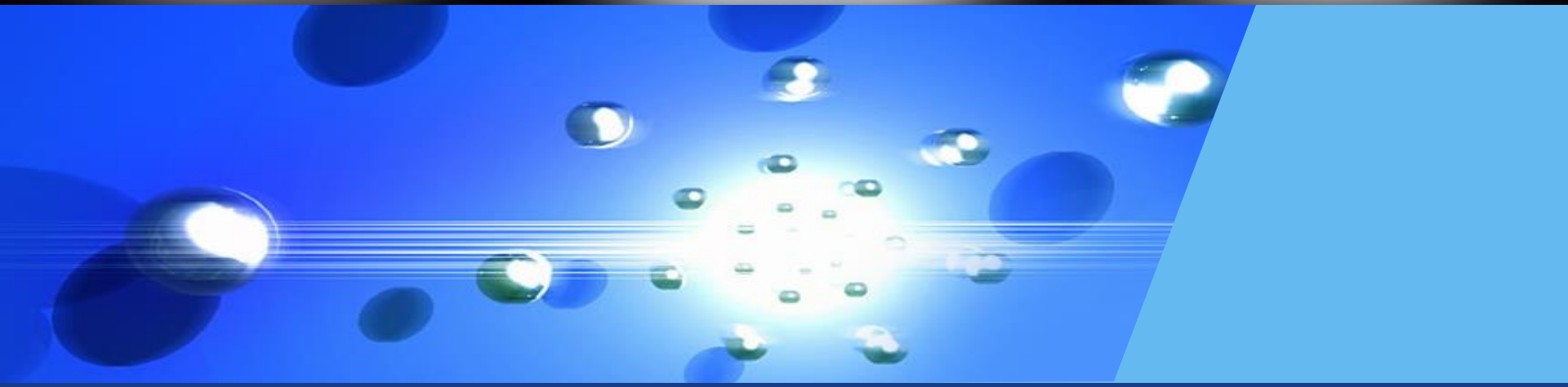


Development of the Information Systems for online and offline data processing in the NICA experiments

K. Gertsenberger, I. Alexandrov, I. Filozova, E. Alexandrov, A. Nozik, A. Moshkin, A. Chebotov, P. Klimai, M. Mineev, D. Pryahina, G. Shestakova, A. Yakovlev

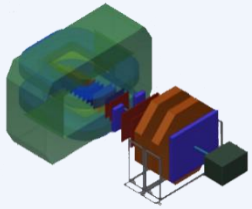
Supported by RFBR Grant №18-02-40125

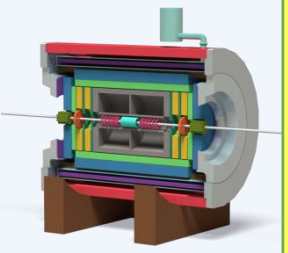
The Conference «RFBR Grants for NICA»

20-23 October 2020

Joint Institute for Nuclear Research, Dubna, Russia

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

Digitizer

ROOT Format
Mapping
Alignment

REC

Event Reconstruction

PA

Physics Analysis

NICA distributed clusters

TDS

Transient Data Storage

PDS

Permanent Data Storage

FT REC

Fast Event Reconstruction

HIST

Online Histograms

EvM

Event Monitor

IS on DataBases

DAQ

Online Processing

Offline Processing

matched from “Data Acquisition System TDR”, DAQ Collaboration

Prerequisites of the data management with IS

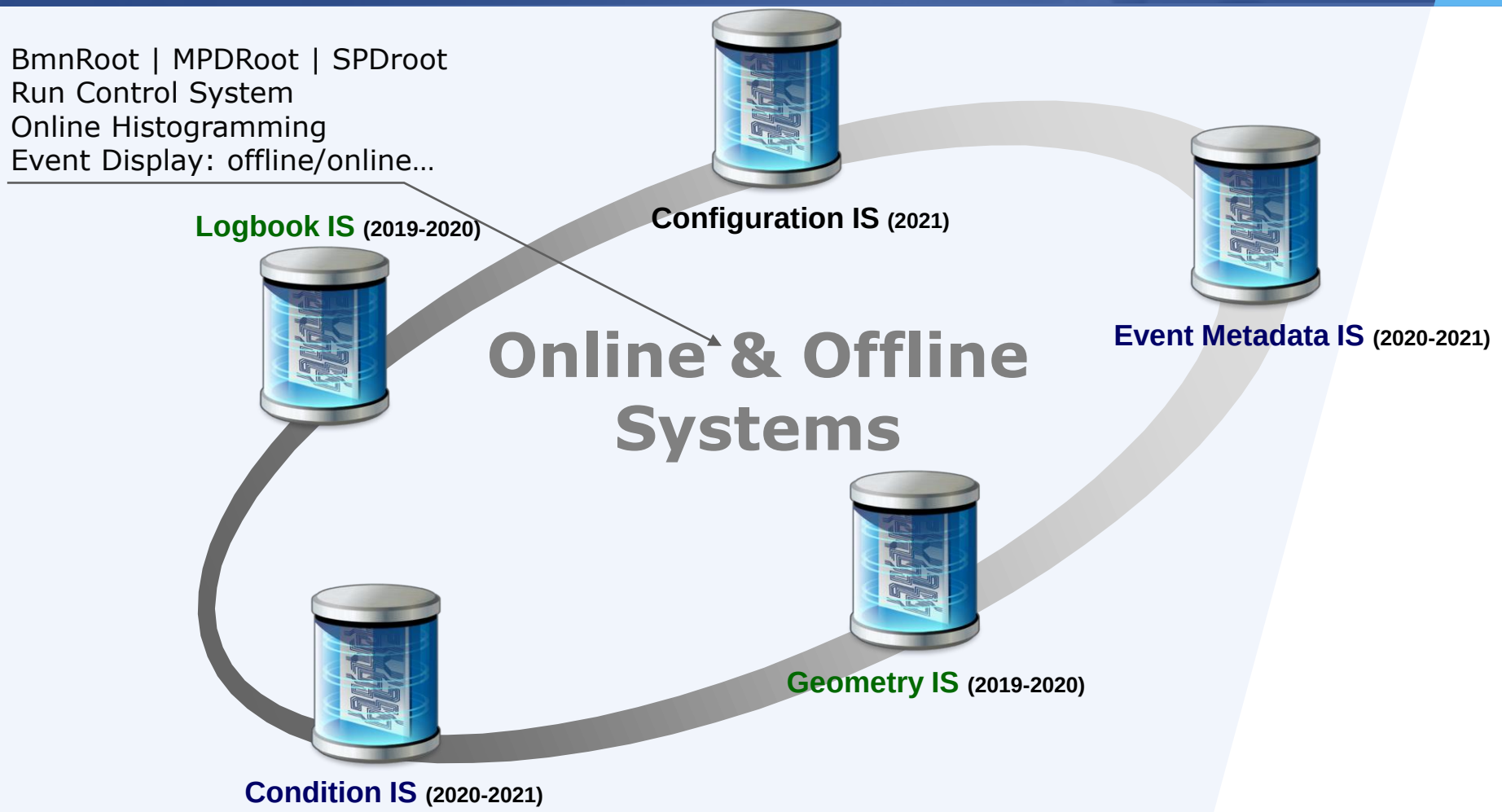
- high interaction rate, e.g. BM@N: up to 50 kHz
- high particle multiplicity, up to 1000 charged particles for central collisions at the NICA energies
- NICA large data stream:

  is estimated up to 20 PB of raw data per year

  500m simulated events ~ 0.5 PB (with digits: ×7)
- Performed review of modern IS in HEP experiments showed: “IS are used in all large physics experiments and have become an important part of the software, but the existing solutions are highly dependent on specifics of conducted experiments and are an inseparable part of them”

E. Alexandrov, I. Alexandrov, K. Gertsenberger, and et. al, Information Systems for Online and Offline Data Processing in Modern High-Energy Physics Experiments, Modern Information Technologies and IT-Education, No. 15(3), 654-671 (2019)

Information Systems for online & offline processing

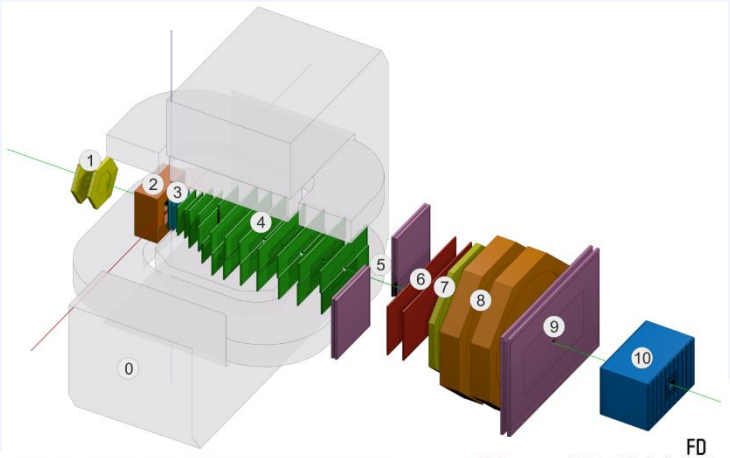


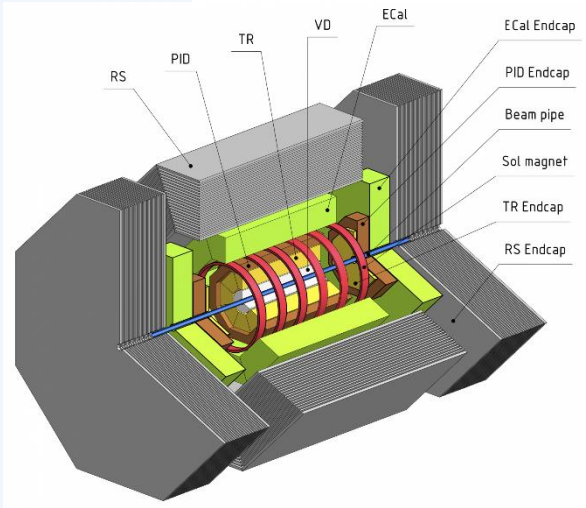
RFBR Grant 2019 – 2021: Development of Information Systems for Online and Offline Data Processing for the Experimental Setups of the NICA Complex

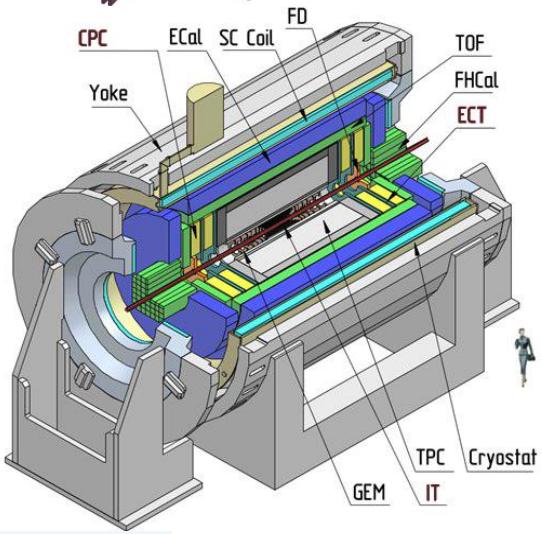
Integration with the NICA frameworks

The software BmnRoot, MPDRoot and SPDroot are developed for event simulation, reconstruction of experimental or simulated data and following physics analysis of particle collision events.

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







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

Electronic Logbook System

Electronic Logbook (e-Log)

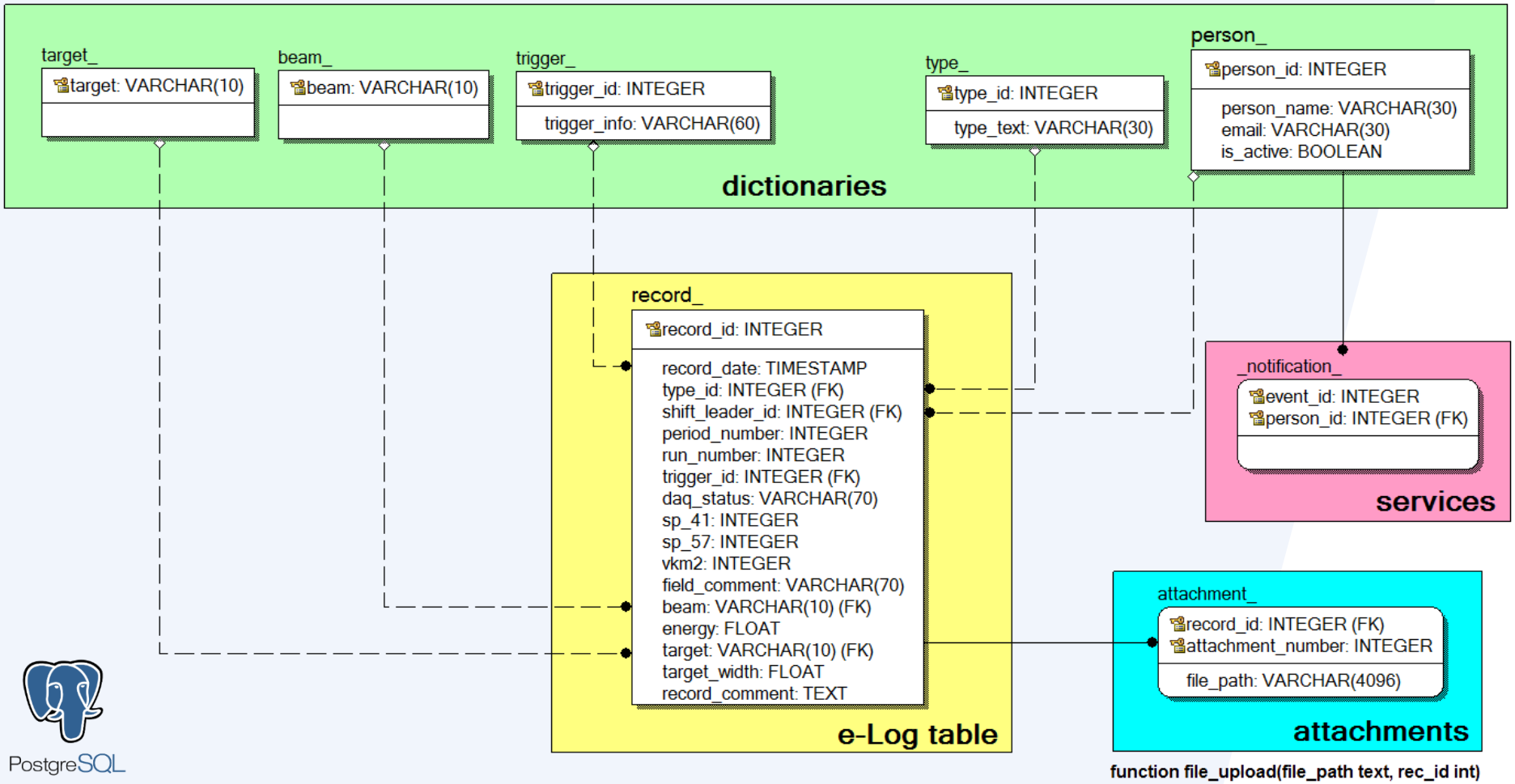
- e-Log platform is a collaborative tool which provides shift crews with an interface to store and share information with offline users on various events or problems occurred in the experiment during its operation.
- The e-Log system uses a developed Logbook Database based on PostgreSQL which ensures correct multi-user access, data consistency, integrity and automatic backup of the stored data.
- A part of e-Log data is automatically transferred to the offline database of the experiment to use in offline analysis.
- Developed interfaces provide a unified access to required logbook data for various online and offline systems, and convenient viewing, transparent managing and searching for required information by users.

Electronic Logbook Data

- record number
- record type
- record time
- current shift leader
- period number
- run number
- DAQ status
- trigger type
- magnetic field
- beam particle
- beam energy
- target / second particle
- record comment

record information during experiment runs on current events, states of various systems, operation conditions of detectors and many others which can be further used in the processing and physics analysis of the particle collision events

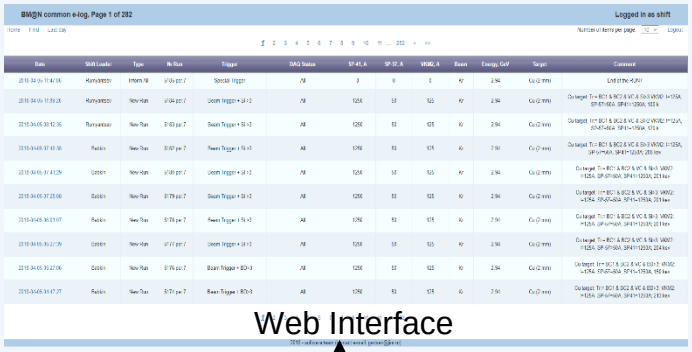
e-Log Database scheme



e-Log Communication Scheme

shift operators

users



Web Interface

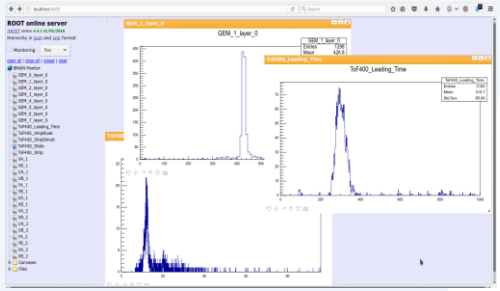
Application Programming Interface

Application Programming Interface

Online

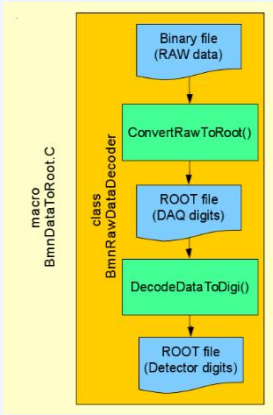
Offline

Online Histogramming



…

Raw Data Converter



Event Data Analysis

…

Web-interface of the e-Log Platform

BM@N Electronic Logbook https://bmn-elog.jinr.ru **Logged in as shift**

Home New Find Last day Account Reference Book

Page: 1 of 282 Number of items per page: 10 Logout

Date	Shift Leader	Type	№ Run	Trigger	DAQ Status	SP-41, A	SP-57, A	VKM2, A	Beam	Energy, GeV	Target	Comment	Attachment
2018-04-05 11:47:06	Rumyantsev	Inform All	5185 per.7	Special Trigger	All	0	0	0	Kr	2.94	Cu (2 mm)	End of the RUN7	
2018-04-05 11:09:20	Rumyantsev	New Run	5184 per.7	Beam Trigger + Si >3	All	1250	50	125	Kr	2.94	Cu (2 mm)	Cu target; Tr.= BC1 & BC2 & VC & Si>3 VKM2: I=125A, SP-57=50A, SP41=1250A; 100 k	
2018-04-05 08:12:35	Rumyantsev	New Run	5183 per.7	Beam Trigger + Si >3	All	1250	50	125	Kr	2.94	Cu (2 mm)	Cu target; Tr.= BC1 & BC2 & VC & Si>2 VKM2: I=125A, SP-57=50A, SP41=1250A; 120 k	
2018-04-05 07:46:35	Babkin	New Run	5182 per.7	Beam Trigger + Si >3	All	1250	50	125	Kr	2.94	Cu (2 mm)	Cu target; Tr.= BC1 & BC2 & VC & Si>3 VKM2: I=125A, SP-57=50A, SP41=1250A; 208 kev	
2018-04-05 07:41:29	Babkin	New Run	5180 per.7	Beam Trigger + Si >3	All	1250	50	125	Kr	2.94	Cu (2 mm)	Cu target; Tr.= BC1 & BC2 & VC & Si>3; VKM2: I=125A, SP-57=50A, SP41=1250A; 201 kev	
2018-04-05 07:25:08	Babkin	New Run	5179 per.7	Beam Trigger + Si >3	All	1250	50	125	Kr	2.94	Cu (2 mm)	Cu target; Tr.= BC1 & BC2 & VC & Si>3; VKM2: I=125A, SP-57=50A, SP41=1250A; 201 kev	
2018-04-05 06:01:07	Babkin	New Run	5178 per.7	Beam Trigger + Si >3	All	1250	50	125	Kr	2.94	Cu (2 mm)	Cu target; Tr.= BC1 & BC2 & VC & Si>3; VKM2: I=125A, SP-57=50A, SP41=1250A; 201 kev	
2018-04-05 05:27:39	Babkin	New Run	5177 per.7	Beam Trigger + Si >3	All	1250	50	125	Kr	2.94	Cu (2 mm)	Cu target; Tr.= BC1 & BC2 & VC & Si>3; VKM2: I=125A, SP-57=50A, SP41=1250A; 204 kev	
2018-04-05 05:27:06	Babkin	New Run	5176 per.7	Beam Trigger + BD>3	All	1250	50	125	Kr	2.94	Cu (2 mm)	Cu target; Tr.= BC1 & BC2 & VC & BD>3; VKM2: I=125A, SP-57=50A, SP41=1250A; 150 kev	
2018-04-05 04:47:27	Babkin	New Run	5174 per.7	Beam Trigger + BD>3	All	1250	50	125	Kr	2.94	Cu (2 mm)	Cu target; Tr.= BC1 & BC2 & VC & BD>3; VKM2: I=125A, SP-57=50A, SP41=1250A; 213 kev	

2020 - software team (contact e-mail: gertsen@jinr.ru)

Common FreeIPA Authentication: Administrator, Editor, Reader roles

File Attachments (text description, photo)

Email Subscription to selected event types

Multi-Column Sorting

Logbook Monitoring

User Cabinet

Easy Searching

e-Log: Notification Service

e-mail notifications

different types of events:

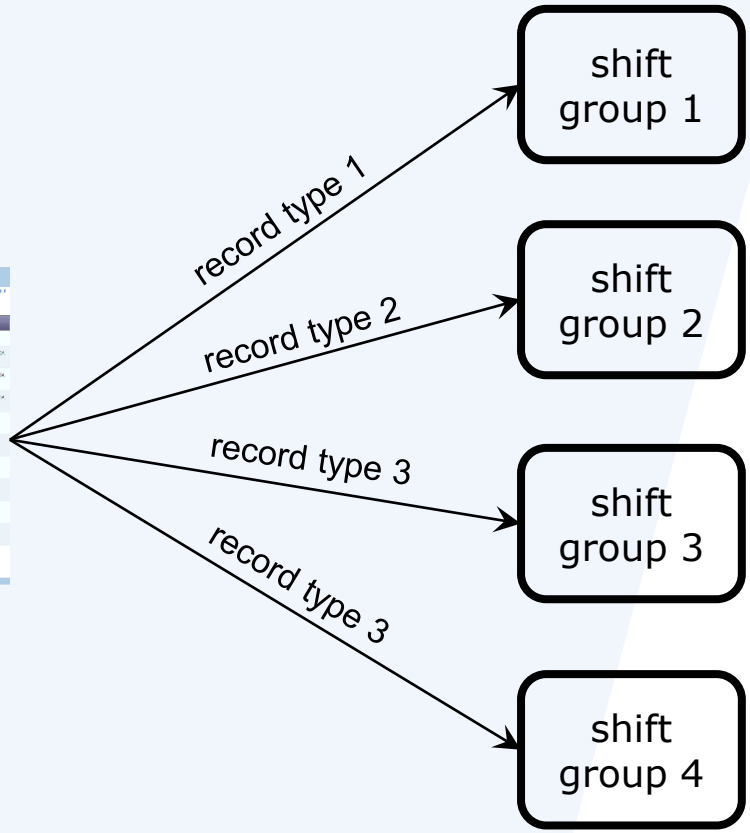
“shift started”

“problem report”

“configuration”

“inform all”

…



C++ database interface (autogenerated API)

Class wrappers for logbook object with specific functions allow to access and manage the data without SQL statements in the experiment frameworks

ElogDbRecord – records written by a shift crew during the experiment runs which describe operating modes of various systems and detectors and different types of events

ElogDbType – record types: ‘Shift started’, ‘Problem report’, ‘Configuration’, ‘New Run’, etc.

ElogDbPerson – a list of the experiment staff

ElogDbTrigger – dictionary of all possible trigger types

ElogDbBeam – dictionary of all possible beam particles

ElogDbTarget – dictionary of all possible targets

ElogDbAttachment – files attached to a record for detailed description of an event

ElogDbConnection – serves to open and close connections to the e-Log database

ElogDbSearchCondition – forms criteria for selection of e-Log records

The main functions of the e-Log interface:

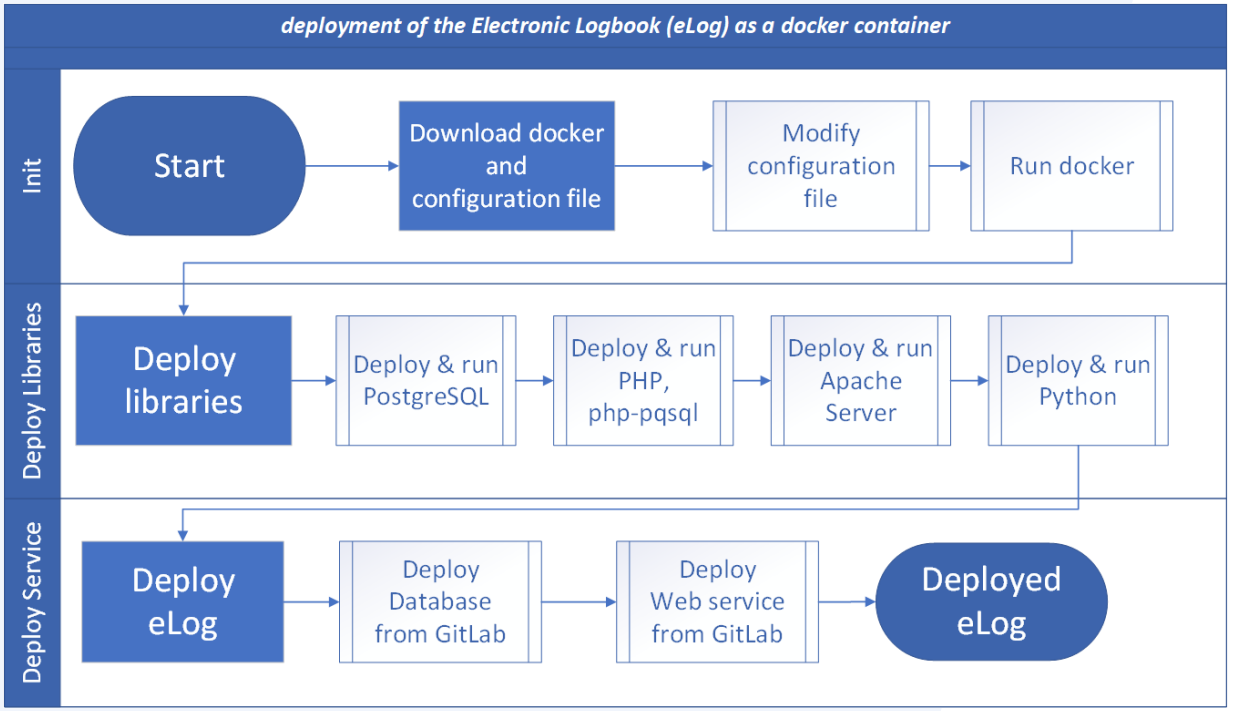
for data objects (static): *Create*, *Delete, Get*, *Search*, *PrintAll.*

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

e-Log: Configuration and Deployment

Configuration File

```
{ "host" : "nc13.jinr.ru",    //Database host
"port" : "5432",              //Database port
"dbname" : "bmn_elog",        //Database name
"dbA":true,                   //Authorization type
//Experiment specific data columns
"colVal": {  "sp_41" : "SP-41, A",
    "sp_57" : "SP-57, A",
    "vkm2"  : "VKM2, A"},
"columns" : [ {"column" : "sp_41 int null"},
{"column" : "sp_57 int null"},
{"column" : "vkm2  int null"}],
"expName":"MPD",                        //Experiment name
"loginImage" :"login/images/bmn2.png", //Login image
"loginLink":"http://bmn.jinr.ru",       //Link to Web site
"fillRunDB":true,                       //whether fill DB
"sendNotif":true,                       //Notifications
```



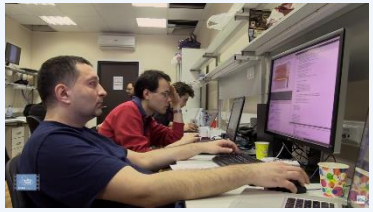
Deployment Scheme

The Common Deployment System is based on Docker containers and shell scripts

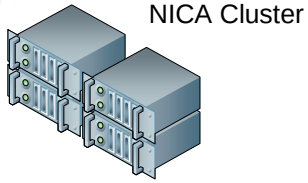
It allows to install the Online Logbook System for all the experiments of the NICA project taking into account some specifics of the experiments

Electronic Logbook Completeness

- The first prototype of the Electronic Logbook was employed in the latest session of the BM@N experiment, and at present it contains records for all conducted BM@N runs.
- The Electronic Logbook System has been deployed on the NICA cluster and approved 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.
- 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

Detector Geometry Definition

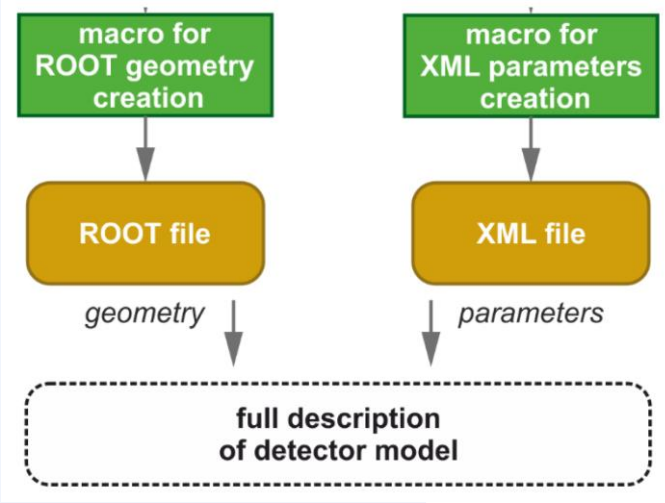
- **Geometry File**

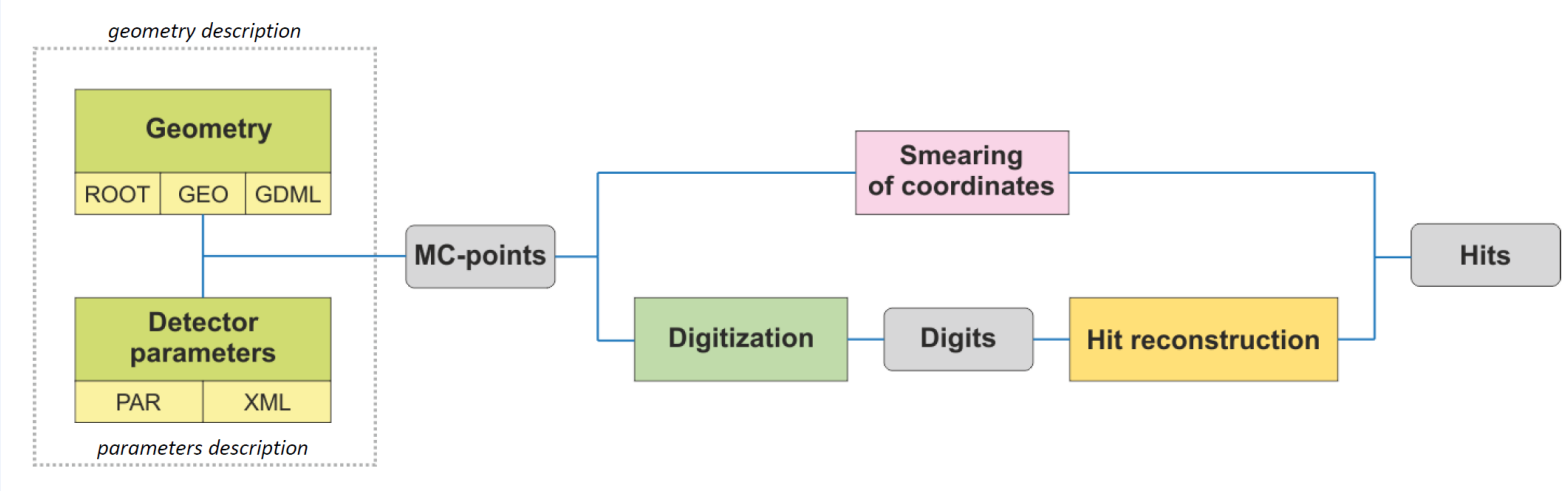
Description of basic detector volumes (sensitive + passive):
dimensions, position, orientation, materials

- **Parameter File** (optional)

Description of the detailed structure of the detector
required for realistic simulation, digitization, hit reconstruction:

  - the internal structure of the detector (stations, modules, layers, dead zones ...), as well as their orientation, position and layout..
  - reading elements, e.g. strips (strip angle, strip width, reading order, etc.)
  - other parameters required for simulation and reconstruction (width of gaps, direction of electron drift, etc.)





Basic Geometry Definitions

Geometry Module

File in the ROOT format containing a detector geometry

Setup Module

Consists of geometry module, link to the parent geometry module, its placement in the parent module (a transformation matrix or object of the TGeoMatrix class)

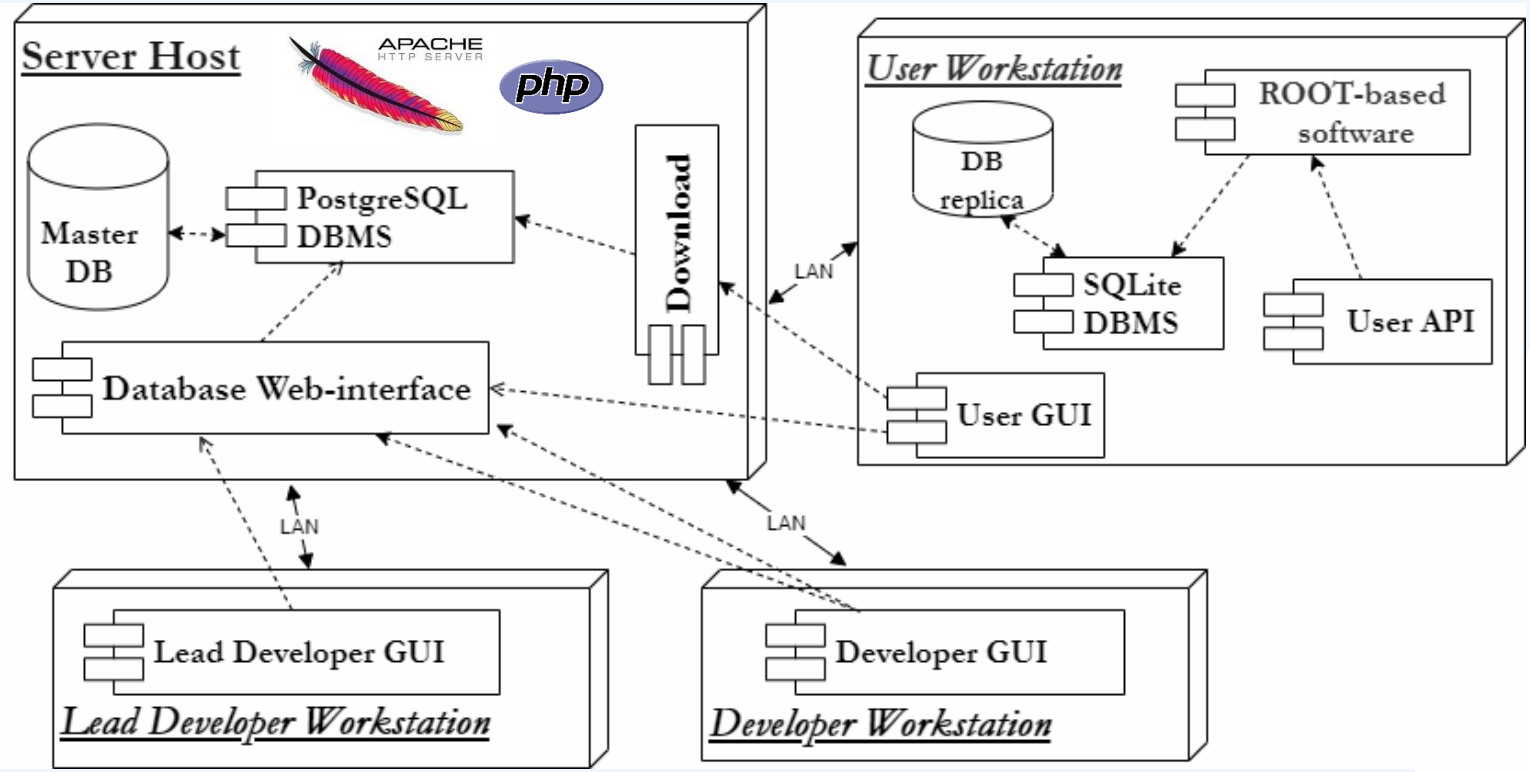
Setup

Combination of setup modules, which represents the full setup geometry

Geometry Information System

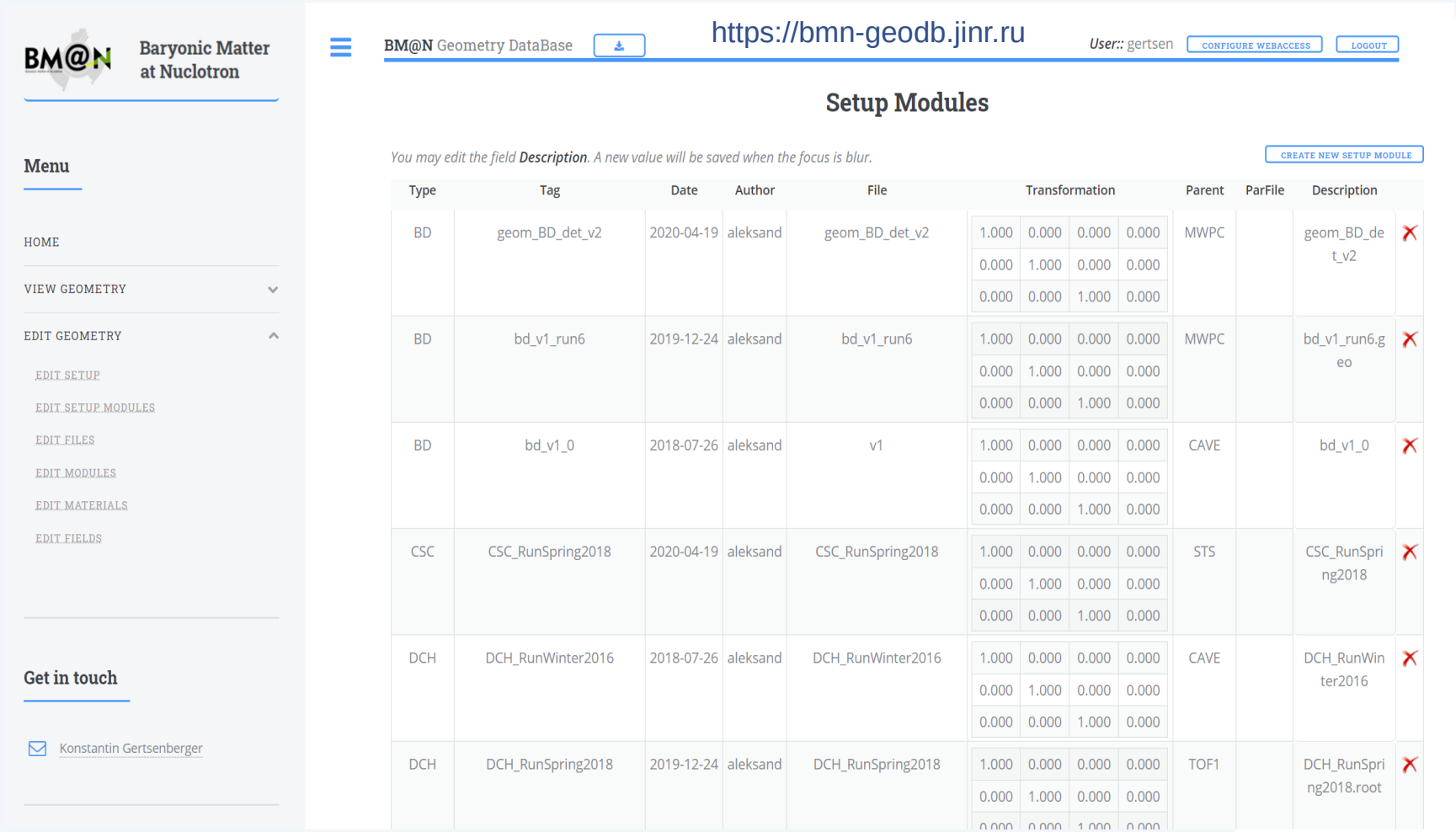
- is based on the Geometry Database to work with detector geometries of the NICA experiments and intended for storing and managing information on the geometry models of the detectors
- manages geometry modules as ROOT binary objects
- each setup module stores a tag, version, transformation matrix, link to the parent module
- manages full setups as a combination of geometry setup modules, magnetic field and materials
- manages versions of the modules and setups
- provide detector geometries for online (e.g. monitoring the current events) and offline (e.g. event reconstruction and analysis) systems

Client-Server Architecture of the Geometry IS



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

Web-interface of the Geometry Database



BM@N Geometry Database has filled with setup geometries for Run 7 and 6

Graphical User Interface Functions:

View Edit Download

Application Programming Interface (API)

Geometry API provides a set of ROOT macros 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

calling API interface functions of the developed macros

Signature	*Description*	*Call Example*	*Comment*
`void getSetupList();`	***Get the list of available setups.*** Print the list of available setups including tag, date of creation, author and description parameters for each approved setup.	getSetupList.c();	Return the available setups' list
bool loadSetup(const char* setupTag, const char* moduleName);	***Load setup into the Fair framework.*** The Geometry can be used in ROOT framework afterwards. Return FALSE if setup is not loaded, and TRUE if the loading is successful.	bool res = loadSetup("sis100_electron", "*");	"*" – all setup modules to be loaded
void installLocalDB.C();	***Install local database from server to client.*** Download replica of central database to client computer.	**installLocalDB**();	Require set variable **DBL_FILE_PATH** before use.
void installServerDB.C();	***Install new server instance.*** Install and init PostgreSQL database server	**installServerDB**();	Required config file with name **geodb.config.xml**

Geometry Information System Completeness

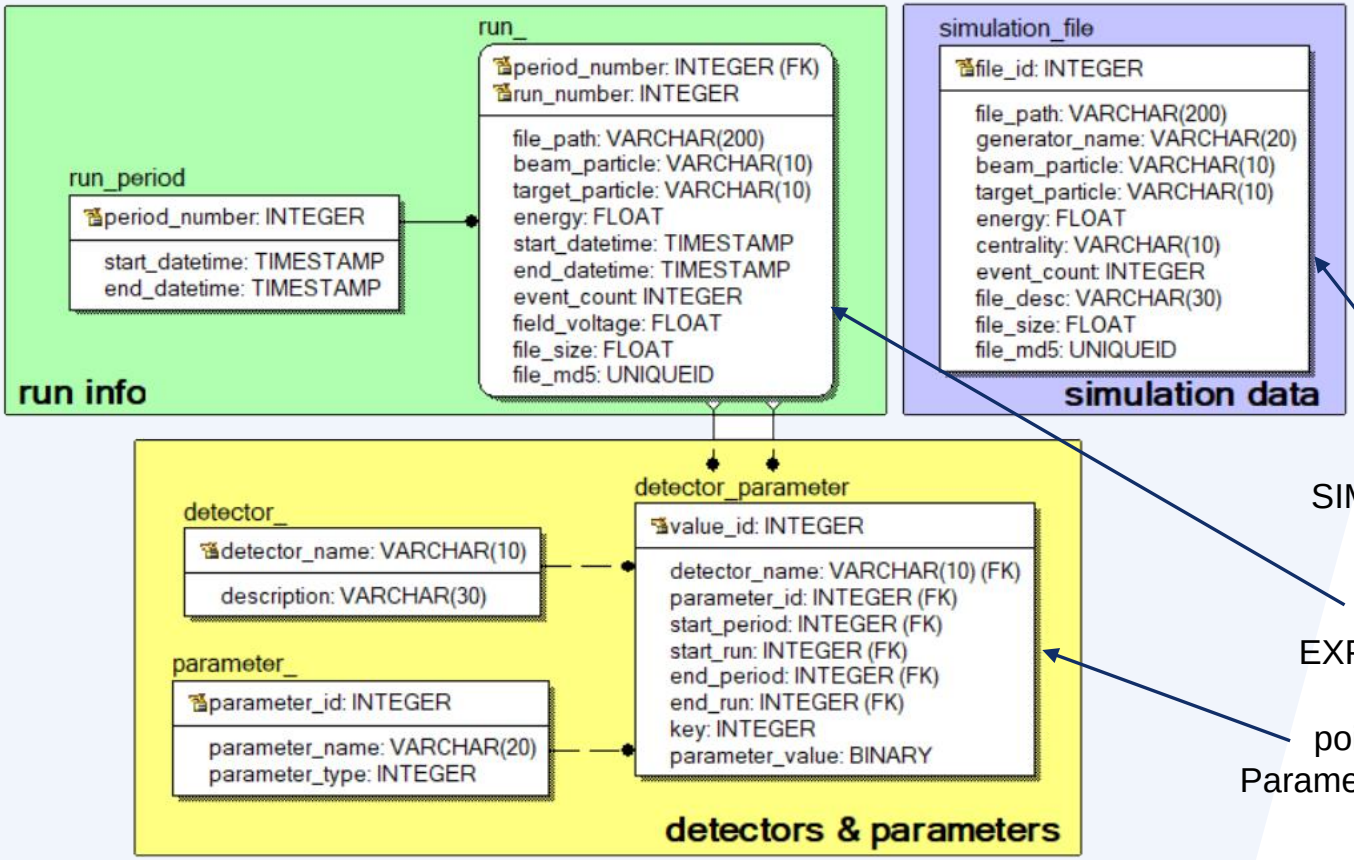
- The developed Geometry Information System provides a central storage of the detector geometries and a set of convenient tools to manage geometry modules and software assemblies of various setup versions in the form of a combination of these modules and auxiliary files containing a description of magnetic 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 experiment.
- It can be employed in the other NICA experiments using developed unified installation script and customized interfaces.

Condition Database

Purposes of the Condition Database

- central data storage for offline data analysis in the high-energy physics experiments
- 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

Condition Database Diagram



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

PostgreSQL

Stored parameters classification

- Configuration data is concerned with detector running mode, i.e. voltage settings as well as some programmable parameters for frontends electronics.
- Calibration data describes the calibration and the alignment of different subdetectors. Usually quantities are evaluated by running dedicated offline algorithms.
- Parameter data presents the state of detector subsystems. They include a variety of detector settings including the geometry and material definitions.
- Algorithm data is used to control the way algorithms operates. It includes, for example, cuts for selection noise channels excluded from the data processing.

User interfaces for the Condition Database

Application Programming Interface (API):

- integration the database with the experiment frameworks based on the ROOT package
- using stored information in online and offline event data processing

Web service:

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

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

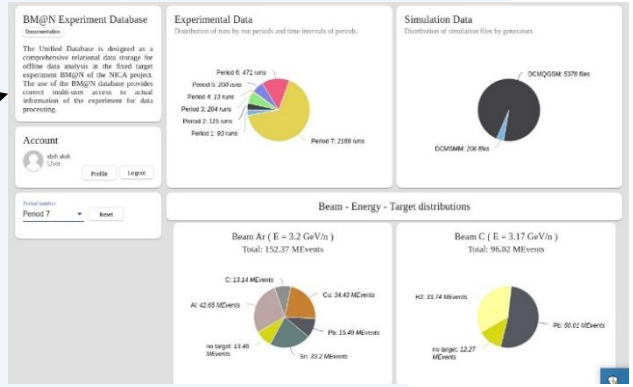
Node.JS

Condition Database

Local Condition Database
as a local replica
of the central Condition Database
is under closer consideration

users

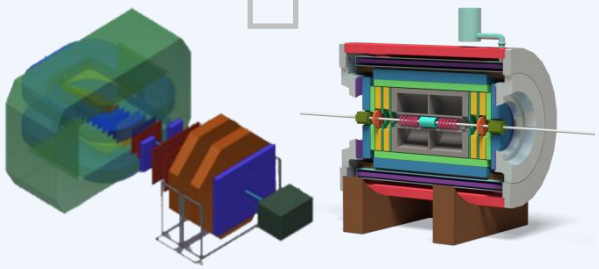
FreeIPA authentication



Web interface

configuration calibration

parameter and algorithm data



Event Metadata System

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 required set of events by various criteria and parameters for further processing and physics analysis
- is responsible for creating, maintaining and checking the quality of the catalogue of physics events

Metadata Structure

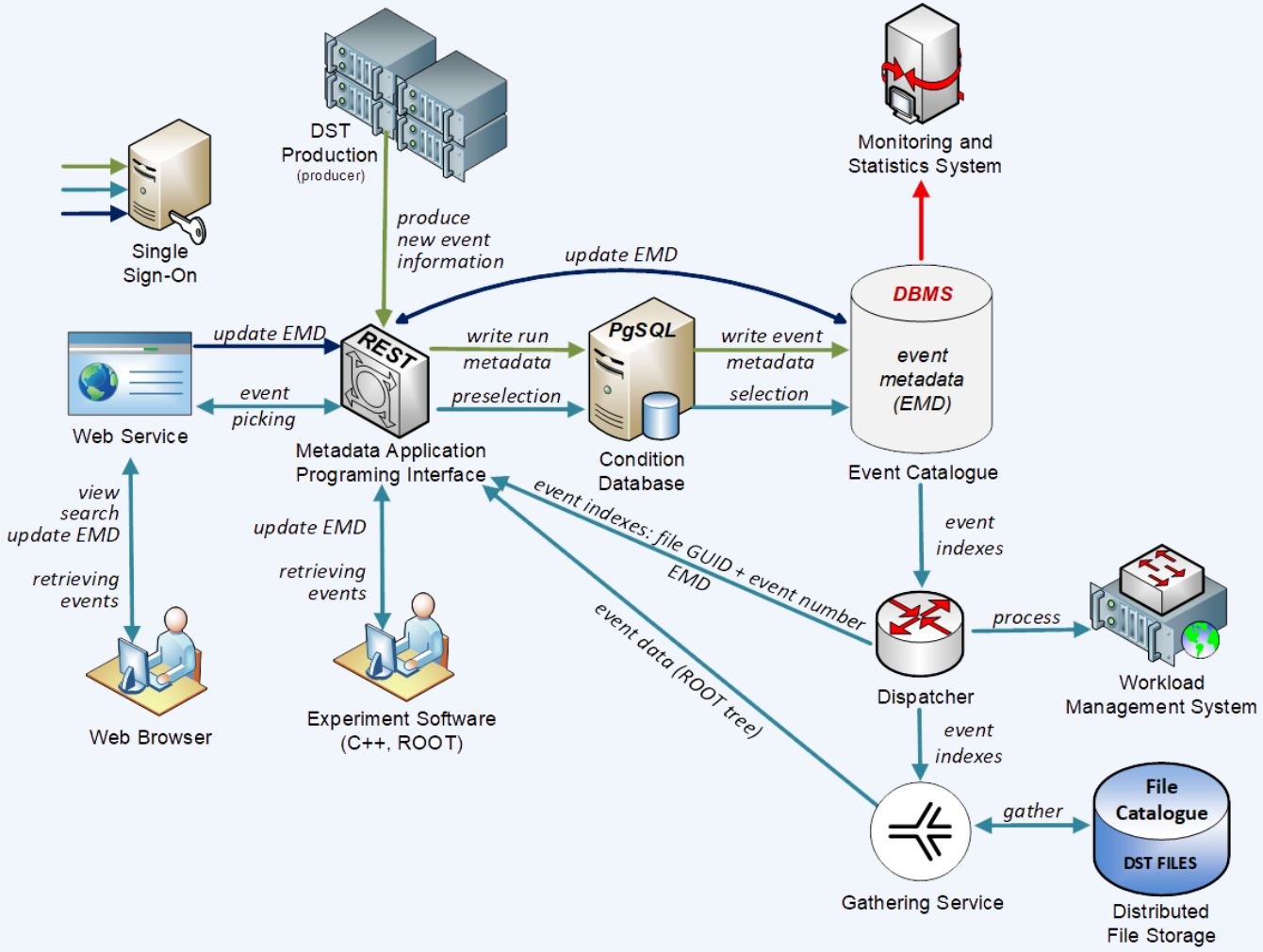
- period and run number

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

- event time
- number of primary and all reconstructed tracks
- track number of positively and negatively charged particles
- primary and secondary particles found
- number of hits by detectors
- total input and output charge in the event
- software version…

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, management and access to the metadata, organization of online and offline interfaces for selecting events of interest

Architecture of the Event Metadata System



Web interface

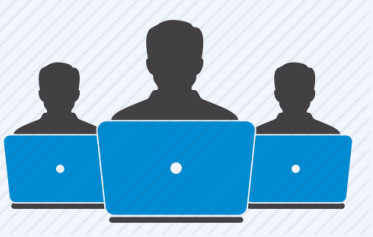
for viewing and searching for event metadata stored in the Event Catalogue and retrieving events which satisfy given user parameters

Metadata API

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

Ecosystem Services

FreeIPA: Single Authentication & Autorization



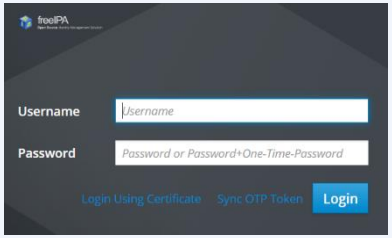
Collaboration members

Single Account

FreeIPA

register

request special rights



bmn-ipa.jinr.ru

admins *writers* *users*

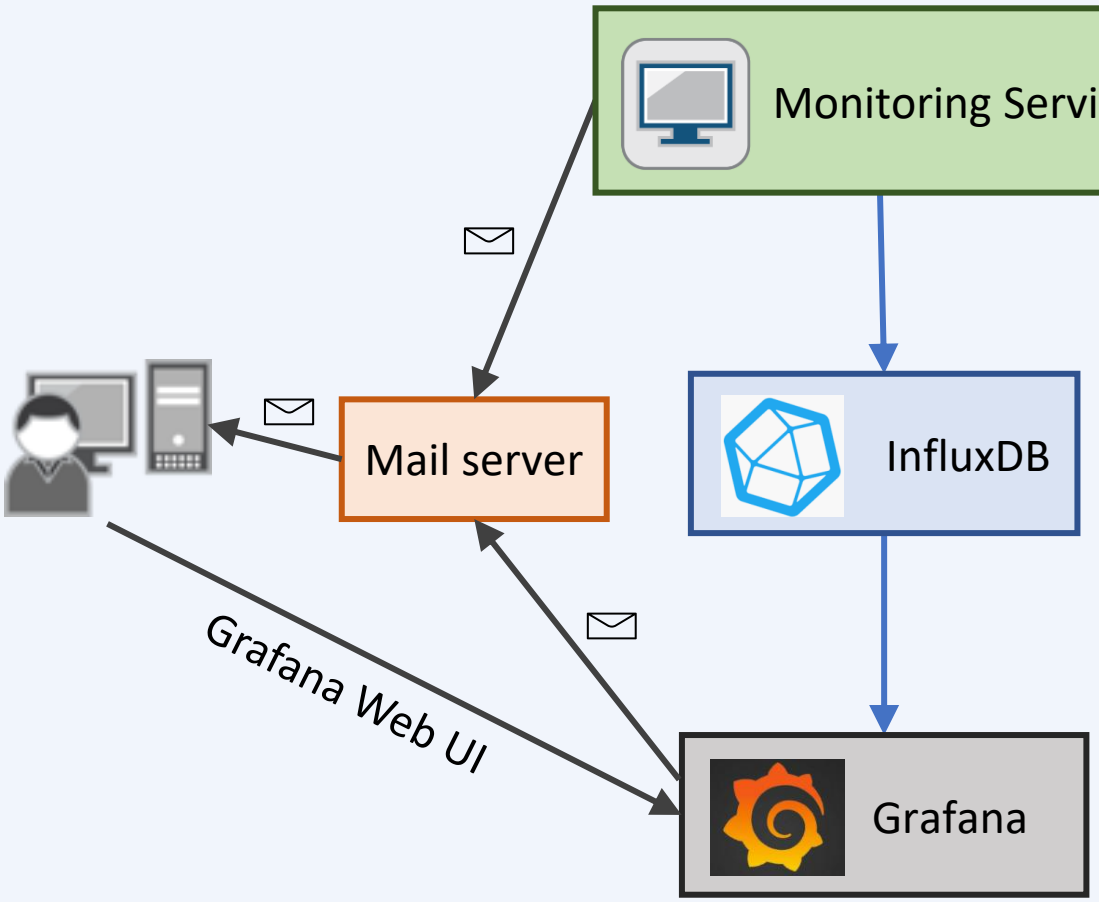
Electronic Logbook

Condition Database

Geometry Database

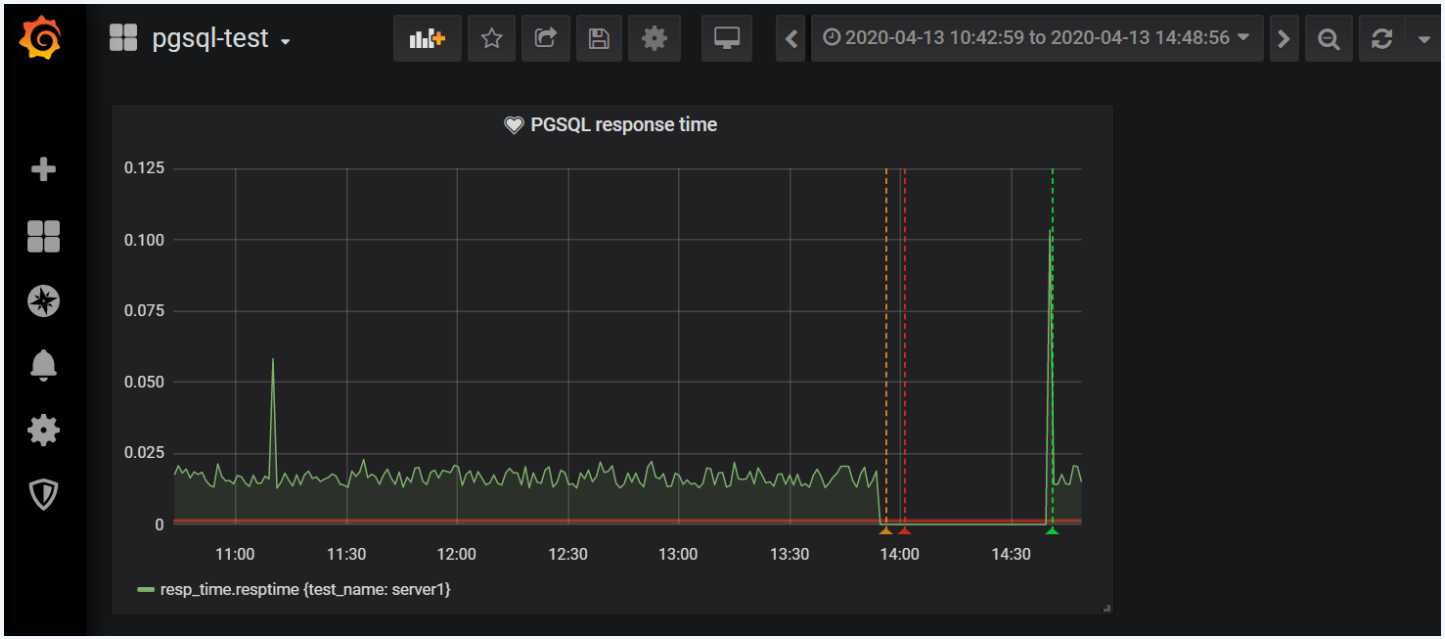
. . .

Monitoring Service

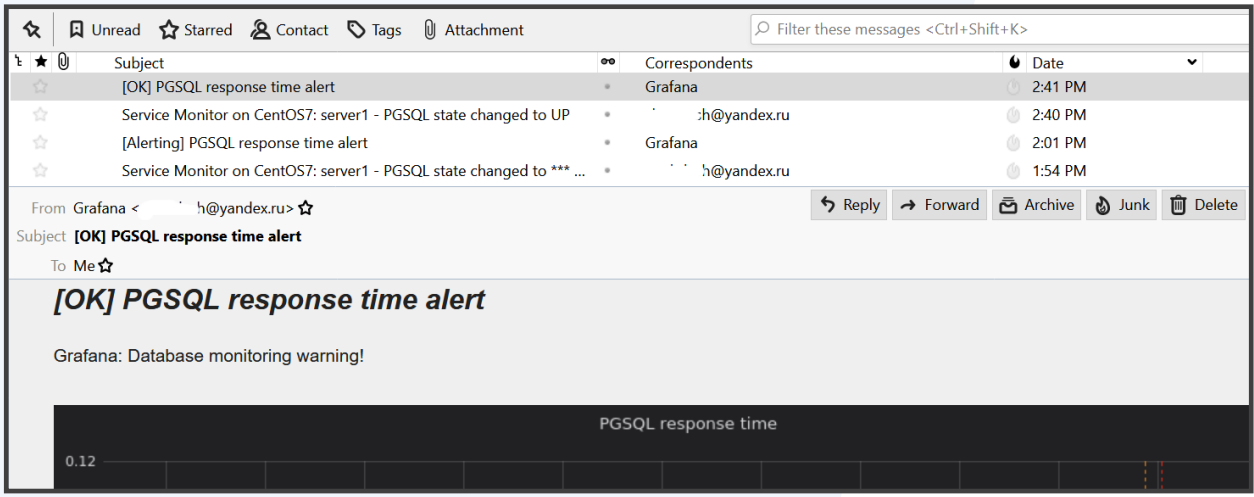


- Monitoring Service Features
  - Ping and PG-SQL request to check database status
  - Configurable via JSON file
  - Email notifications
  - Response time stored in InfluxDB
  - Use Grafana for visualization and additional alerting

Monitoring Information Systems



Grafana View



Email Notifications

Summary

- The complex of the information systems solving the issues of unified multi-user access, data management, actuality and consistency of the information, and data backup will automate the processes of storing, processing and analyzing data, and provide a set of convenient information services for collaboration members.
- The development of the Electronic Logbook Platform and Geometry Information System has been completed, and the design of the Event Metadata System and Condition Database has been finished.
- RFBR support with the NICA grant enables to develop and significantly improve the Information Systems for experimental and simulated data processing.
- The implementation of the information systems is a necessary step for the successful operation of the complex, and high-quality data management will accelerate the achievement of scientific results.

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

thanks to the HybriLIT and NICA cluster teams for computing support

contact email: *gertsen@jinr.ru*