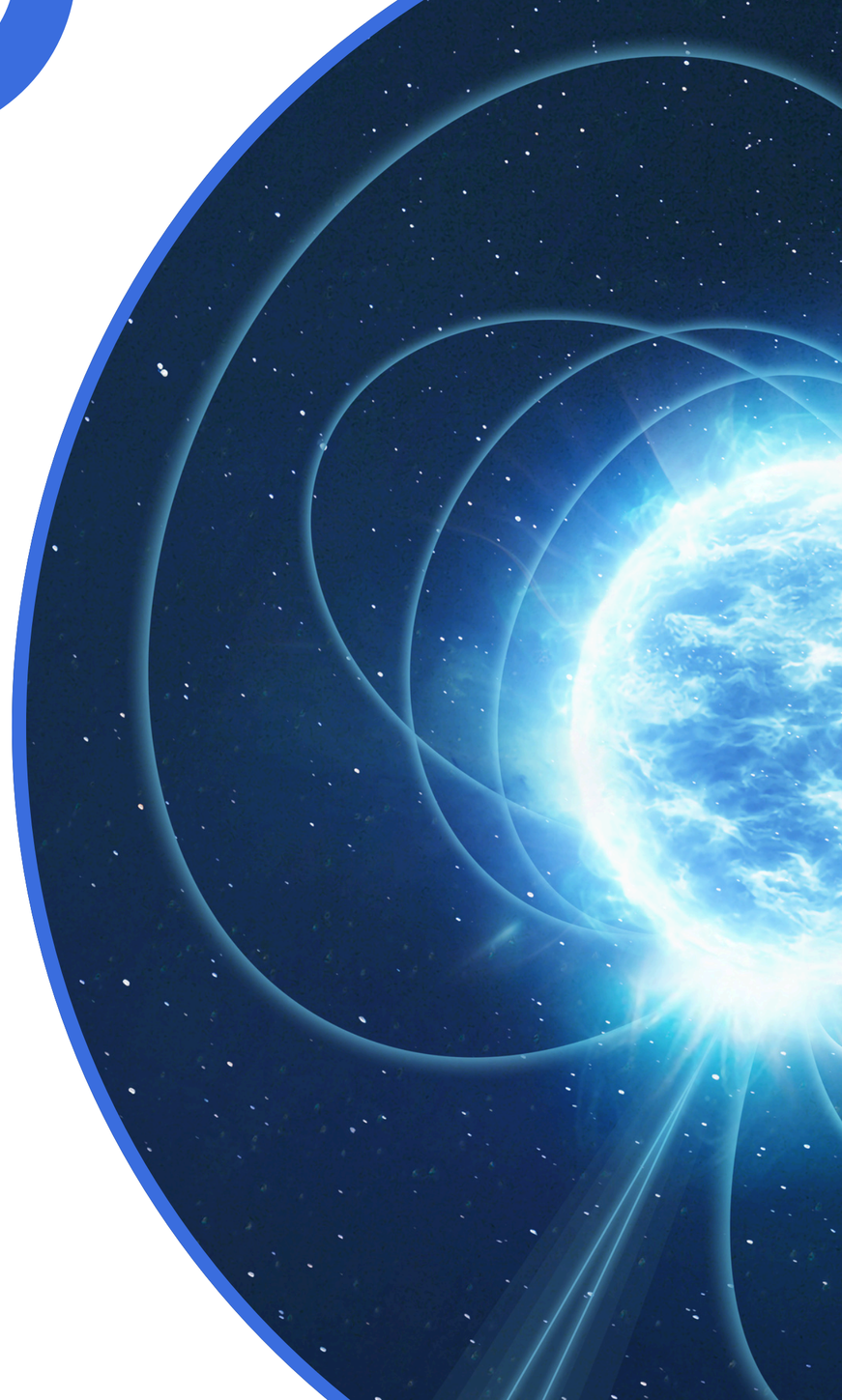
Kubernetes-based online TPC data processing and monitoring for NICA/MPD

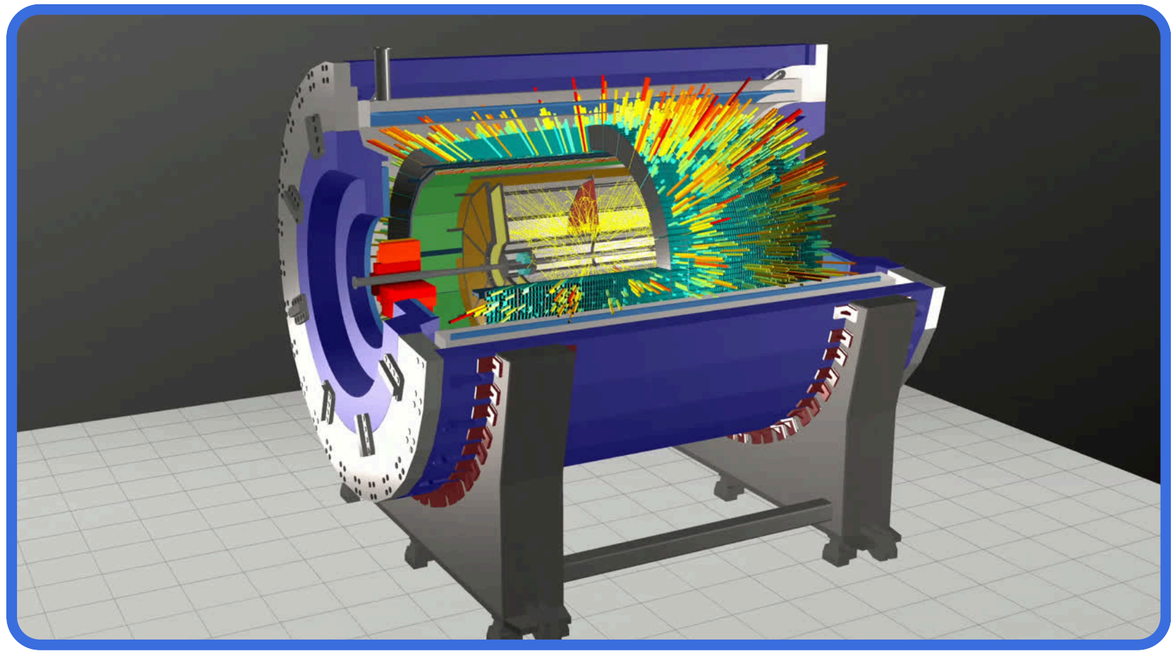
Andrey Budnikov, Voronezh State University
Alexander Krylov, LHEP JINR
Victor Krylov, LHEP JINR
Oleg Rogachevsky, LHEP JINR

22-26 June 2026, MMCP'2026, Dubna, Russia



Introduction

Multi-Purpose Detector (MPD)

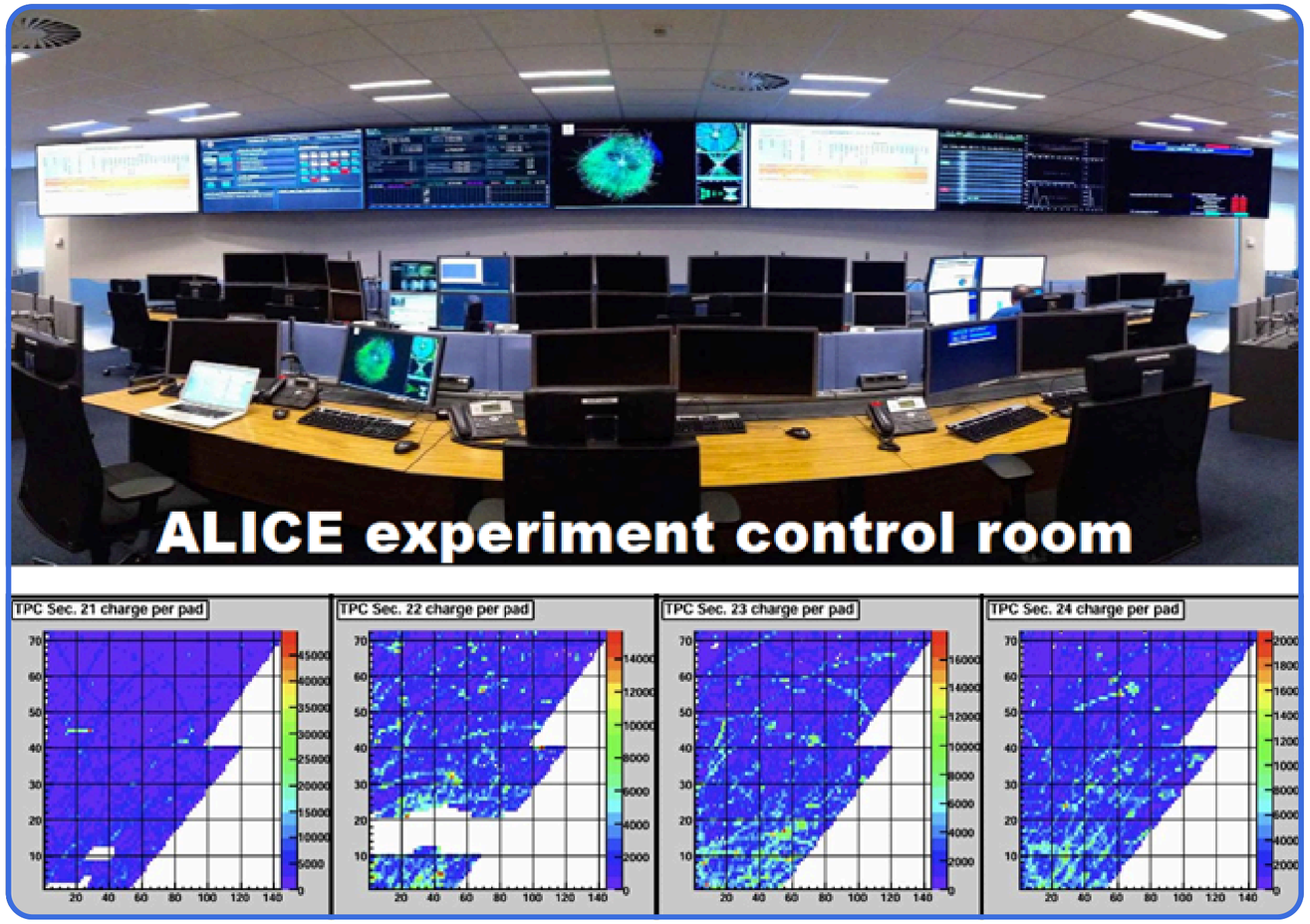


MPD Experiment is one of megascience projects on NICA collider.

MPD Detectors:

- **TPC - Time Projector Chamber**
- TOF - Time of Flight
- ECAL - Electromagnetic Calorimeter
- FFD – Fast Forward Detector
- FHCAL - Forward Hadron Calorimeter

Introduction



Online data monitoring task

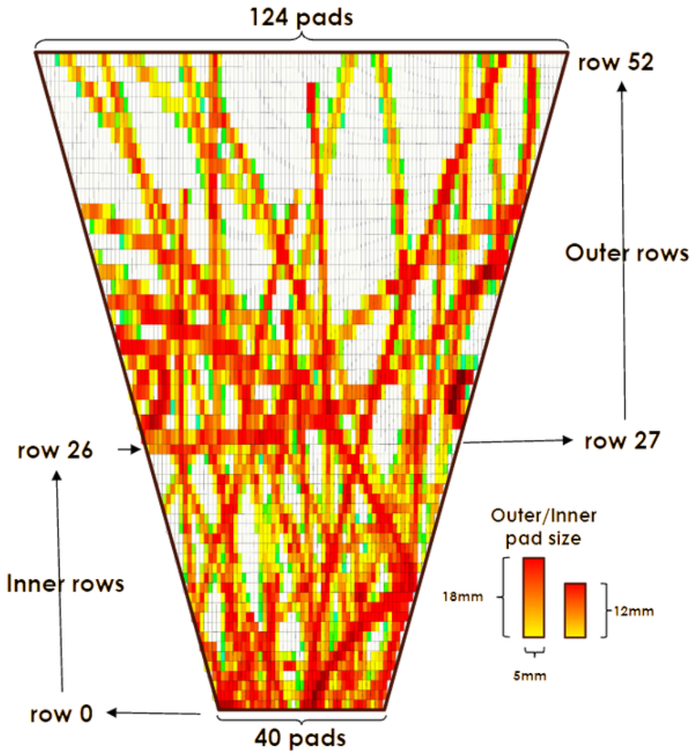
For tasks of **Event Display** and **Data Quality Assurance (DQA)** it is necessary to implement a **distributed online data processing system** that satisfies the following demands:

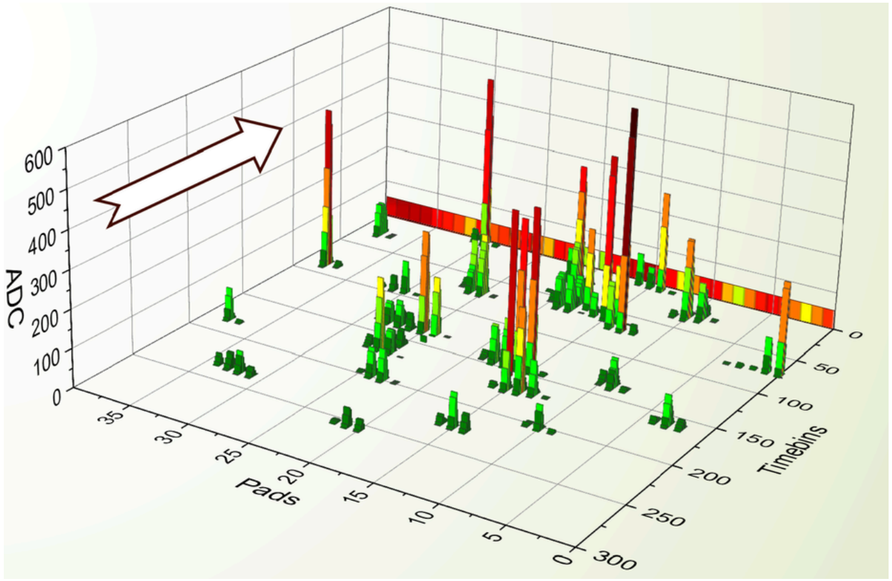
- **scalability**
- **no-downtime insurance**
- **reactivity**
- **observability**

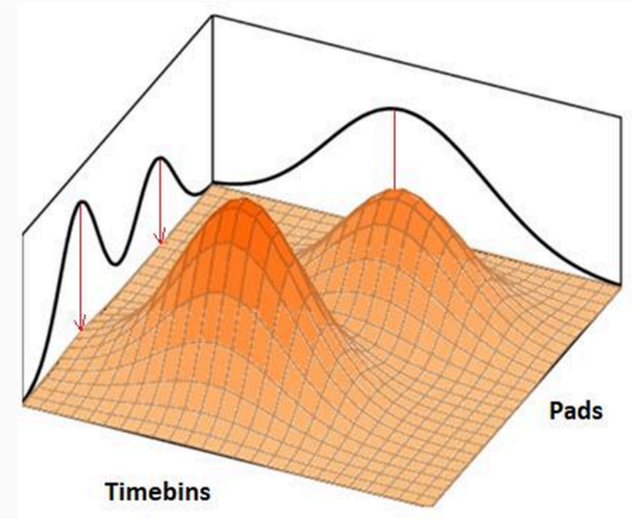
This requirements can be easily met using modern technologies of **containerization** and **orchestration.**

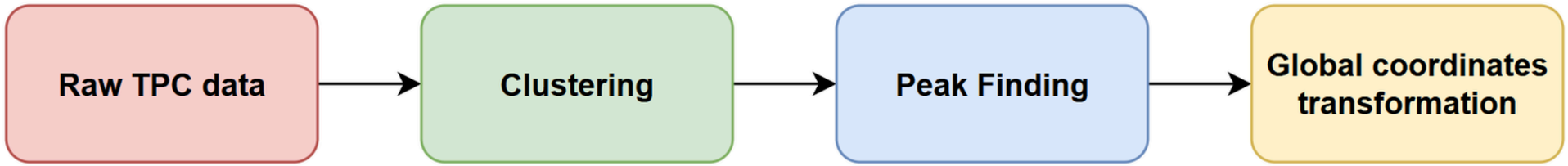
Online data processing

Fast clustering workflow









Online data processing

Peak finding algorithms

For TPC data processing **fast Gaussian fitting algorithms** as a basis are used.

$$\ln(y) = \ln(A) - \frac{\mu^2}{2\sigma^2} + \frac{2\mu x}{2\sigma^2} - \frac{x^2}{2\sigma^2} = a + bx + cx^2, \quad [1]$$

By solving the system of equations [2] (Guo's algorithm) it is possible to **restore form of Gauss function.**

$$\begin{bmatrix} \sum y_n^2 & \sum x_n y_n^2 & \sum x_n^2 y_n^2 \\ \sum x_n y_n^2 & \sum x_n^2 y_n^2 & \sum x_n^3 y_n^2 \\ \sum x_n^2 y_n^2 & \sum x_n^3 y_n^2 & \sum x_n^4 y_n^2 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \sum y_n^2 \ln(y_n) \\ \sum x_n y_n^2 \ln(y_n) \\ \sum x_n^2 y_n^2 \ln(y_n) \end{bmatrix}, \quad [2]$$

There three main **limitations**:

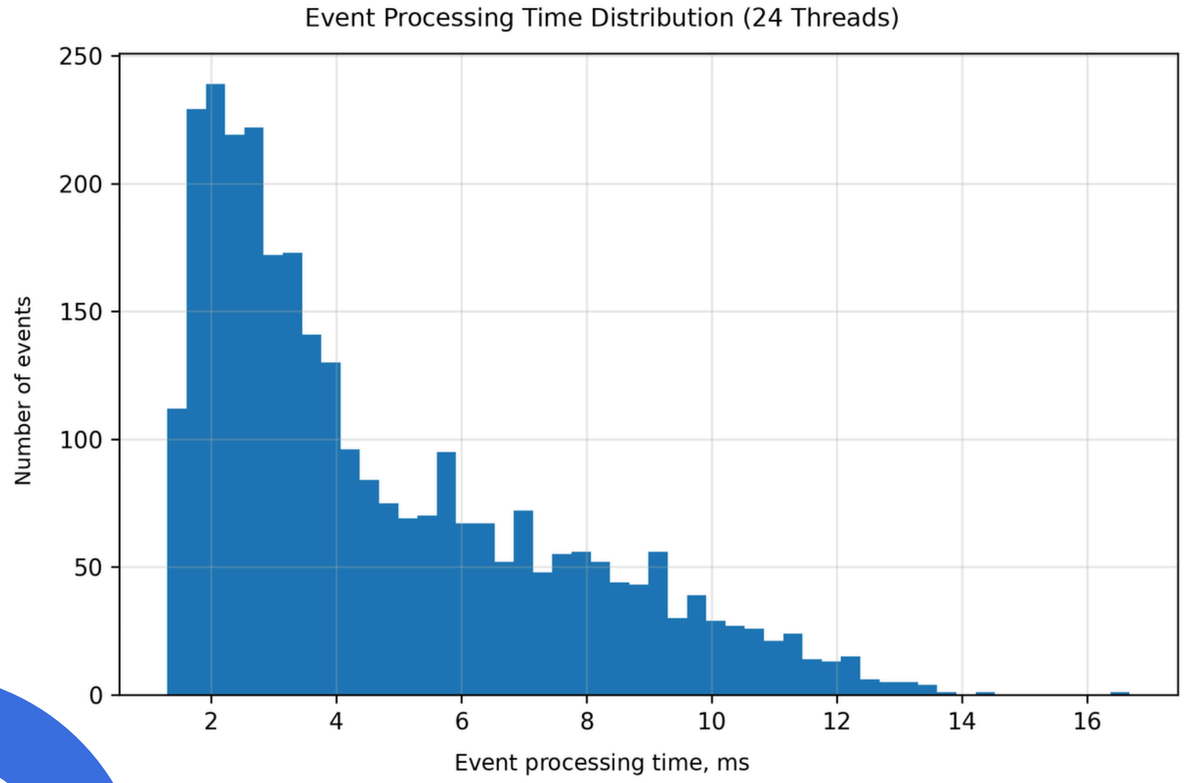
- **Too few fitting points** (4-7 points per cluster);
- **Noise** significance;
- **Points distribution** on Gaussian.

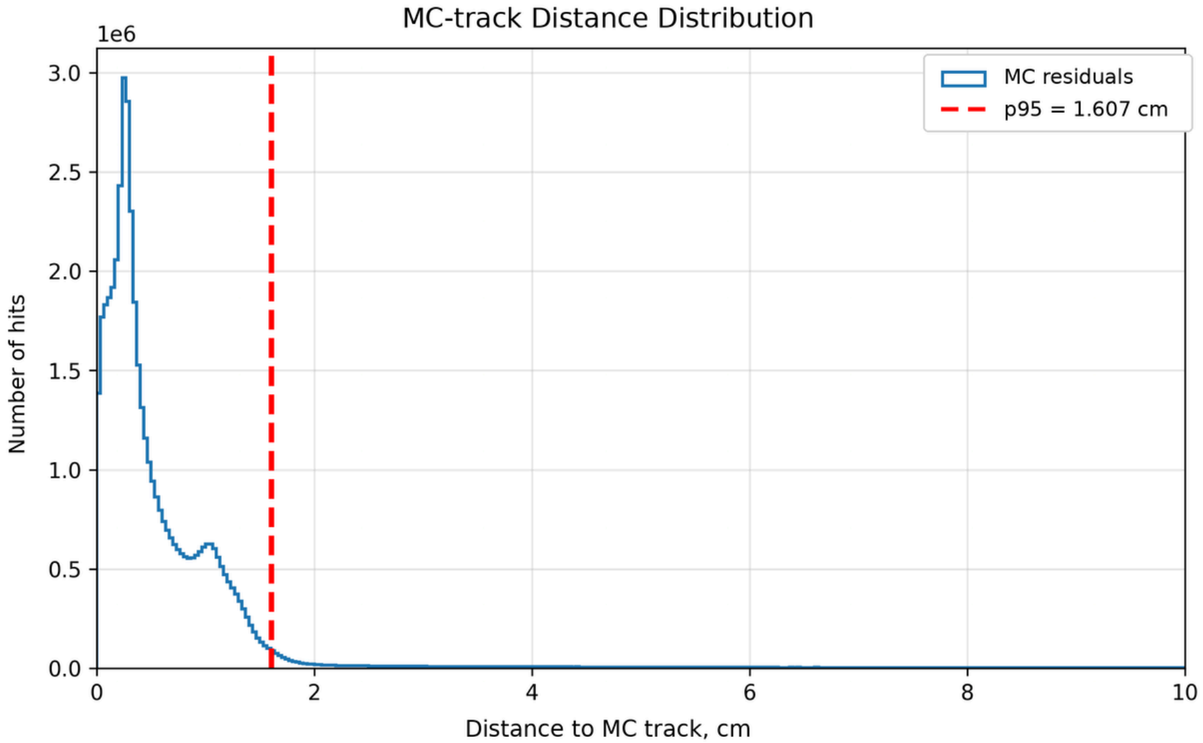
$$A = e^{\,a - \frac{b^2}{4c}}, \mu = -\frac{b}{2c}, \sigma = \sqrt{-\frac{1}{2c}}, \quad [3]$$

Al-Nahhal I. [et al.] / A Fast, Accurate, and Separable Method for Fitting a Gaussian Function
URL: https://arxiv.org/abs/1907.07241v1

Online data processing

Peak finding algorithms

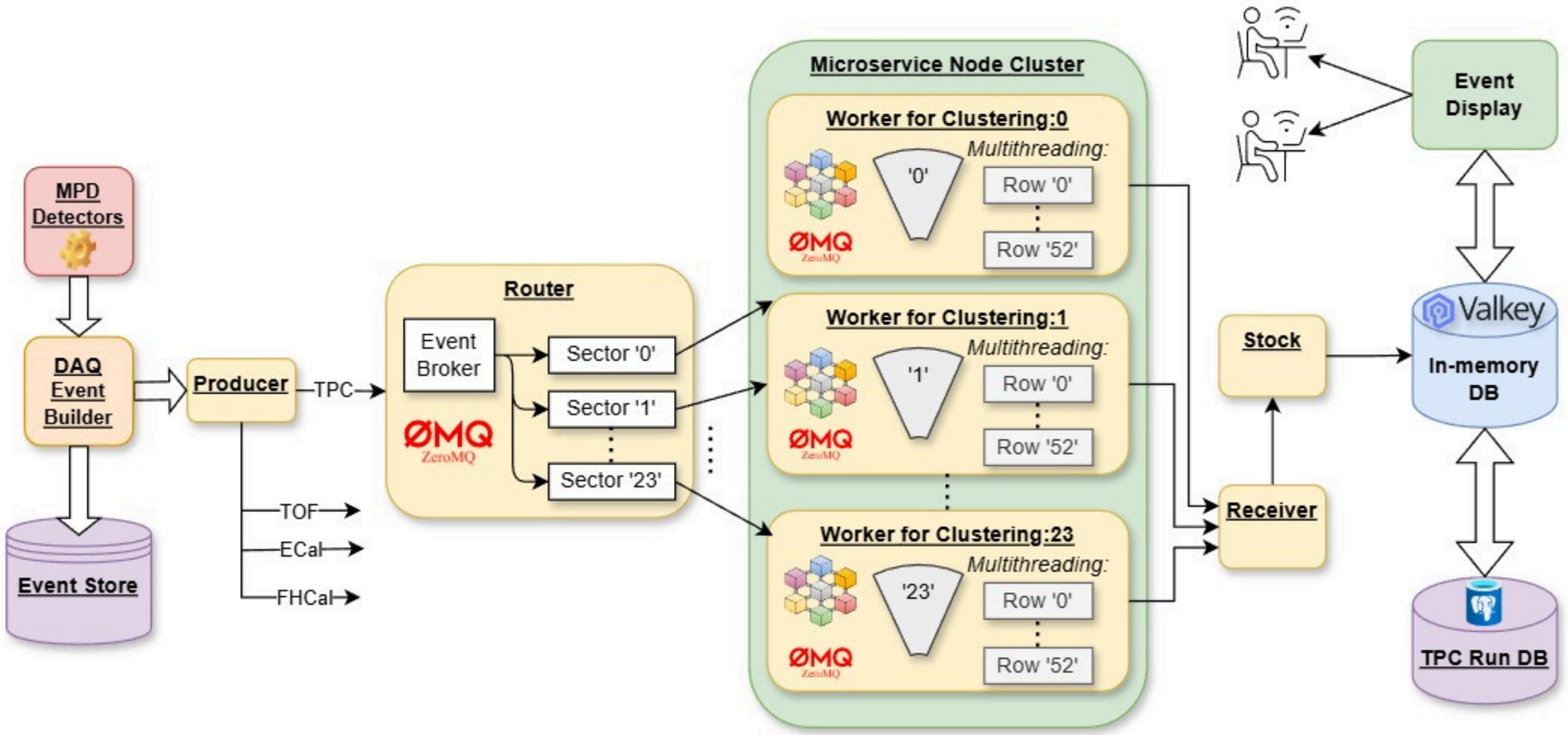




Online data processing

Conception:

1. Producer gets **raw events** from detectors;
2. Events are **processed** within data pipeline;
3. Results are **stored** in fast database **(Valkey)**;
4. Visualization in **Event Display** service.



Microservice architecture based on containers **demands** orchestrators!

Online data processing

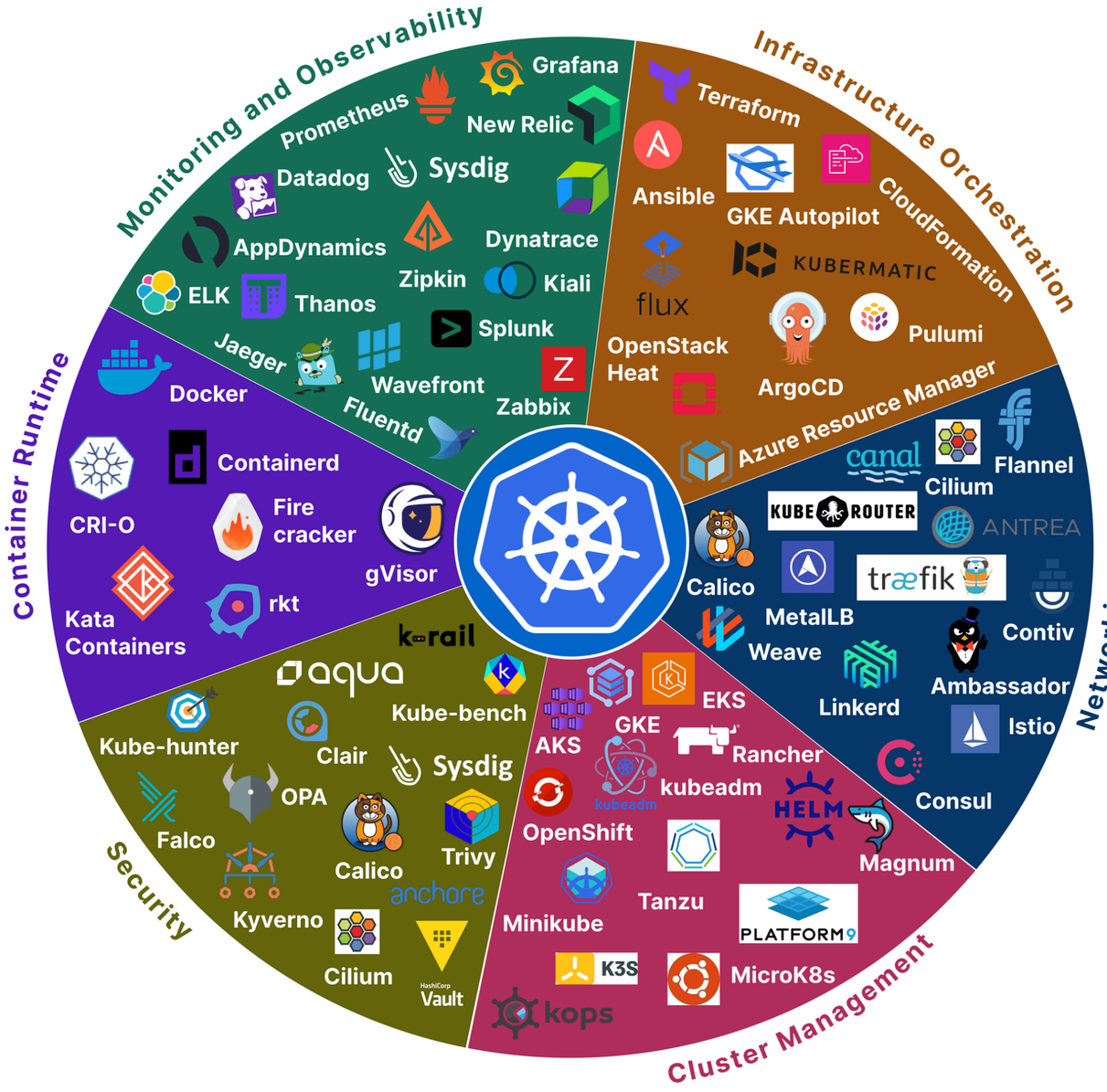
Kubernetes (k8s)

is an open-source **orchestrator** for deploying and managing containers.

The Kubernetes declarative API-driven infrastructure allows **optimizing and automating** most of the processes related to service and cluster management.

Kubernetes is the **de facto standard** among orchestrators. The technology is widely used in cloud computing, e.g. it is used as **part of the CERN computing site** (ATLAS, CMS, ALICE).

Kubernetes Tools Stack Wheel



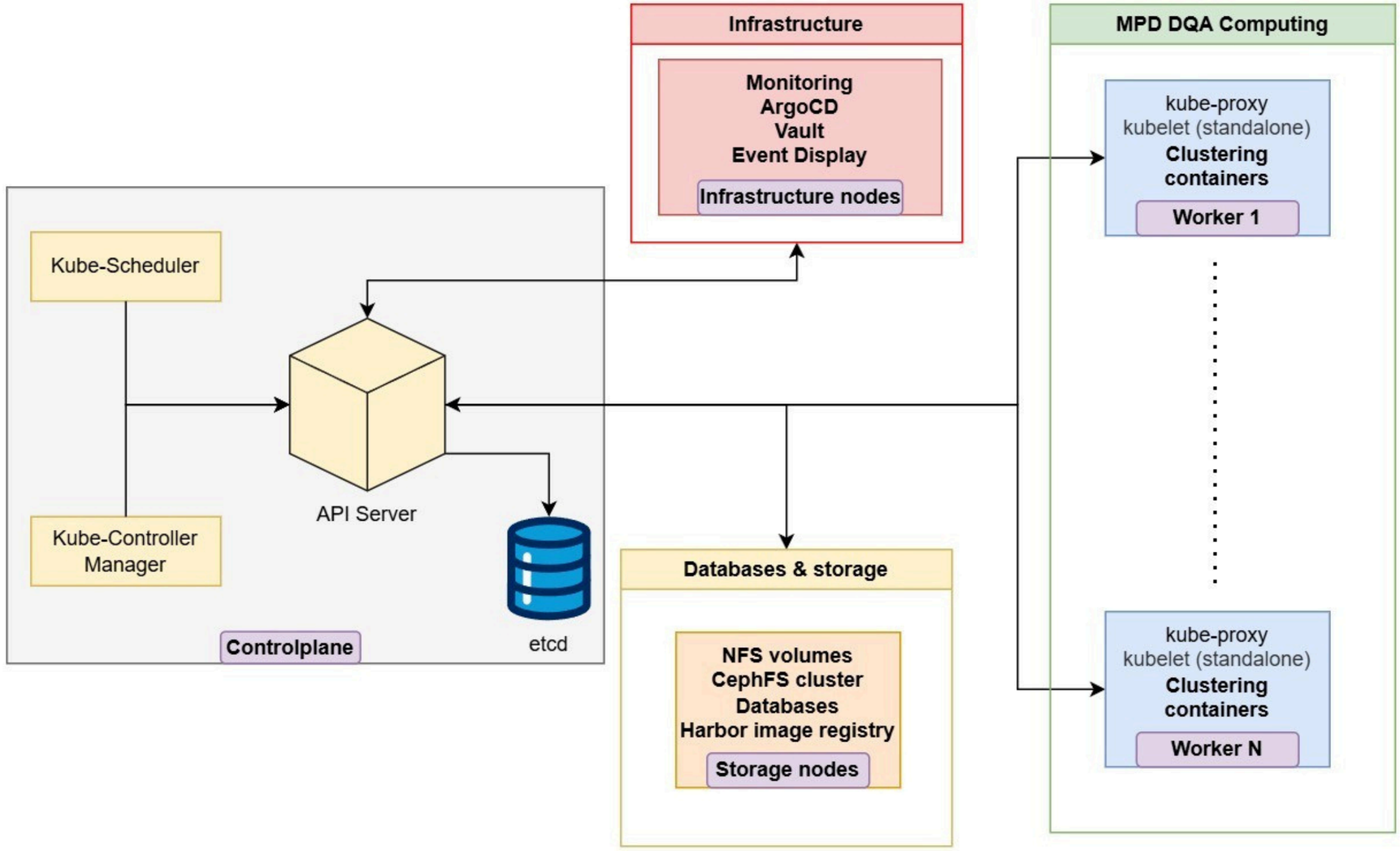
Implementation

MPD DQA Cluster

The Kubernetes cluster is implemented based on the **PROXMOX** virtualization platform.

The cluster divided into **several groups**:

- **Computing nodes**
- **Infrastructure nodes**
- **Data storage nodes**



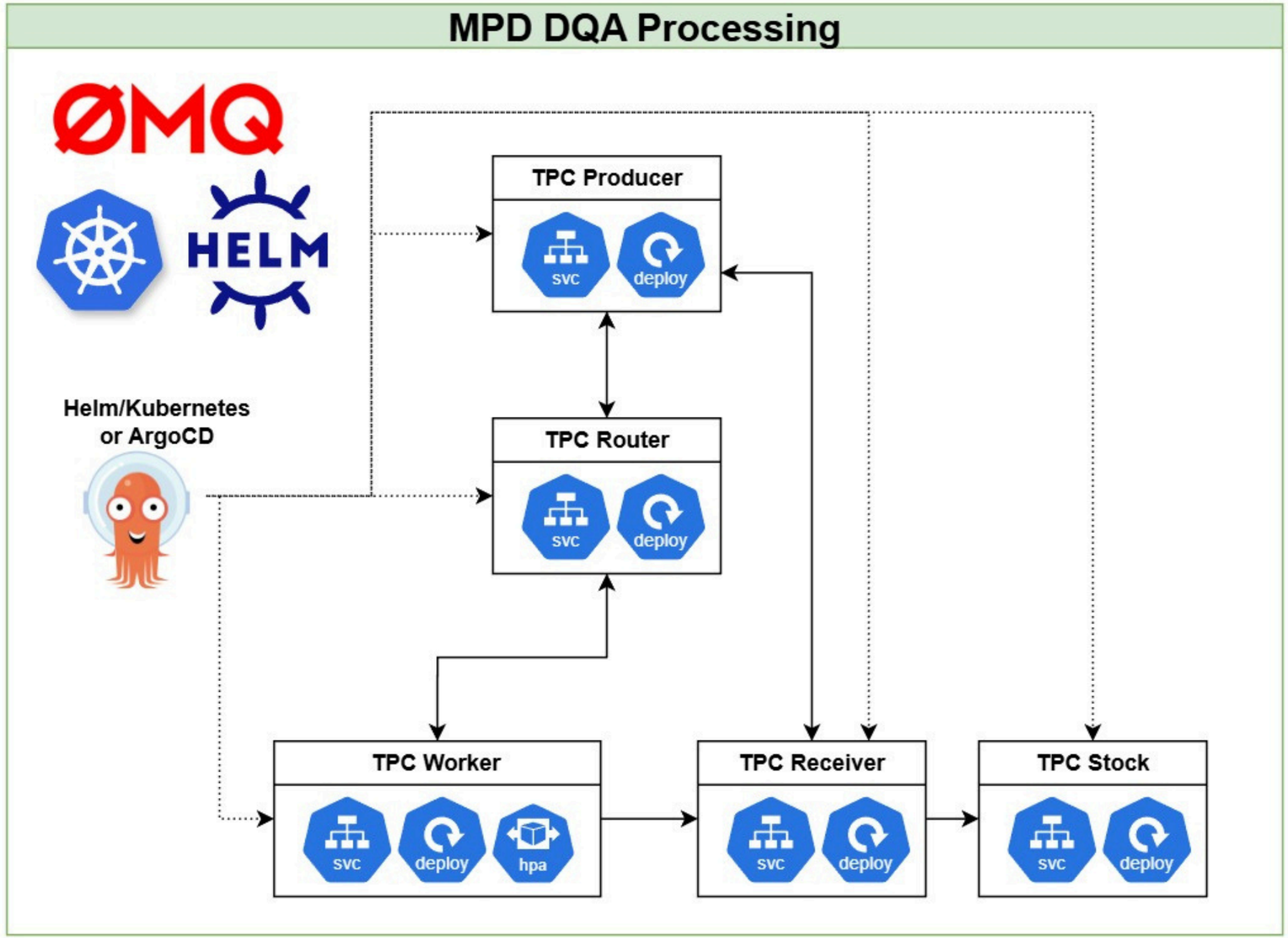


Implementation

Clustering services

Microservices (**Docker containers**) are organized into a single **Helm chart**. Messaging technology is **ZeroMQ**.

The architecture allows to change dynamically the number of replicas of each microservice via the **Horizontal Pod Autoscaler (HPA)** so system adapt to the incoming dataflow.

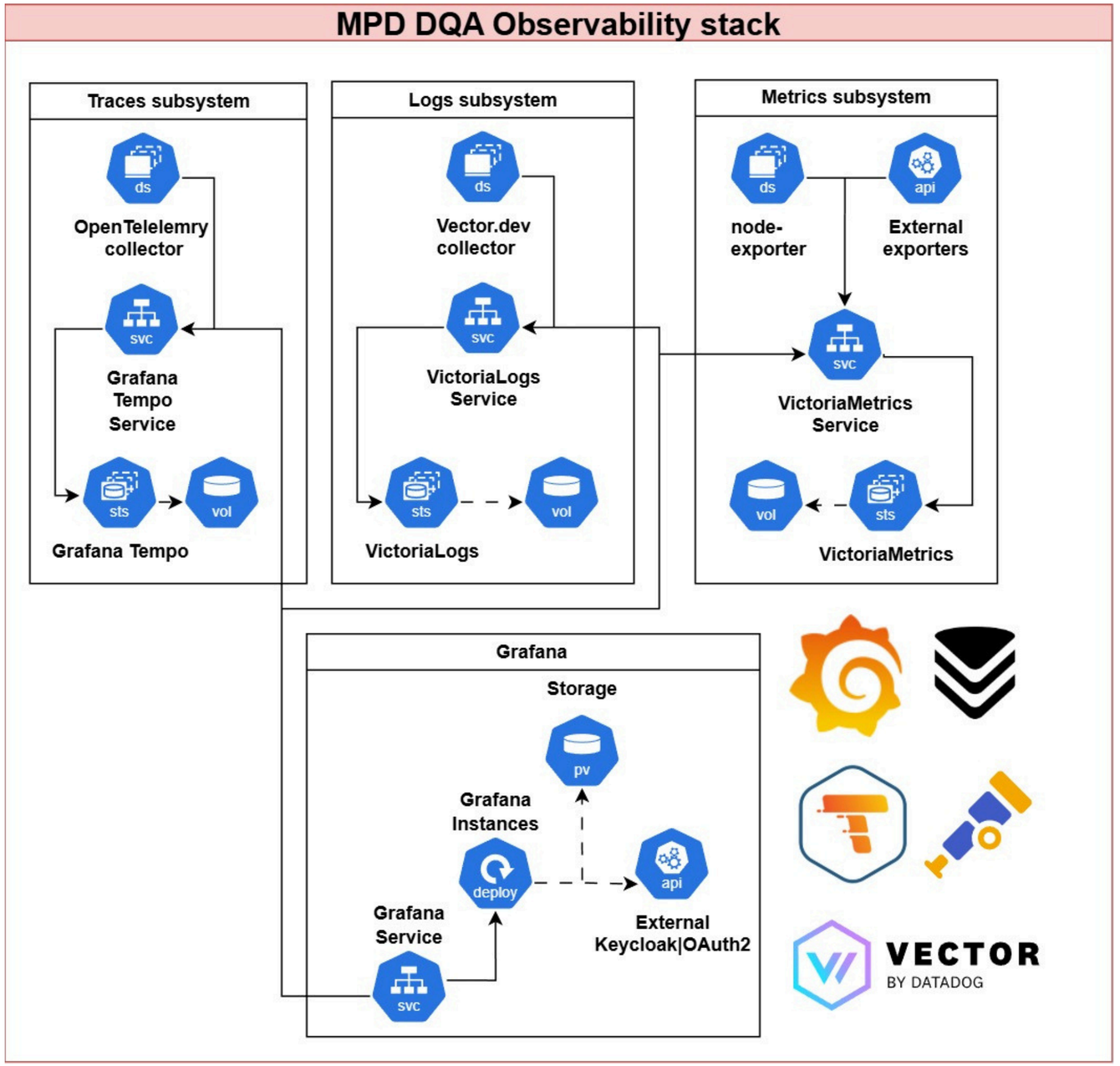


Implementation

Monitoring

The monitoring subsystem collects **telemetry (metrics, logs, traces)** from applications and services, i.e. a complete information about the system's behavior.

Metrics: VictoriaMetrics, Node Exporter
Logs: VictoriaLogs, Vector.dev
Traces: Grafana Tempo, OpenTelemetry Collector
Dashboards: Grafana

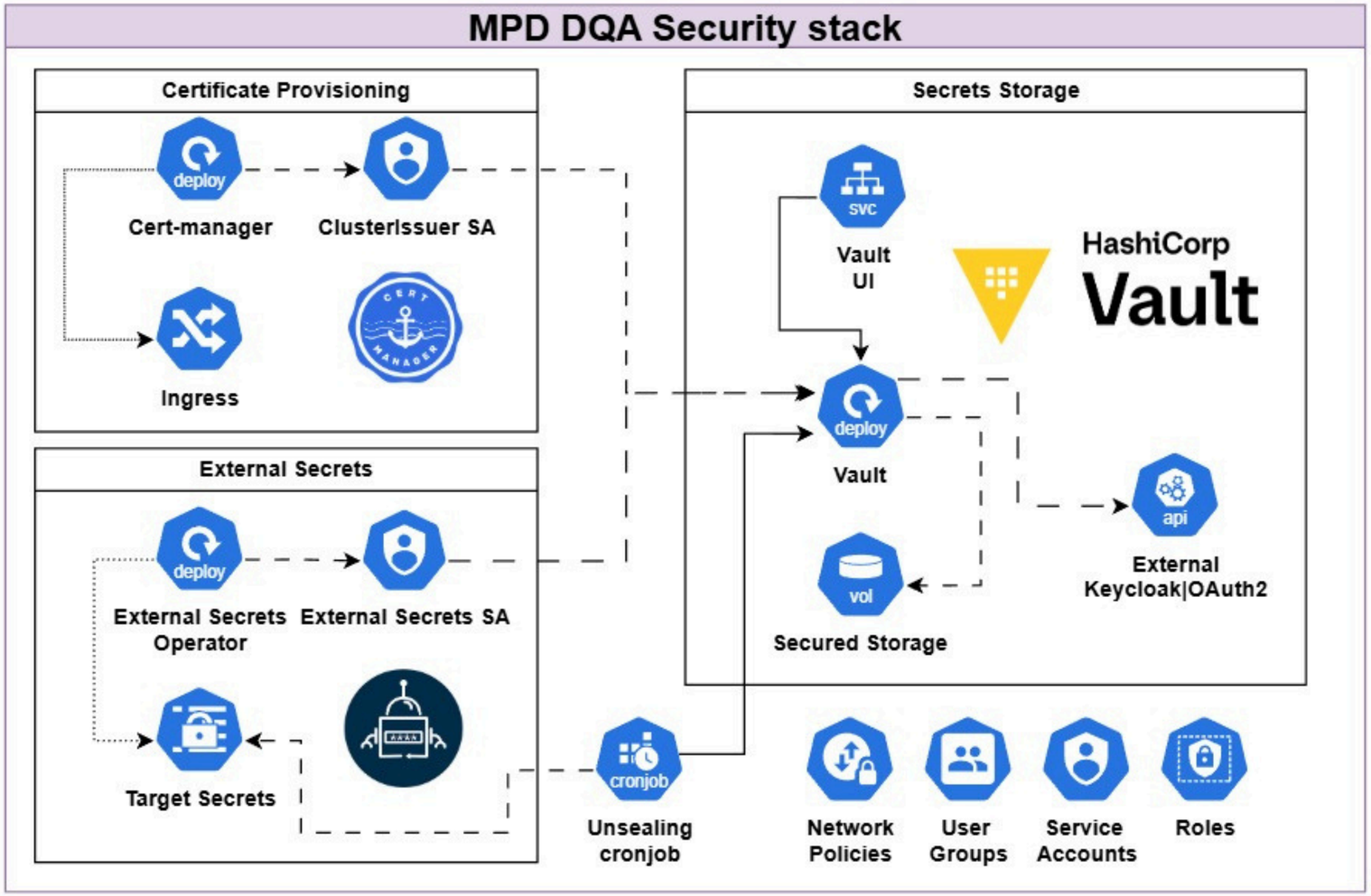


Implementation

Security stack

The cluster has a subsystem responsible for storing secrets and TLS certificates based on **Hashicorp Vault** technology.

- **External Secrets Operator** - injects and synchronizes secrets in services.
- **Cert-manager** - automatically creates, injects and updates TLS certificates in Ingress objects.



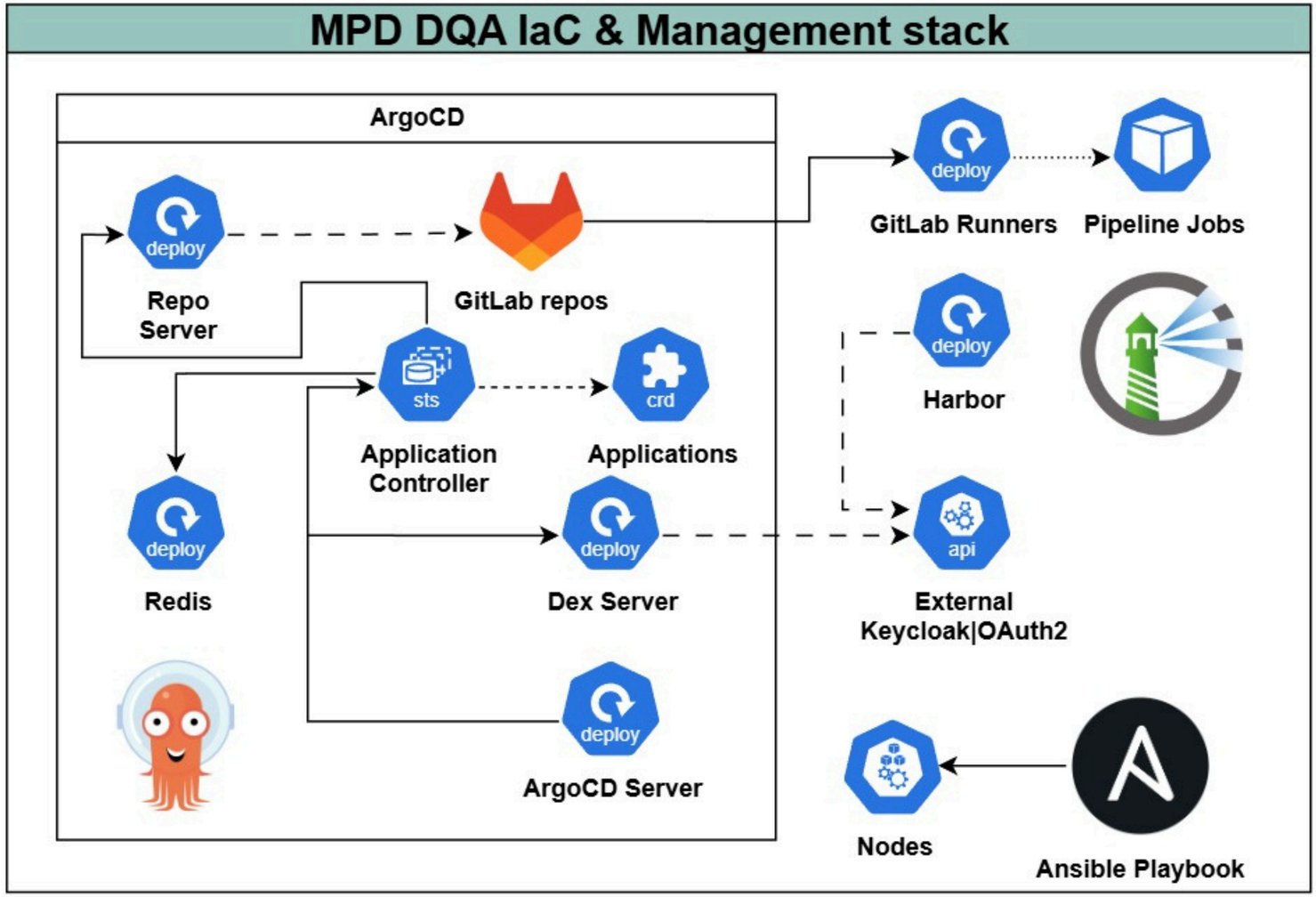
Implementation

Infrastructure as Code (IaC)

Subsystems are managed through **ArgoCD** following the **GitOps** methodology.

The images for the online processing system are built via **GitLab CI/CD pipelines**, images/charts (artifacts) are stored in **Harbor**.

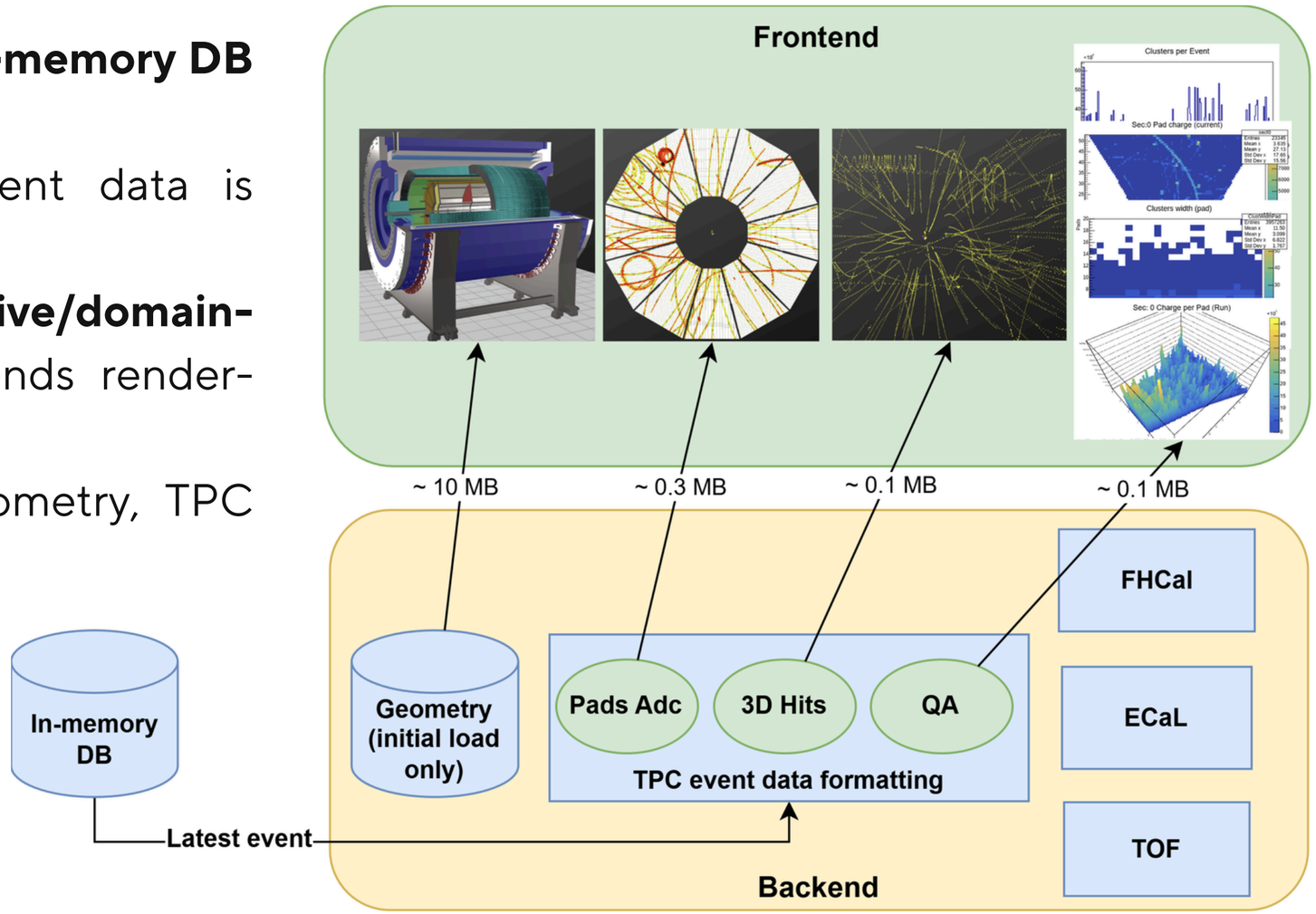
Ansible is used to configure computing nodes for Kubernetes **(Kubespray)**.



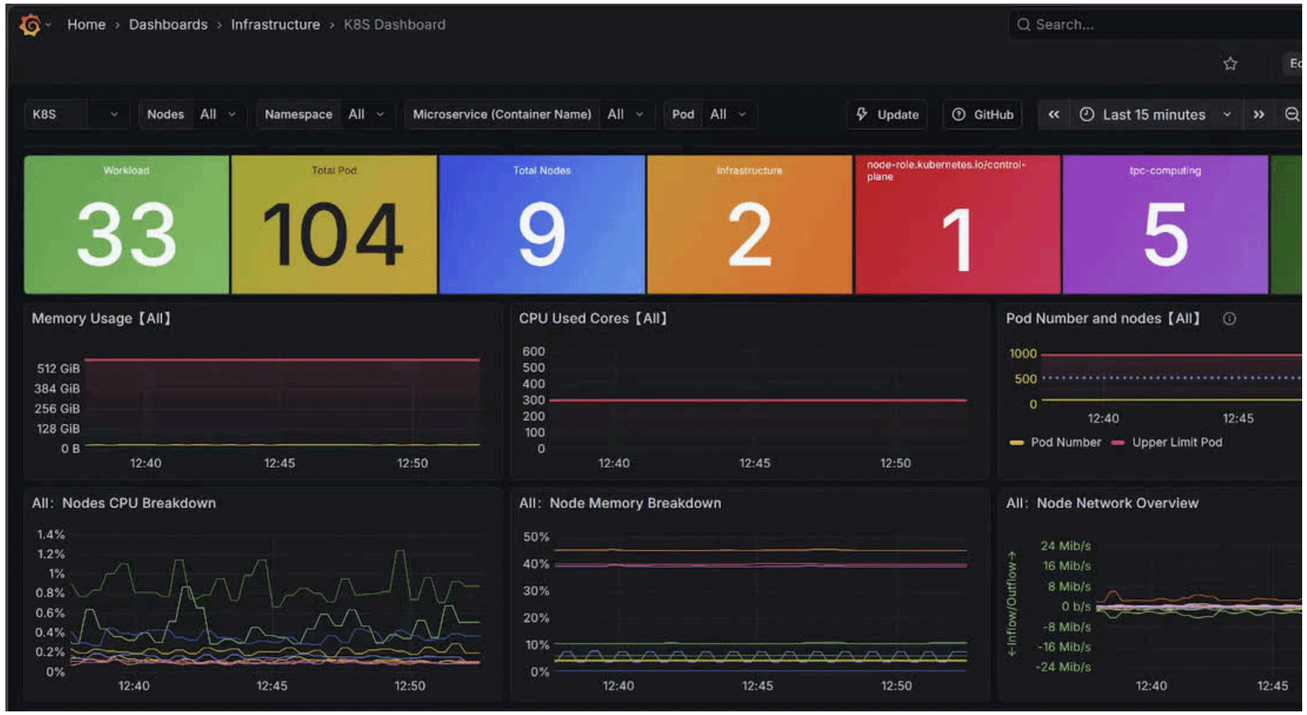
Implementation

QA histograms + Event Display

- Latest event is taken from the **in-memory DB (Valkey)**.
- Geometry is loaded **once**, event data is streamed **separately**.
- Backend performs the **expensive/domain-specific transformation** and sends render-ready buffers to the frontend.
- Frontend **displays** detector geometry, TPC hits, and QA histograms.

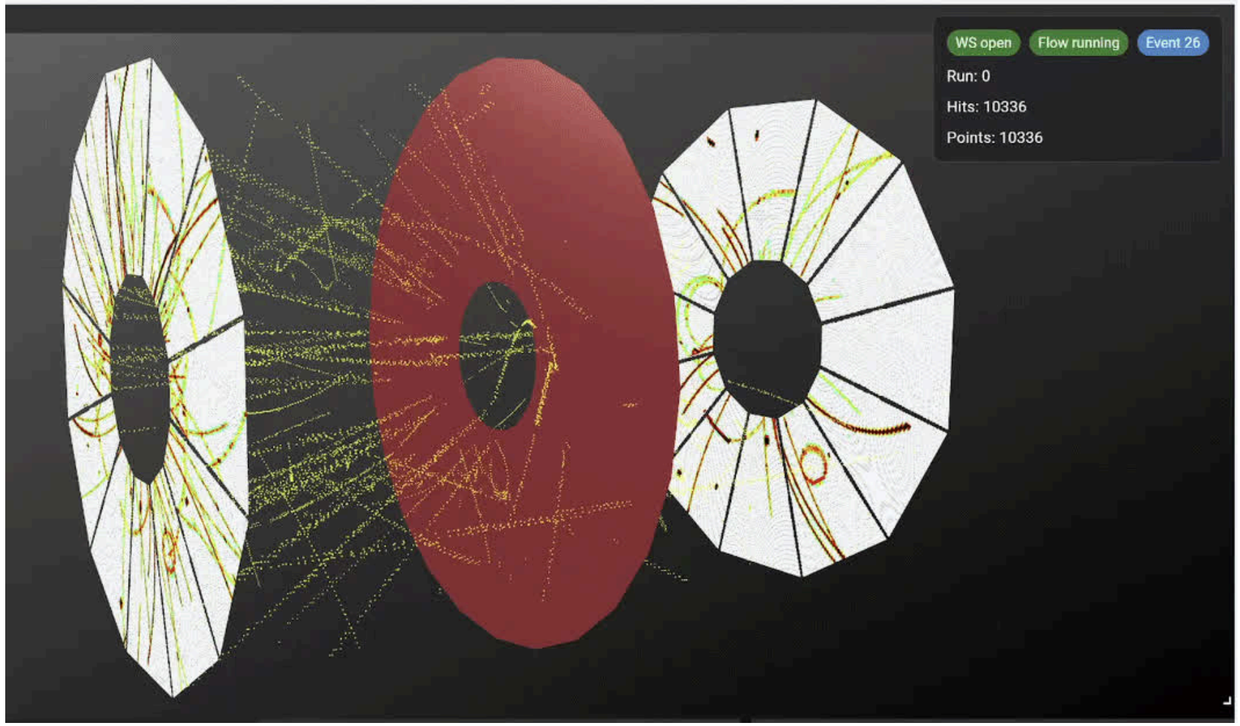


Implementation



Grafana dashboards

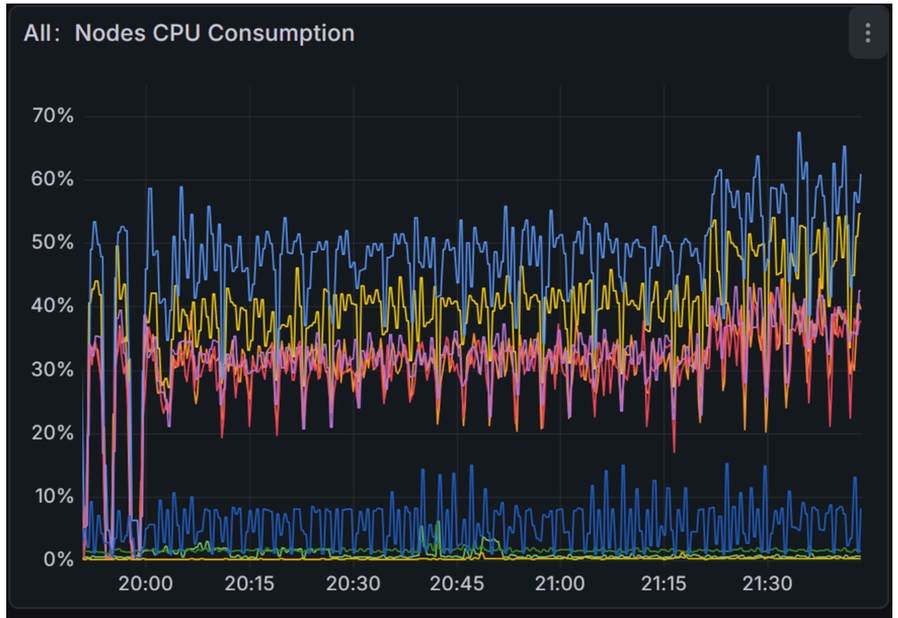
Event Display

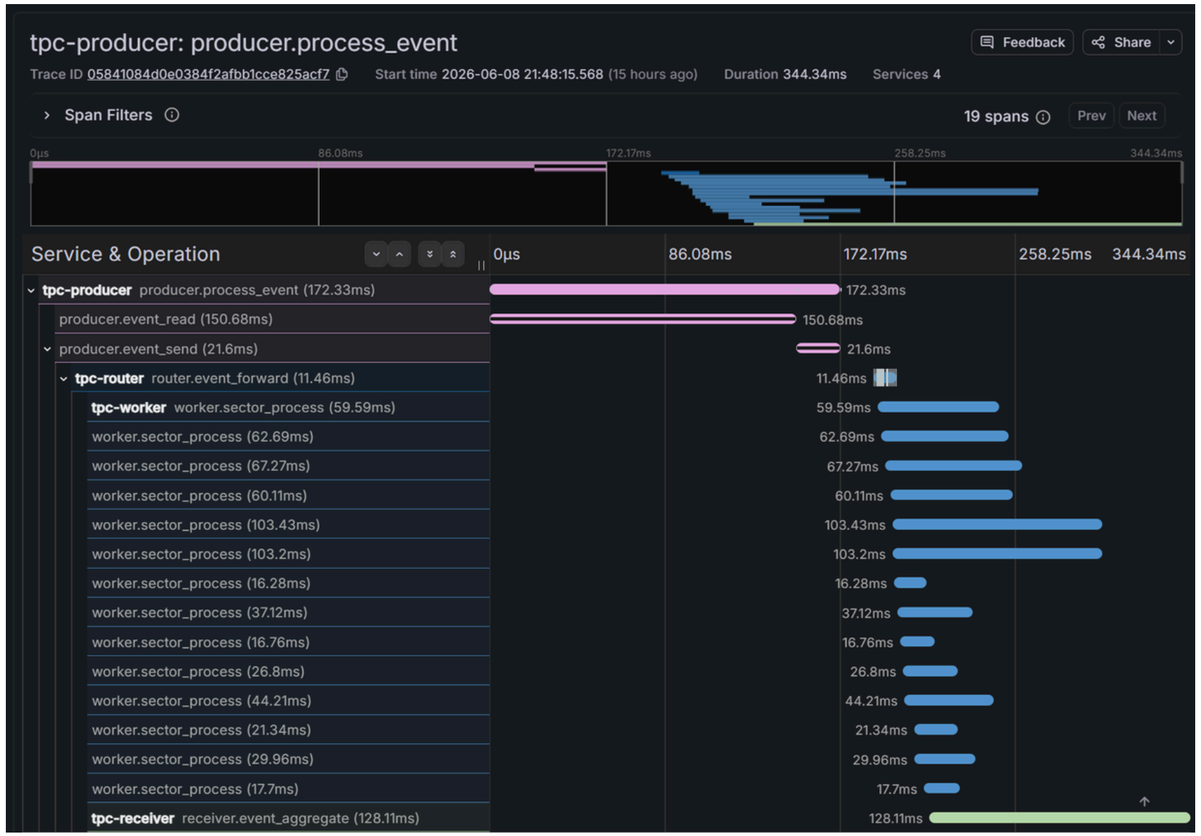


Implementation

All: Node Information Detail

Node	Pod Limit	Pod Number of	CPU Usage%	Memory Usage%	Disk Usage%	CPU Total Cores	CPU Core Usage
mpd-qa-valkey.he.jinr.ru	110	9	4%	3%	2%	8	0.329
mpd-qa-obs.he.jinr.ru	110	12	0%	8%	2%	40	0.0774
mpd-qa-k8s.he.jinr.ru	110	13	2%	8%	45%	16	0.269
mpd-qa-evd5.he.jinr.ru	110	13	35%	25%	12%	40	13.8
mpd-qa-evd4.he.jinr.ru	110	14	30%	27%	12%	40	12.0
mpd-qa-evd3.he.jinr.ru	110	14	29%	26%	12%	40	11.7
mpd-qa-evd2.he.jinr.ru	110	17	51%	14%	45%	40	20.2
mpd-qa-evd1.he.jinr.ru	110	16	40%	5%	18%	40	16.1
mpd-eds.he.jinr.ru	110	25	0%	15%	19%	40	0.177





Main pipeline is executed **within 100-150 ms**



Conclusion

Current results

In preparation for the launch of the MPD detector a **Kubernetes cluster for online data processing was implemented**.

MPD DQA cluster has:

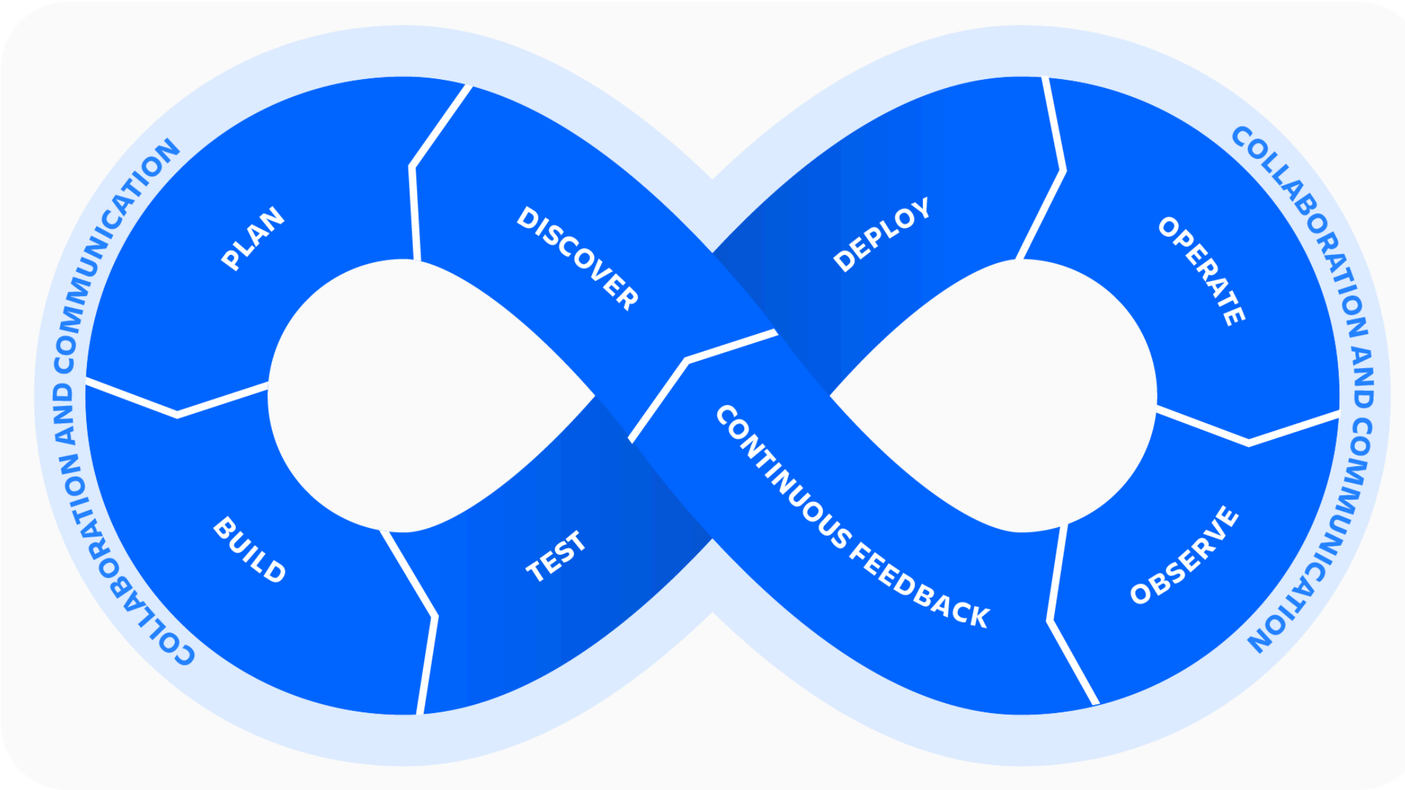
- **Online processing system** (data pipeline + Event Display);
- **Monitoring system** (metrics, logs, traces);
- **Security system** (Vault, certificates and secrets);
- **Automated infrastructure management System** (IaC);



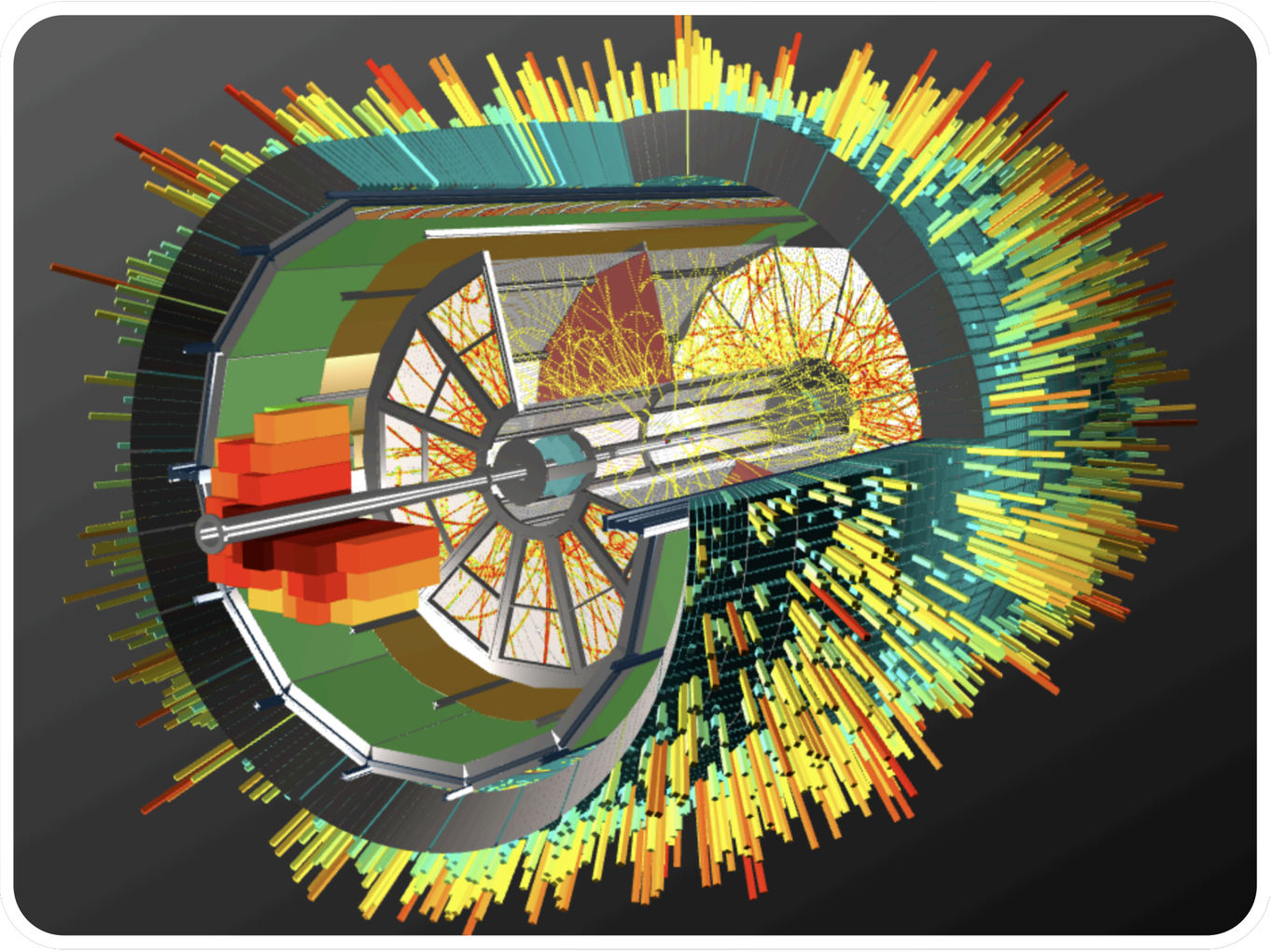
Conclusion

Future plans:

- **Optimization** of network interaction and data storage;
- Connecting **Keycloak authorization** to services from the UI.
- **Launching and testing** a full cycle of online processing on real data;
- **Adding monitoring** of other detectors.



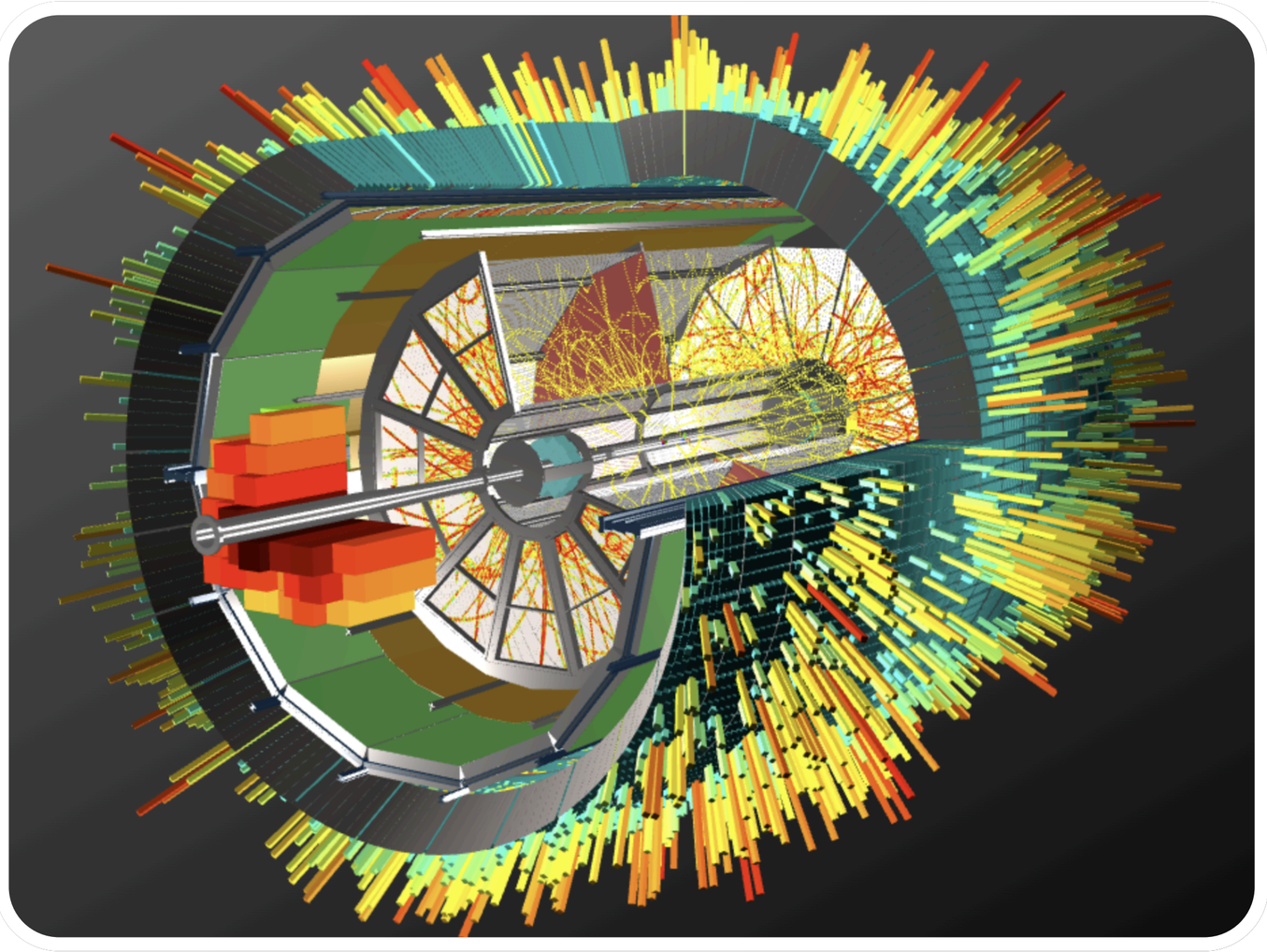
Thank you for your attention!



Andrey Budnikov, budnikov@jinr.ru
Alexander Krylov, avkrylov@jinr.ru

JINR

Backup slides





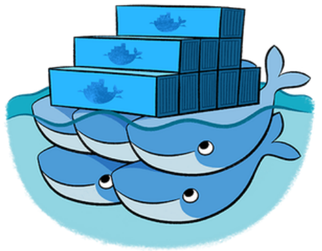
Introduction

Orchestrators

are tools that **automate, manage and coordinate** complex IT workflows, **containerized** (moslly) applications, and infrastructure.

Popular orchestration platforms:

- Docker Swarm
- **Kubernetes**
- HashiCorp Nomad
- Apache Mesos
- OpenShift
- Google Container Engine (GKE)
- etc...





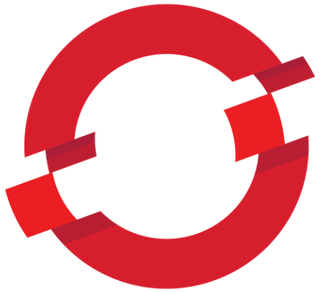
kubernetes



HashiCorp
Nomad



Apache
MESOS

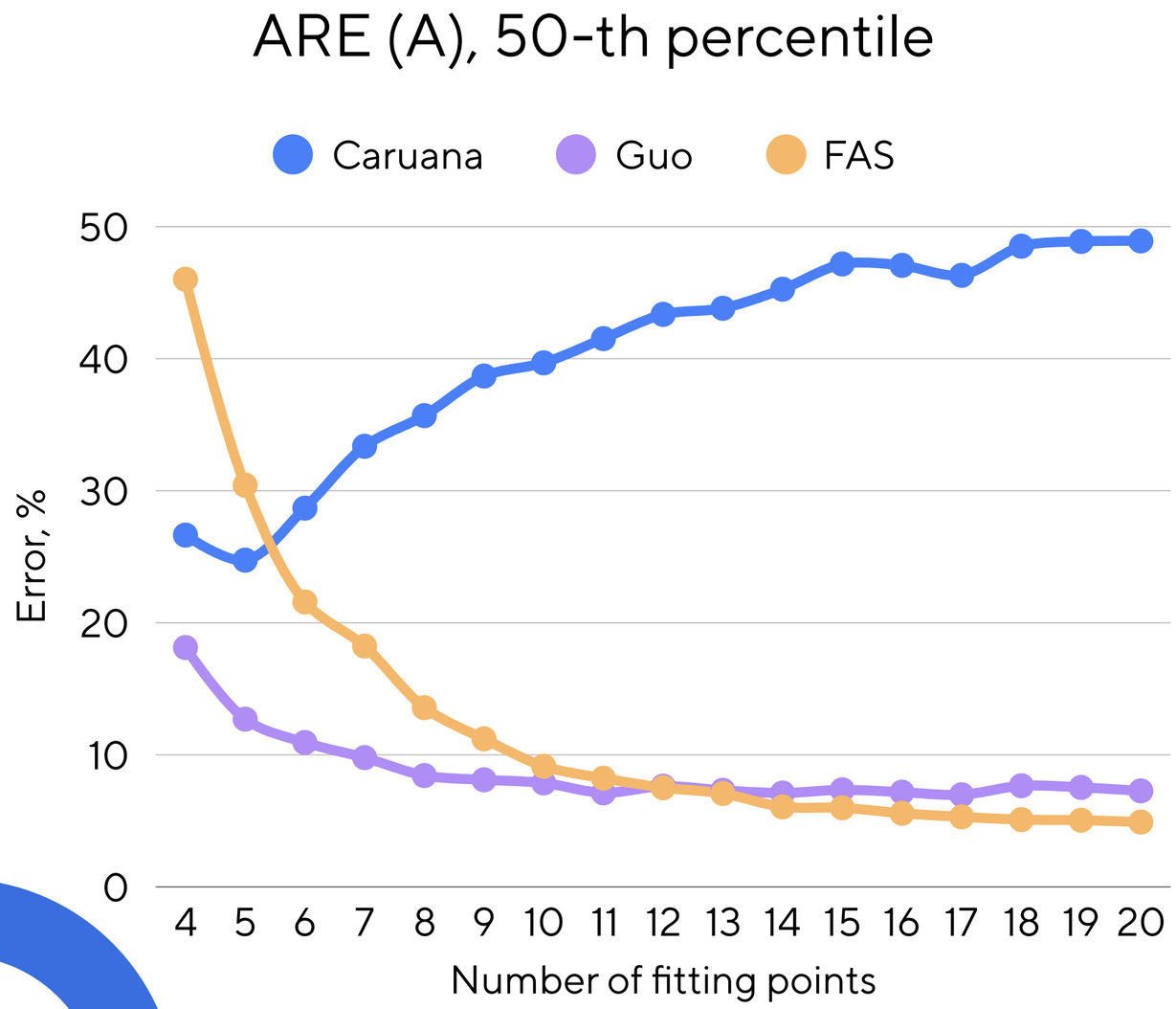


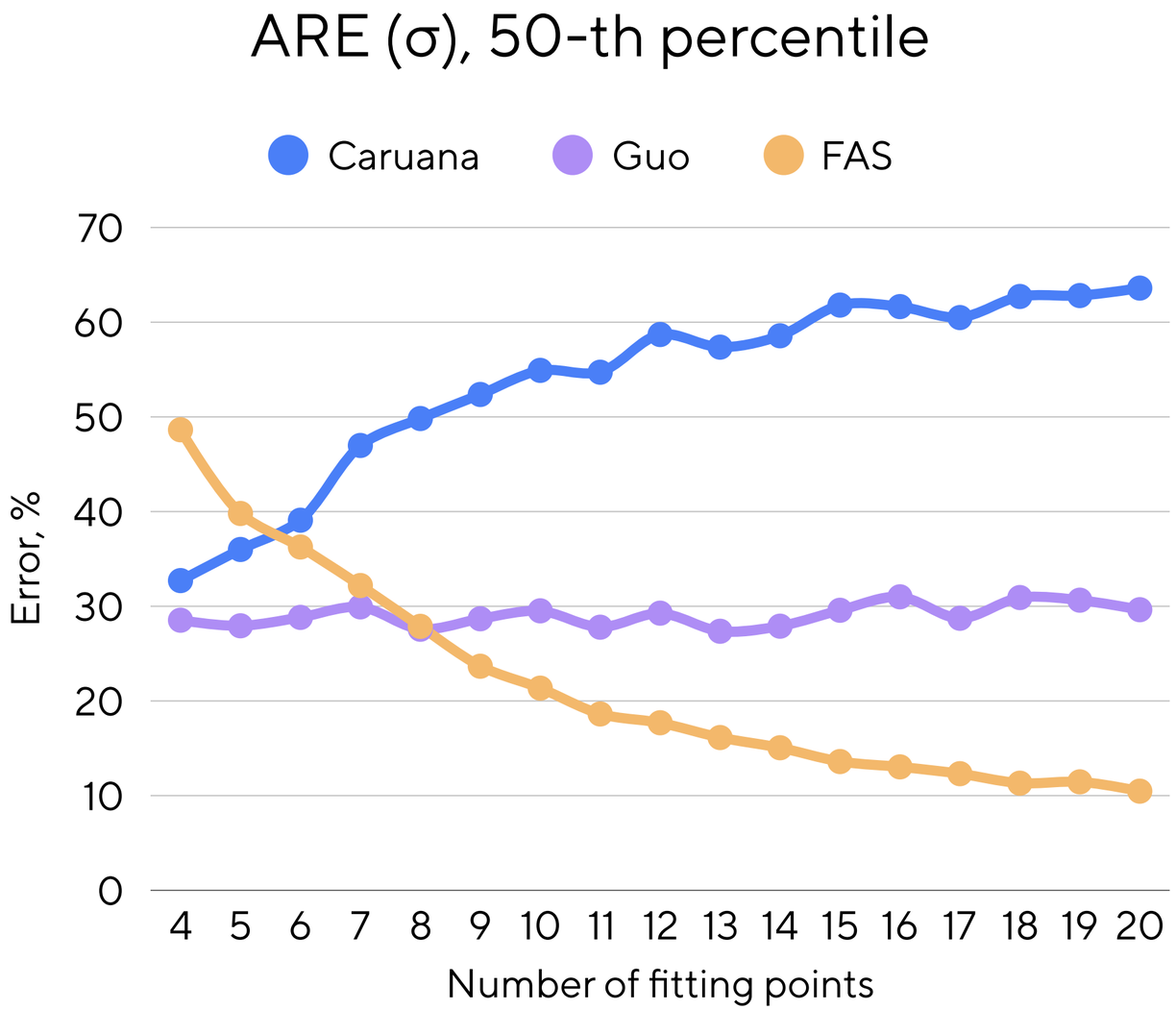
OPENSHIFT



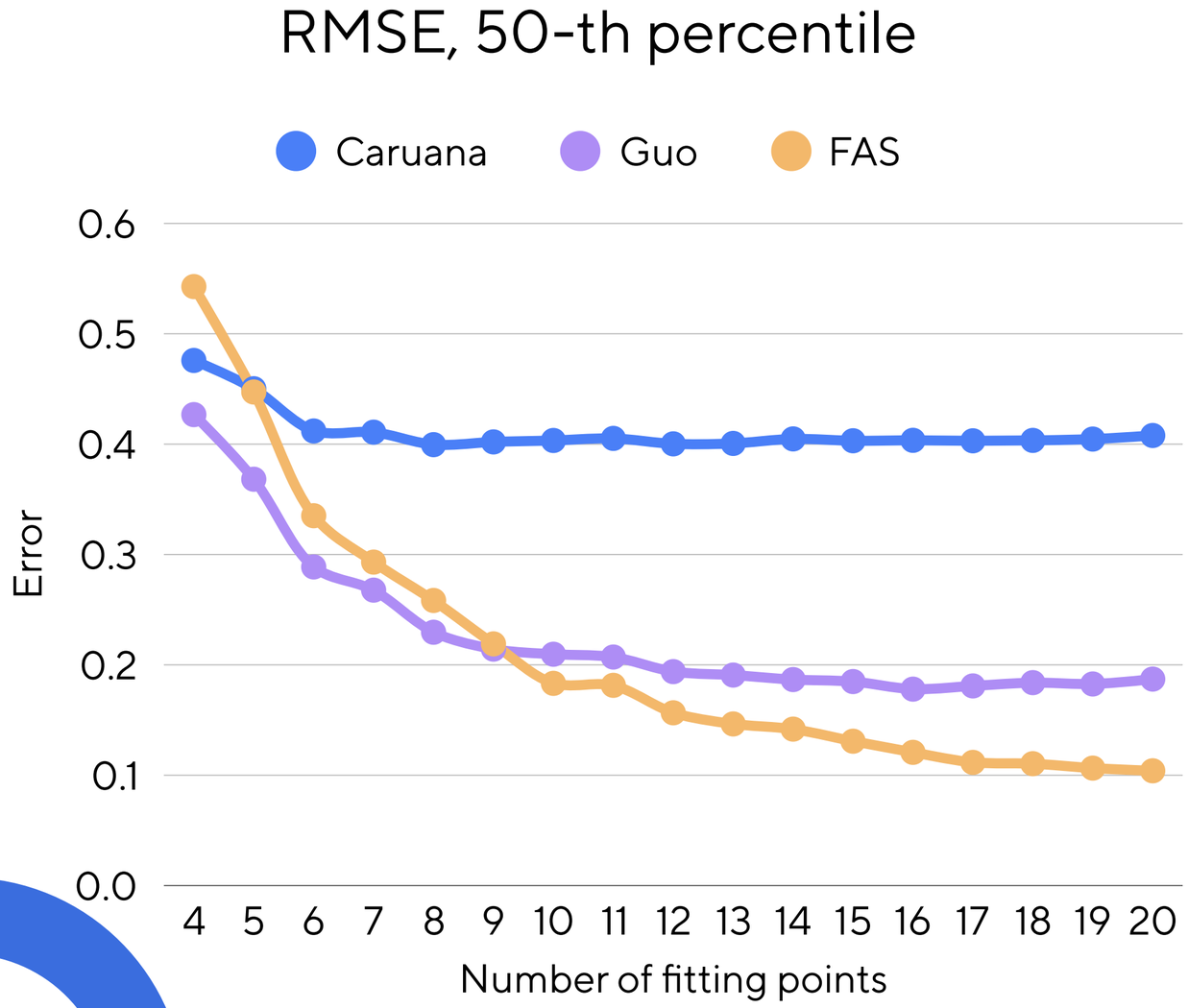
Google Kubernetes Engine

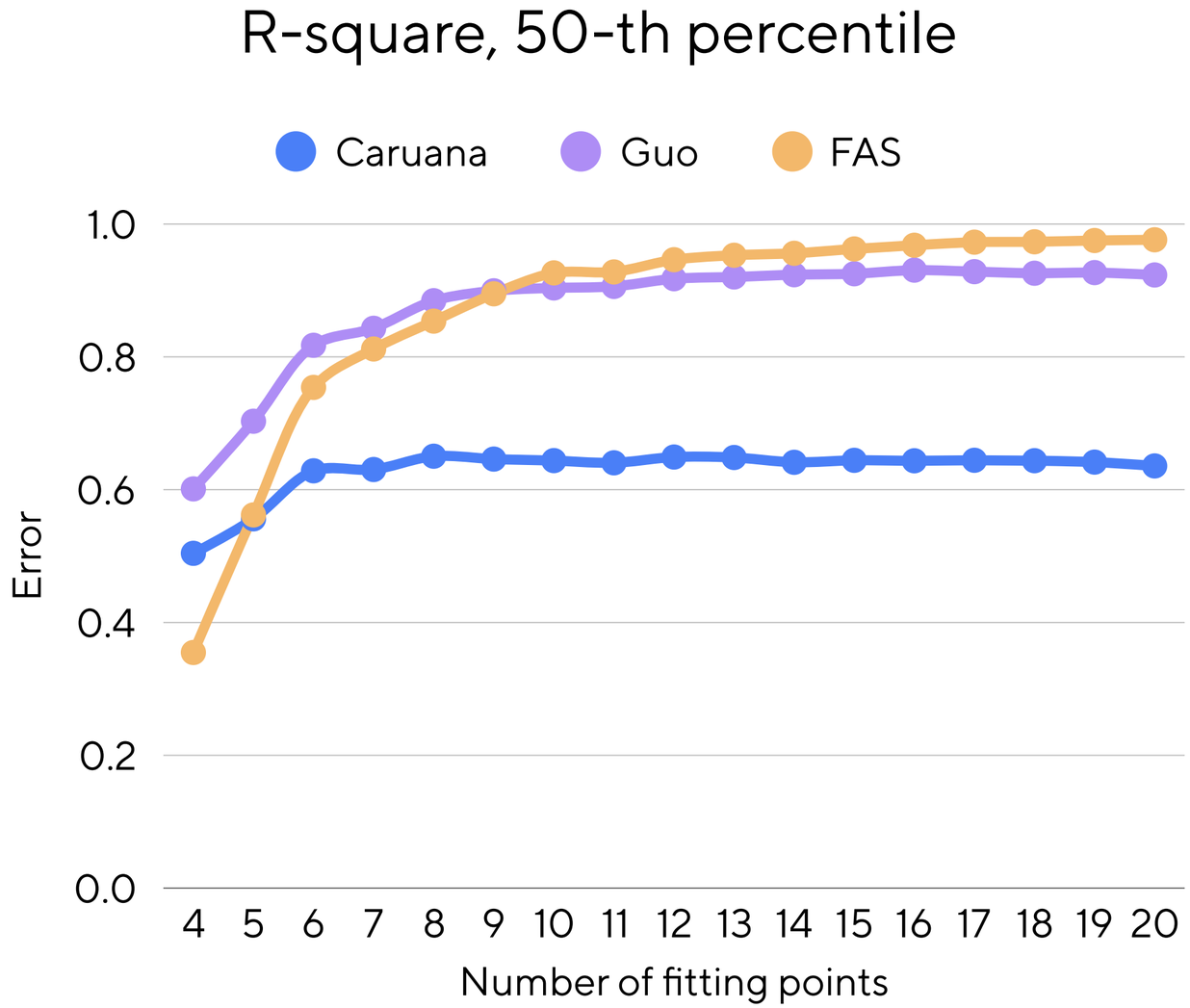
Online data processing





Online data processing





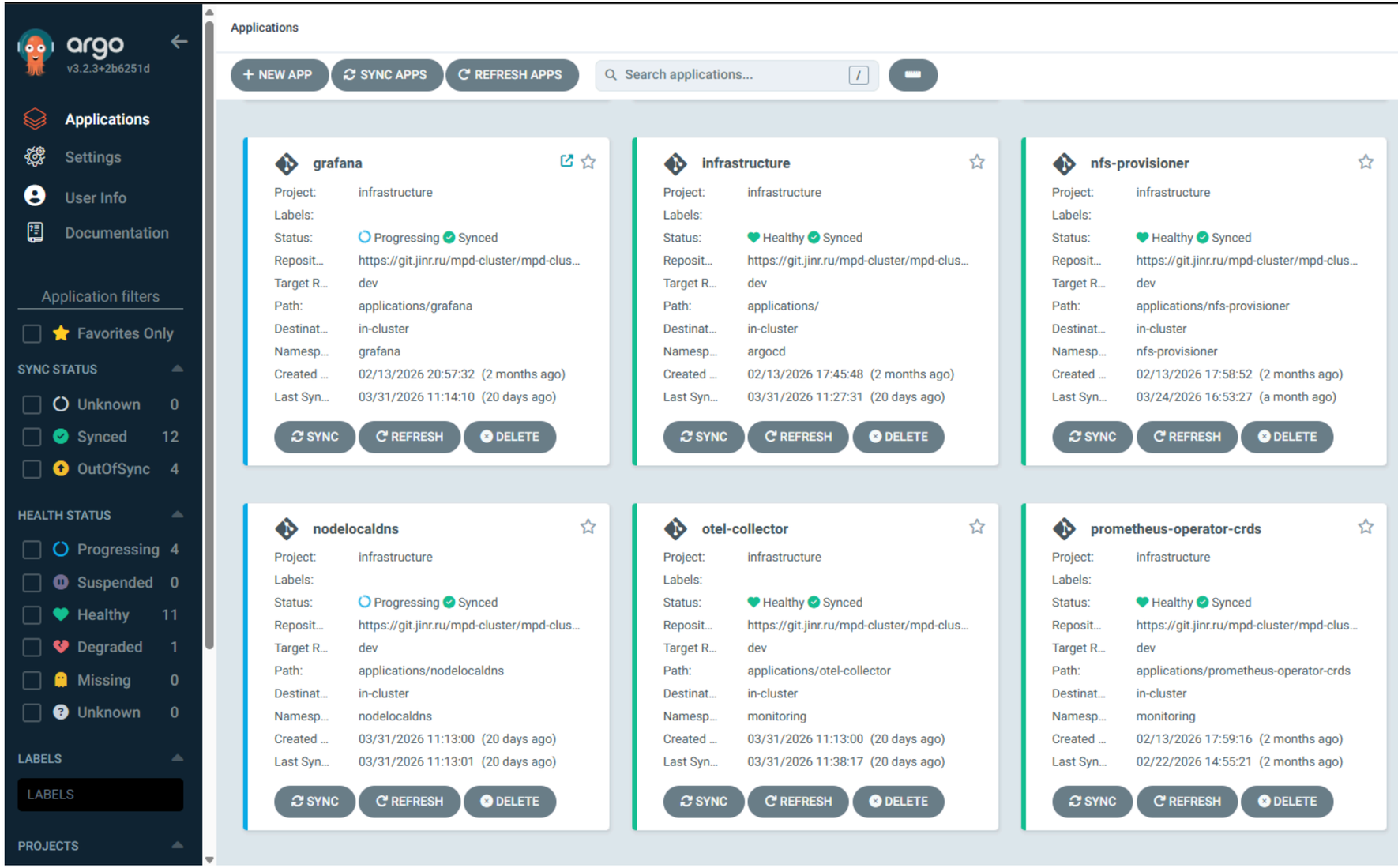
Implementation

Argo CD

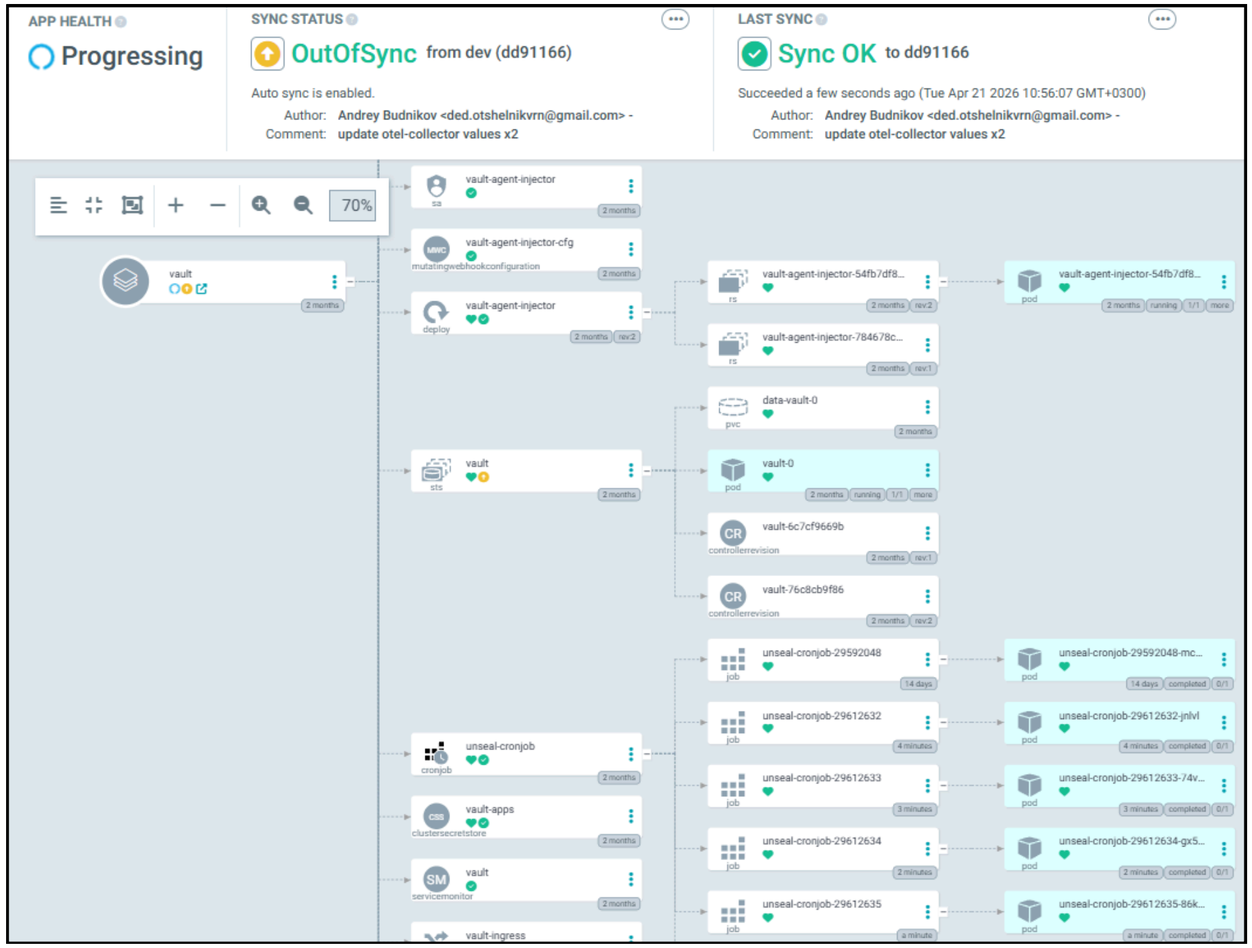
is a declarative, **GitOps** continuous delivery tool for Kubernetes.

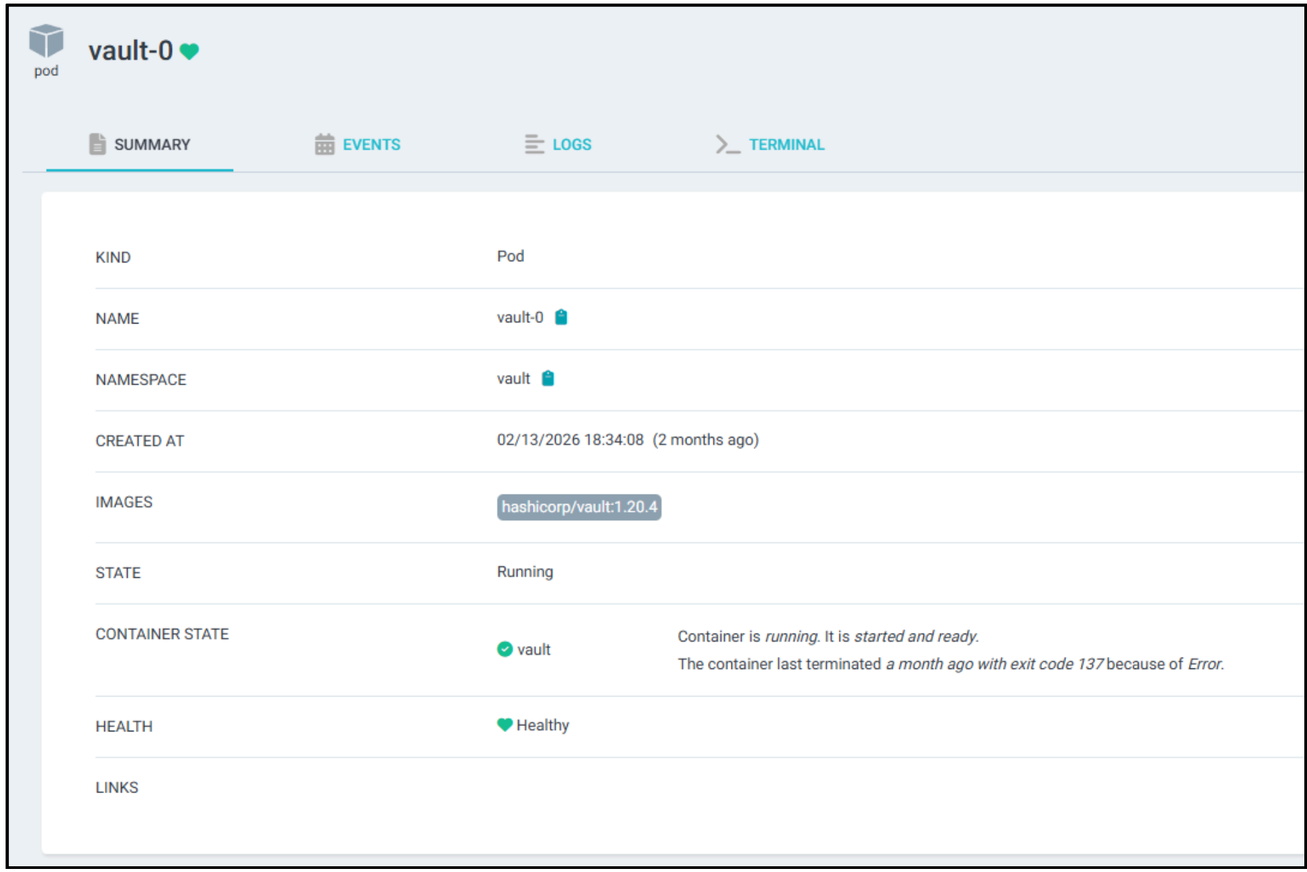
Main idea: configuration and environments **must** be declarative and version controlled (Git repos).

ArgoCD automate the deployment of the **desired** application states in the specified target environments or (!) clusters.



Implementation







Implementation

Harbor

Problem: users cannot connect to external artifacts registry from internal network.

Solution: deploy **Harbor** service that can:

- pull necessary artifacts from external registries and store it in cache;
- be used as independent and comprehensive registry.

