ZABBIX SUMMIT

OCTOBER 8 · 10, 2025
RIGA · LATVIA

Multi-Tenancy Zabbix Metrics Processing and Streaming

with Kafka, KSQL, and Grafana

About this talk

Real-world use case

The story of a solution we built and keep developing for one of our customers that had a special set of challenges and requirements.

Example of integration

Especially in larger environments, even the best of all monitoring utilities have to play the role of being just a little part of something.

Integration

A showcase of how data-driven observability can be built as built as a modern, microservice-based, extendible platform. platform.

In…Q…what?

Incubating tomorrow's IT today

Trusted Zabbix Partner & Trainer

Deep expertise from countless projects — we know Zabbix inside-out and help customers get the most out of it.

Observability for Modern IT

Modern applications have observability built in. It' s the key to assuring quality, guiding development, and keeping full control of infrastructure and apps.

Automation & Cloud-Native Practices

No manual work, no fragile setups – everything must be automatable and declarative. We design microservices and integrate them with leading open-source tech.

We're a specialized IT consulting company focused IT Automation, Monitoring, Observability and Container Platforms.

We help our customers adopt and develop custom solutions, built on open-source, cloud native technologies.

Christian Anton

Owner | Consultant | Trainer

ZABBIX CERTIFIED PARTNER

ZABBIX CERTIFIED TRAINER

The Customer and Their Requirements

Health Care Organization

Consumes IT services from a service provider

Also runs some of their own infrastructure

Metrics View

Organization-specific metrics needed to be made available to the customer

Extendable and Multi-tenant

The solution should be capable of onboarding additional service provider' s monitoring as customer-owned equipment

Shared Service Zabbix

Zabbix is used by the Service Provider to monitor all kinds of IT equipment

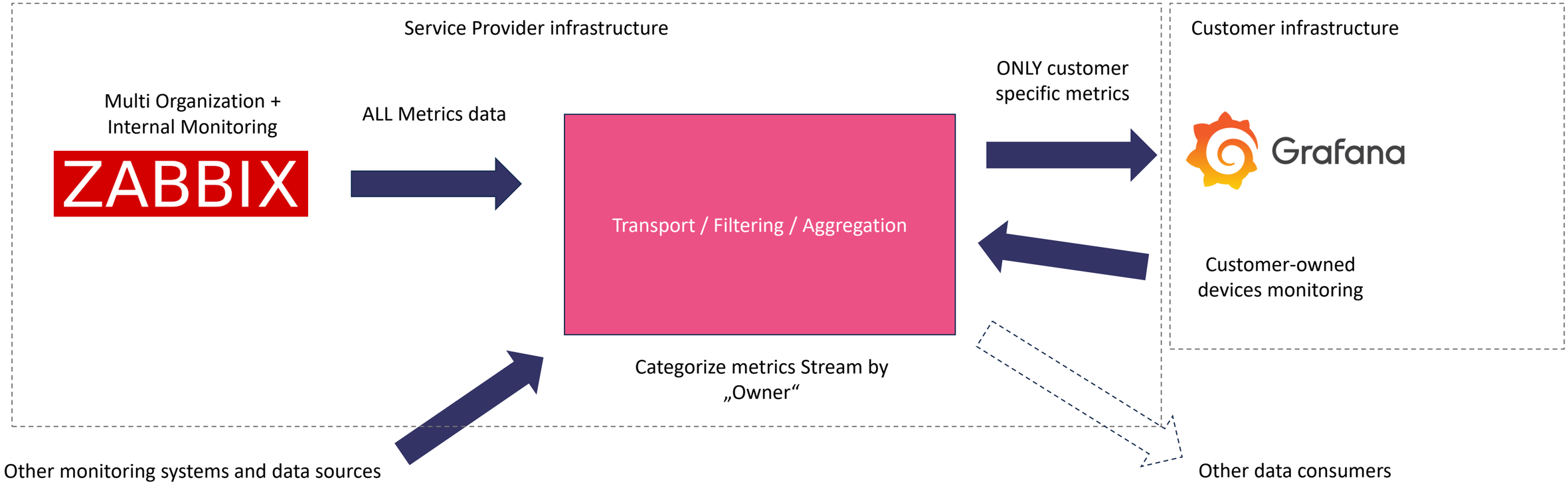
Visualization layer

Grafana was chosen as the desired platform for visualization and dashboards

Platform of choice

Stack should run in Red Hat Openshift, automated with ArgoCD

Basic Architecture



Turning the Idea into Architecture

Stream Processing

Apache Kafka – a robust and high performing data streaming platform with a large ecosystem

Target Format

Use Prometheus metrics format at the destination end - best to consume for Grafana

Kafka Deployment

Strimzi operator offers fully declarative management of Kafka Instances in Kubernetes + Openshift

Storage

Thanos is a high-performance long-term metrics store – S3 buckets downsampling, housekeeping

Getting Data into the Stack

Time-Series Data

High Volume (~6k NVPS)

Timestamp + Item-ID + Value

Datadog Vector

Future: Native HTTP Exporter + Kafka Bridge

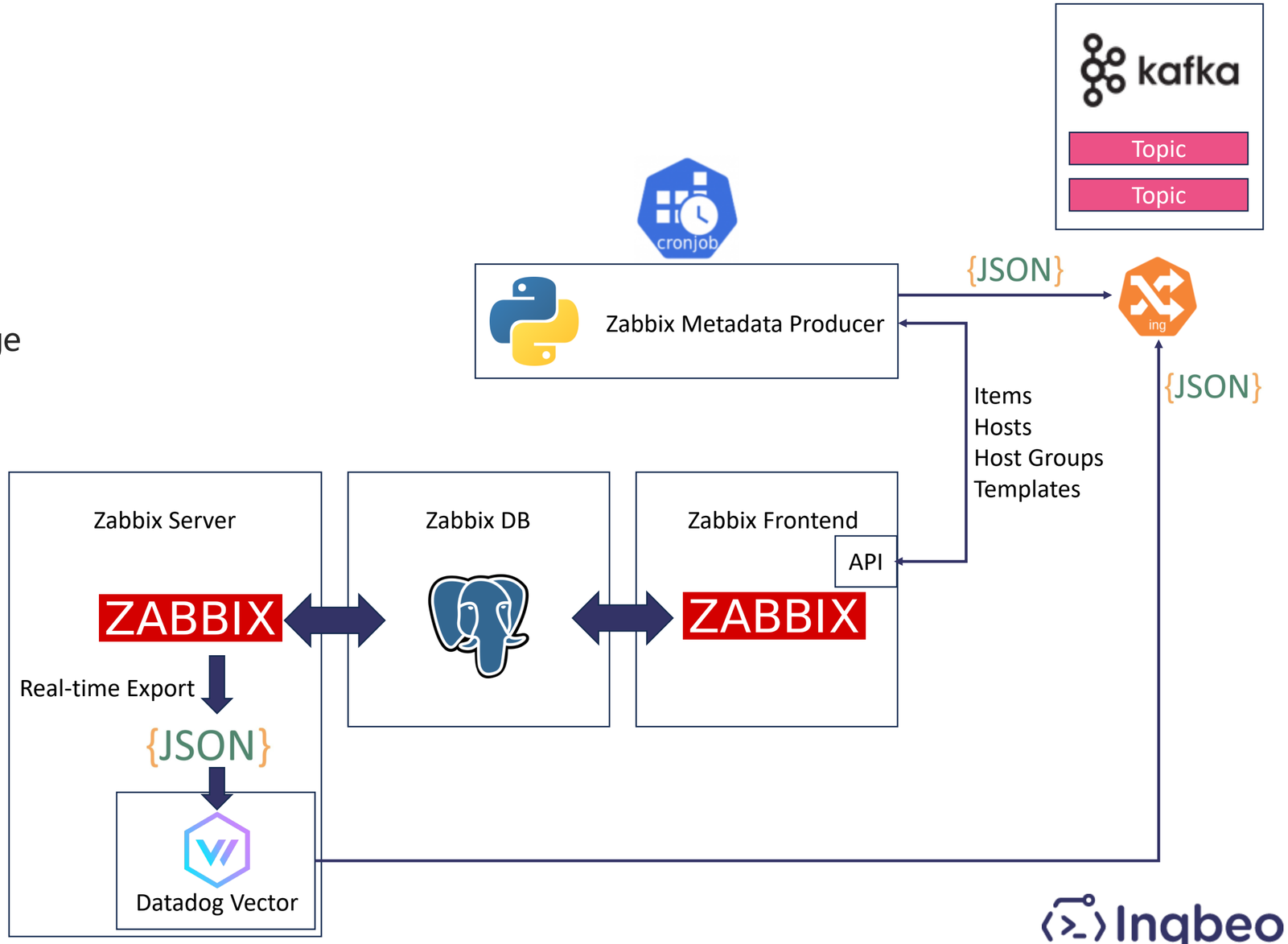
Metadata Producer

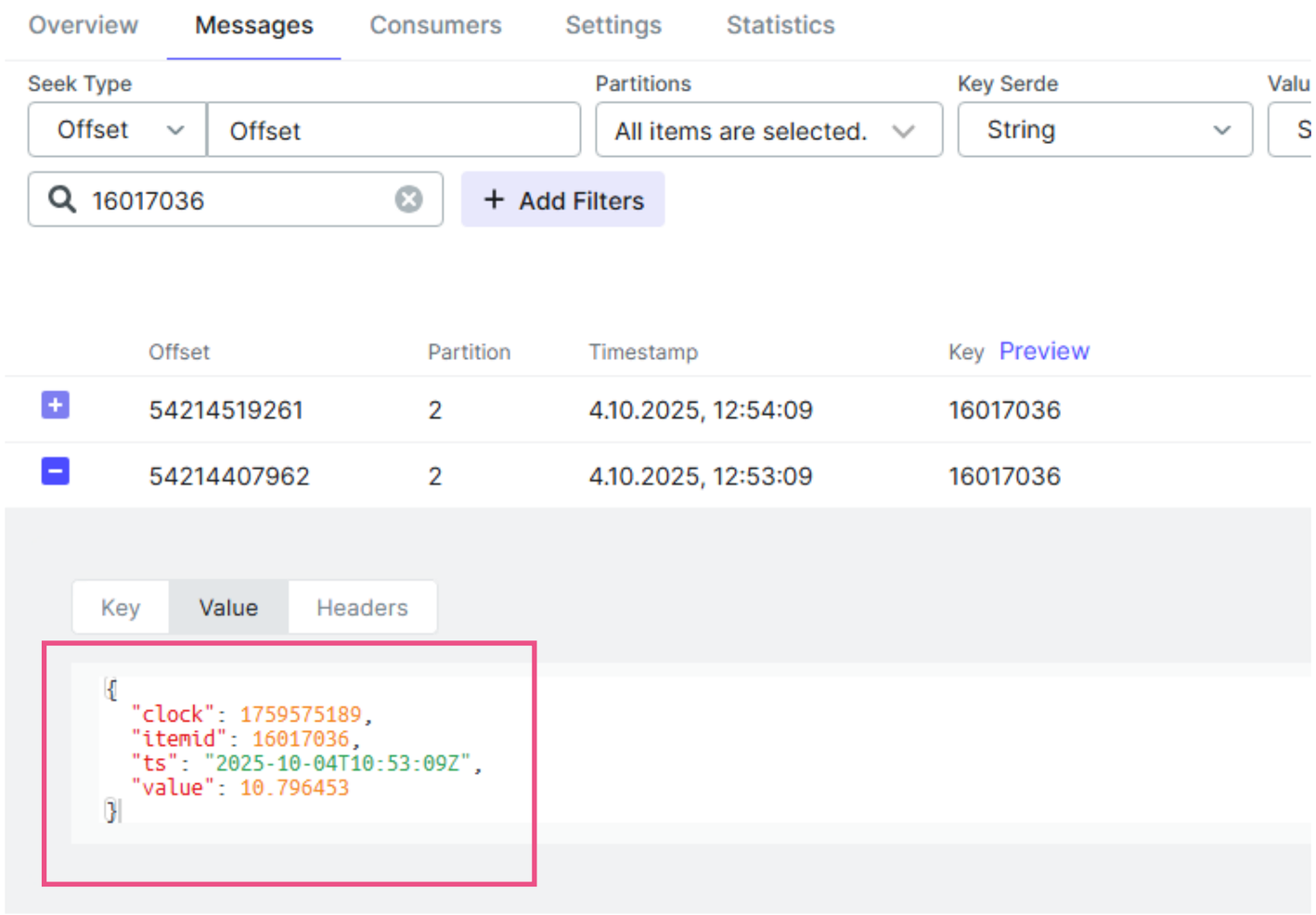
Python program, runs as Cron job

Mapping of Item-ID with Item Name, Key, Host Name, Group, Template, Tags, …

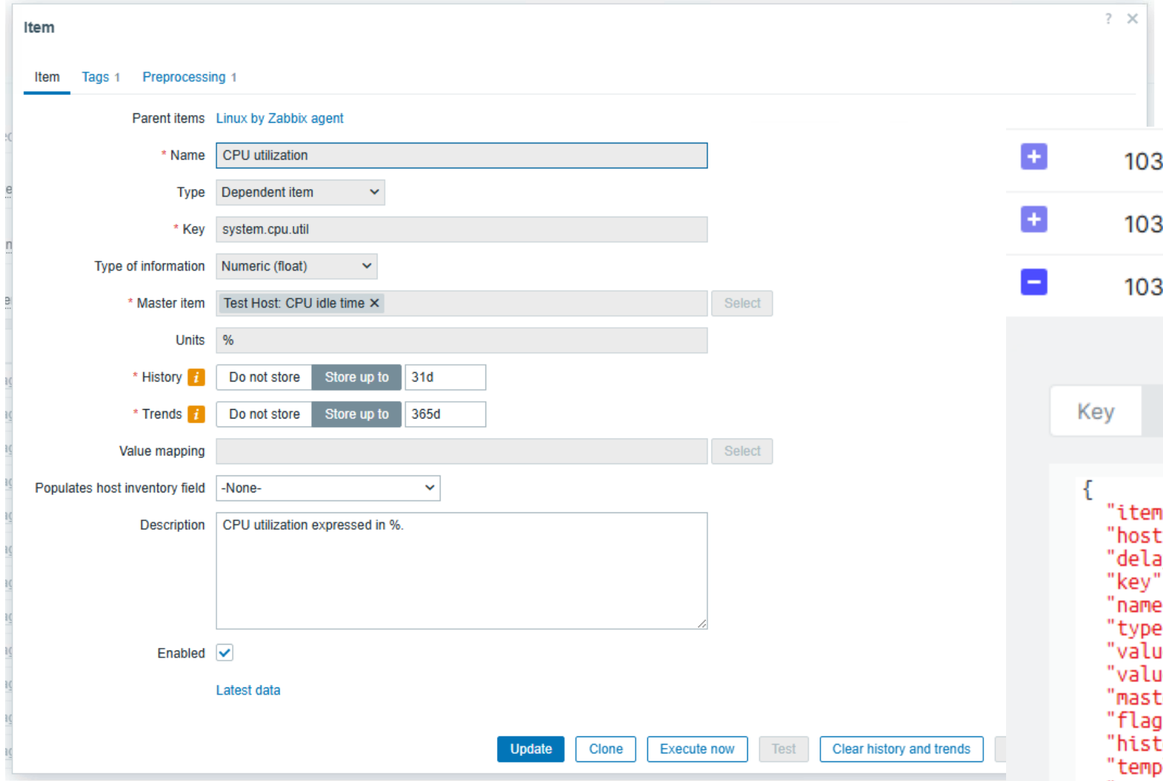
Categorization Ruleset

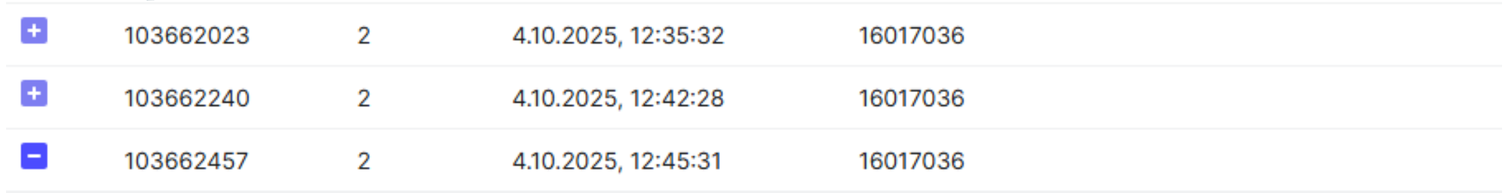
- Tag selection
- Regex on Item Keys, Names
- Host Names and Host Groups



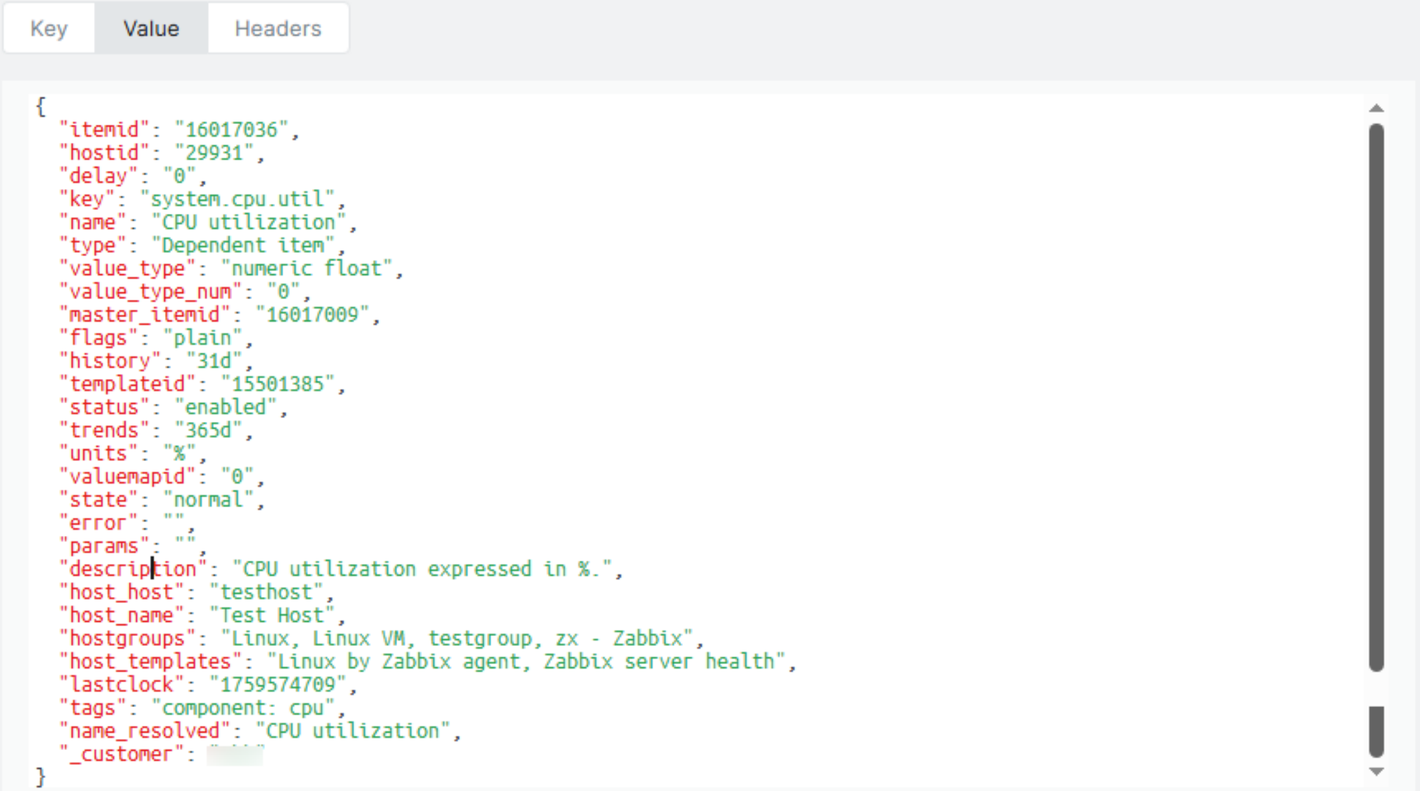


Item Value Message





+	103662023	2	4.10.2025, 12:35:32	16017036
+	103662240	2	4.10.2025, 12:42:28	16017036
−	103662457	2	4.10.2025, 12:45:31	16017036




```
501          - name: Test Host
502            attribute_name: _customer
503            attribute_value:
504            filters:
505            - item_attr: host_host
506              regex: testhost
507            - item_attr: hostgroups
508              regex: testgroup
509            - item_attr: key
510              regex: ^(icmpping|system.cpu.util.*|agent.ping|vm.memory.utilization|vm.memory.si
```

Item configuration, assignment filter, metadata message

Processing Data

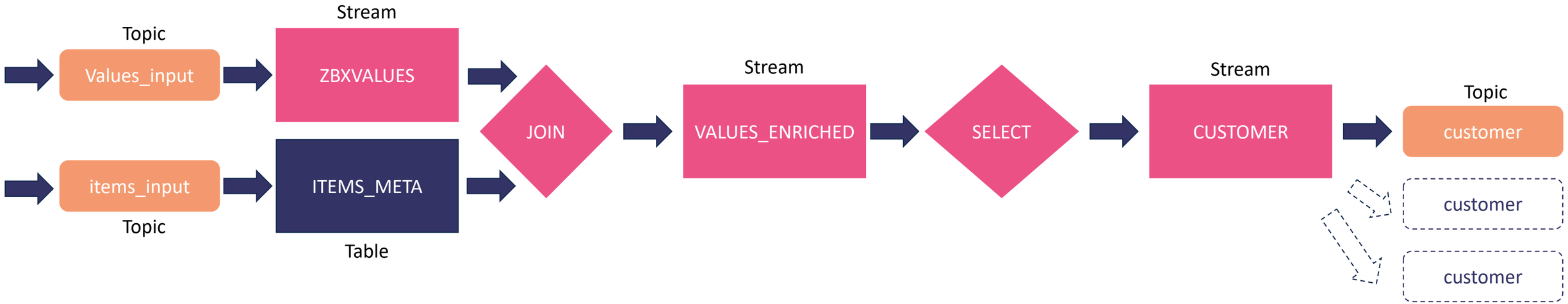
ksqlDB

SQL-like interface on top of Kafka

Generates Kafka Streams Applications internally

Real-Time Stream Processing

No JAVA Code



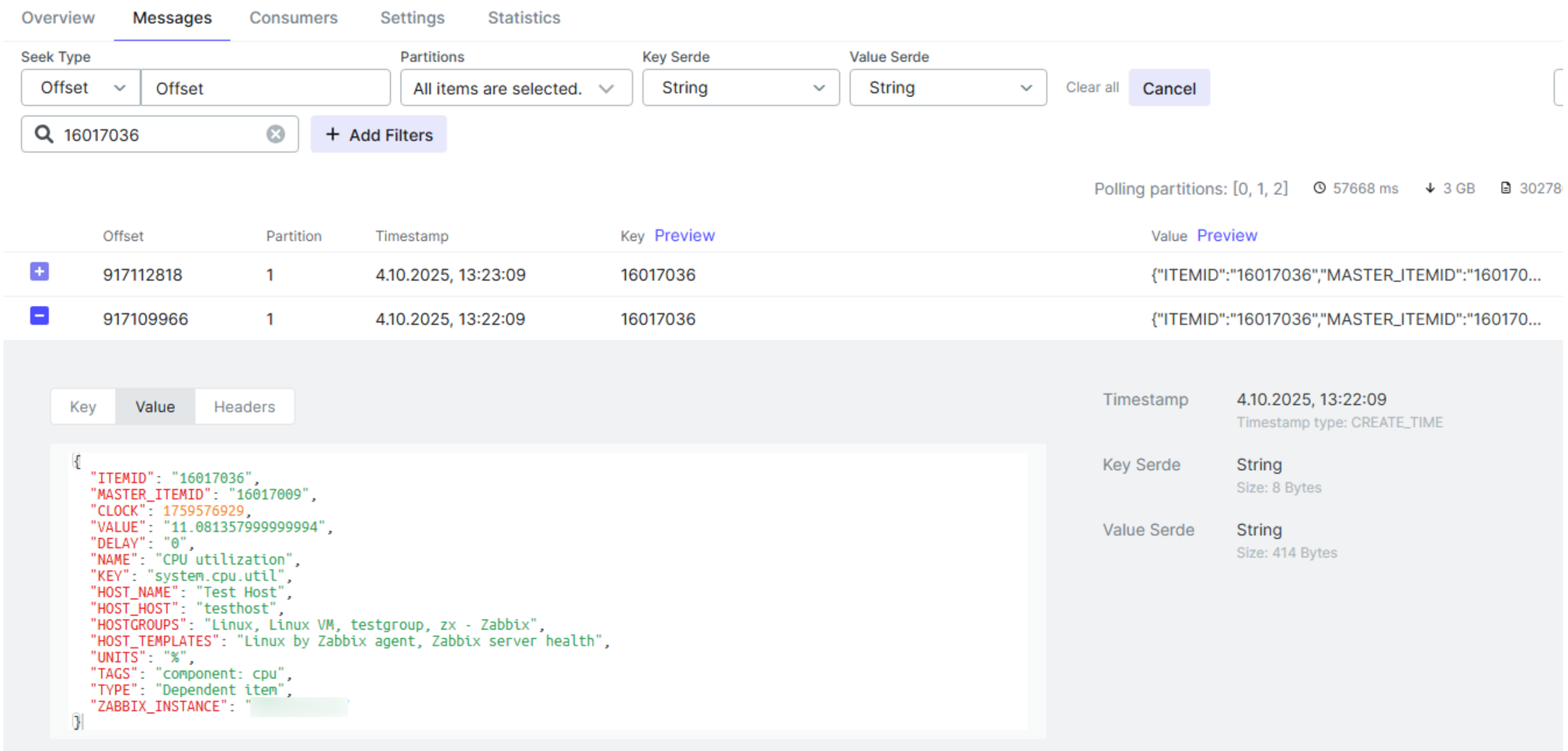
```
sqlQueries: |
  CREATE STREAM ZBXVALUES (itemid STRING KEY, clock INT, value STRING) WITH (KAFKA_TOPIC='values_input', VALUE_FORMAT='json');

  CREATE TABLE ITEMS_META (itemid STRING PRIMARY KEY, master_itemid STRING, delay VARCHAR, name VARCHAR, key VARCHAR, type VARCHAR, hostid INT, host_name
  VARCHAR, host_host VARCHAR, hostgroups VARCHAR, host_templates VARCHAR, units VARCHAR, description VARCHAR, tags VARCHAR, _customer VARCHAR) WITH
  (KAFKA_TOPIC='items_input', VALUE_FORMAT='json');

  CREATE STREAM VALUES_ENRICHED WITH (VALUE_FORMAT='json', PARTITIONS=3) AS SELECT i.itemid AS itemid, i.master_itemid, v.clock, v.value, i.delay, i.name,
  i.key, i.host_name, i.host_host, i.hostgroups, i.host_templates, i.units, i.tags, i.type, i._customer, 'zabbix' AS zabbix_instance FROM ZBXVALUES v LEFT
  JOIN ITEMS_META i ON v.itemid = i.itemid;

  CREATE STREAM CUSTOMER WITH (KAFKA_TOPIC='customer', VALUE_FORMAT='json', PARTITIONS=3) AS SELECT itemid AS itemid_key, AS_VALUE(itemid) AS itemid,
  master_itemid, clock, value, delay, name, key, host_name, host_host, hostgroups, host_templates, units, tags, type, zabbix_instance FROM VALUES_ENRICHED
  WHERE _customer = 'ourcustomer';
```

KsqlDB queries



Combined Kafka message

Getting Data Out

Metrics Receiver

Python program, running as a service

Consume data from Kafka topic

Conversion into Prometheus metrics format

Meta information as labels

Push data using Prometheus remote_write protocol

Thanos

S3 bucket (on prem) as data storage

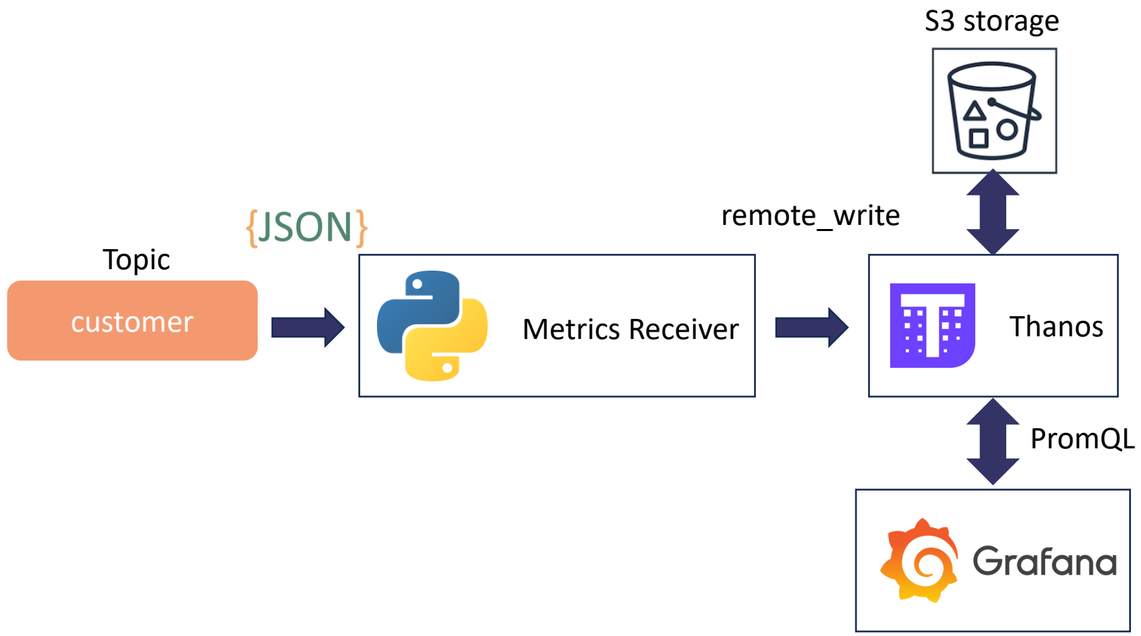
Supports downsampling, "housekeeping"

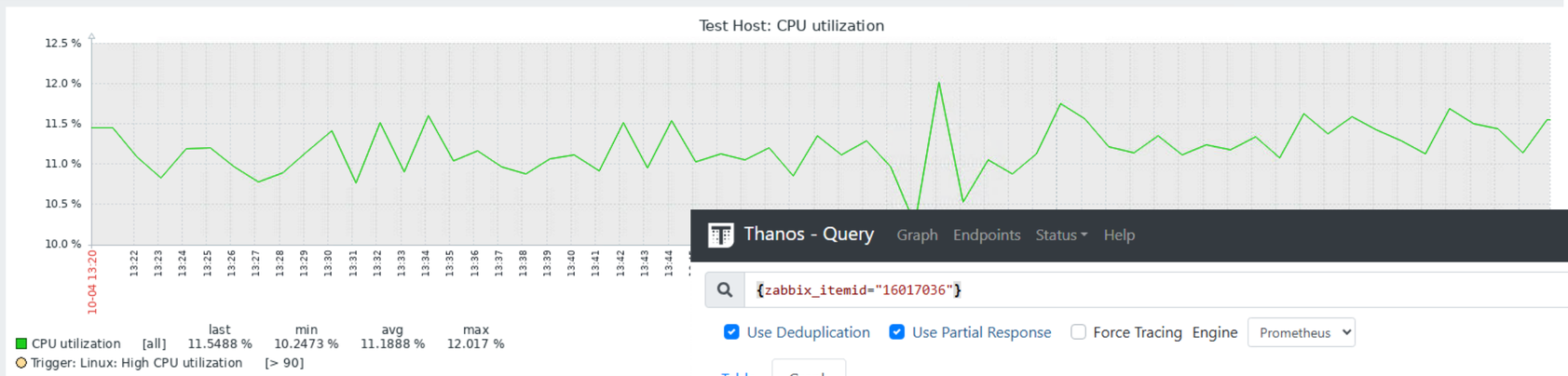
Provides Query API for Grafana

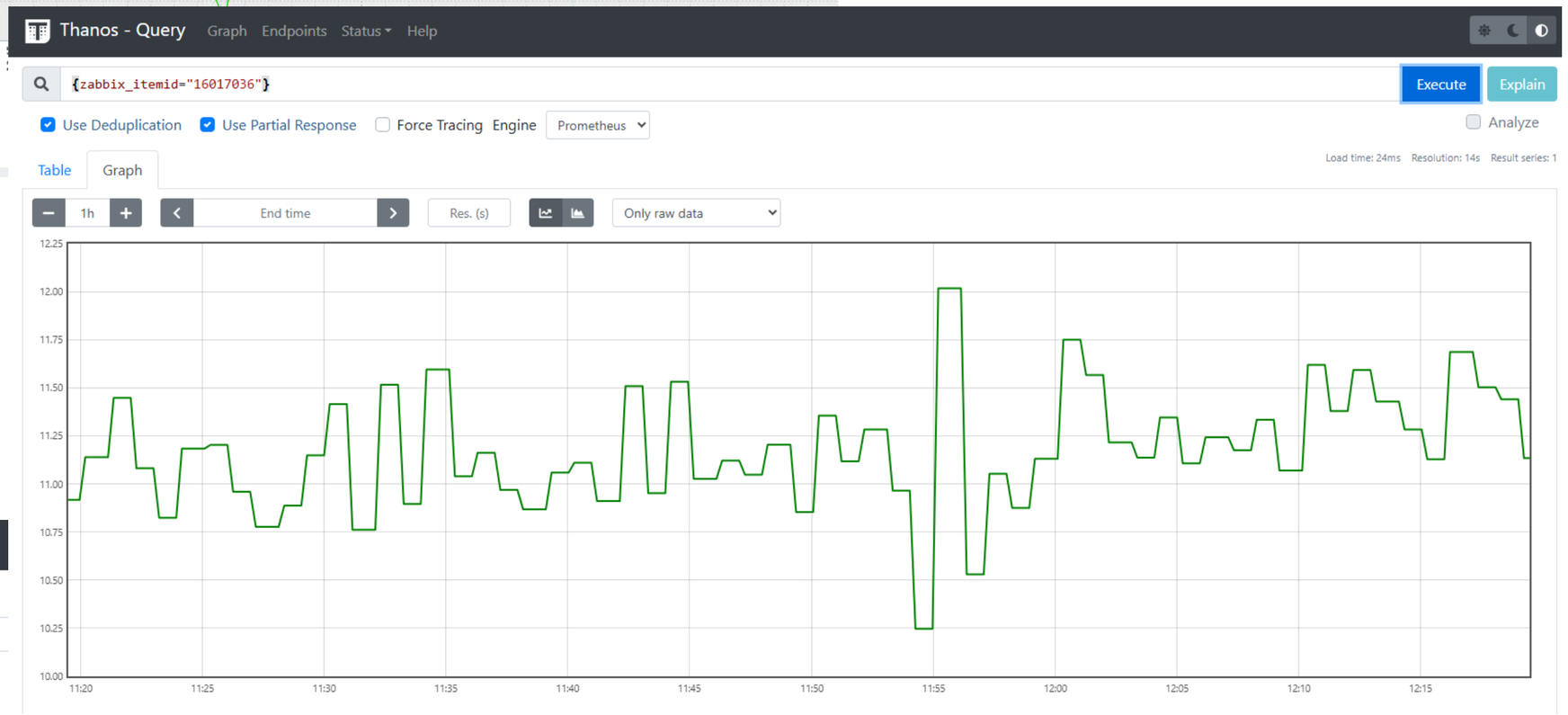
Grafana

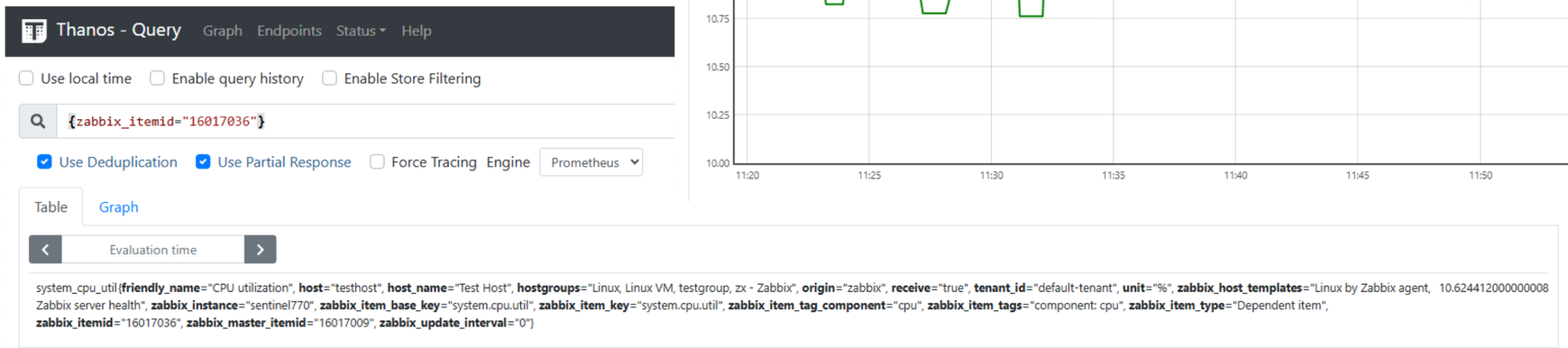
Presentation layer

Rich dashboarding capabilities using PromQL









A single metric in Zabbix and Thanos

There's so much more…

Redis exporter

Generation of Prometheus metrics out of Redis HASH objects

Redis feeders

Query inventory data of Servers, Databases, Network Devices, Printers… from a CMDB and generate Hash entries in the Redis DB.

Also generate entries of " Locations" with Geo coordinates, addresses …

Messaging transport bridge

A little microservice implementing a "Subscriber Model" so that users can just invite a Bot that emits specific alerts to a personal or group chat

Conclusion

Successfully

built an enterprise-ready, cloud-native observability pipeline with Zabbix as the central data generator

High-Performance

filtering and enrichment of Zabbix monitoring data to enable secure, scalable multi-tenant usage

Full Multi-Tenancy

both for additional data sources (e.g. Zabbix, Prometheus) and for customer-specific data isolation

Reusable

modular architecture that can be adapted or extended as a blueprint for other large-scale observability platforms

Thank you!!!

christian.anton@inqbeo.de