Результаты проделанной работы в рамках Программы от группы МФТИ

Разработка информационных систем и сервисов для эксперимента BM@N

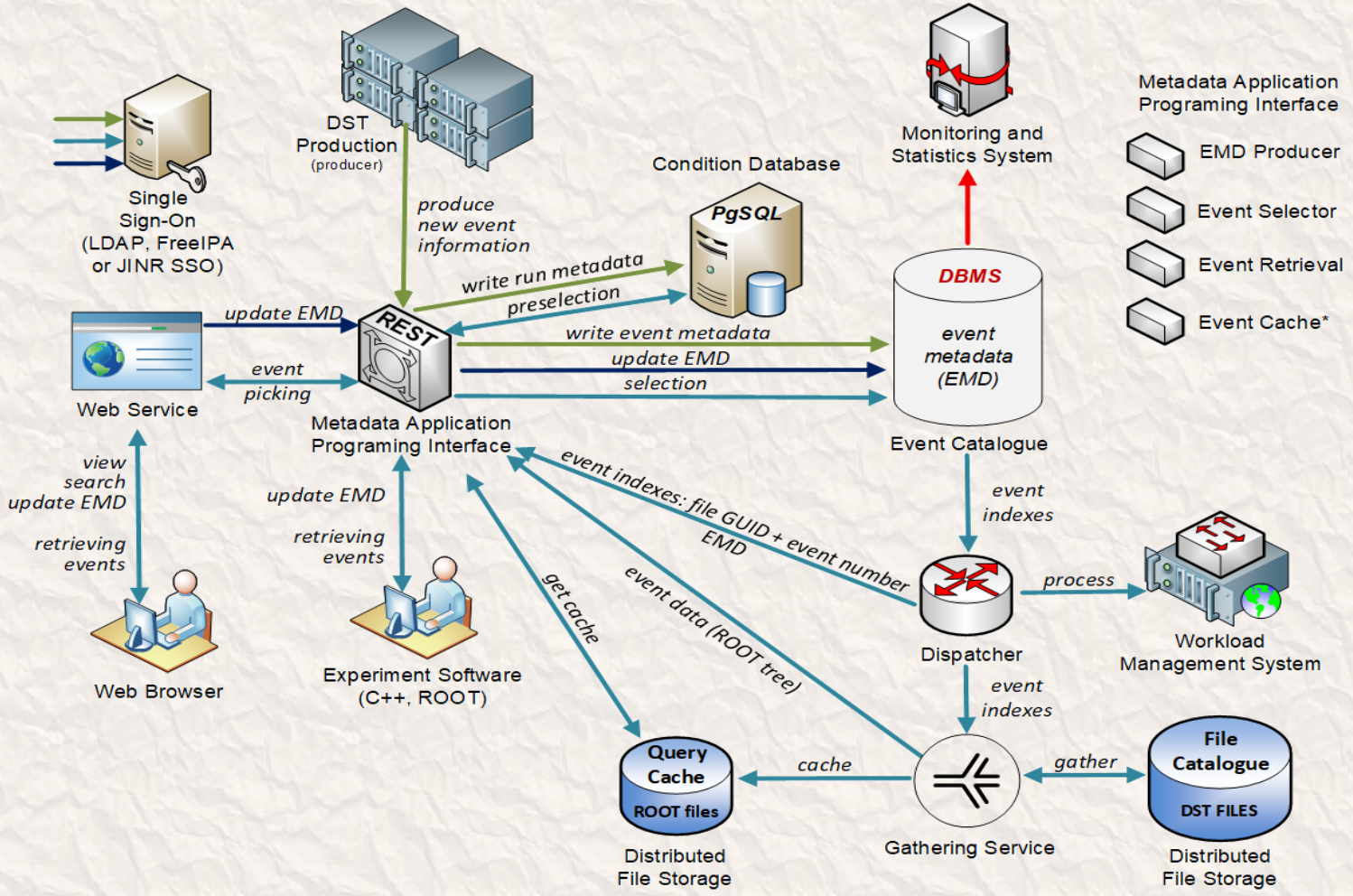
Петр Климай

Current Projects Summary

Project	URL
Разработка компонент системы метаданных физических событий и сервиса мониторинга программных систем эксперимента BM@N	https://git.jinr.ru/nica_db/emd https://git.jinr.ru/pklimai/ems-stat-collector https://git.jinr.ru/pklimai/mon-service-deploy
Разработка решений по автоматическому восстановлению и автоматизированному развертыванию информационных систем эксперимента BM@N	https://git.jinr.ru/pklimai/ems-deploy
Разработка серверной части и сервиса визуализации современной системы графического представления физических событий для эксперимента BM@N	https://git.jinr.ru/idunaev/visionforge
Разработка подсистемы чтения ROOT-файлов с геометрией и событиями эксперимента BM@N для новой системы визуализации физических событий	https://git.jinr.ru/pklimai/visapi

Разработка компонент системы метаданных физических событий и сервиса мониторинга программных систем эксперимента BM@N

Система метаданных физических событий



- Event Metadata System
  - Event Catalogue is based on PostgreSQL
  - Integrates with BM@N Condition database
  - REST API and Web UI developed based on Kotlin multiplatform
  - Configurable to support different metadata
  - ROOT macro to write BM@N events in the catalogue
  - Role-based access control implemented
  - Monitoring

For more details:
E. Alexandrov, I. Alexandrov, A. Chebotov, A. Degtyarev, I. Filozova, K. Gertsenberger, P. Klimai and A. Yakovlev, "Implementation of the Event Metadata System for physics analysis in the NICA experiments", J. Phys.: Conf. Ser. 2438, 012046 (2023).

Новый REST API системы метаданных

- The new scheme is unified for different BM@N Information Systems

GET
POST
DELETE

https://bmn-event.jinr.ru/event_api/v1/event?

run_number=3950:4000&beam_particle=Ar&target_particle=Al
energy=3.16:3.18&target_particle=SRC%20Lead

HOSTNAME / SERVICE / VERSION / ENTITY?parameter_set

HOSTNAME=https://bmn-[SYSNAME].jinr.ru

SERVICE=[SYSNAME]_api

VERSION=v1 (v2…)

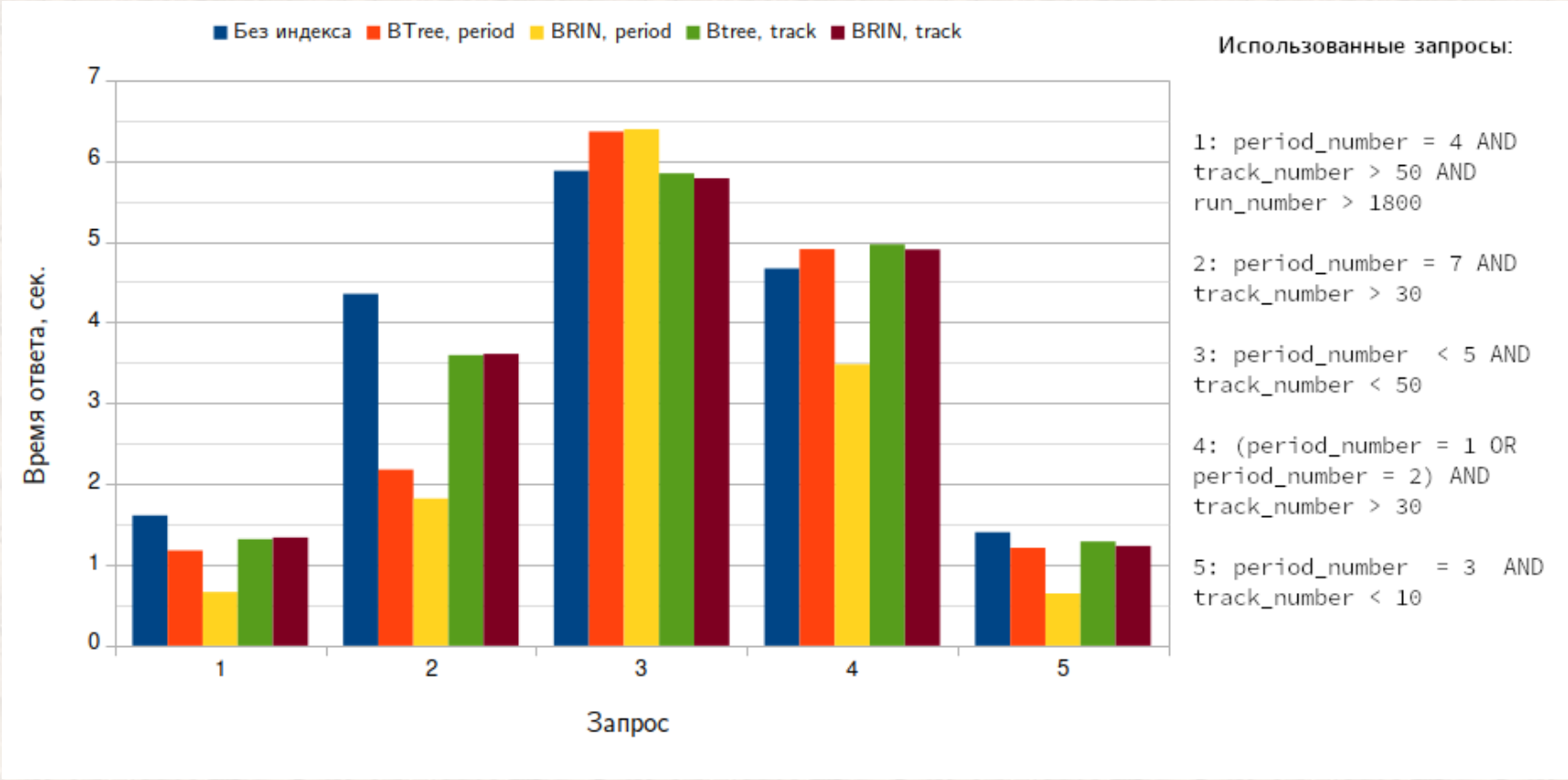
ENTITY=tablename without last '_' (if present)

parameters are separated by '&'
ranges: min:max → >=min AND <=max
min: → >=min :max → <=max

For the Unified Condition Database (UniConDa), SYSNAME = uniconda
For the Event Metadata System (EMS), SYSNAME = event

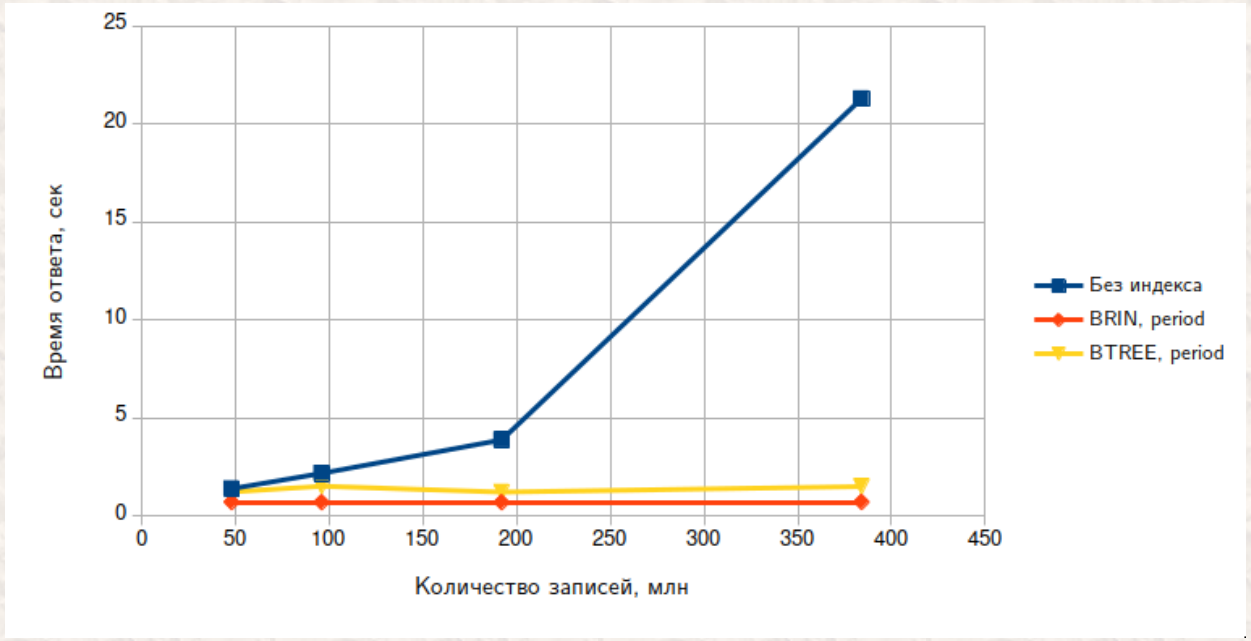
Оптимизация БД Каталога Событий

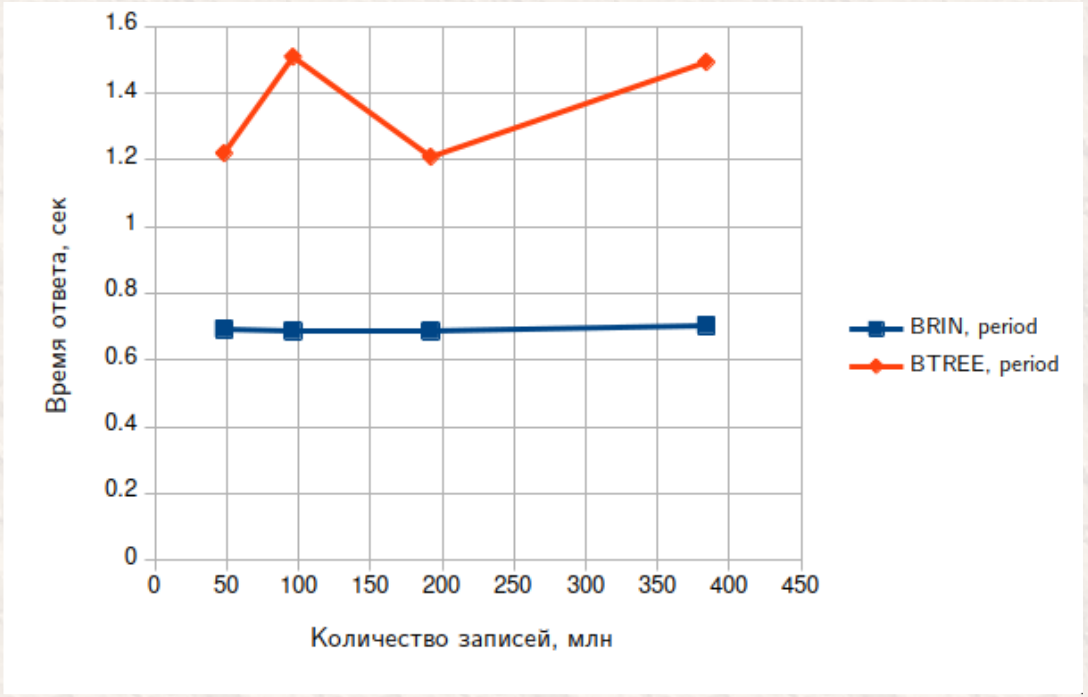
- Measurements with test database instance are shown (50M events)



Время отклика

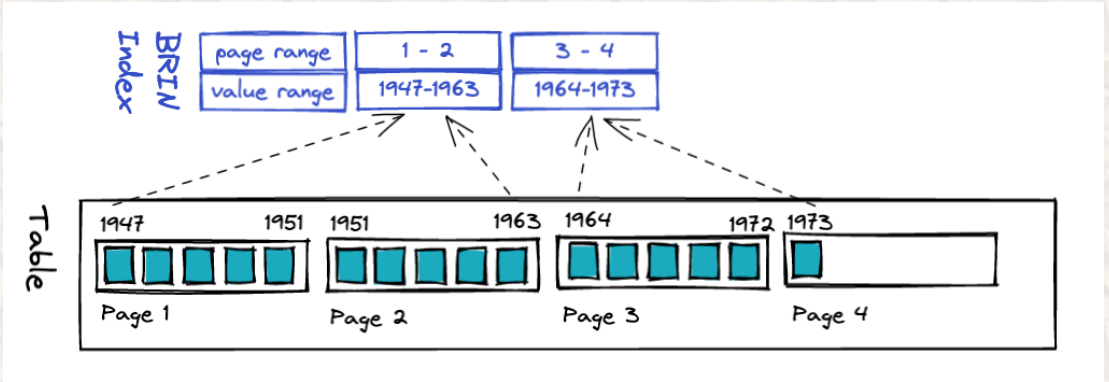
- Adding more periods to test database



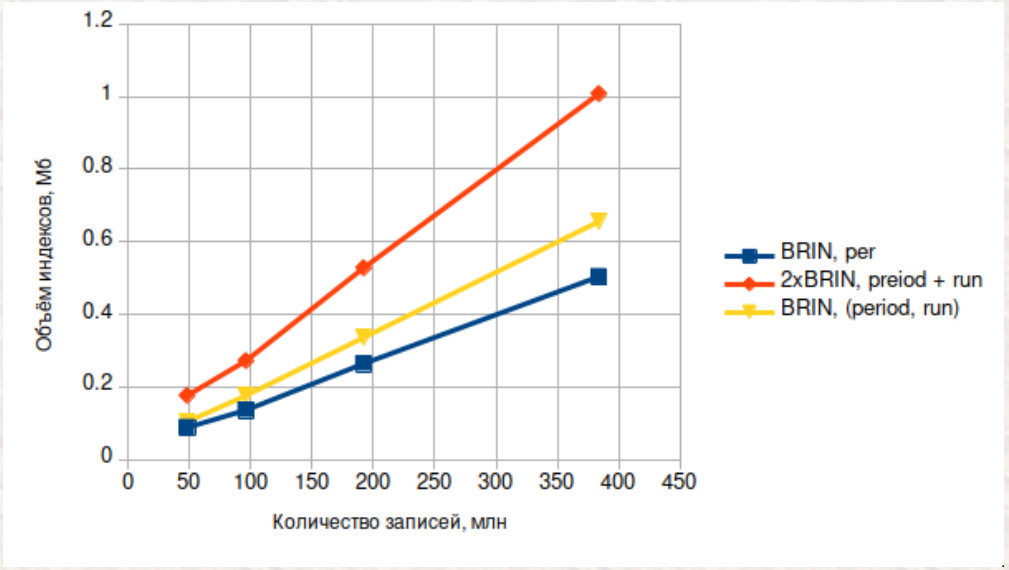


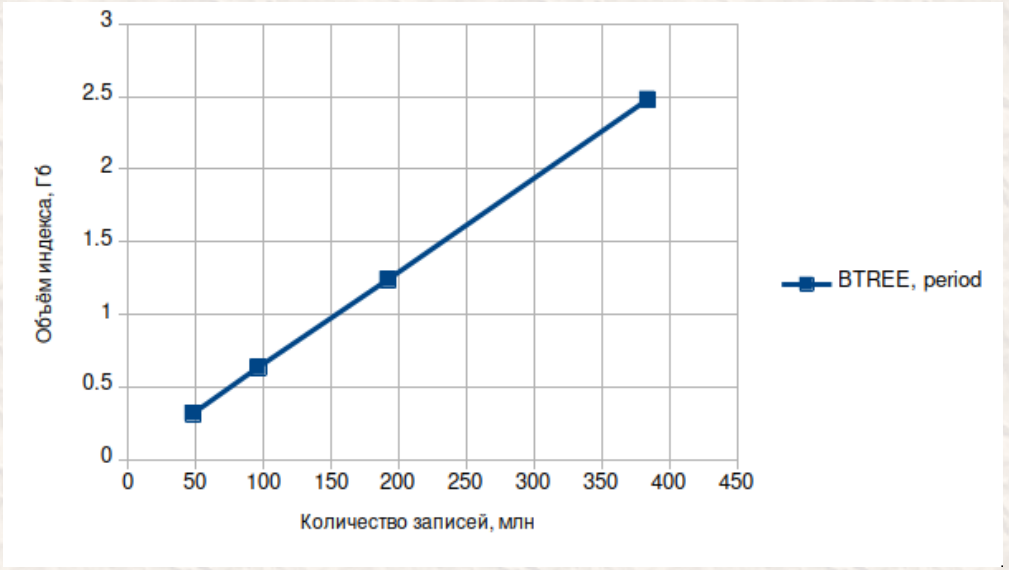
Размеры индексов на диске

- BRIN vs. BTREE
  - Overall, BRIN (Block Range Index) works better for indexing columns having some natural correlation with their physical location within the table



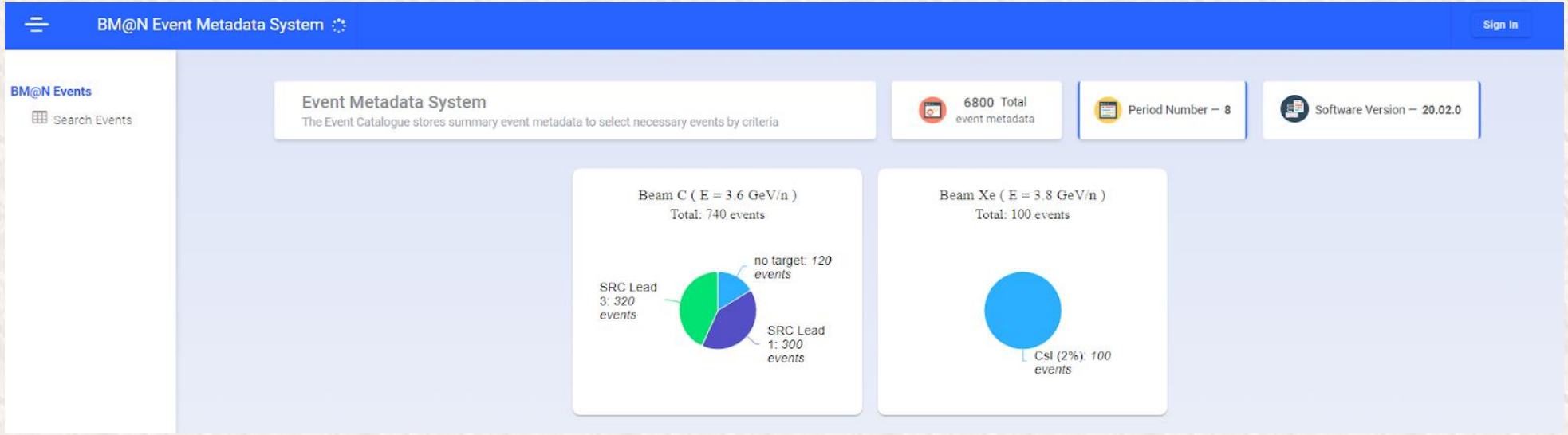
https://www.crunchydata.com/blog/postgres-indexing-when-does-brin-win
https://www.postgresql.org/docs/current/brin-intro.html





Сбор статистики

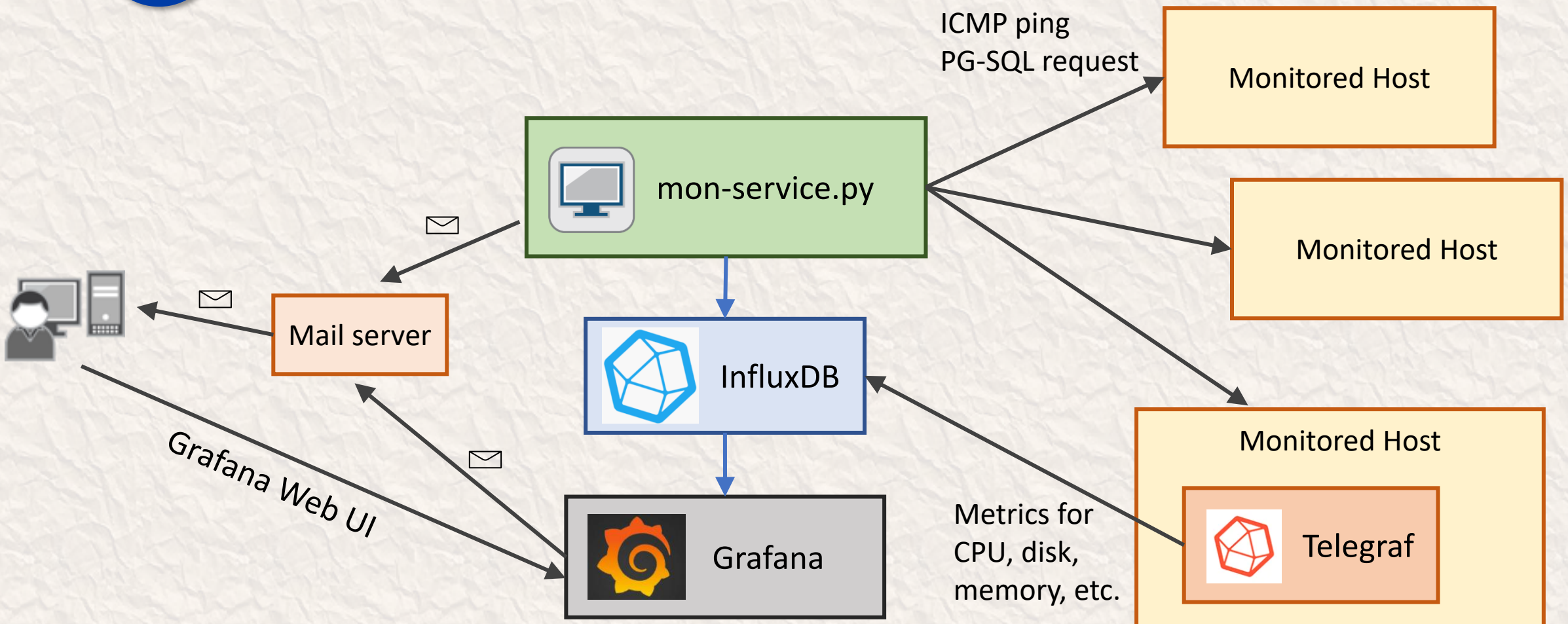
- Разработан скрипт сбора статистики по метаданным в Каталоге событий
  - Статистика сохраняется в БД и отображается в веб-интерфейсе



Monitoring Service - Task

- Monitoring Service Features
  - Ping, PG-SQL, or HTTP request to check 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.

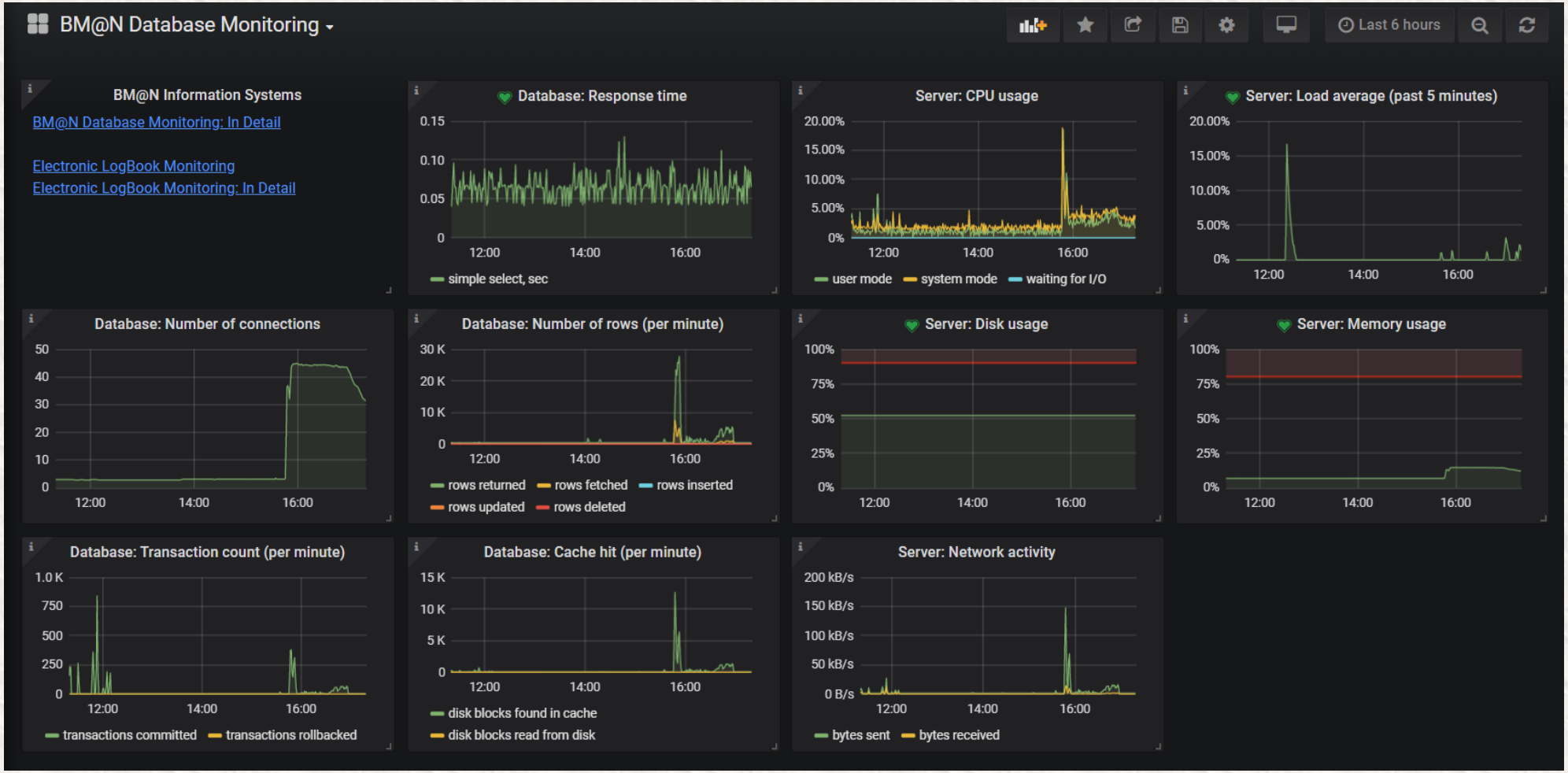
Архитектура сервиса мониторинга



https://mon-service.jinr.ru

Monitoring Service View Example

https://mon-service.jinr.ru



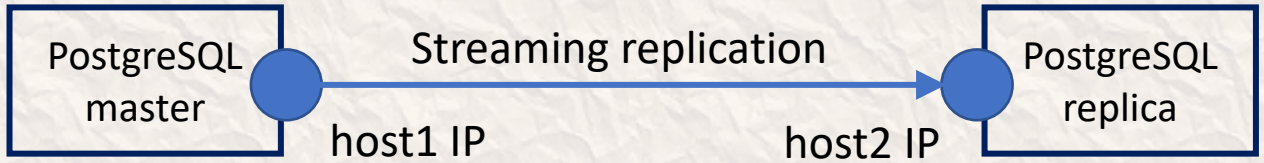
Разработка решений по автоматическому восстановлению и автоматизированному развертыванию информационных систем эксперимента BM@N

High Availability – Task

- Need for HA
  - EMS as well as other IS are essential for timely obtaining physical results of the experiment
  - From client point of view, connection must be initiated to single IP / domain name
    - We do not want to ask client to keep several addresses like primary/secondary ones
  - Considering 2 to 1, active/passive redundancy
  - Need to avoid split brain and no brain scenarios

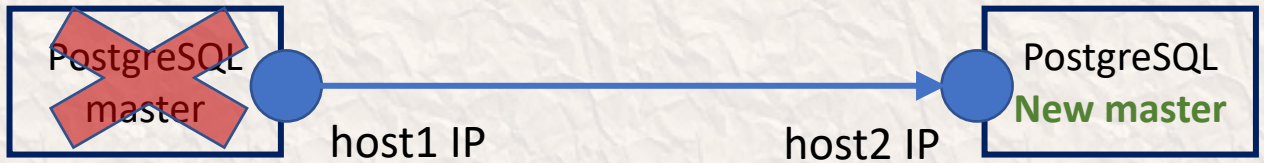
High Availability – Solution

- Base for HA solution
  - PostgreSQL supports streaming replication out of the box (one master to one/many replica servers)
    - https://www.postgresql.org/docs/current/warm-standby.html#STREAMING-REPLICATION
  - Completely synchronous replication is also available (at a performance price)
    - https://www.postgresql.org/docs/current/warm-standby.html#SYNCHRONOUS-REPLICATION



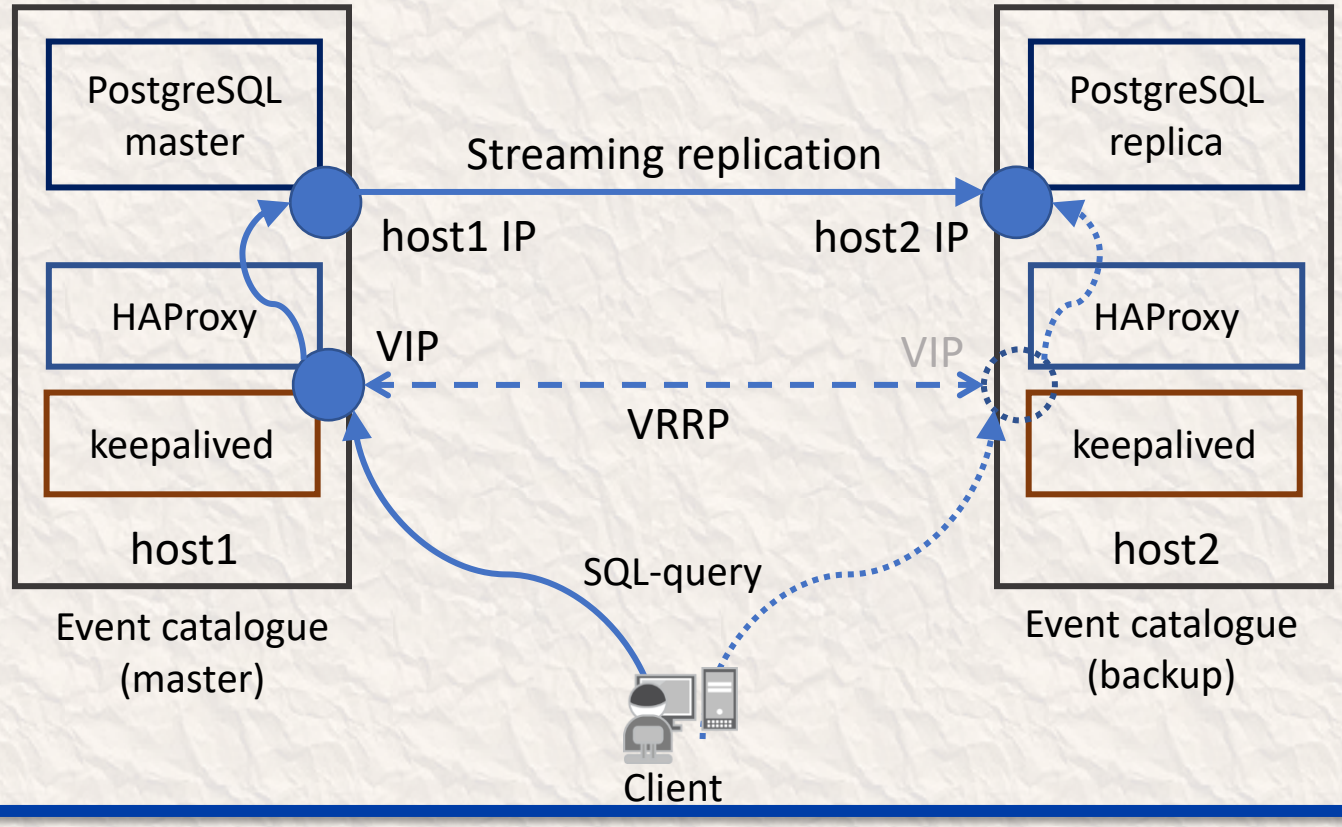
Switchover to new master

- Switchover
  - One command on replica - `pg_ctl promote`
  - Old master must be turned off to avoid split brain
  - Monitoring system can perform switchover (WIP), or it can be done manually
  - It works, but the big question is – where does a client connect?



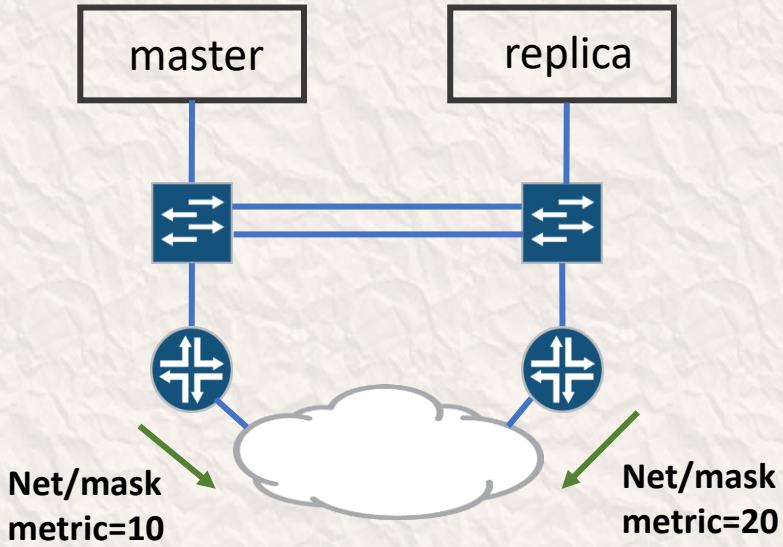
Solution based on VRRP (single L2 domain)

- Keepalived provides virtual IP address for client connection
- This works when both servers are in the same L2 (broadcast) domain



Avoiding single point of failure

- VRRP-based solution can be considered final if:
  - L2-segment is built with redundancy (both for links and switches)
  - VIP's network is announced from at least two routers
  - Not possible to implement without access to network infrastructure



EMS Automated Deployment

- Why automated deployment?
  - Manual deployment of a distributed system is slow and error-prone
  - Automation increases speed and predictability
  - Avoids issue of “forgotten step” in documentation
  - EMS instance may be deployed by other NICA experiments
- Main components of solution
  - Ansible
  - Docker
- Inputs
  - EMS configuration as YAML template
  - Deployment configuration as Ansible variables in hosts file
    - To be replaced by unified JSON config (WIP)

Ansible Playbook example (abbreviated)

```
(env) [lab@alma1 ems-deploy]$ cat deploy-pgsql.pb.yaml
---
- name: Deploy PostgreSQL on Event Catalogue hosts
  hosts: event_catalogue
  become: yes

  tasks:

  - name: Install packages
    dnf: "name={{ item }} state=present"
    with_items:
      - postgresql
      - postgresql-server

  - name: Install Python packages
    pip: "name={{ item }}  state=present"
    with_items:
      - psycopg2-binary

…
```

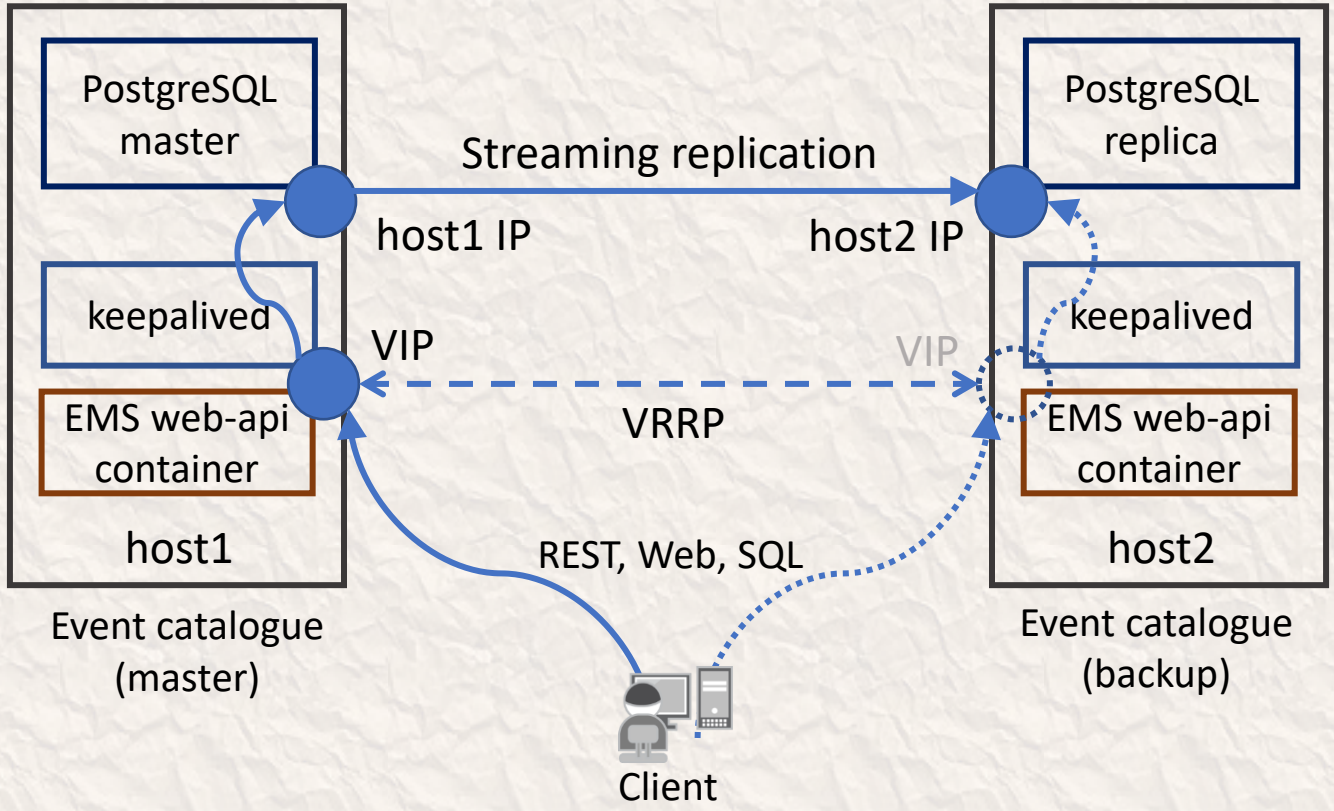
```
…

 - name: Check if PostgreSQL is initialized
    ansible.builtin.stat:
      path: "/var/lib/pgsql/data/pg_hba.conf"
    register: postgres_data

 - name: Initialize PostgreSQL
    command: "postgresql-setup initdb"
    when: not postgres_data.stat.exists

 - name: Start and enable services
    service: "name={{ item }} state=started enabled=yes"
    with_items:
      - postgresql

…
```

Deployment example (abbreviated)

```
[lab@alma1 ems-deploy]$ source env/bin/activate
(env) [lab@alma1 ems-deploy]$ ansible-playbook deploy-pgsql.pb.yaml

PLAY [Deploy PostgreSQL on Event Catalogue hosts] ******************************************

TASK [Gathering Facts] *********************************************************************
ok: [ems2]
ok: [ems1]

TASK [Install packages] *********************************************************************
ok: [ems1] => (item=postgresql)
ok: [ems2] => (item=postgresql)
ok: [ems1] => (item=postgresql-server)
ok: [ems2] => (item=postgresql-server)

…

TASK [Apply SQL schema file] ****************************************************************
changed: [ems1]

PLAY RECAP **********************************************************************************
ems1                       : ok=13   changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
ems2                       : ok=16   changed=4    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0

(env) [lab@alma1 ems-deploy]$ ansible-playbook deploy-vrrp.pb.yaml
(env) [lab@alma1 ems-deploy]$ ansible-playbook deploy-web-api-docker.pb.yaml
```

The Result

- After running the three playbooks:



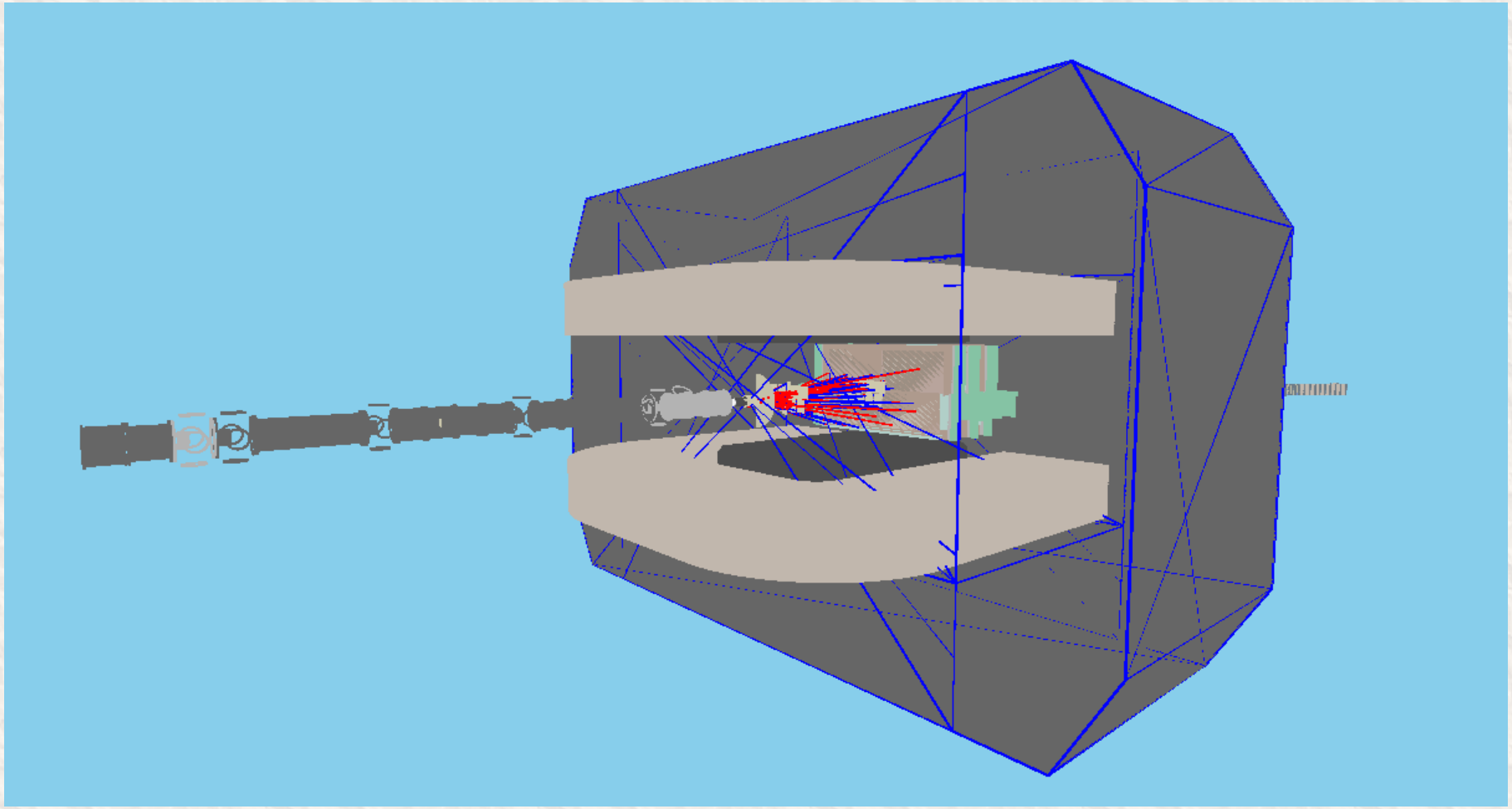
Разработка серверной части и сервиса визуализации современной системы графического представления физических событий для эксперимента BM@N

Разработка подсистемы чтения ROOT-файлов с геометрией и событиями эксперимента BM@N для новой системы визуализации физических событий

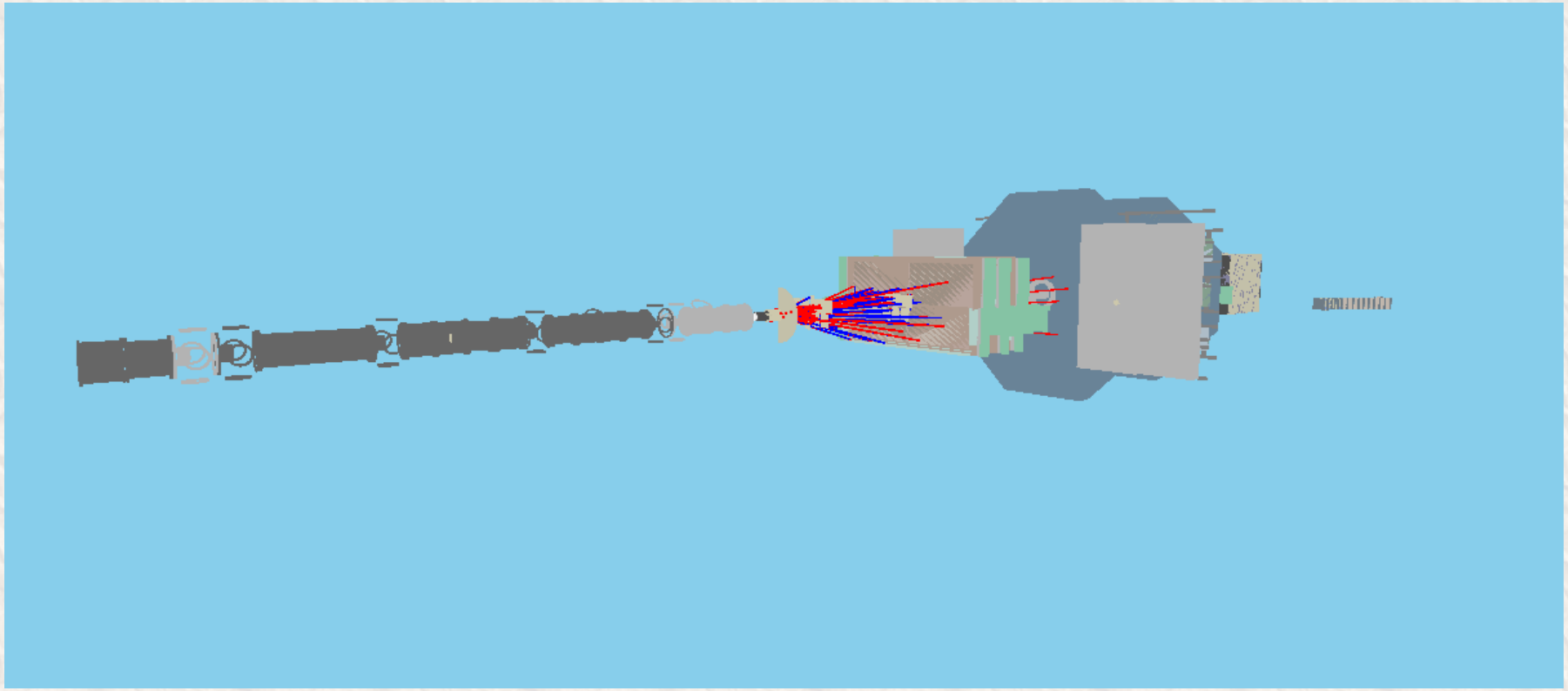
Проект VisionForge

- VisionForge - платформа для создания систем визуализации нового поколения
  - Распределенная динамическая система
    - Модель для визуализации может быть создана на одном узле, передана на другой узел и отрисована там.
    - Узлы могут обмениваться обновлениями модели данных
    - Если в дереве изменилось только одно значение, то только оно будет передано при обновлении
  - Оптимизации и производительность
    - Модель геометрии BM@N включает более 400 000 примитивов
    - Геометрия или группа геометрий может быть определена как прототип - тогда множество объектов может использовать этот прототип, не копируя геометрию при отрисовке, а только добавляя свойства, такие как цвет
  - Разработана на Kotlin-Multiplatform

Пример 1



Пример 2

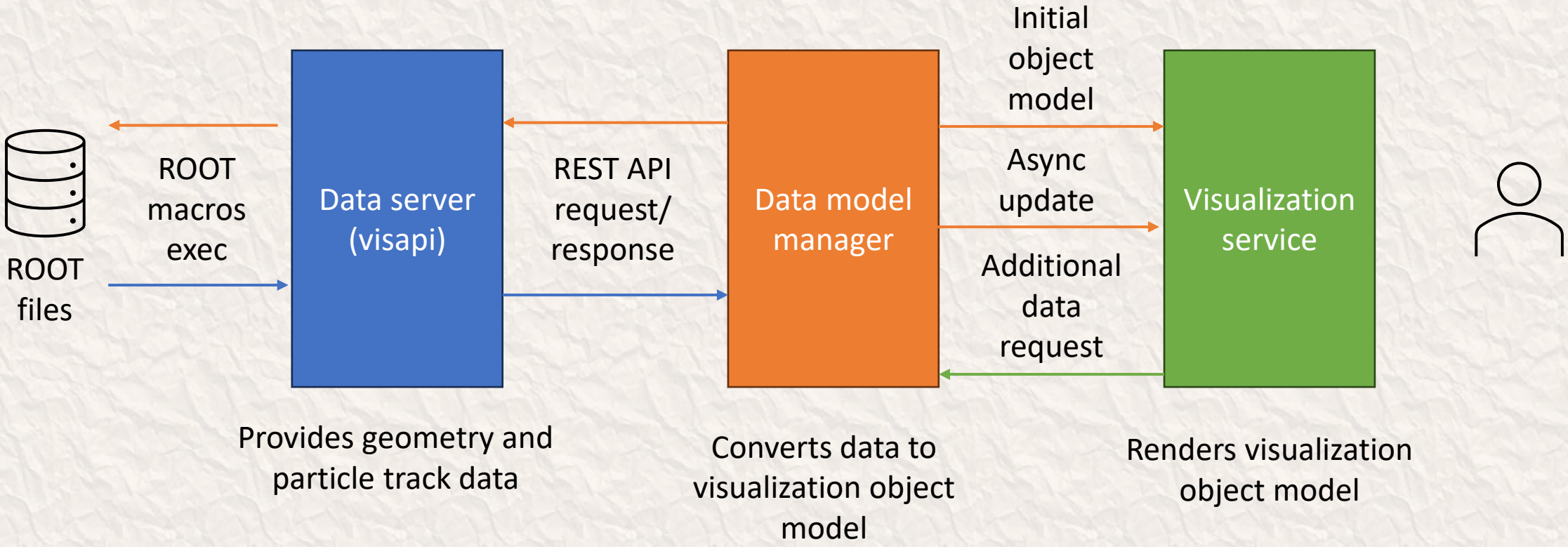


CERN ROOT integration

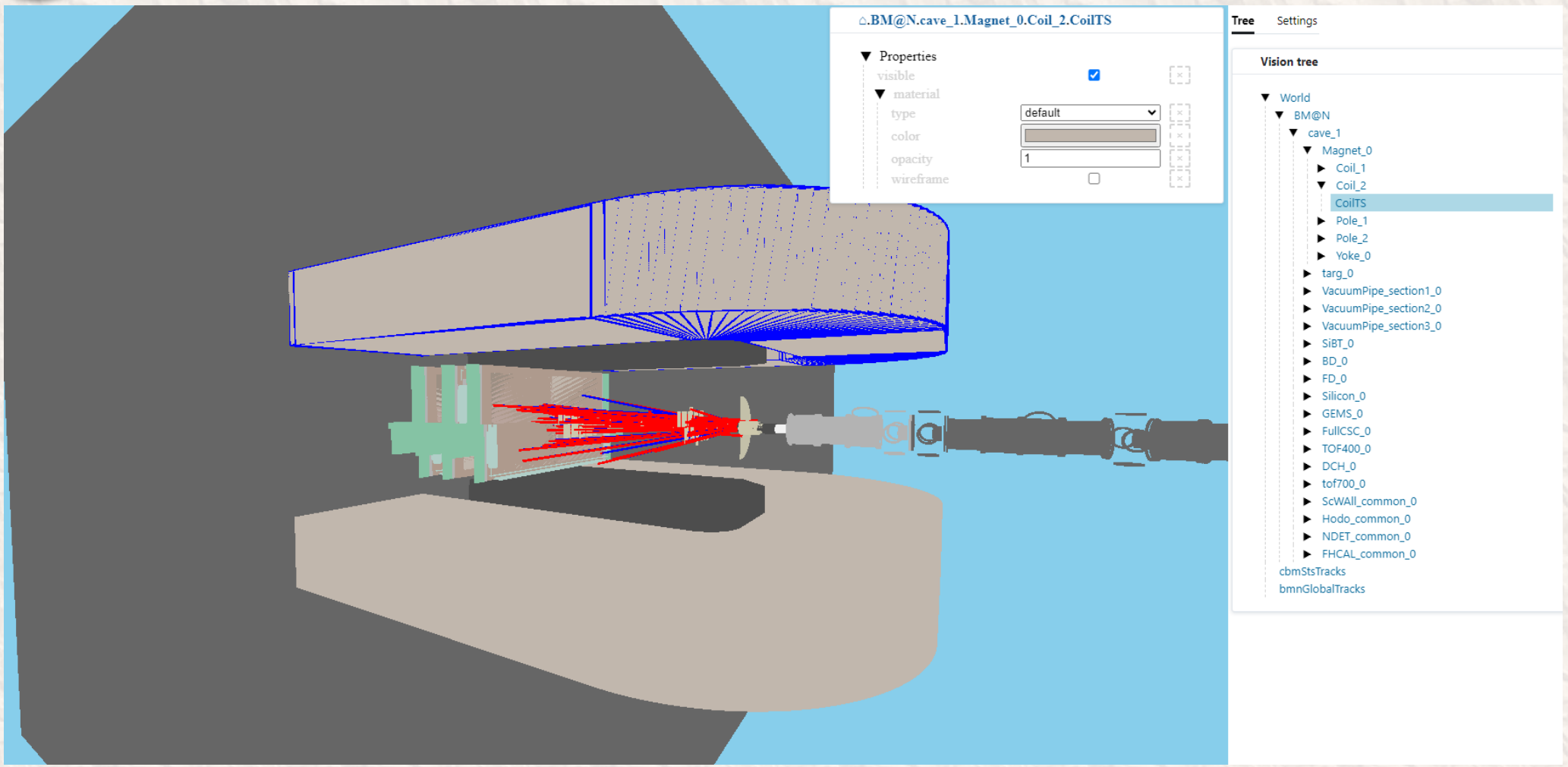
Possible approaches:

- Read ROOT format and convert it to Vision Object Model (**KRootIO** project)
- Create a ROOT plugin that converts TGeoManager to VOM on the ROOT side.
- Convert TGeoManager to JSON via TBufferJSON (**visapi** project)

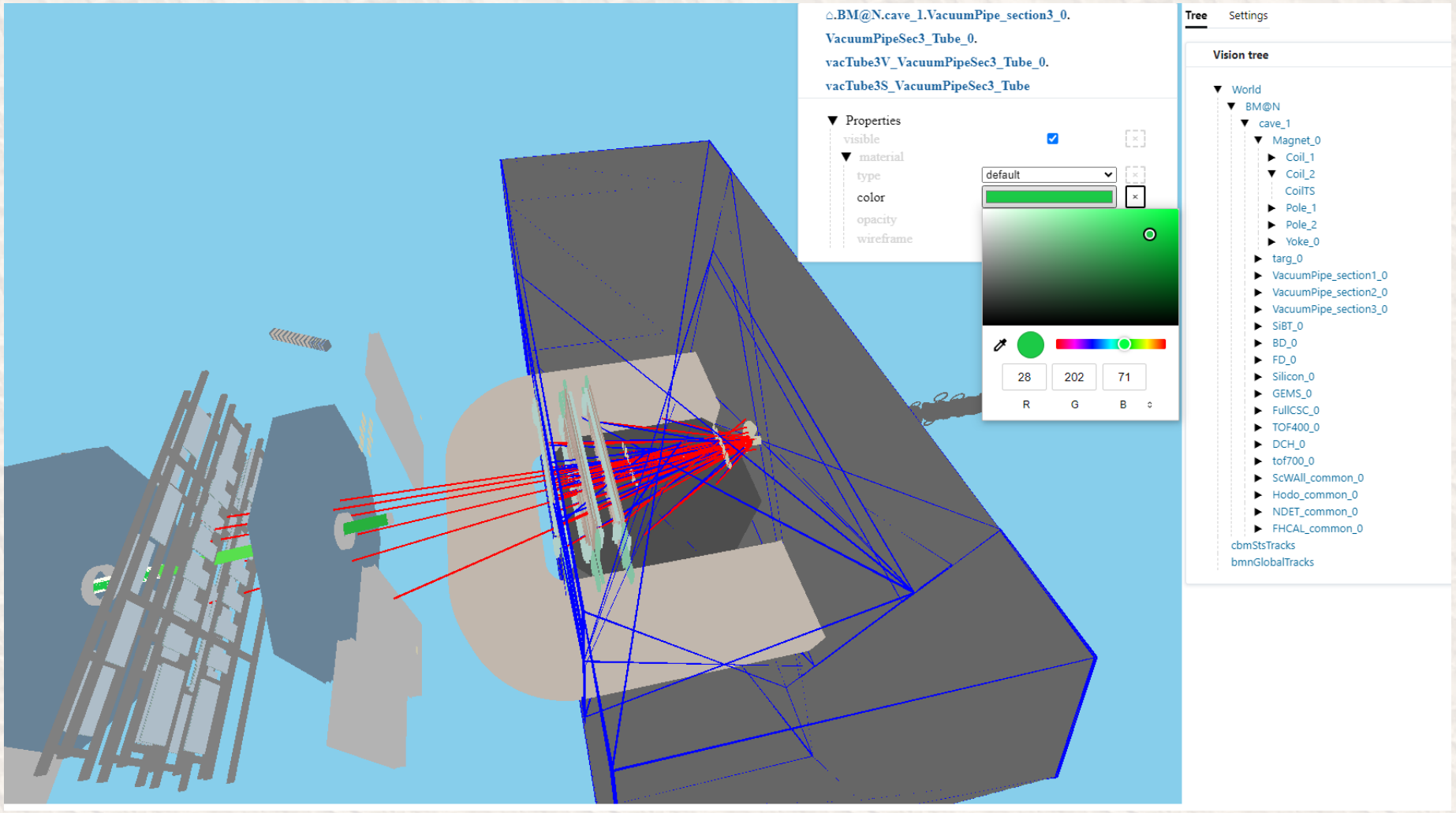
Общая схема работы системы визуализации VisionForge при отображении данных из ROOT-файлов



Scene graph


⌂.BM@N.cave_1.Magnet_0.Coil_2.CoilTS
Properties
visible
material
type
default
color
opacity
1
wireframe
Tree
Settings
Vision tree
World
BM@N
cave_1
Magnet_0
Coil_1
Coil_2
CoilTS
Pole_1
Pole_2
Yoke_0
targ_0
VacuumPipe_section1_0
VacuumPipe_section2_0
VacuumPipe_section3_0
SiBT_0
BD_0
FD_0
Silicon_0
GEMS_0
FullCSC_0
TOF400_0
DCH_0
tof700_0
ScWAll_common_0
Hodo_common_0
NDET_common_0
FHCAL_common_0
cbmStsTracks
bmnGlobalTracks

Настройки объектов (невидимость и перекраска)



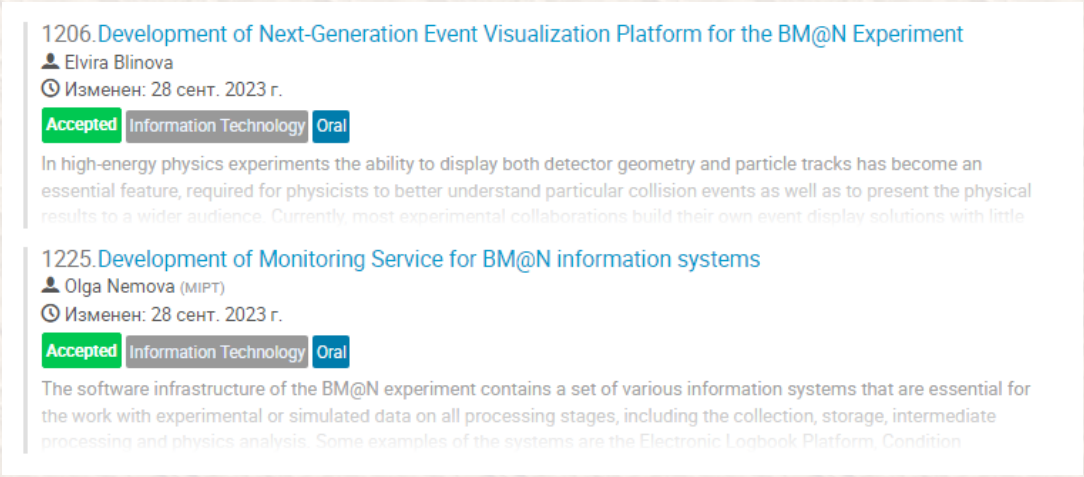
Статьи и доклады

Статьи:

- K. Gertsenberger, I. Pelevanyuk, P. Klimai, A.Chebotov, “Computing software architecture of the BM@N experiment”, направлена в ЭЧАЯ.
- И.А. Дунаев, К.В. Герценбергер, П.А. Климай, “Статус разработки каталога физических событий для эксперимента BM@N”, будет опубликовано в Трудах 65-й Всероссийской научной конференции МФТИ.

Доклады:

- Доклады П. Климая и А. Нозика на 10th BM@N collaboration meeting (Май 2023)
- И.А. Дунаев, К.В. Герценбергер, П.А. Климай - “Статус разработки каталога физических событий для эксперимента BM@N”, доклад на 65-й Всероссийской научной конференции МФТИ (3-8 апреля 2023 года)
- Доклады, поданные на AYSS-2023 (30.10-3.11.2023):


1206. Development of Next-Generation Event Visualization Platform for the BM@N Experiment
Elvira Blinova
Изменен: 28 сент. 2023 г.
Accepted Information Technology Oral
In high-energy physics experiments the ability to display both detector geometry and particle tracks has become an essential feature, required for physicists to better understand particular collision events as well as to present the physical

1225. Development of Monitoring Service for BM@N information systems
Olga Nemova (MIPT)
Изменен: 28 сент. 2023 г.
Accepted Information Technology Oral
The software infrastructure of the BM@N experiment contains a set of various information systems that are essential for the work with experimental or simulated data on all processing stages, including the collection, storage, intermediate

Спасибо!