Activities and workplans of the MIPT Group for development of BM@N software systems

Peter Klimai < pklimai@gmail.com >

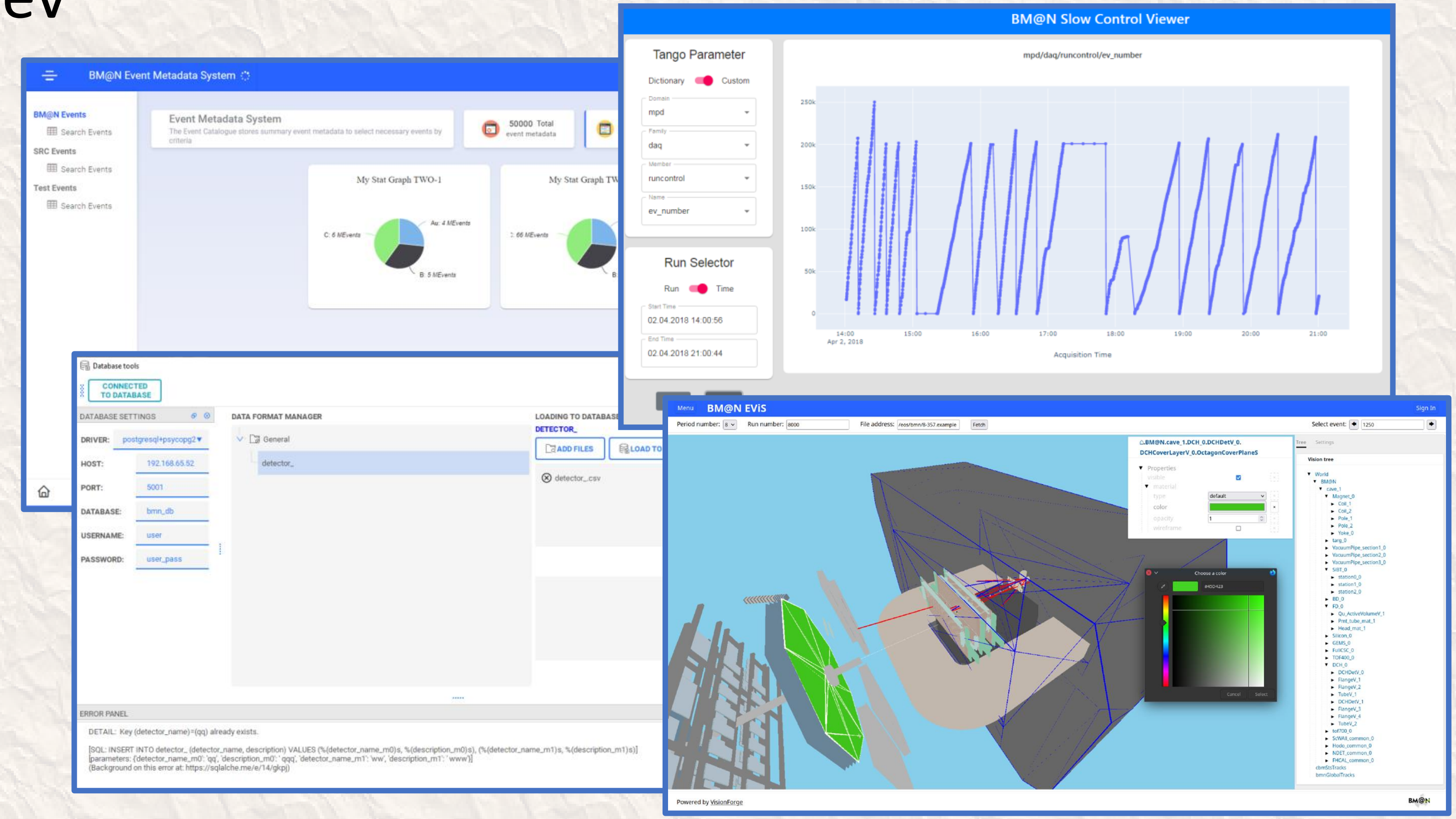
the MIPT team for the BM@N collaboration

MIPT Software for BM@N – Team

Supervision: T. A.-Kh. Aushev

Team members:

- P. Klimai
- A. Nozik
- O. Nemova (student 6y)
- I. Dunaev (student 6y)
- V. Kaplenko (student 6y)
- A. Degtyarev (PhD st. 2y)
- S. Efimov (graduated)



BM@N Event Metadata System

Event Metadata System

The Event Catalogue stores summary event metadata to select necessary events by criteria

50000 Total event metadata

BM@N Slow Control Viewer

Tango Parameter

Dictionary ● Custom

Domain: mpd

Family: daq

Member: runcontrol

Name: ev_number

Run Selector

Run ● Time

Start Time: 02.04.2018 14:00:56

End Time: 02.04.2018 21:00:44

Database tools

CONNECTED TO DATABASE

DATABASE SETTINGS

DRIVER: postgresql+psycopg2

HOST: 192.168.65.52

PORT: 5001

DATABASE: bmn_db

USERNAME: user

PASSWORD: user_pass

```
DETAIL: Key (detector_name)=(qq) already exists.

[SQL: INSERT INTO detector_ (detector_name, description) VALUES (%(detector_name_m0)s, %(description_m0)s), (%(detector_name_m1)s, %(description_m1)s)]
[parameters: {'detector_name_m0': 'qq', 'description_m0': ' qqq', 'detector_name_m1': 'ww', 'description_m1': ' www'}]
(Background on this error at: https://sqlalche.me/e/14/gkpj)
```

BM@N EVIS

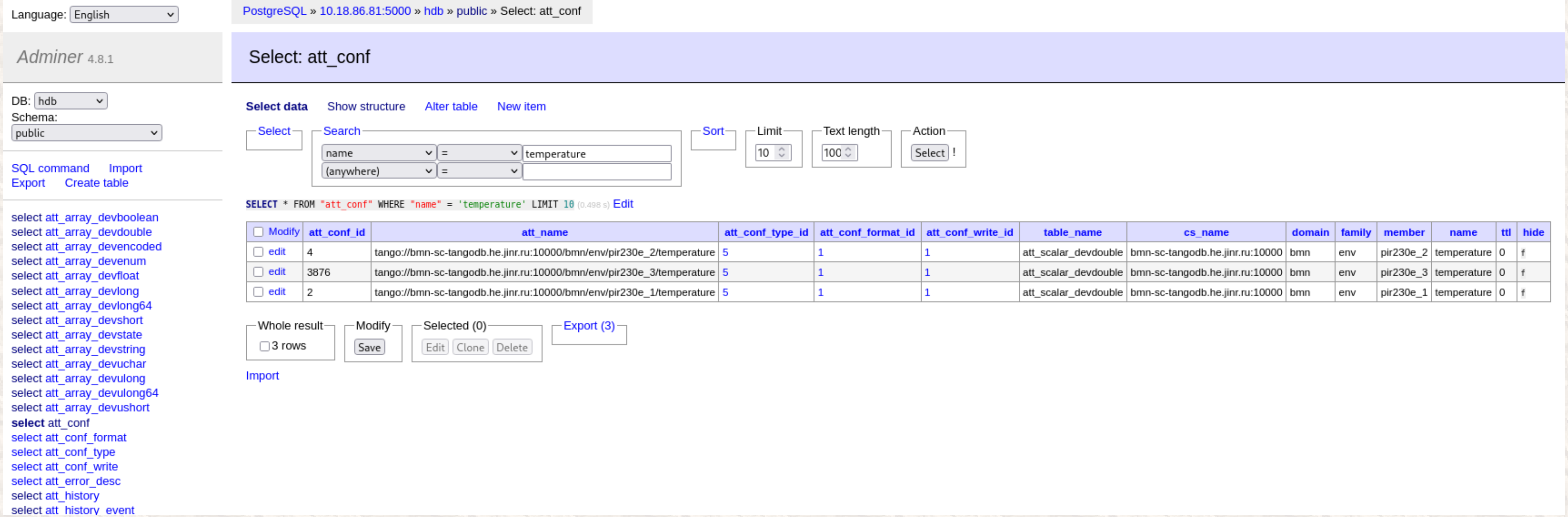
Main Projects Summary

Project	URL
Event Metadata System	https://git.jinr.ru/nica_db/emd https://git.jinr.ru/pklimai/ems-stat-collector https://git.jinr.ru/pklimai/ems-deploy
Next-generation Event Display	https://git.jinr.ru/idunaev/visionforge https://git.jinr.ru/pklimai/visapi
Monitoring Service	https://git.jinr.ru/pklimai/mon-service-deploy
Development of REST API and Web interfaces for slow control system	https://git.jinr.ru/pklimai/architect https://git.jinr.ru/pklimai/tango-api

Development of REST API and Web interfaces for slow control system

BM@N slow control system database

- Updated version of Tango slow control database uses PostgreSQL
- Convenient REST API access is required



Search: name = temperature; (anywhere) =
Limit: 10
Text length: 100
Select !

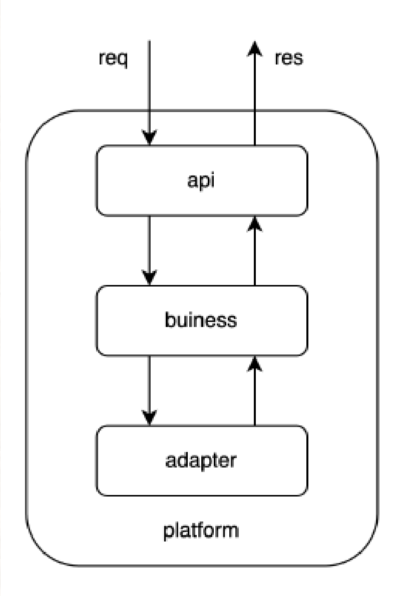
```
SELECT * FROM "att_conf" WHERE "name" = 'temperature' LIMIT 10 (0.498 s) Edit
```

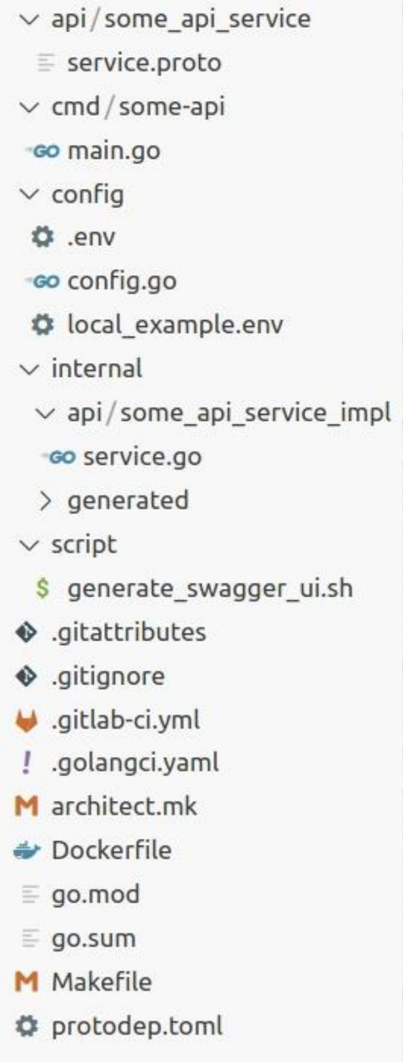
□ Modify	att_conf_id	att_name	att_conf_type_id	att_conf_format_id	att_conf_write_id	table_name	cs_name	domain	family	member	name	ttl	hide
□ edit	4	tango://bmn-sc-tangodb.he.jinr.ru:10000/bmn/env/pir230e_2/temperature	5	1	1	att_scalar_devdouble	bmn-sc-tangodb.he.jinr.ru:10000	bmn	env	pir230e_2	temperature	0	f
□ edit	3876	tango://bmn-sc-tangodb.he.jinr.ru:10000/bmn/env/pir230e_3/temperature	5	1	1	att_scalar_devdouble	bmn-sc-tangodb.he.jinr.ru:10000	bmn	env	pir230e_3	temperature	0	f
□ edit	2	tango://bmn-sc-tangodb.he.jinr.ru:10000/bmn/env/pir230e_1/temperature	5	1	1	att_scalar_devdouble	bmn-sc-tangodb.he.jinr.ru:10000	bmn	env	pir230e_1	temperature	0	f

□ 3 rows

API Service Development

- “Architect” service was developed by Sergey Efimov
  - Creates a skeleton for API service
  - Used technologies: Go, Docker, GitLab CI
  - Supported API types: REST and gRPC
  - https://gitlab.com/zigal0/architect
- Actual TANGO API microservice
  - https://gitlab.com/zigal0-group/nica/tango-api
- Considering to use this approach for other services as well

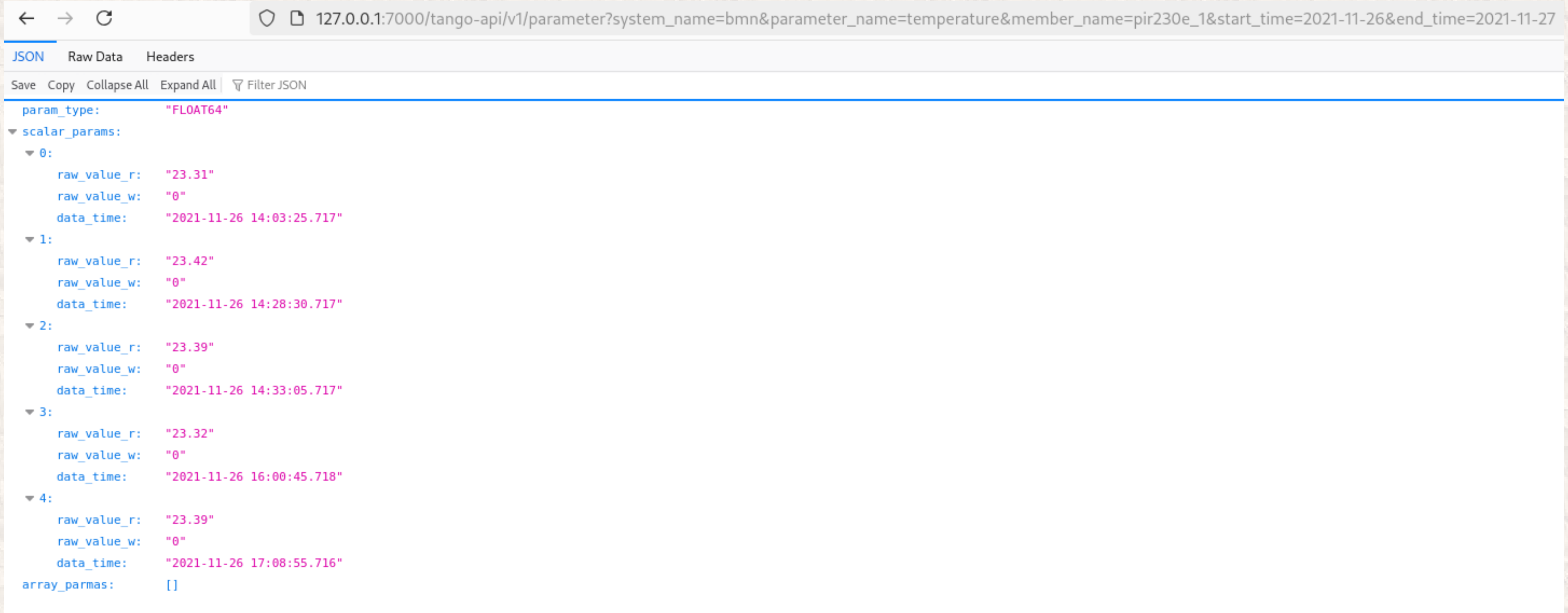




```
∨ api / some_api_service
    service.proto
∨ cmd / some-api
    main.go
∨ config
    .env
    config.go
    local_example.env
∨ internal
  ∨ api / some_api_service_impl
      service.go
  > generated
∨ script
    generate_swagger_ui.sh
  .gitattributes
  .gitignore
  .gitlab-ci.yml
  .golangci.yaml
  architect.mk
  Dockerfile
  go.mod
  go.sum
  Makefile
  protodep.toml
```

REST API call example

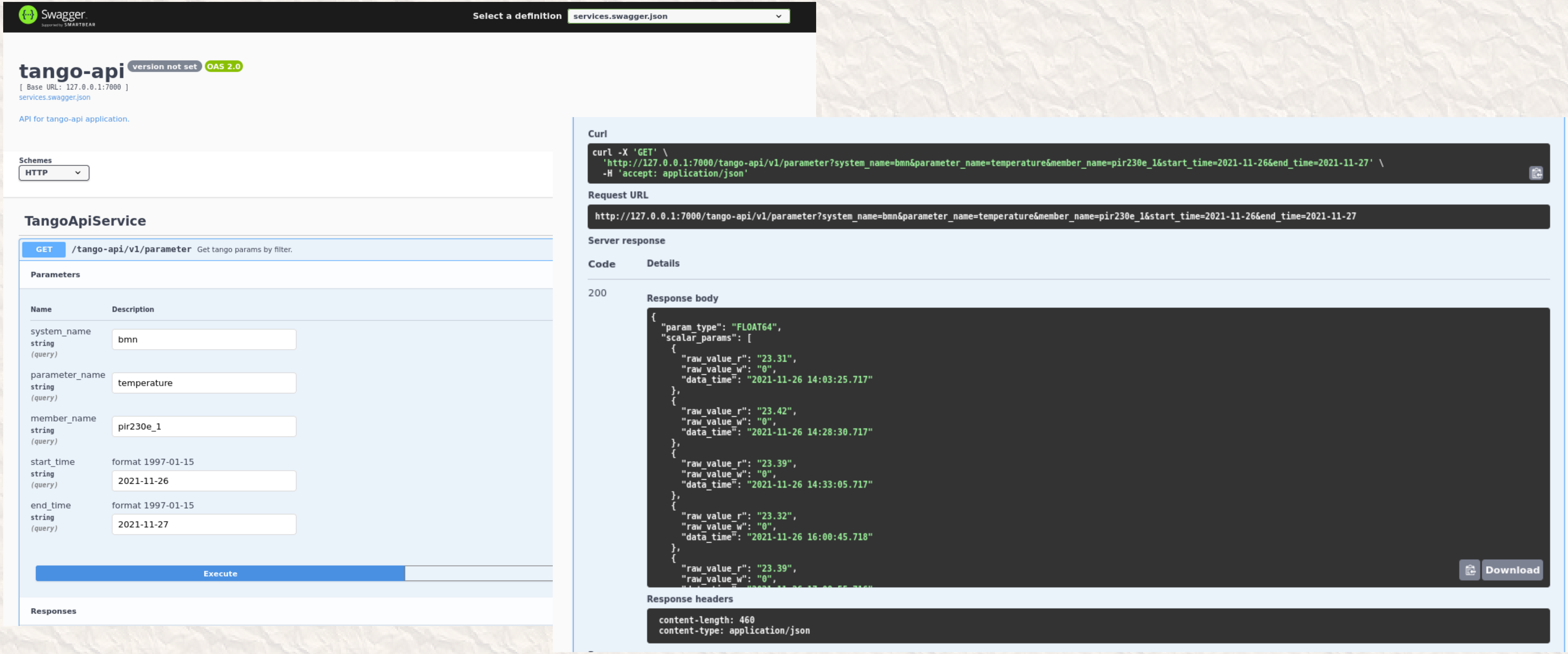
- **http(s)://<host>:7000/tango-api/v1/parameter**?**system_name=**bmn&**parameter_name=**temperature&**member_name=**pir230e_1&**start_time**=2021-11-26&**end_time**=2021-11-27



127.0.0.1:7000/tango-api/v1/parameter?system_name=bmn&parameter_name=temperature&member_name=pir230e_1&start_time=2021-11-26&end_time=2021-11-27

```
param_type:          "FLOAT64"
▾ scalar_params:
  ▾ 0:
      raw_value_r:   "23.31"
      raw_value_w:   "0"
      data_time:     "2021-11-26 14:03:25.717"
  ▾ 1:
      raw_value_r:   "23.42"
      raw_value_w:   "0"
      data_time:     "2021-11-26 14:28:30.717"
  ▾ 2:
      raw_value_r:   "23.39"
      raw_value_w:   "0"
      data_time:     "2021-11-26 14:33:05.717"
  ▾ 3:
      raw_value_r:   "23.32"
      raw_value_w:   "0"
      data_time:     "2021-11-26 16:00:45.718"
  ▾ 4:
      raw_value_r:   "23.39"
      raw_value_w:   "0"
      data_time:     "2021-11-26 17:08:55.716"
array_parmas:        []
```

OpenAPI / Swagger page



tango-api

[Base URL: 127.0.0.1:7000]
services.swagger.json

API for tango-api application.

Schemes: HTTP

TangoApiService

GET `/tango-api/v1/parameter` Get tango params by filter.

Parameters

Name	Description
system_name string *(query)*	bmn
parameter_name string *(query)*	temperature
member_name string *(query)*	pir230e_1
start_time string *(query)*	format 1997-01-15 2021-11-26
end_time string *(query)*	format 1997-01-15 2021-11-27

Responses

Curl

```
curl -X 'GET' \
  'http://127.0.0.1:7000/tango-api/v1/parameter?system_name=bmn&parameter_name=temperature&member_name=pir230e_1&start_time=2021-11-26&end_time=2021-11-27' \
  -H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:7000/tango-api/v1/parameter?system_name=bmn&parameter_name=temperature&member_name=pir230e_1&start_time=2021-11-26&end_time=2021-11-27
```

Server response

Code **Details**

200

Response body

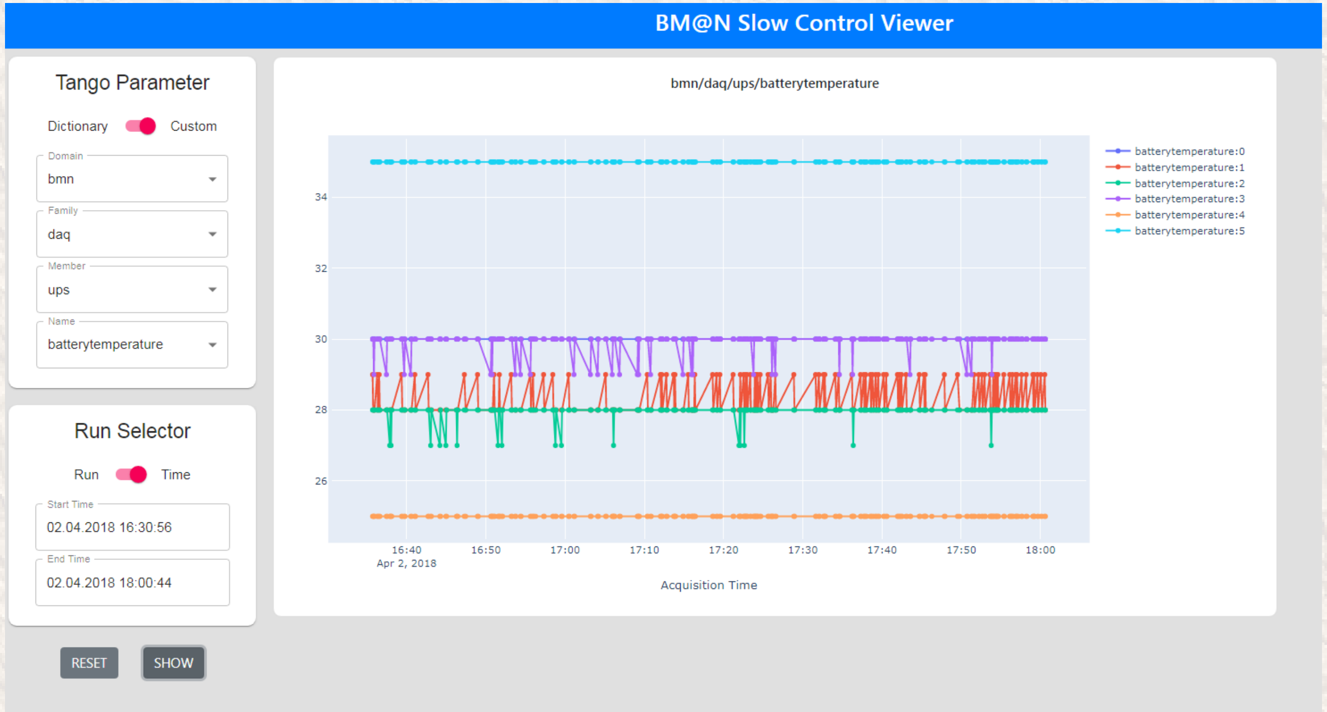
```
{
  "param_type": "FLOAT64",
  "scalar_params": [
    {
      "raw_value_r": "23.31",
      "raw_value_w": "0",
      "data_time": "2021-11-26 14:03:25.717"
    },
    {
      "raw_value_r": "23.42",
      "raw_value_w": "0",
      "data_time": "2021-11-26 14:28:30.717"
    },
    {
      "raw_value_r": "23.39",
      "raw_value_w": "0",
      "data_time": "2021-11-26 14:33:05.717"
    },
    {
      "raw_value_r": "23.32",
      "raw_value_w": "0",
      "data_time": "2021-11-26 16:00:45.718"
    },
    {
      "raw_value_r": "23.39",
      "raw_value_w": "0",
```

Response headers

```
content-length: 460
content-type: application/json
```

Work in progress

- In addition to API, a Web-based viewer for SCS is being developed
- Old SCS system viewer developed previously (BM@N Runs 1-7) is shown:



BM@N Slow Control Viewer

Tango Parameter

Dictionary / Custom

Domain: bmn

Family: daq

Member: ups

Name: batterytemperature

Run Selector

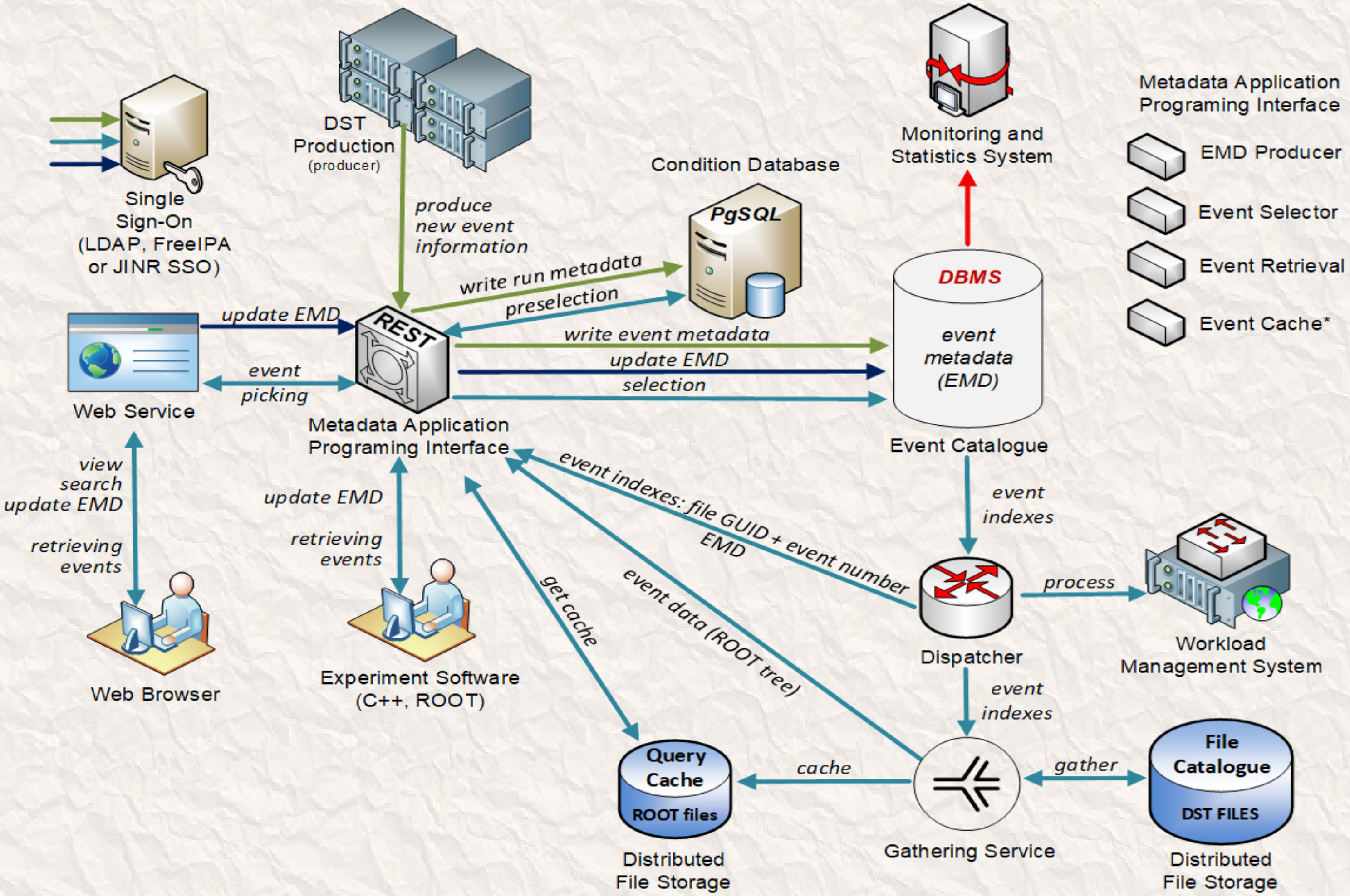
Run / Time

Start Time: 02.04.2018 16:30:56

End Time: 02.04.2018 18:00:44

Event Metadata System
(an update)

BM@N Event Metadata System



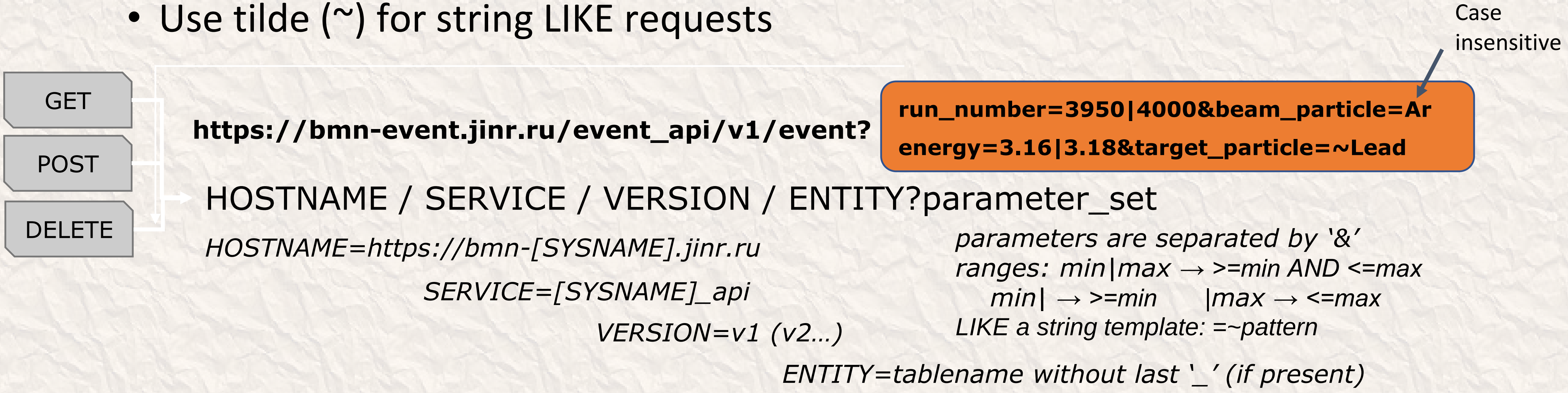
- Event Metadata System
  - Event Catalogue is based on PostgreSQL
  - Integrates with BM@N Condition database
  - REST API and Web UI developed based on Kotlin multiplatform
  - Configurable to support different metadata
  - ROOT macro to fill in the catalogue
  - Automatic deployment
  - High Availability solution available
  - Statistics collection and display
  - 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).

Updated REST API scheme for EMS

- The new scheme is unified for different BM@N Information Systems
  - Use pipe (|) for ranges
  - Use tilde (~) for string LIKE requests



GET

POST

DELETE

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

run_number=3950|4000&beam_particle=Ar
energy=3.16|3.18&target_particle=~Lead

Case insensitive

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
LIKE a string template: =~pattern

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

KeyCloak Integration

- Authentication and authorization in EMS
  - KeyCloak token-based authentication and authorization is now supported
    - Bug that came out after KeyCloak migration/upgrade was fixed
  - Database-based authentication is supported as before
  - FreeIPA / LDAP support has been dropped

```
keycloak_auth:
  server_url: "https://bmn-user.jinr.ru"
  realm: "BMN"
  client_id: "emd_api"
  client_secret: "*****"
  writer_group_name: "bmneventwriter"
  admin_group_name: "bmneventadmin"

# database_auth: True
```

Development of Next-Generation Event Visualization Platform for BM@N (an update)

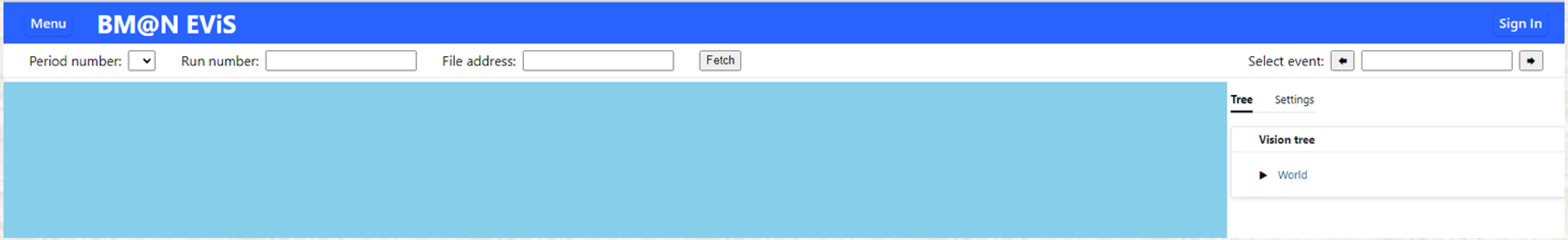
VisionForge Project Overview

- VisionForge – platform for creating next-gen visualization systems
  - Distributed dynamic system
    - Visualization model can be created on one node, transferred to another node and rendered there
    - Nodes can exchange **updates** to the model
    - Changing one element or attribute only requires sending this small change
  - Performance and optimizations
    - BM@N geometry model includes more than 400 000 elements
    - Geometry can be defined as **prototype** that is used by a set of objects, in this case rendering is simplified – only required properties can be changed if needed
  - Using Kotlin-Multiplatform

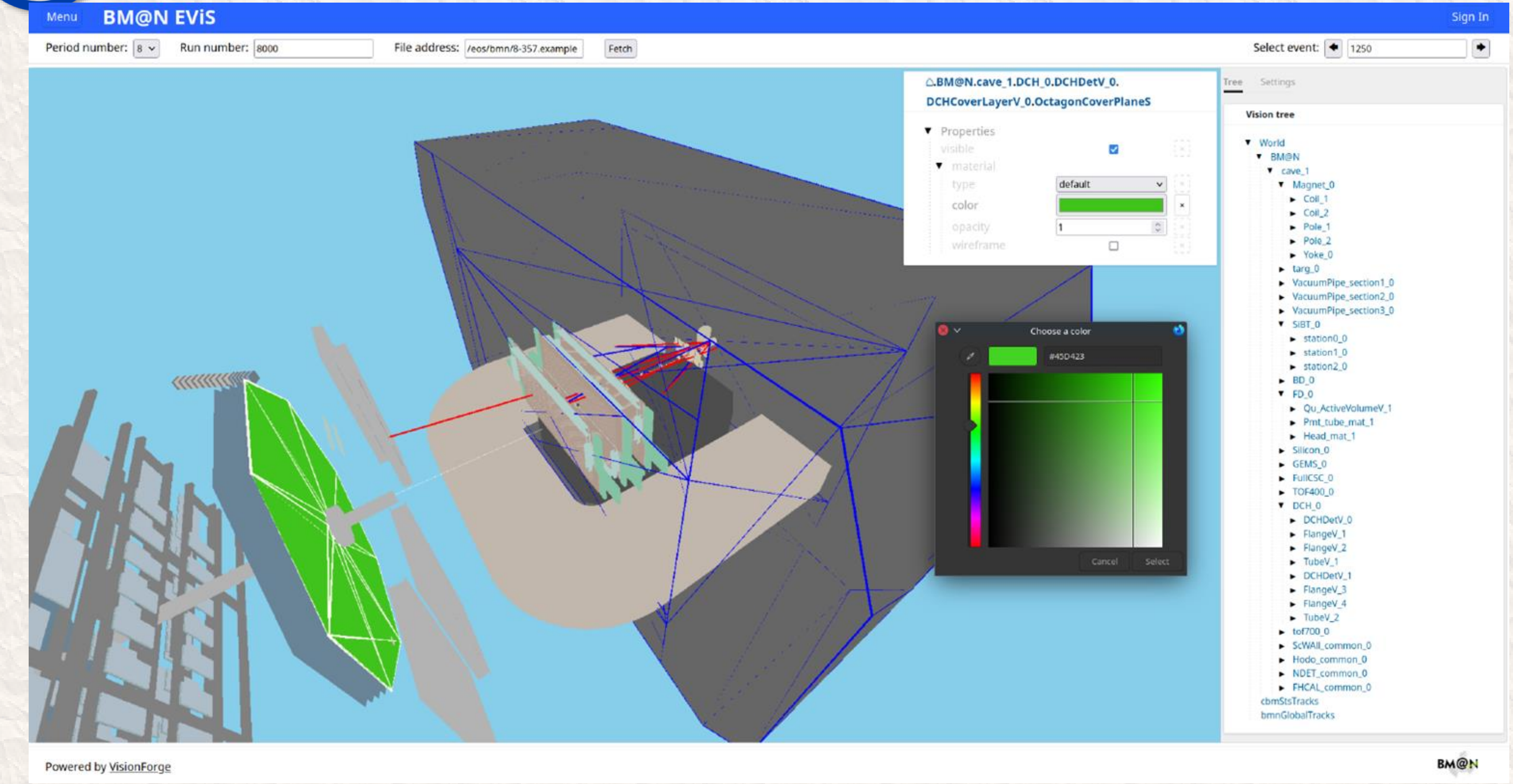
See also: Alexander Nozik — Unbearable lightness of data visualization in Kotlin full stack
https://www.youtube.com/watch?v=uT5j-xOXC3E&ab_channel=JPoint%2CJoker%D0%B8JUGru

Available for test now!

- Available online at http://10.220.16.81:8080/
- Example entry:
  - Period number: **8**
  - Run number: **8000**
  - File address: **/home/lab/events/mpd_run_Top_8000_ev1_p8.root**
  - Select event: **1, 2, 3,…**
- Possible to run it on your own as well (not so simple right now)
- Please send us feedback (contacts on the title slide)!



Geometry, tracks, scene graph, tuning



Period number: 8

Run number: 8000

File address: /eos/bmn/8-357.example

Select event: 1250

△.BM@N.cave_1.DCH_0.DCHDetV_0.
DCHCoverLayerV_0.OctagonCoverPlaneS

- Properties
  - visible ☑
  - material
    - type: default
    - color: 🟩
    - opacity: 1
    - wireframe ☐

Choose a color

#45D423

Vision tree

- World
  - BM@N
    - cave_1
      - Magnet_0
        - Coil_1
        - Coil_2
        - Pole_1
        - Pole_2
        - Yoke_0
      - targ_0
      - VacuumPipe_section1_0
      - VacuumPipe_section2_0
      - VacuumPipe_section3_0
      - SiBT_0
        - station0_0
        - station1_0
        - station2_0
      - BD_0
      - FD_0
        - Qu_ActiveVolumeV_1
        - Pmt_tube_mat_1
        - Head_mat_1
      - Silicon_0
      - GEMS_0
      - FullCSC_0
      - TOF400_0
      - DCH_0
        - DCHDetV_0
        - FlangeV_1
        - FlangeV_2
        - TubeV_1
        - DCHDetV_1
        - FlangeV_3
        - FlangeV_4
        - TubeV_2
      - tof700_0
      - ScWAll_common_0
      - Hodo_common_0
      - NDET_common_0
      - FHCAL_common_0
  - cbmStsTracks
  - bmnGlobalTracks

Powered by VisionForge

WIP Items

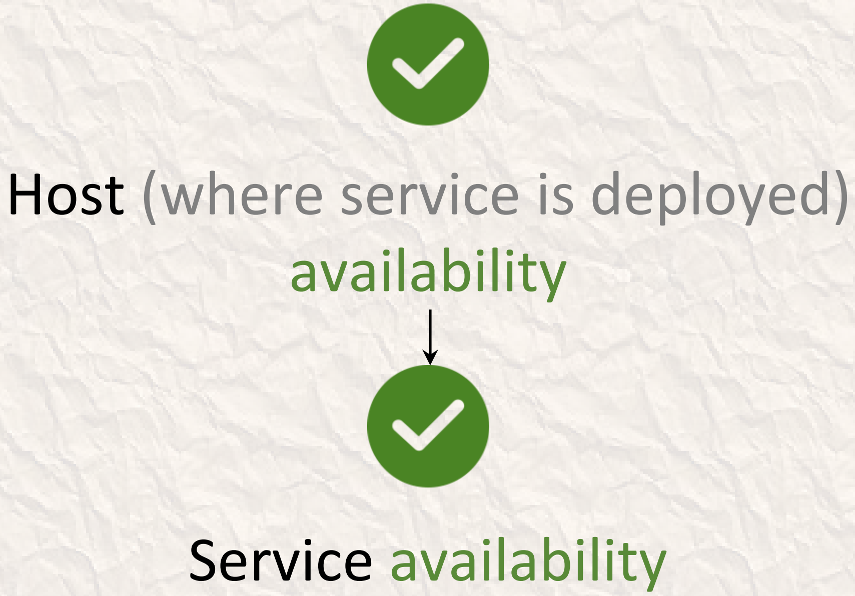
- Visualization of the detectors geometry with a choice of the detail level
- Working with the scene: the ability to scale, shift, rotate, display coordinate axes, coordinate grid (optional), section by plane or parallelepiped, choice of background color. Saving an image to a file, it is possible to create a GIF animation (optional). Optionally, the ability to display projections on separate tabs or windows in a common window.
- Show/hide geometric elements, set color, transparency. For a solid detector, we loaded from a prepared scheme (XML or JSON) to replace the default.
- Ability to create buttons to which functionality can be attached (examples: light/dark background changes; show/hide magnet)
- Visualization of particle collision events: display of tracks and hits, activated calorimeter towers. The source is either a file (initially ROOT), or a data stream from the socket for online monitoring.
- Selection of event objects with viewing of their properties, editing of color, visibility, marker, size/thickness. Selection/scrolling of transferred events in case of the source from a file. Event objects are presented as a hierarchical tree, with tracks grouped by particle type. When an object is selected in the tree, the object is highlighted, and vice versa, when an object is selected in the view, its properties are opened.
- Filter of displayed event objects: particles by their code, energy range, only primary tracks. Show/hide separately simulated tracks/particles (before reconstruction), reconstructed tracks/particles
- Output general information: selected setup geometry, event number, number of events (if from file), number of displayed geometry objects.

Development of a service for monitoring software systems of the BM@N experiment

Monitored parameters

For checking stability and reliability of BM@N systems (Unified Condition database, Configuration database, Integrity Inspector, Electronic LogBook,…):

- **Endpoints state**:
  - network interfaces,
  - memory,
  - disk,
  - CPU.
- **Database** (e.g. PostgreSQL):
  - latency
- **Web interfaces**:
  - HTTP requests checks (e.g. GET-request).



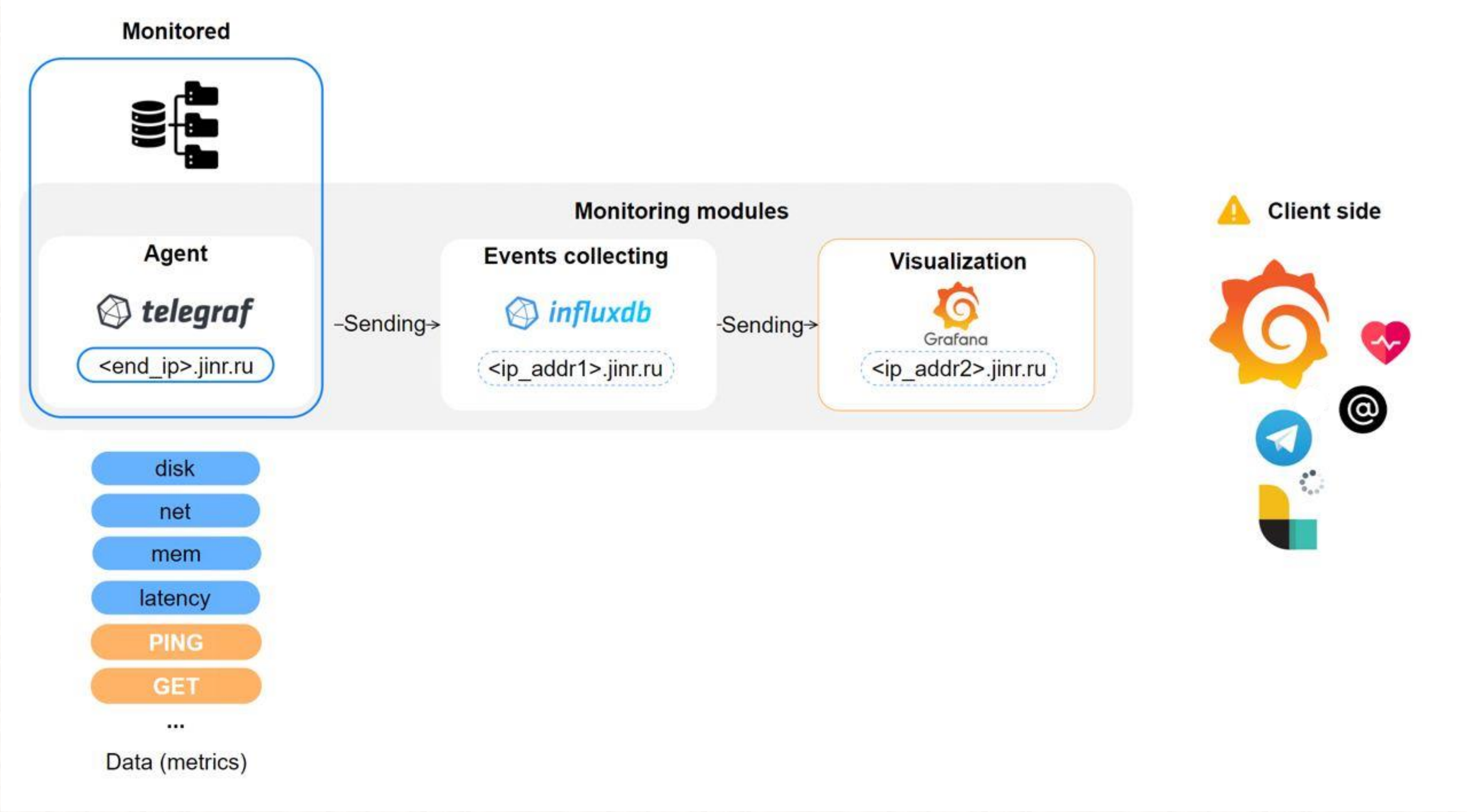
Host (where service is deployed) availability

Service availability

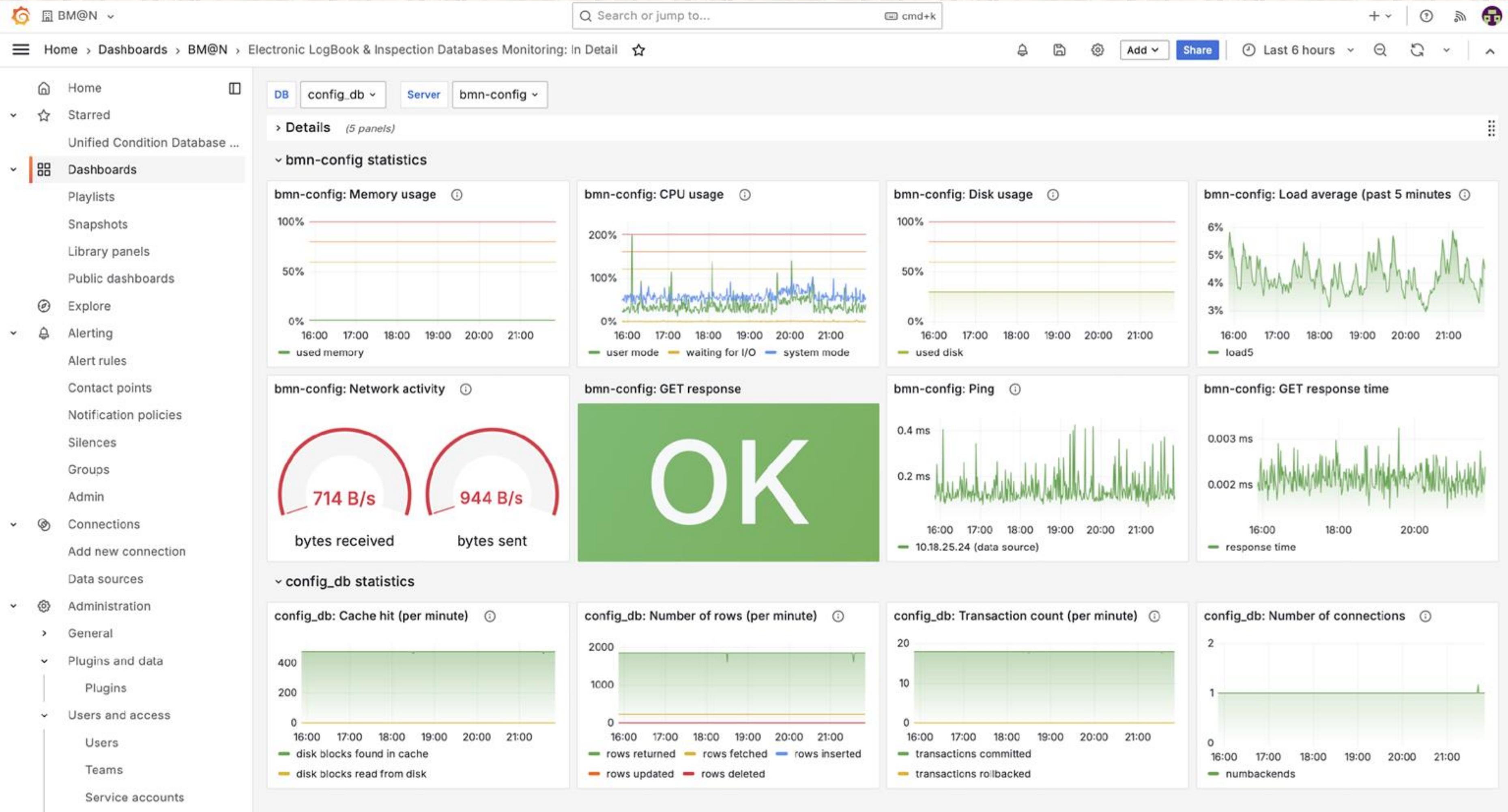
Using **TIG** (Telegraf + InfluxDB + Grafana) stack.

Architecture for monitoring of the software systems

- **Automated** deployment of components with **Ansible playbooks**
- **Automated configuration** generation (Jinja2 + JSONs: Alerts and Dashboard)
- Ease of scaling because of **module architecture**
- Failure **alerting** with Grafana



BM@N monitoring client’s view (Dashboard)

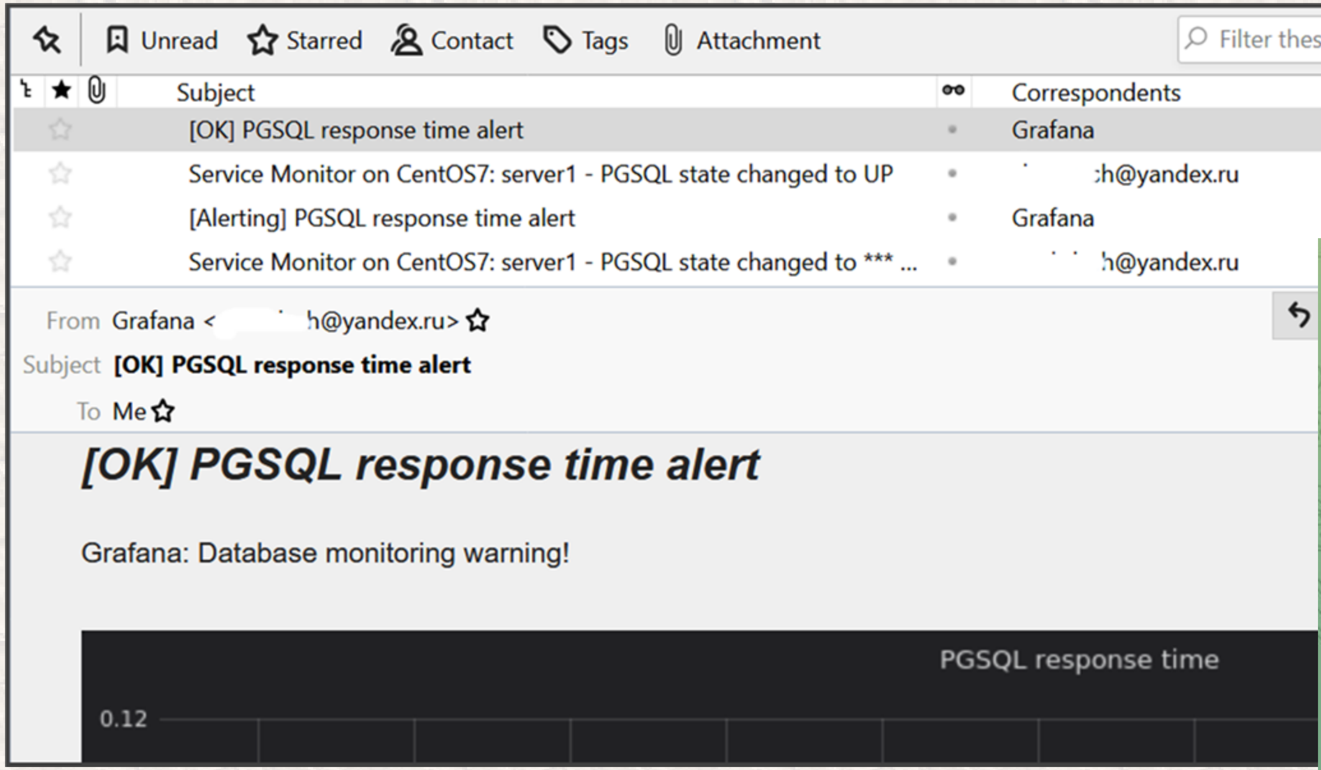


DB config_db

Server bmn-config

Last 6 hours

BM@N monitoring alerting



Subject	Correspondents	Date
[OK] PGSQL response time alert	Grafana	2:41 PM
Service Monitor on CentOS7: server1 - PGSQL state changed to UP	[redacted]:h@yandex.ru	2:40 PM
[Alerting] PGSQL response time alert	Grafana	2:01 PM
Service Monitor on CentOS7: server1 - PGSQL state changed to *** ...	[redacted]h@yandex.ru	

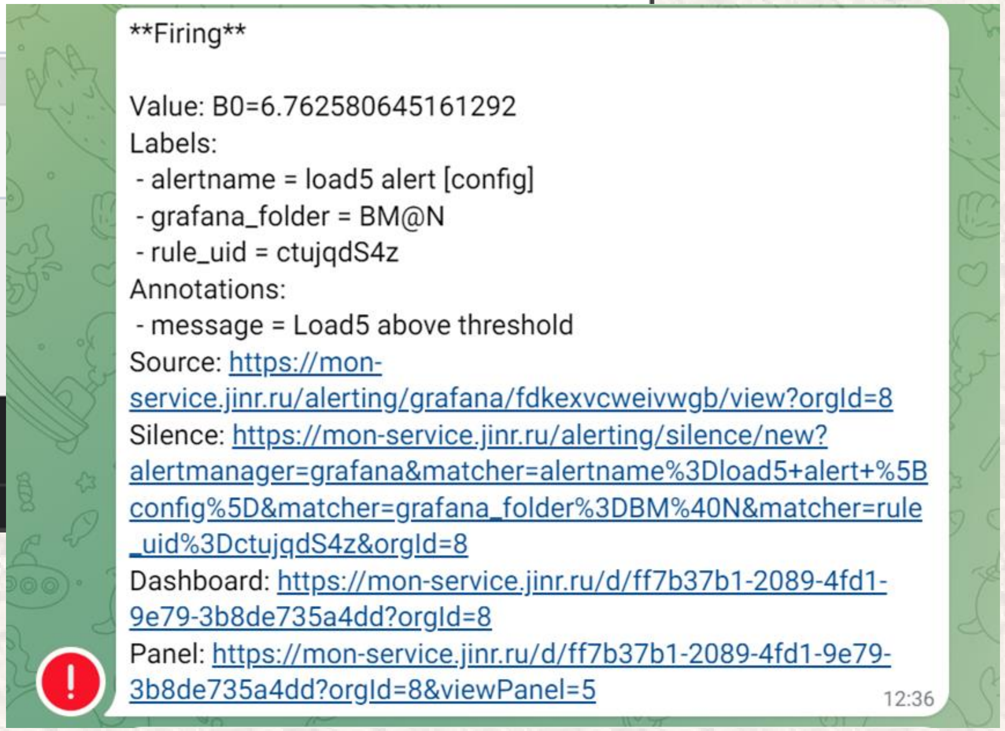
From Grafana <[redacted]h@yandex.ru>

Subject **[OK] PGSQL response time alert**

To Me

[OK] PGSQL response time alert

Grafana: Database monitoring warning!



Firing

Value: B0=6.762580645161292
Labels:
- alertname = load5 alert [config]
- grafana_folder = BM@N
- rule_uid = ctujqdS4z

Annotations:
- message = Load5 above threshold

Source: https://mon-service.jinr.ru/alerting/grafana/fdkexvcweivwgb/view?orgId=8
Silence: https://mon-service.jinr.ru/alerting/silence/new?alertmanager=grafana&matcher=alertname%3Dload5+alert+%5Bconfig%5D&matcher=grafana_folder%3DBM%40N&matcher=rule_uid%3DctujqdS4z&orgId=8
Dashboard: https://mon-service.jinr.ru/d/ff7b37b1-2089-4fd1-9e79-3b8de735a4dd?orgId=8
Panel: https://mon-service.jinr.ru/d/ff7b37b1-2089-4fd1-9e79-3b8de735a4dd?orgId=8&viewPanel=5
12:36

Thank You!