GRID 2021
July 5-9 Dubna
9th International Conference
Distributed Computing and Grid-technologies in Science and Education

Development of the Condition Database for the experiments of the NICA complex

Konstantin Gertsenberger, Alexander Chebotov,
Igor Alexandrov, Irina Filozova, Evgeny Alexandrov

Joint Institute for Nuclear Research, Dubna, Russia

Supported by RFBR Grant №18-02-40125

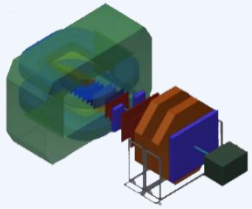
NICA

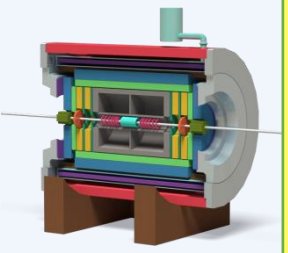
JOINT INSTITUTE FOR NUCLEAR RESEARCH 1956

Nuclotron-based Ion Collider fAcility

- Beams: from $p, d^{\uparrow}$ to Au^{79+}
- Luminosity: 10^{27} (Au^{79+}), 10^{32} (p) $cm^{-2}s^{-1}$
- Collision energy: $\sqrt{S_{NN}}_{AU}$ = *4 – 11 Gev* E_{lab} = *1 – 6 AGev*
- Fixed target experiment: BM@N (2018)
- 2 interaction points: MPD (2023) & SPD (>2025)
- Official site: *nica.jinr.ru*

Data Processing Pipeline





DRE

LDC

FLP

First Level Processor

Data Check
Flow Control
Formatting

EvB

Event Builder

Buffering
Sorting
Distribution

HLT

High Level L3 Trigger

RQ

Raw Data Quality Check

TDS

Transient Data Storage

PDS

Permanent Data Storage

FT REC

Fast Event Reconstruction

HIST

Online Histograms

EvM

Event Monitor

Digitizer

ROOT Format
Mapping
Alignment

REC

Event Reconstruction

PA

Physics Analysis

frameworks **@ NICA distributed systems**

IS on databases

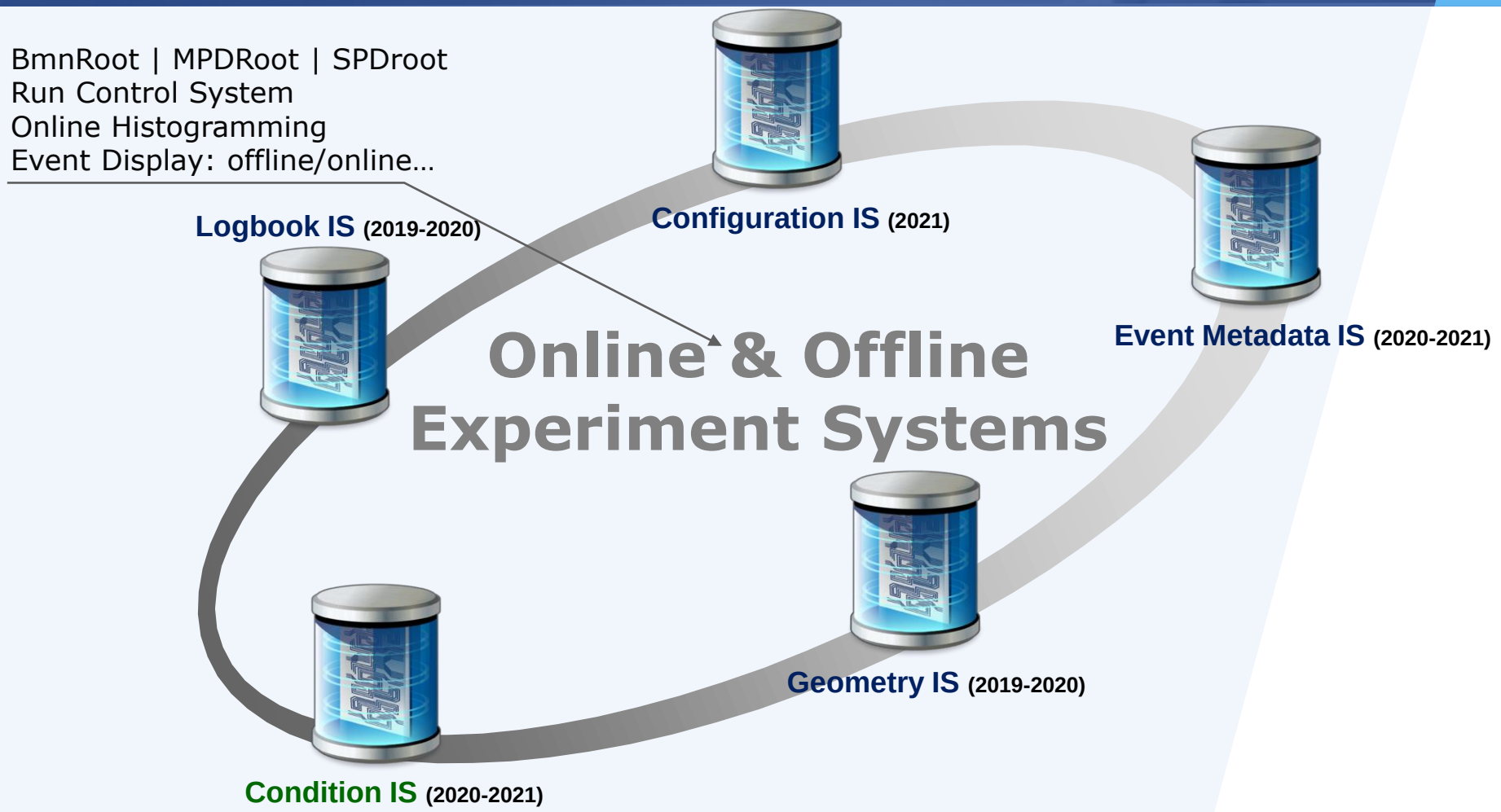
DAQ

Online Processing

Offline Processing

matched from “Data Acquisition System TDR”, DAQ Collaboration

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

Multiple formats of the BM@N parameters (2015)

~1000 raw files, ~2000 ROOT files, various parameter data

#	File name	Events	Start Time	End Time	Average Rate	BField	detales
1	run688.data	49766	2015-03-15 08:05:39	2015-03-15 08:24:28	44.08	0.05	detales
2	run687.data	3643	2015-03-15 07:56:49	2015-03-15 08:04:39	7.75	0.06	detales
3	run686.data	744	2015-03-15 07:49:25	2015-03-15 07:56:16	1.81	0.07	detales
4	run685.data	6599	2015-03-15 07:39:01	2015-03-15 07:45:37	16.66	55.30	detales
5	run684.data	303520	2015-03-15 07:11:16	2015-03-15 07:38:37	184.96	59.79	detales
6	run682.data	10796	2015-03-15 06:58:37	2015-03-15 06:59:02	431.84	59.80	detales
7	run681.data	49976	2015-03-15 06:55:17	2015-03-15 06:58:03	301.06	59.78	detales
8	run680.data	61053	2015-03-15 06:51:37	2015-03-15 06:55:05	293.52	59.78	detales
9	run675.data	9633	2015-03-15 06:30:37	2015-03-15 06:30:44	1376.14	59.80	detales
10	run672.data	59162	2015-03-15 03:44:00	2015-03-15 03:48:14	232.92	59.75	detales
11	run651.data	59161	2015-03-15 03:44:00	2015-03-15 03:48:14	232.92	59.75	detales
12	run650.data	500572	2015-03-15 02:41:14	2015-03-15 03:36:26	151.14	59.72	detales
13	run649.data	517970	2015-03-15 01:47:00	2015-03-15 02:40:26	161.56	59.73	detales
14	run648.data	540761	2015-03-15 00:49:47	2015-03-15 01:45:38	161.37	59.71	detales
15	run647.data	524991	2015-03-15 00:04:47	2015-03-15 00:48:37	199.62	59.70	detales
16	run646.data	508602	2015-03-14 23:17:23	2015-03-14 23:59:13	202.63	59.70	detales
17	run645.data	378469	2015-03-14 22:25:44	2015-03-14 23:10:37	140.54	59.69	detales
18	run644.data	512338	2015-03-14 20:26:54	2015-03-14 22:01:55	89.87	59.63	detales
19	run643.data	505324	2015-03-14 19:34:33	2015-03-14 20:26:36	161.81	59.61	detales
20	run642.data	454358	2015-03-14 18:48:03	2015-03-14 19:34:15	163.91	59.57	detales
21	run641.data	1765	2015-03-14 18:47:43	2015-03-14 18:47:45	882.50	0.00	detales

HTML

Target	Field Curr., A	Field Fact.	File size
Cu	900	1,00	113 M
Cu	900	1,00	289 M
-	600	0,67	221 M
-	800	0,89	223 M
-	900	1,00	226 M
-	1000	1,11	231 M
-	1100	1,22	223 M
-	1200	1,33	217 M
-	1100	1,22	218 M
-	1100	1,22	244 M
-	1100	1,22	158 M
-	1100	1,22	196 M
-	1000	1,11	211 M
-	1000	1,11	210 M
-	900	1,00	326 M
Cu	900	1,00	194 K
Cu	1100	1,22	324 M
Cu	1100	1,22	249 K
Cu	1100	1,22	127 M

258	d	-
262	d	Cu
266	d	Cu
270	d	Cu
271	d	Cu

Excel

e.g. noise channels

ise_run1_2.txt

```
run       row
slot      chan.

077        2
 15       33-48
 16       49-64

133        1
 16       49-64

151        2
 15       33-48
 16       49-64

161        1
 16       49-64

172        2
  9       40 41
 10       57
```

Ln : Dos\Windows UTF-8 INS

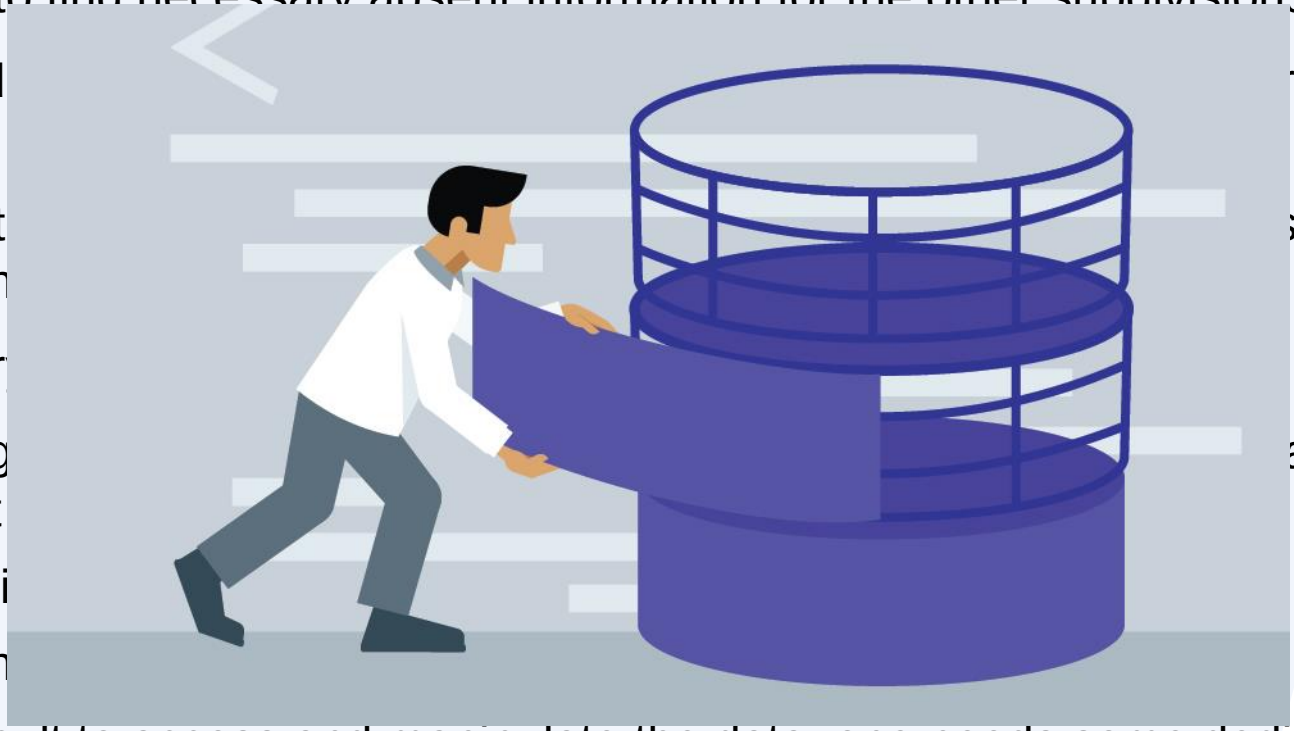
Text

Technical Run 1

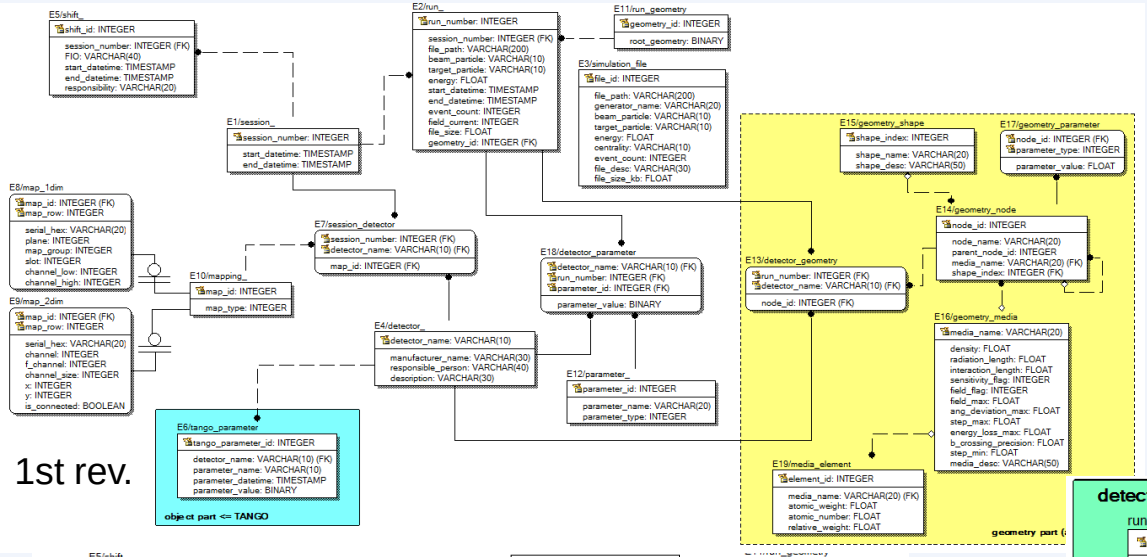
Why should a database system be used?

The lacks of the used file storing approach:

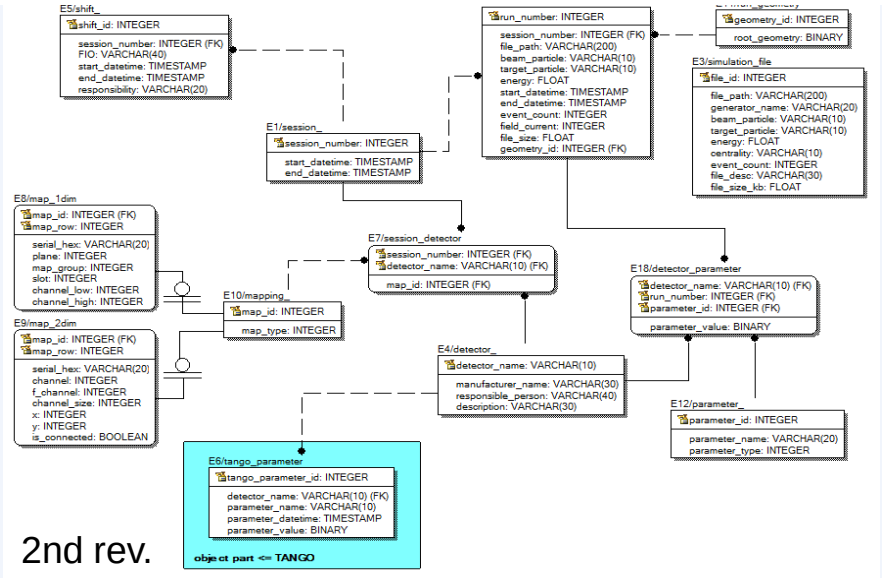
- parameter data of the experiment is distributed between many subdivisions → it is difficult to find necessary absent information for the other subdivisions
- required
- paramet
- geometr
- the usag
- sequenti
- no mech
- it is difficult to access and manipulate the data: one needs some dedicated programs
- uncontrolled concurrent access by multiple users often leads to inconsistencies



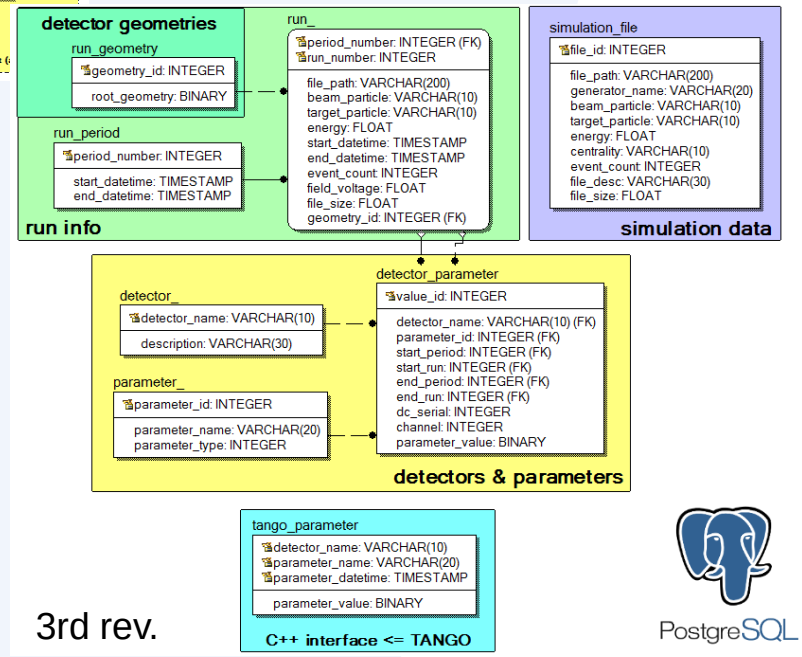
Evolution of the Unified Database (2016-2019)



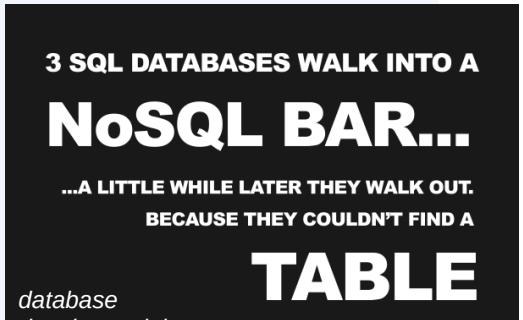
1st rev.



2nd rev.



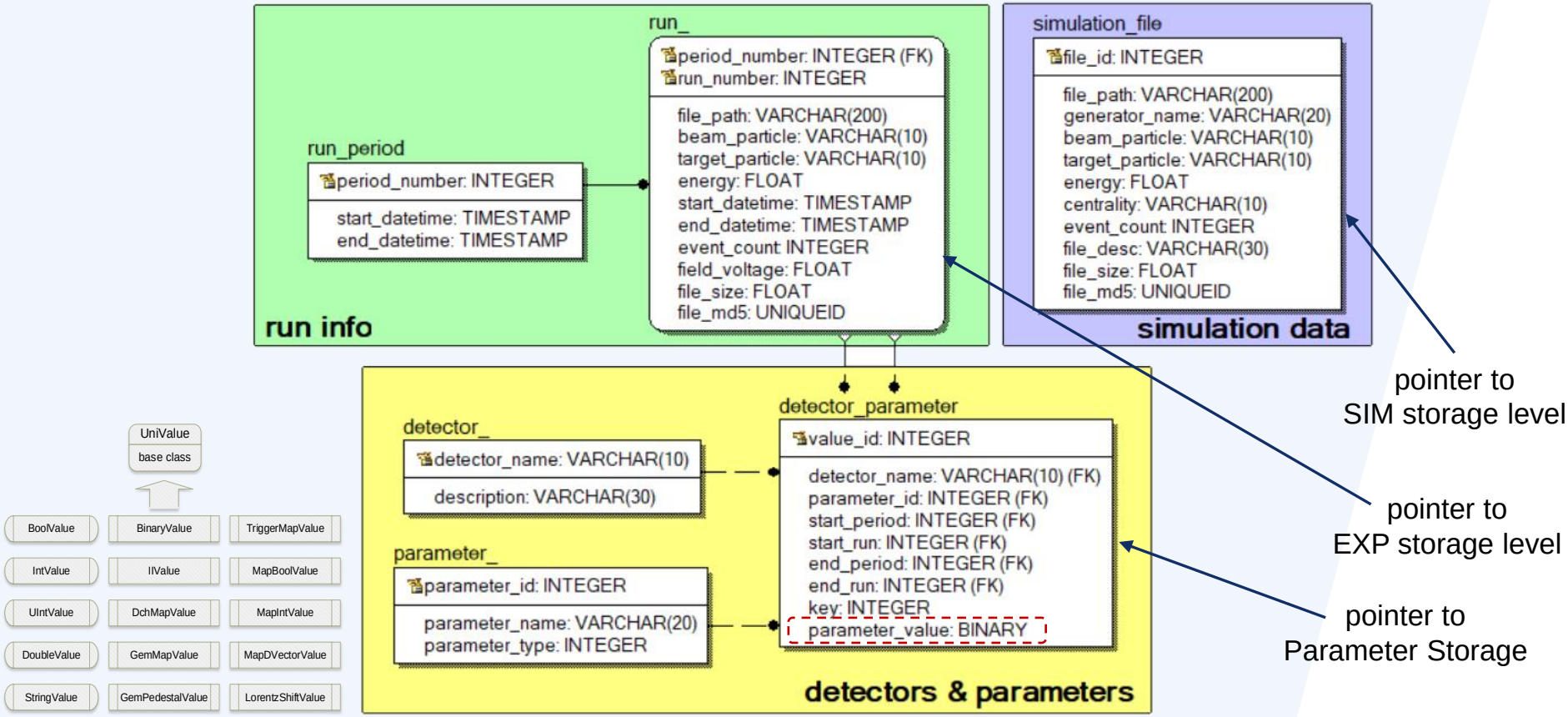
3rd rev.



Purposes of the Condition Database

- central data storage for offline data analysis (and may be online tasks) in the experiments of the NICA project
- unified access and data management for all collaboration members
- correct multi-user data processing
- ensuring the actuality of the information being accessed (sessions and run parameters, technical and calibration data, etc.), data consistency and integrity
- excluding the multiple duplication and use of outdated data
- automatic backup of the stored data

{ Unified → Condition } Database Diagram



The following solutions were considered to replace old packed structures: ZeroMQ, MessagePack, BOOST, Protobuf, FlatBuffers, ROOT/TStreamer, C++ manual serialization.

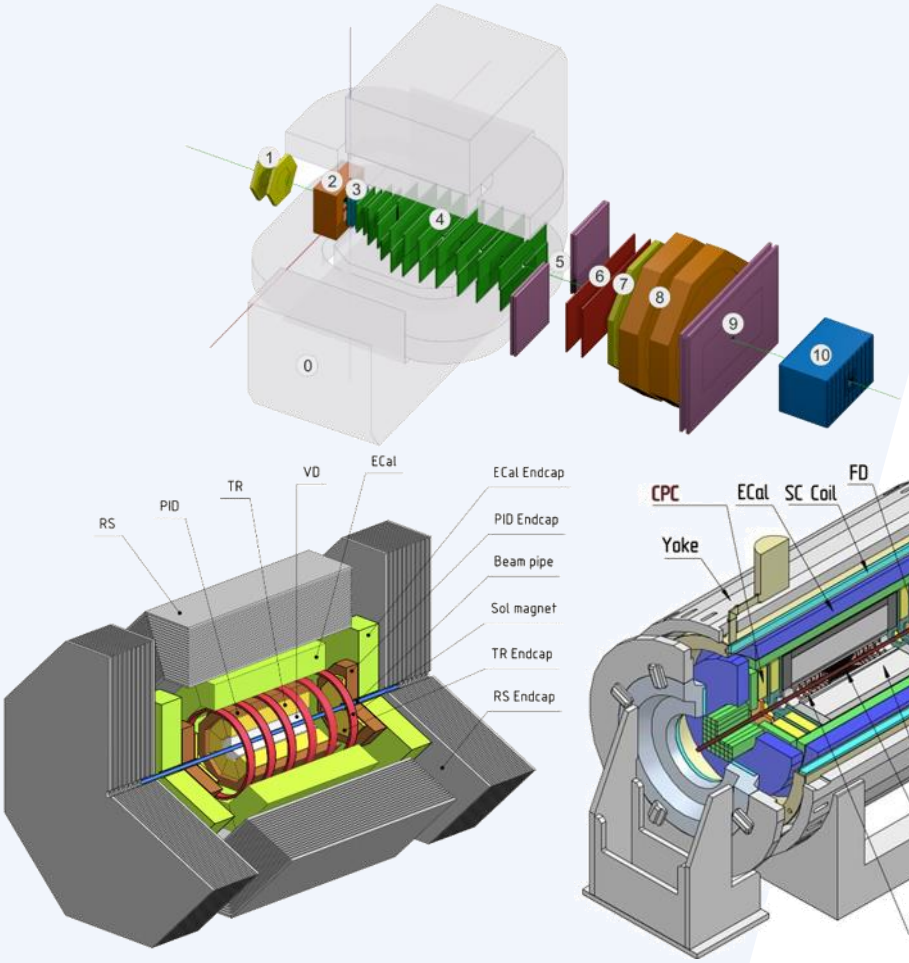
storing information on experiment sessions and runs, detectors, parameters and parameter values, and generated simulation files

PostgreSQL 12

Integration with the NICA frameworks

The software BmnRoot, MPDRoot and SPDroot are developed for event simulation, reconstruction of experimental and simulated data and following physics analysis of particle collision events registered by the detectors at the NICA collider.

C++ classes, Linux OS support, based on ROOT and FairRoot



The frameworks are available at GitLab@JINR: https://git.jinr.ru/nica/

Application Programming Interface (C++ API)

Autogenerated class wrappers for database tables with specific functions allow to access and manage data without SQL statements in experiment software

UniDbRunPeriod – describes run periods (a set of runs) of the experiment

UniDbRun – run parameters (number, time, energy, beam, target, magnet field, file path, etc.)

UniDbDetector – detectors of the experiment (detector dictionary)

UniDbParameter – common information about detectors' parameters presented on the previous slides and stored in the database (parameter dictionary)

UniDbDetectorParameter – values of detector parameters for experiment runs

UniDbSimulationFile – describes a set of generated simulation files

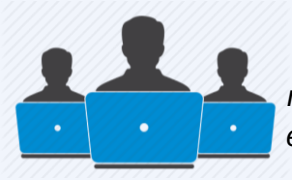
The main functions of the database interface:

for data objects (static): *Create*, *Delete, Get*, ***Search***, *PrintAll.*

for attributes (non-static)*: Getters* and *Setters* functions, *Print*.

- integrating the database with experiment frameworks based on the ROOT package
  - using stored information in offline (and online) event data processing
  - implementing a convenient search for necessary information by various criteria

Condition Database Architecture

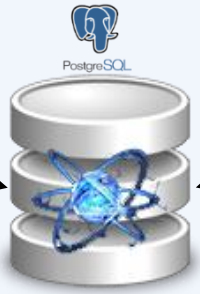


detector simulation
raw data processing
event reconstruction
physics analysis

ROOT-based framework

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

ROOT

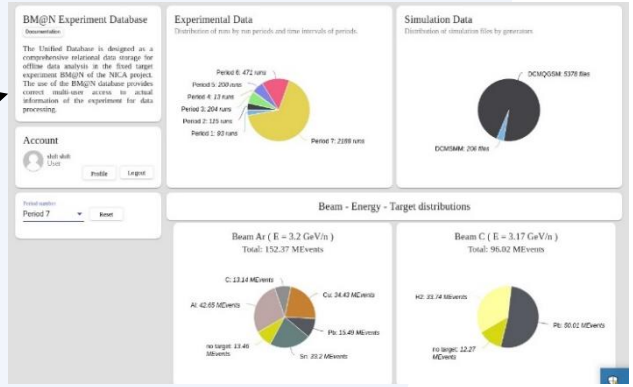


Condition Database

Node.JS

users

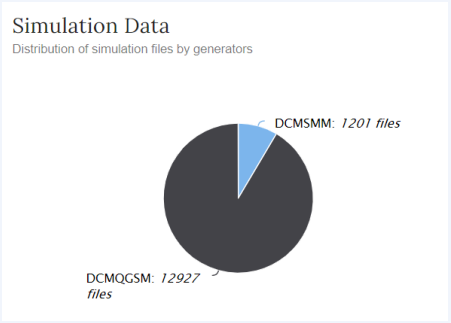
FreeIPA authentication



Web Service

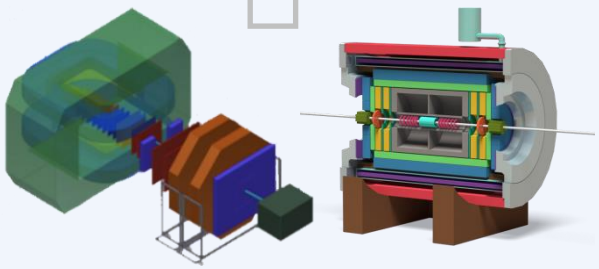
Local Condition Database as a local replica of the central database and mirroring are under closer consideration

python script for auto update of simulation file list

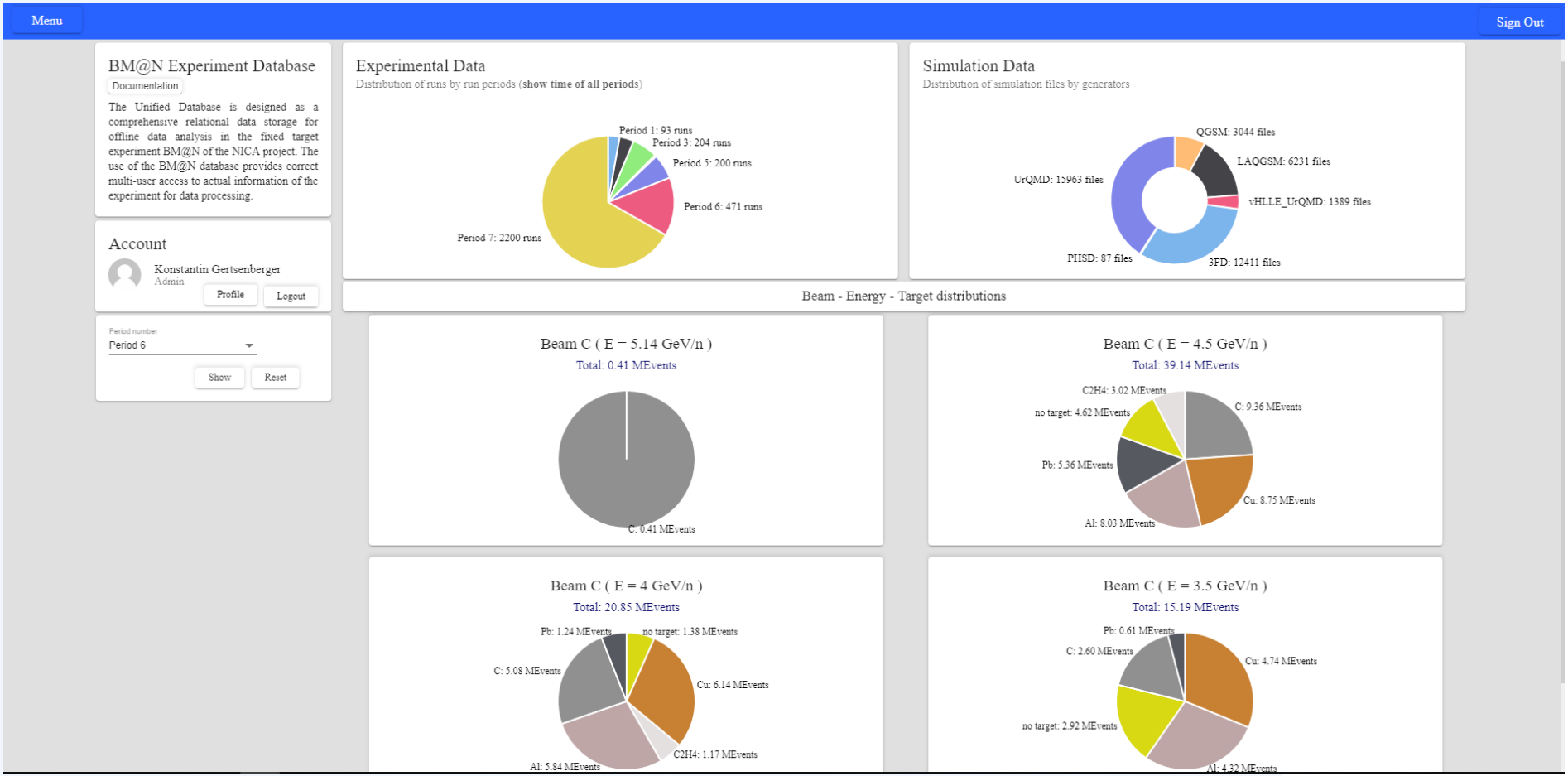


configuration calibration

parameter and algorithm data

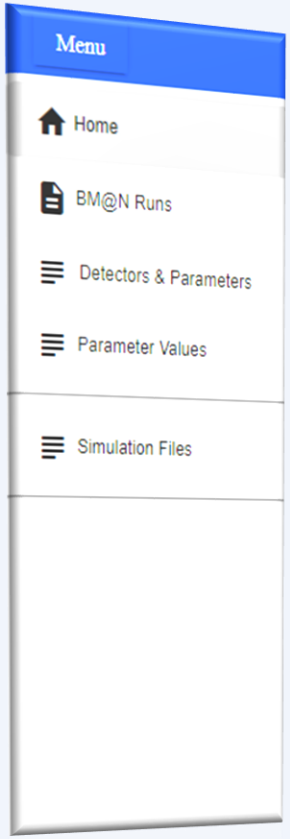


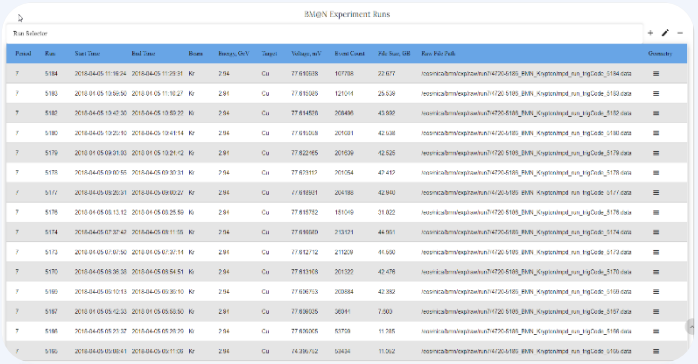
Web service for the Condition Database



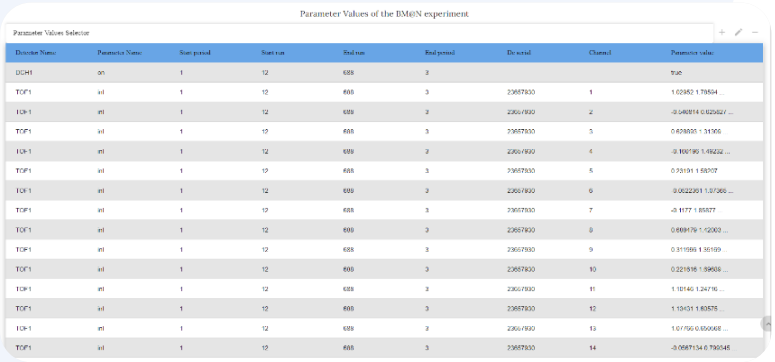
- visualization of summary data in the form of diagrams and charts
- convenient viewing, managing and searching for up-to-date information on the NICA experiments in tabular view by collaboration members

Tabular View of the stored data

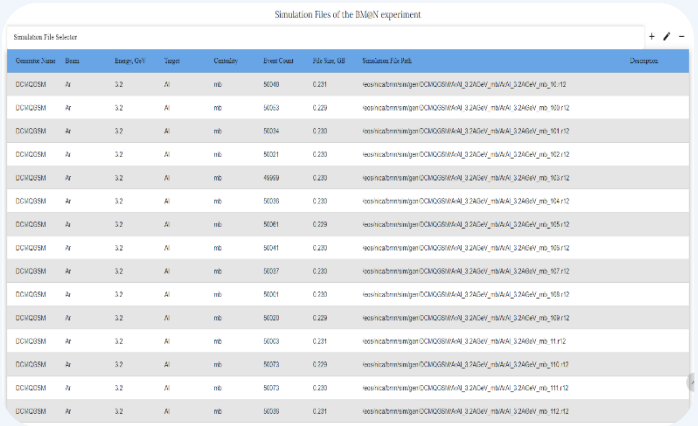




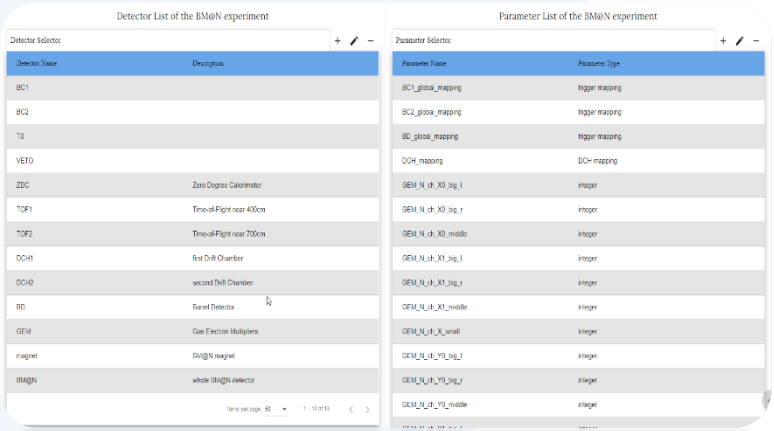
Experiment Runs



Parameter Values

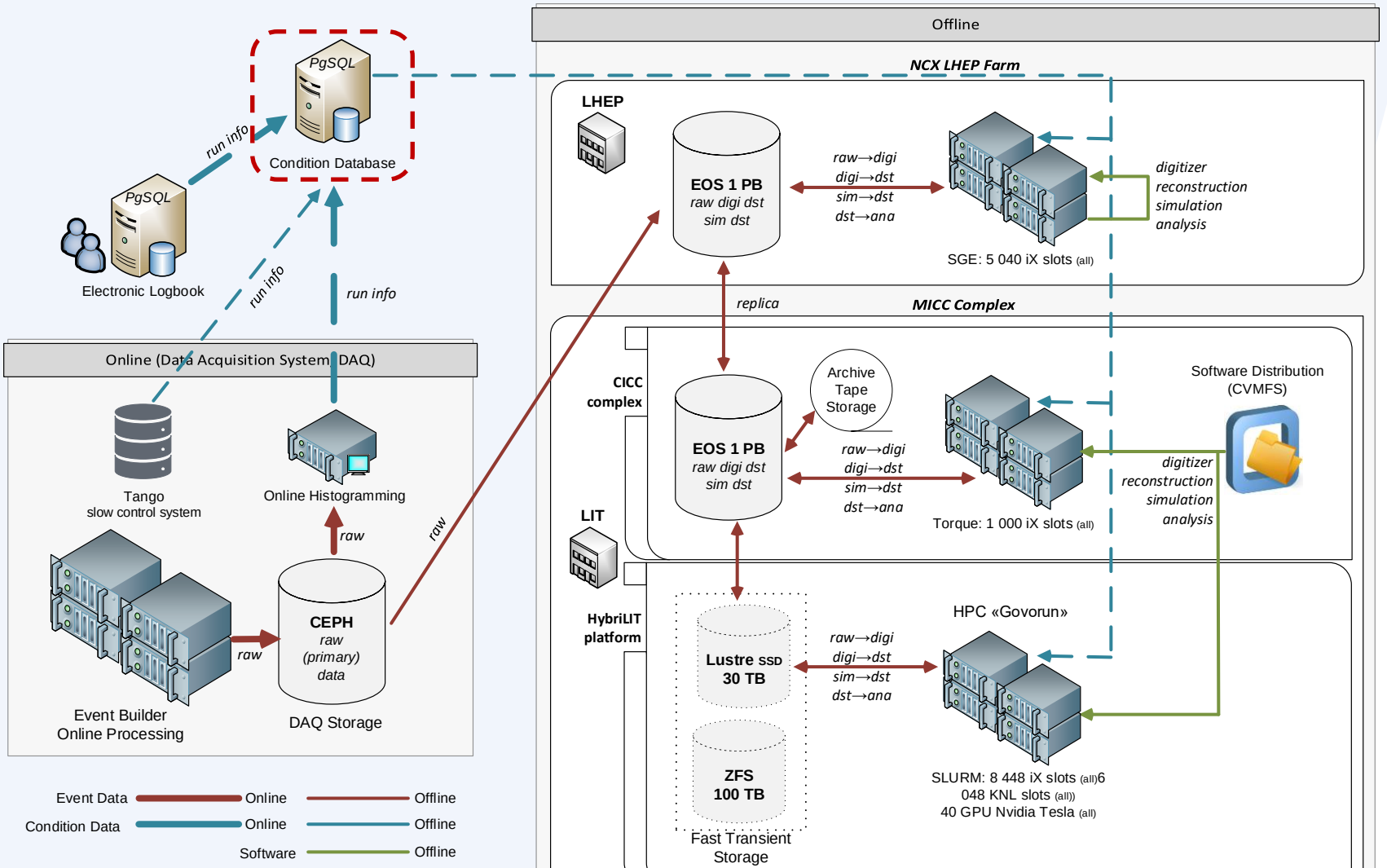


Simulation Files

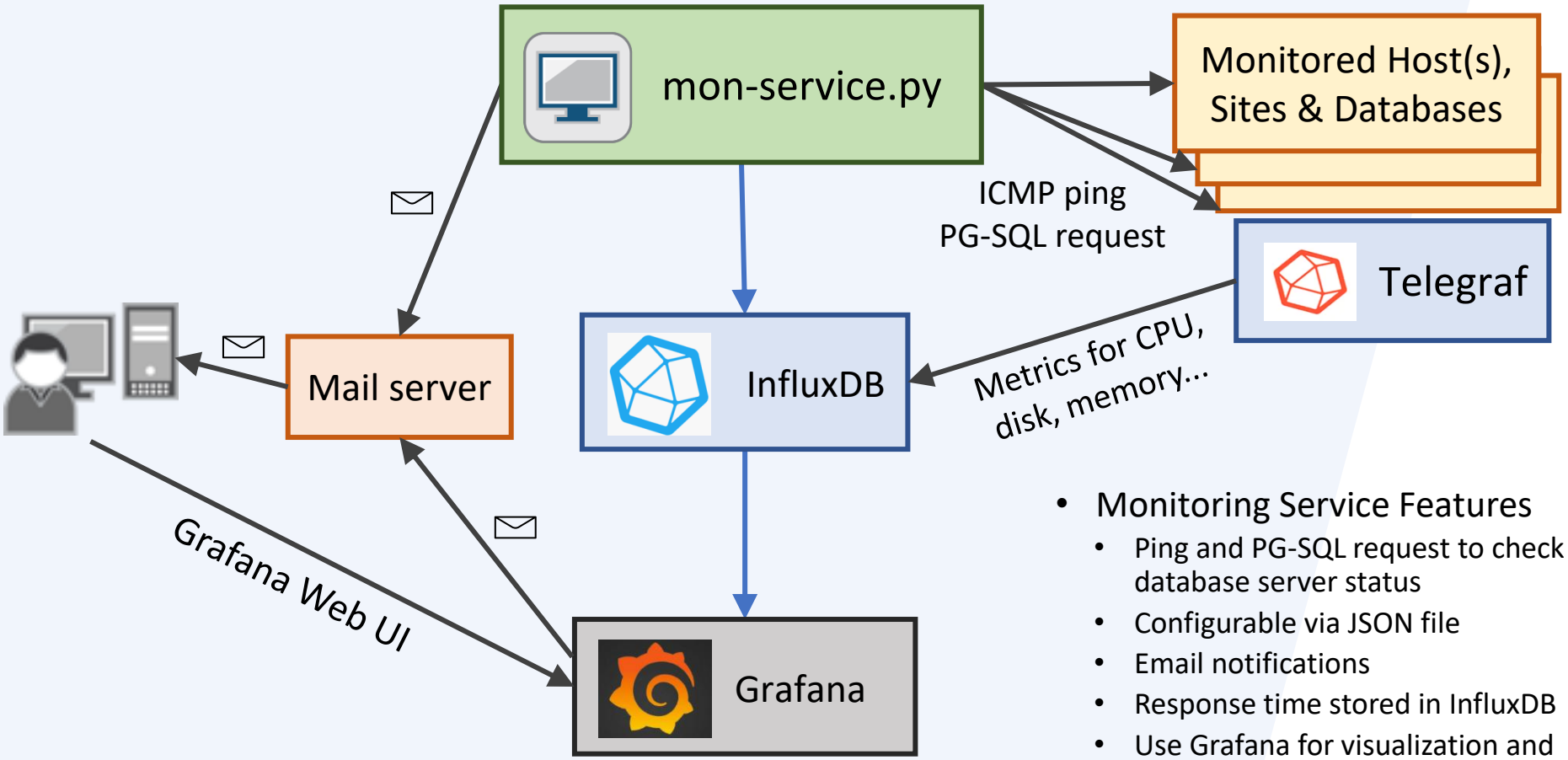


Detector & Parameters

Condition Database in distributed computing

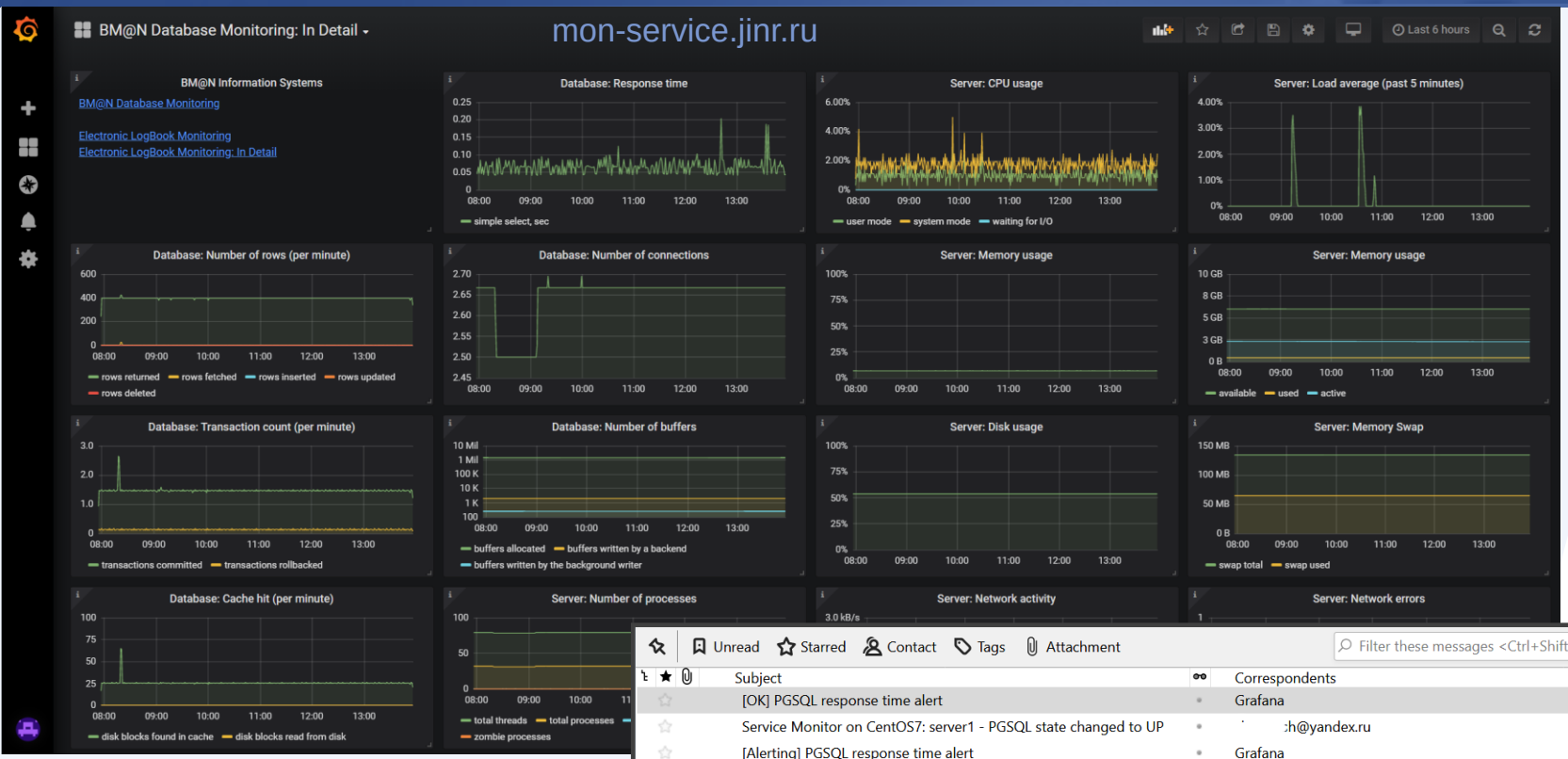


Monitoring Service



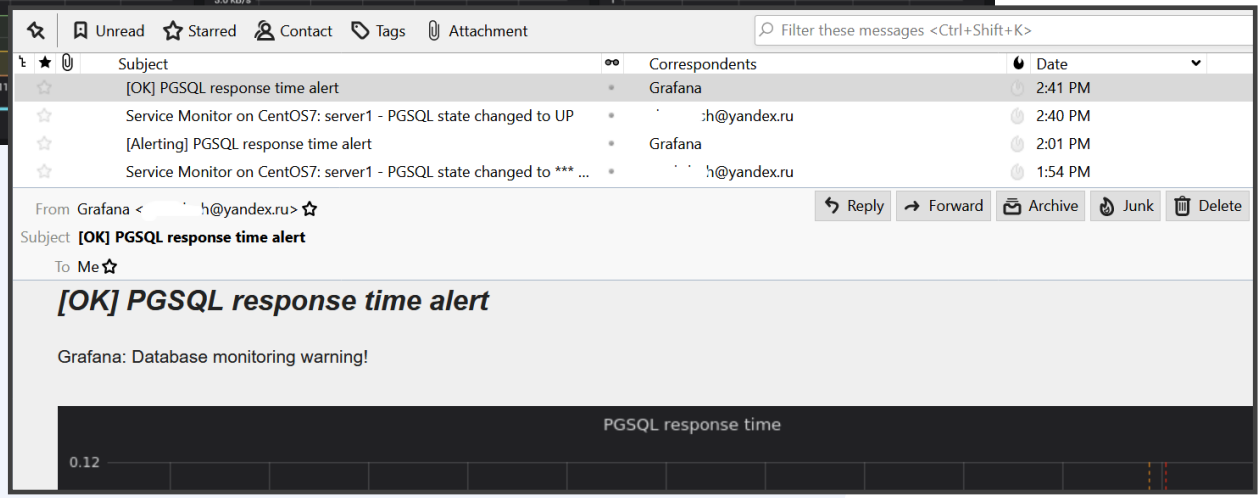
- Monitoring Service Features
  - Ping and PG-SQL request to check database server status
  - Configurable via JSON file
  - Email notifications
  - Response time stored in InfluxDB
  - Use Grafana for visualization and additional alerting
  - Monitor server parameters such as Disk, CPU, Memory, etc.

Monitoring Information Systems



Grafana View

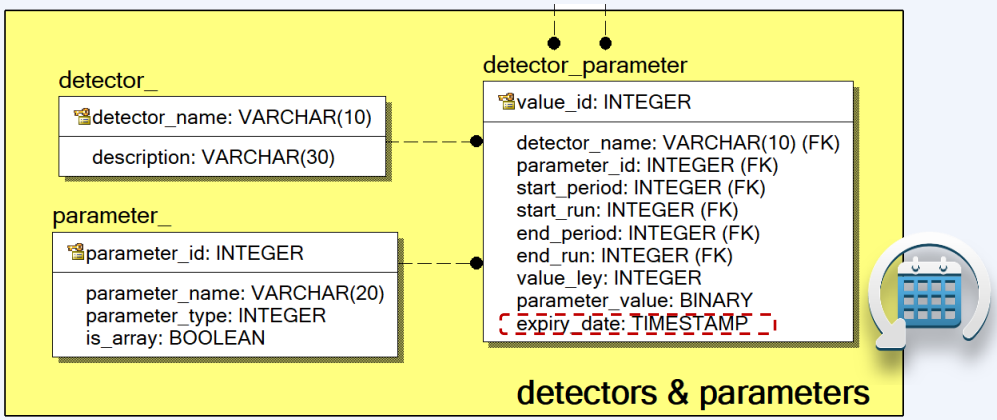
Condition Database + detailed



Email Notifications

Work on the Condition Database in progress

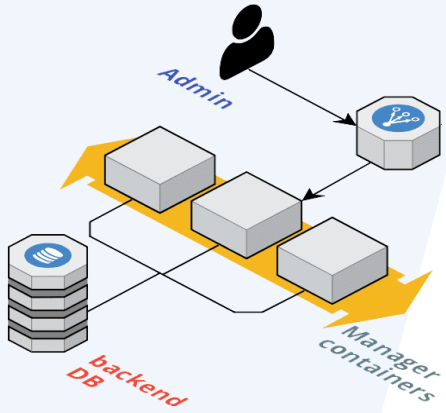
Historical Preservation of Parameter Values



Parameter values need to be retained in case of updating. When parameter values are updated, the database saves the replaced data with the current expiry date.

It allows to repeat event data processing with outdated parameters used in the past

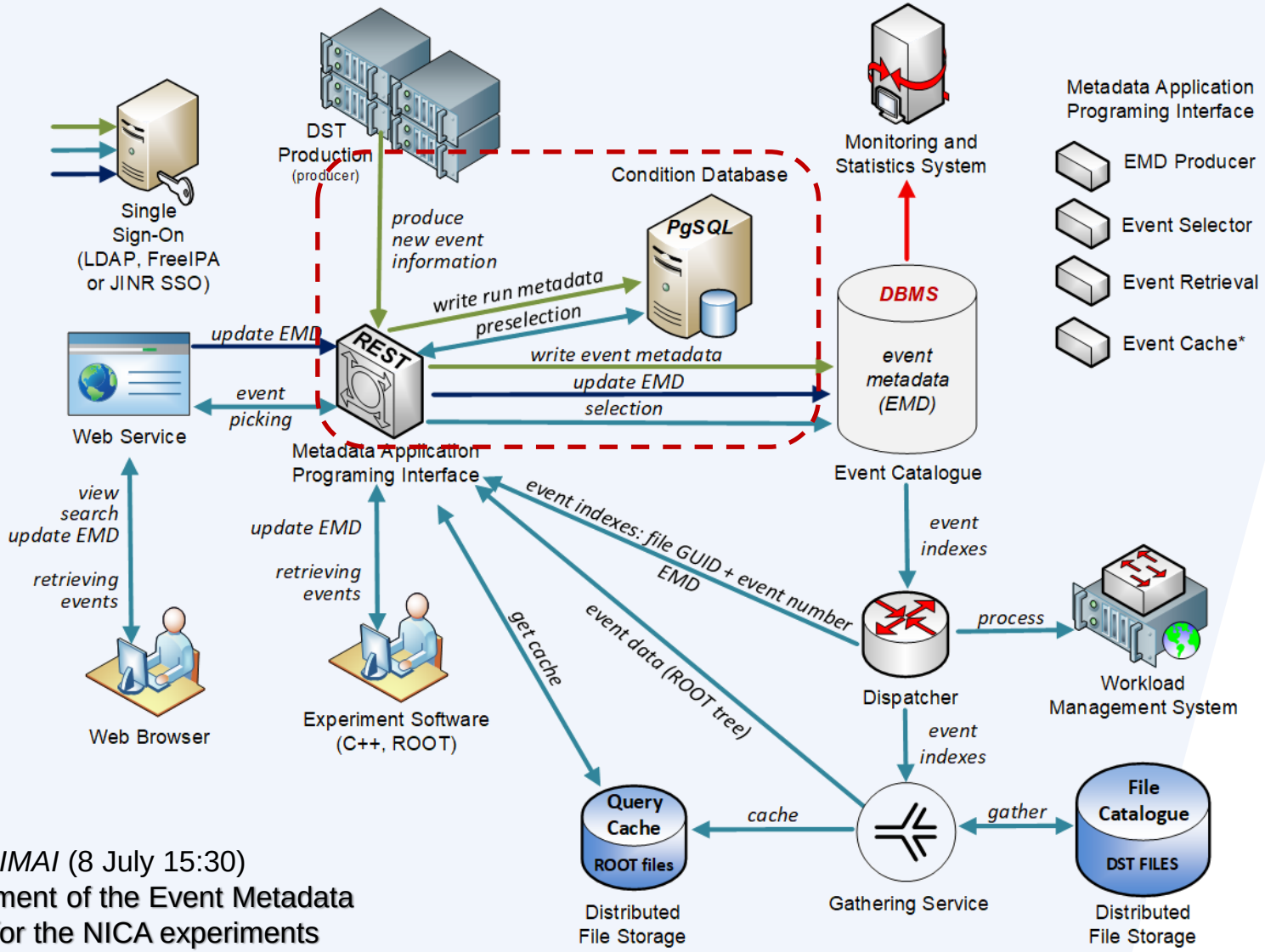
Configuration and Deployment System



The common Configuration and Deployment System is planned to be based on Docker containers and shell scripts

It allows to conveniently deploy the Condition Database and its services for all the experiments of the NICA project taking into account some specifics of the experiments

Condition Database in Event Processing System



Peter KLIMAI (8 July 15:30)
Development of the Event Metadata System for the NICA experiments

Conclusions

- The Condition Database developed on PostgreSQL is an important component of implementing complex of the information systems and a centralized storage to provide unified access and management of various parameters of the experiments for event data processing, including simulation, reconstruction and physics analysis.
- The current version of the Condition Database has been deployed in the NICA cluster of the Laboratory of high-energy physics and is actively used in distributed data processing of the BM@N experiment, the first experiment of the NICA project.
- The C++ interface (API) has been implemented to use information stored in the Condition Database for processing of simulated and experimental data in the experiment frameworks based on the ROOT environment.
- The Web service, single authorization and monitoring systems have been already developed to simplify viewing and managing information on the experiments by users over the Web.
- The Common Deployment System and additional services are under development to install the Condition Database for all the experiments of the NICA project taking into account some specifics of the experiments.

Thank you for your attention!

Alexander NOZIK: PhD, senior researcher
Peter KLIMAI: PhD, researcher…

MIPT participants

JINR LIT participants

Igor ALEXANDROV: PhD, head of sector
Irina FILOZOVA: group leader
Evgeniy ALEXANDROV: researcher
Mikhail MINEEV: researcher
Alexander YAKOVLEV: researcher
Galina SHESTAKOVA: lead developer
Daria PRIAKHINA: development engineer

JINR LHEP participants

Konstantin GERTSENBERGER: PhD, group leader
Andrey MOSHKIN: development engineer
Alexander CHEBOTOV: development engineer

RFBR Grant Collaboration
№18-02-40125

We are open for cooperation!

thanks to the NICA cluster and HybriLIT teams for computing support

contact email: *gertsen@jinr.ru*