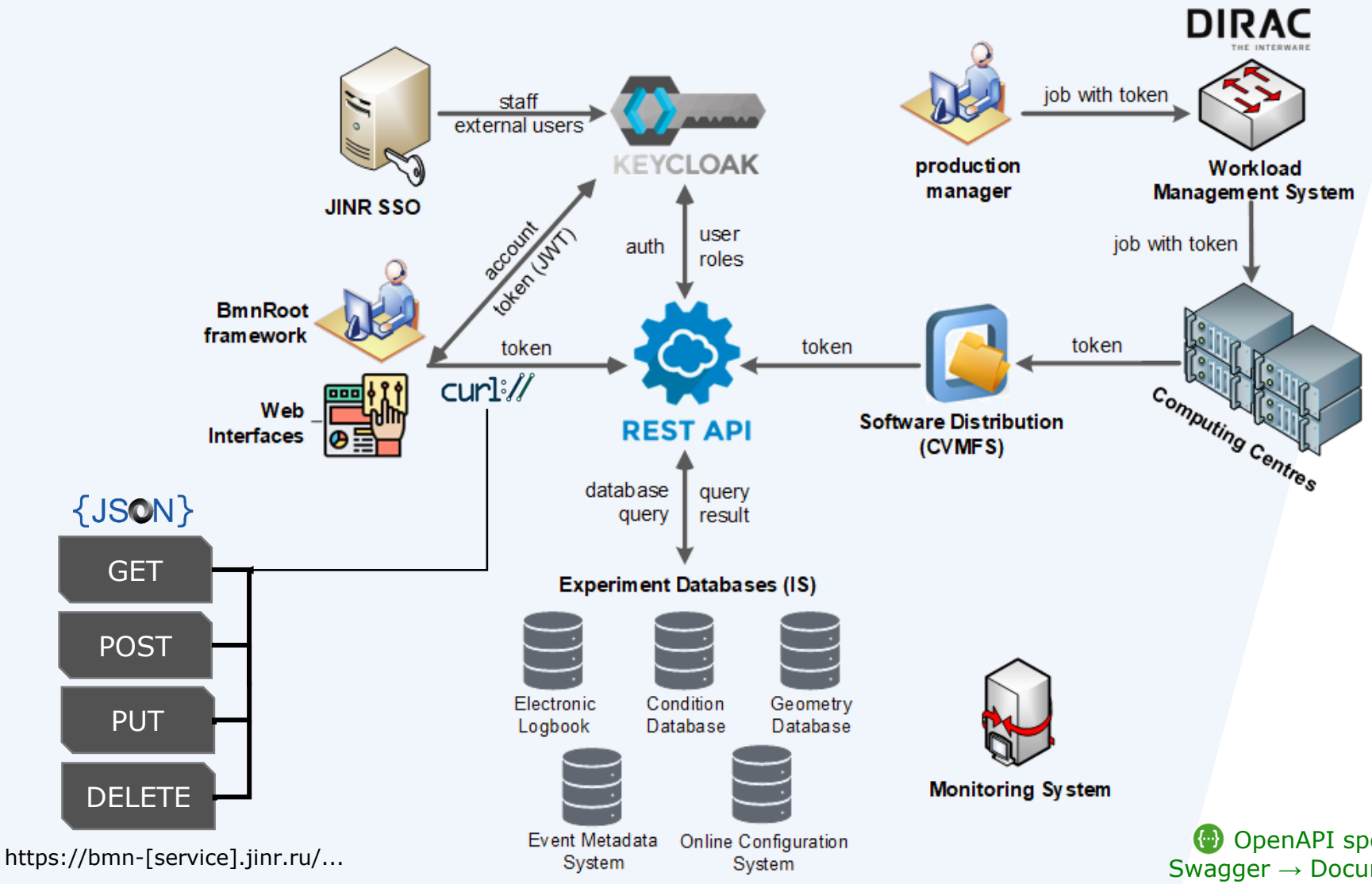
Migration FreeIPA → Keycloak → JINR SSO



BM@N Software Ecosystem



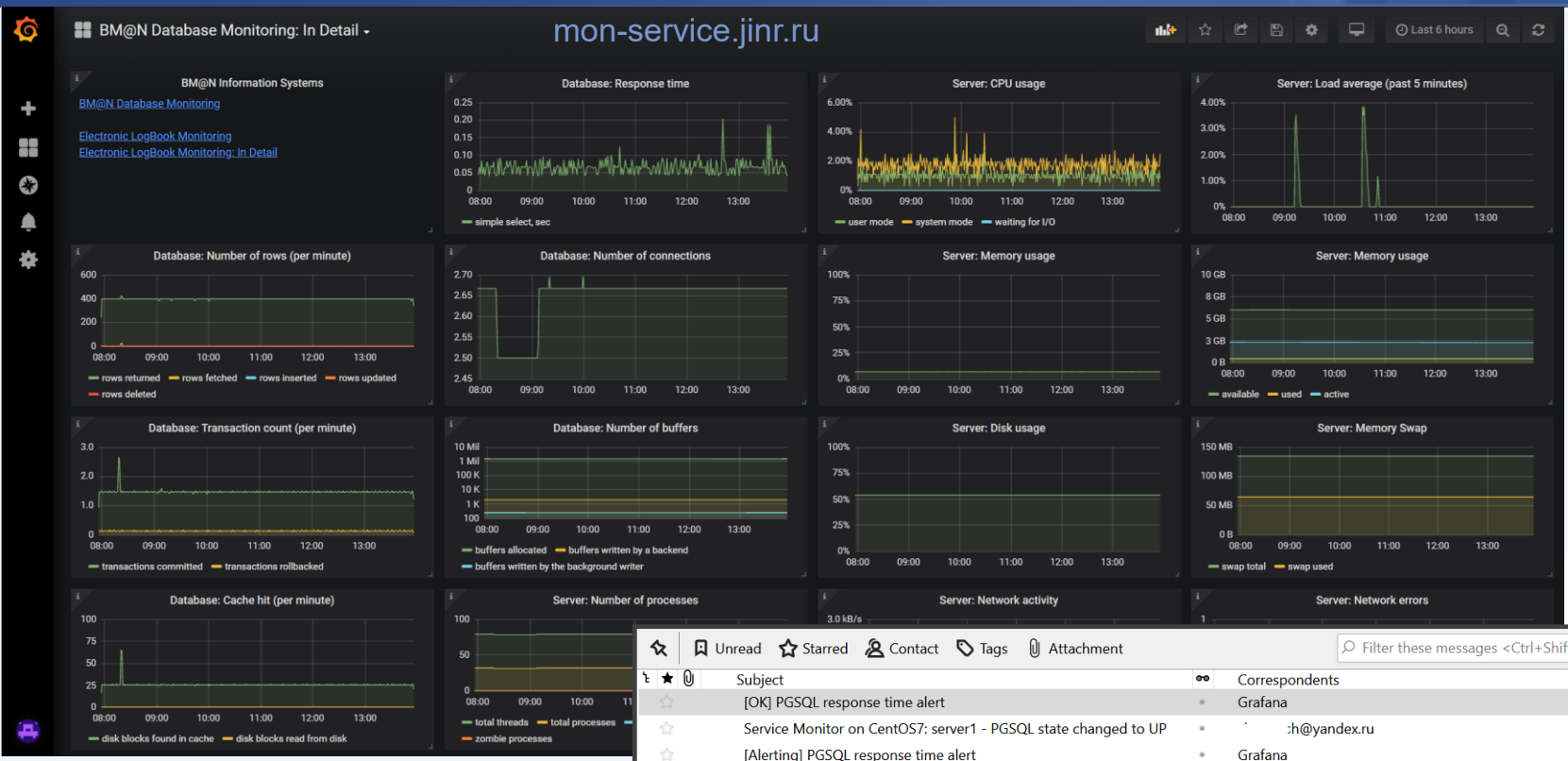
REST APIs for BM@N Information Systems



https://bmn-[service].jinr.ru/...

OpenAPI specification
Swagger → Documentation

Monitoring System for BM@N software complex



Telegraf

influxdb

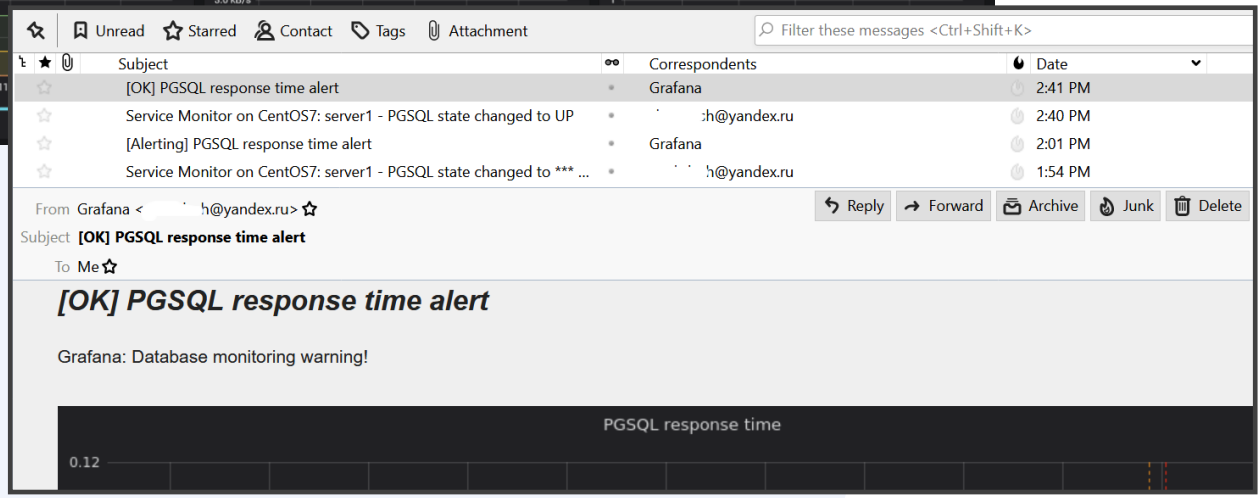
Grafana

- hosts
- databases
- web-sites

- Condition Database
simple or detailed visualization
- Electronic Logbook
simple or detailed visualization

...

Email Alerting



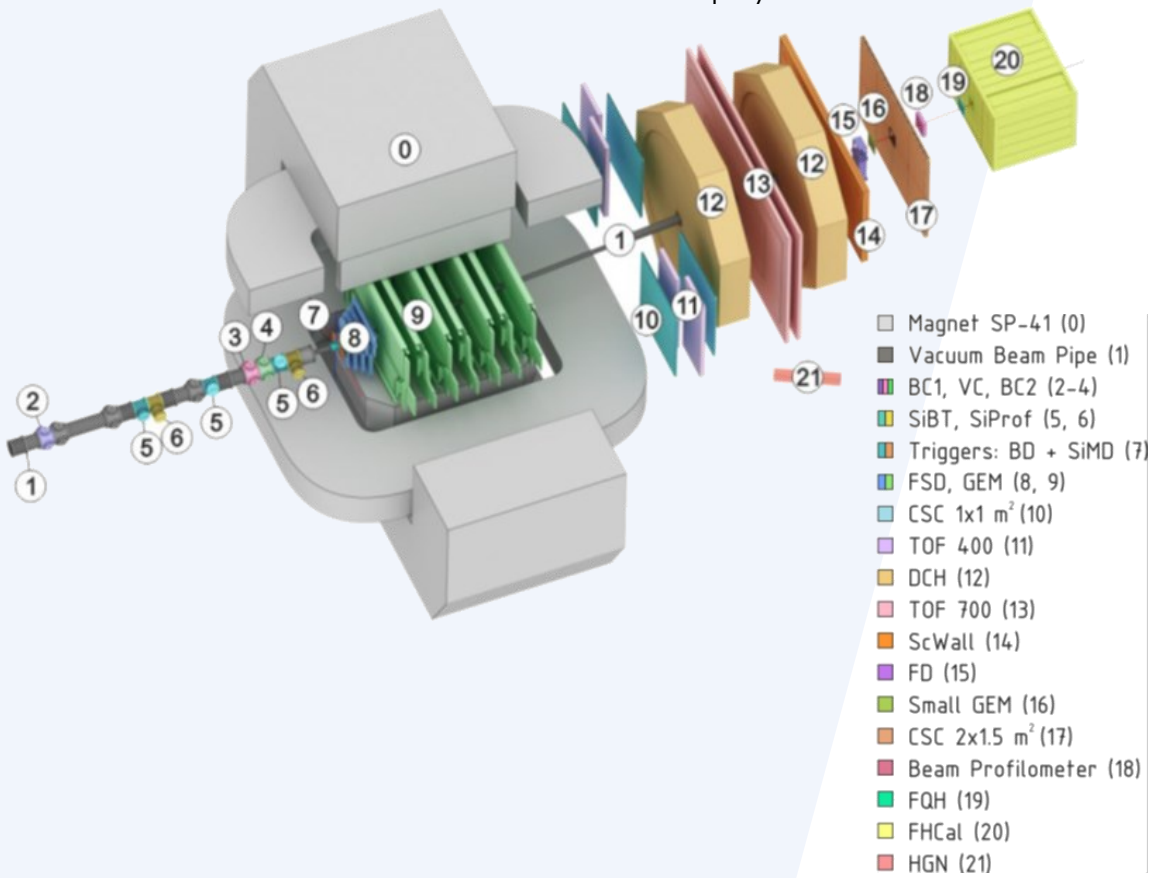
BmnRoot Framework

The BmnRoot framework is developed for realistic event simulation, reconstruction of experimental or simulated data and following physics analysis of ion collisions with a fixed target at the BM@N facility.

single stack for online and offline (FairMQ)

C++ classes, Linux/MacOS, based on ROOT Data Analysis Framework and FairRoot embedded services on Python

BM@N in the 1st physics Run



The BmnRoot software is available in GitLab@JINR: https://git.jinr.ru/nica/bmnroot