initMAX

ZABBIX PREMIUM PARTNER

ZABBIX CERTIFIED TRAINER

Zabbix Online Meetup January 2026

Migrating Enterprise Monitoring: From IBM Netcool to Zabbix

all your microphones are muted

ask your questions in Q&A, not in the Chat

use Chat for discussion, networking or applause

1

Customer request

initMAX

Customer request

Pain points with IBM Netcool (current state)

- **Legacy platform**: solution is outdated; upgrades are difficult/slow and bring high risk
- **Limited flexibility**: hard to adapt to new requirements and evolving operations
- **Vendor lock-in:** many “simple” changes cannot be done internally
  - Every request requires an external vendor and adds cost + delays
- **Customization is painful**: difficult to extend, integrate, or connect external systems
- **High licensing and maintenance costs**: paying for licenses instead of improving the platform
- **Slow time-to-value**: new dashboards/alerts/use cases take too long to deliver
- **More than a basic visualization layer:** it comes with a web client, a desktop client, and a dedicated in-house web application.

initMAX

Customer request

What the customer wants (future state)

- **Easy to maintain** and upgrade with predictable lifecycle
- L1-friendly operations: **similar workflows to Netcool**, but without old limitations
- License-free / open source approach: **invest budget into functionality**, automation, and scale — not licenses
- Modern UI & usability:
  - **simple web app access**
  - easy user management (**roles/permissions**)
  - **dashboards and tailored visualizations for teams**
- Full internal ownership:
  - **understand how it works**
  - develop/customize **internally** without external dependency
- Strong integration capabilities:
  - **robust API** for data export and integration
  - easy connection to TTS, CMDB, automation pipelines

Customer request

Customer: “Zabbix is popular, but everyone tells us our use case **can’t be done** with it. We’d be happy to hear your opinion — we want to give Zabbix a fair chance.” Internally, we’ve already spent about several months **testing it and evaluating other open-source alternatives**.”

initMAX: “After hearing how you operate and what you need, we already have a few ideas on how to implement it in Zabbix. The best next step is a joint, **free PoC to validate it** — and if it succeeds, we’ll move forward together with the full-scale solution.”

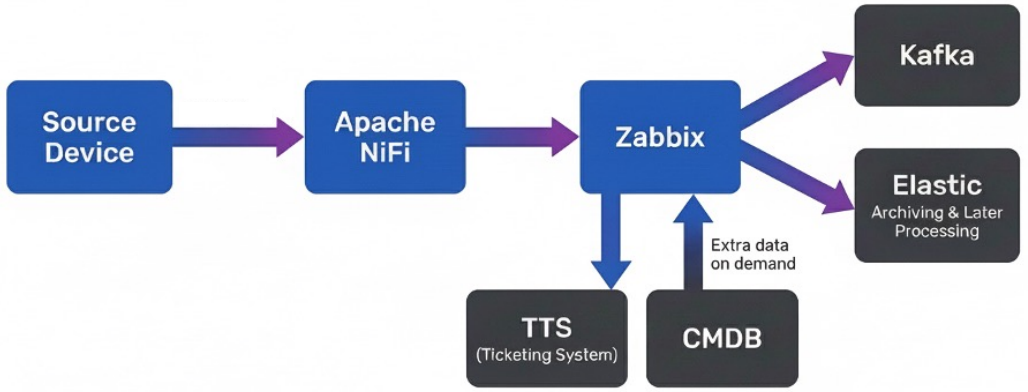
2

Architecture

Architecture

Architecture

- **Source data from SNMP traps** is converted by a custom library using **MIBs** and JSON-based transformation rules.
- **Apache NiFi** handles problem enrichment, data cleaning, and aggregation/counting.
- **Zabbix** receives the processed data via **Zabbix trapper** items and uses it to evaluate triggers and manage the full problem lifecycle (create and close problems).
- **Visualization** is provided through **custom widgets**, combined with additional Zabbix front-end modules.
- We also integrate **TTS** and support ad-hoc retrieval of device information via HTTP calls to the **CMDB**.
- **Kafka and Elasticsearch** are used later for reporting and additional downstream processing, while also preserving the original data stream as a source of truth.



initMAX

Architecture

High Availability & Resilience

- For this specific customer, we operate across three data centers (DCs) — each DC is capable of handling the full load of the entire stack (**active/active-ready** design).
- Every component has **DEV/TEST** environments in addition to production.
- The whole platform is driven by **GitOps**: changes are version-controlled, tested, and validated before deployment, minimizing human error.
- Infrastructure **provisioning is automated**:
  - device preconfiguration ("device prescript")
  - Ansible for VM provisioning and baseline configuration
- Apache NiFi configuration is fully managed as code.
- **Preprocessing:** 2×1 VM (active–standby), **6+6**, **32 GB RAM**, **8 vCPU**
- **Post-processing:** 2×1 VM (active–standby), **6+6**, **32 GB RAM**, **8 vCPU**

initMAX

Architecture

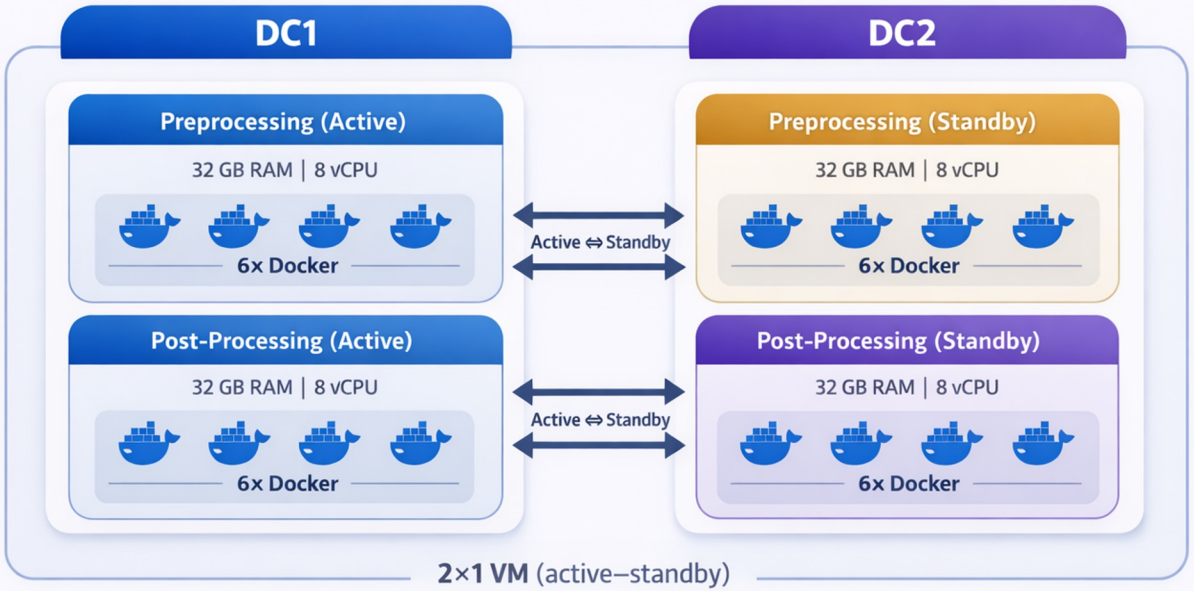
High Availability & Resilience

- Zabbix is deployed with high availability:
  - 3 separate frontends
  - 3 separate backend/server nodes
- PostgreSQL backend runs on a Patroni cluster with 3 data nodes and automatic failover/switchover.
- Traffic handling:
  - frontend traffic is load-balanced
  - backend traffic (port 10051) is routed to the active node
  - both load balancers are implemented on F5 BIG-IP

initMAX

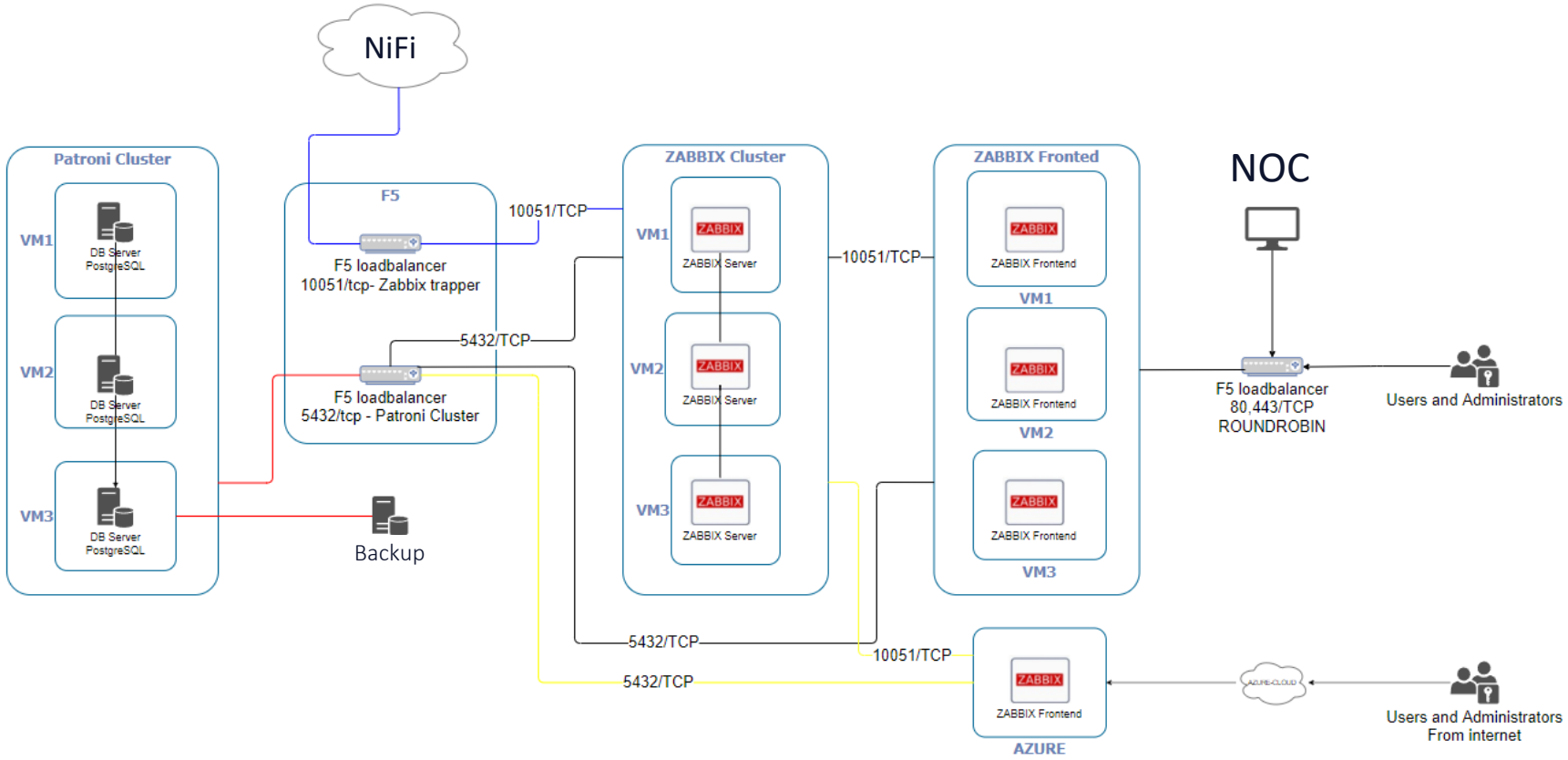
Architecture

High Availability – NiFi



Architecture

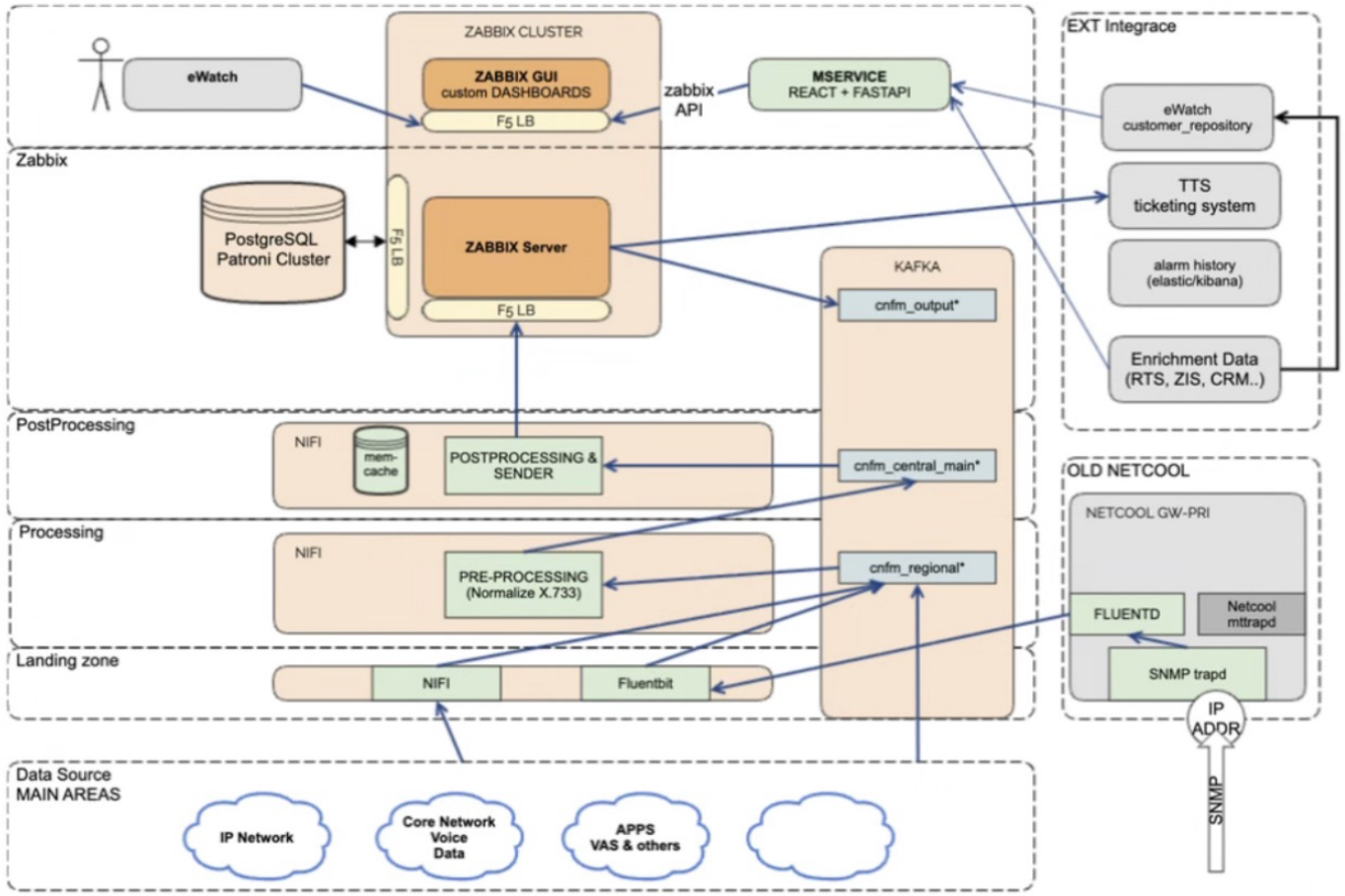
High Availability - Zabbix



initMAX

Architecture

High Availability – All in one



3

Problem Enrichment & Apache NiFi

initMAX

Problem Enrichment & Apache NiFi

Different inputs (heterogeneous sources)

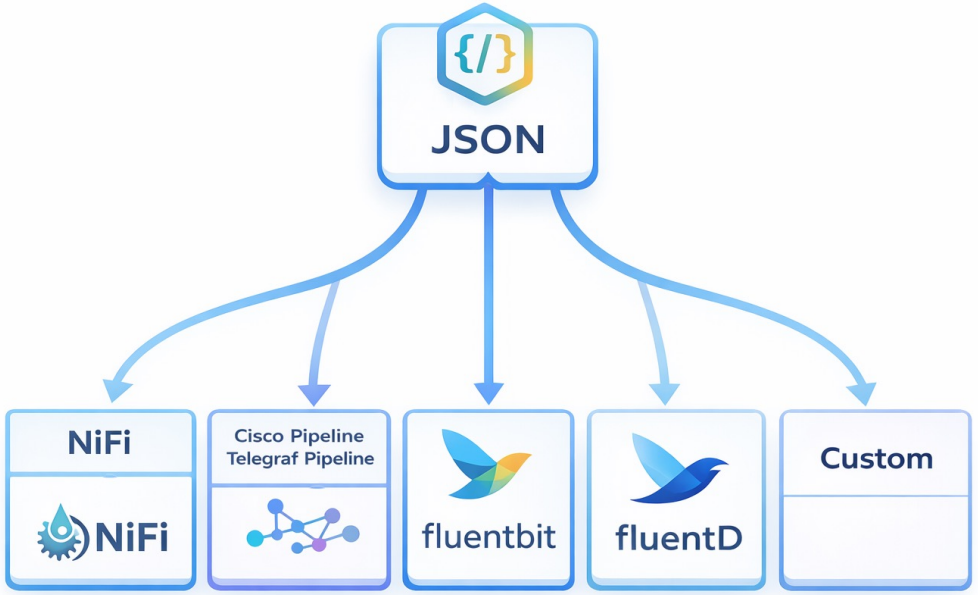
- **NiFi (1.2)** pulling data via **TCP/HTTP (TCP GET)** (e.g., Core Network / MAE)
- **Cisco pipeline / Telegraf pipeline** for **event-driven telemetry** (IP infrastructure)
- **Fluent Bit** reading from **files** (e.g., exports/logs from network/Zabbix-related sources)
- **Fluentd** from **SQL queries** (e.g., openBMP)
- **Fluentd** from **syslog**
- **SNMP polling + SNMP traps** via **“trapo sonda” + MIB/OID translation** (OID → readable fields)
- **Kafka ingestion** (direct writes to Kafka or reading from an existing Kafka source)

initMAX

Problem Enrichment & Apache NiFi

The Output Matters — JSON

- There are **a million ways to collect data** (agents, pipelines, open-source tools, custom scripts, APIs, SNMP, syslog, SQL, Kafka…).
- We don't standardize *how* data is ingested — we standardize **what comes out**.
- **The only requirement:** every source must produce a **JSON event**.

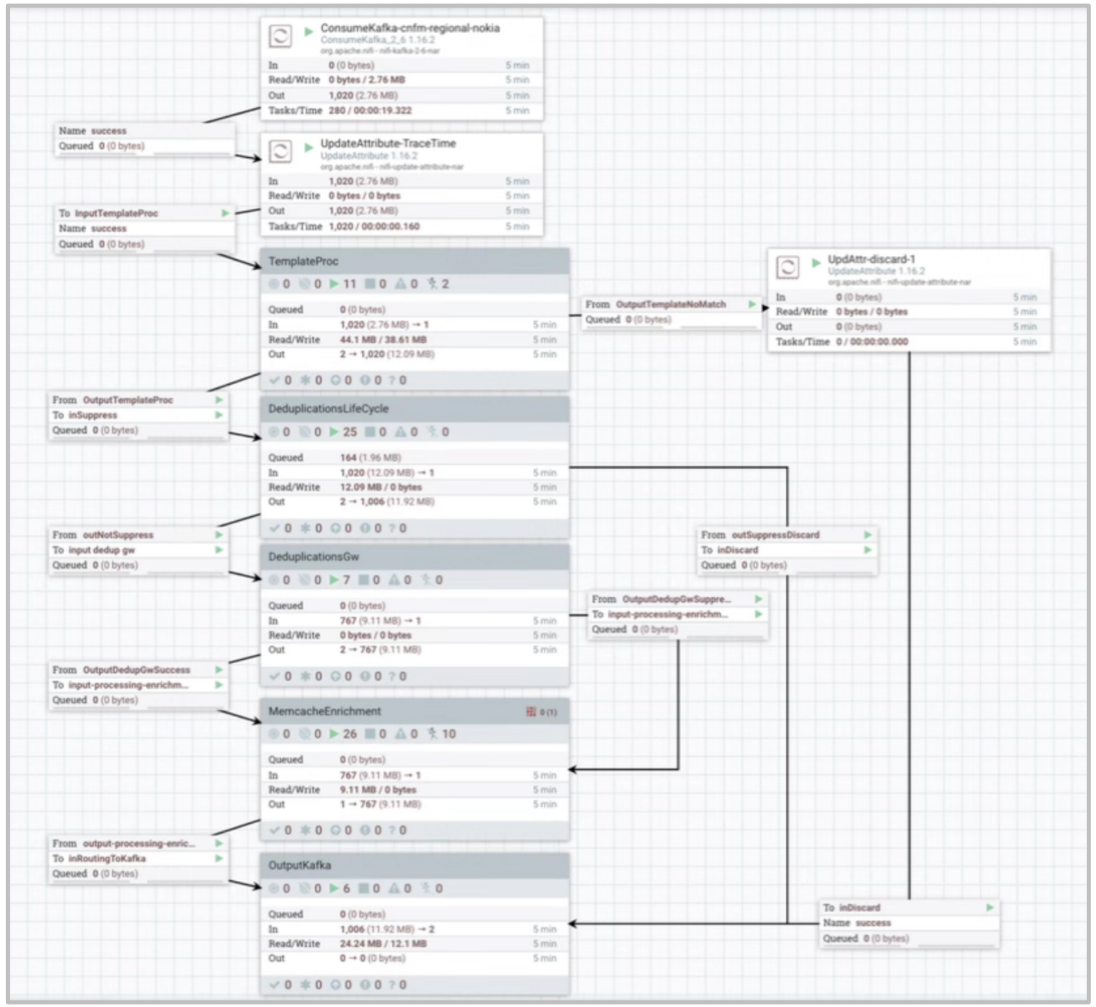


initMAX

Problem Enrichment & Apache NiFi

Problem pre-processing

- **Consume data from Kafka** (source topic)
- **Add a “Kafka ingest timestamp”** (primarily for debugging/traceability)
- **Normalize the data** (unify schema/fields/formats)
- **Deduplicate micro-outage duplicates** (e.g., brief retransmits / short gaps)
- **Deduplicate identical events from multiple sources at the same time window** (e.g., two HA firewalls emitting the same event)
- **Transform/enrich event payload** (add customer-specific data/metadata)
- **Publish the final events to the main Kafka topic** (canonical/output topic)



initMAX

Problem Enrichment & Apache NiFi

Problem post-processing

- **Same concept as preprocessing** (pipeline-style data handling)
- **Check whether the host exists; create it automatically if it doesn't**
- **Call the Talli API** (fetch/enrich required context)
- **Update events via API**: change severity, acknowledge, and add/append an event message
- **Publish data to an external Kafka topic** (for 3rd-party consumers)
- **Send data to Elasticsearch for archiving**

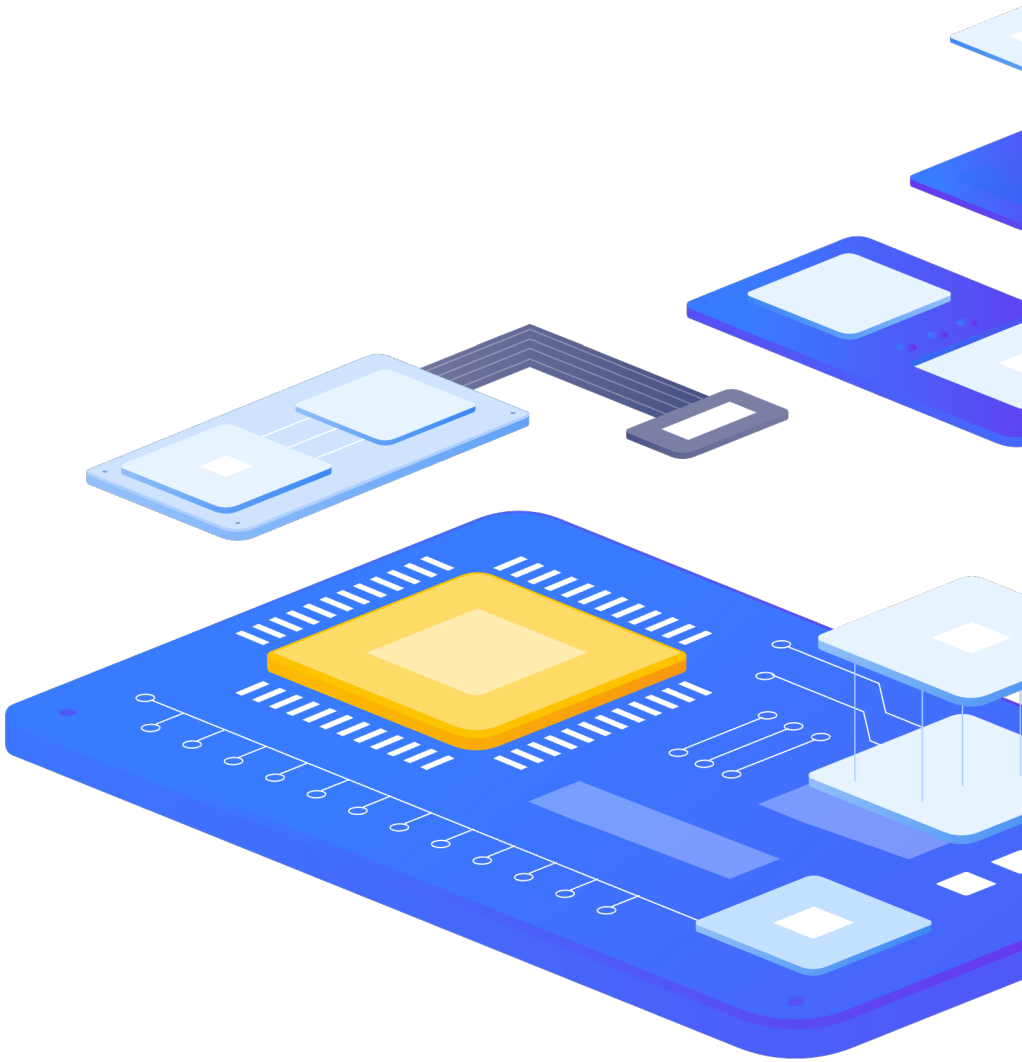
4

Technical Solution in Zabbix

Technical Solution in Zabbix

Zabbix Architecture

- Each host is linked to a template that contains a predefined **Zabbix trapper** item.
- This item is evaluated by triggers that support **multiple event generation**.
- To make each event unique, the event name and additional fields are extracted using **regular expressions**.
- Trigger **tags** are also populated via regular expressions — these tag values carry key enrichment data coming from the upstream processing.
- We use **six similar triggers**, each mapped to a different **severity level**, so the same logic can create problems with the correct priority.

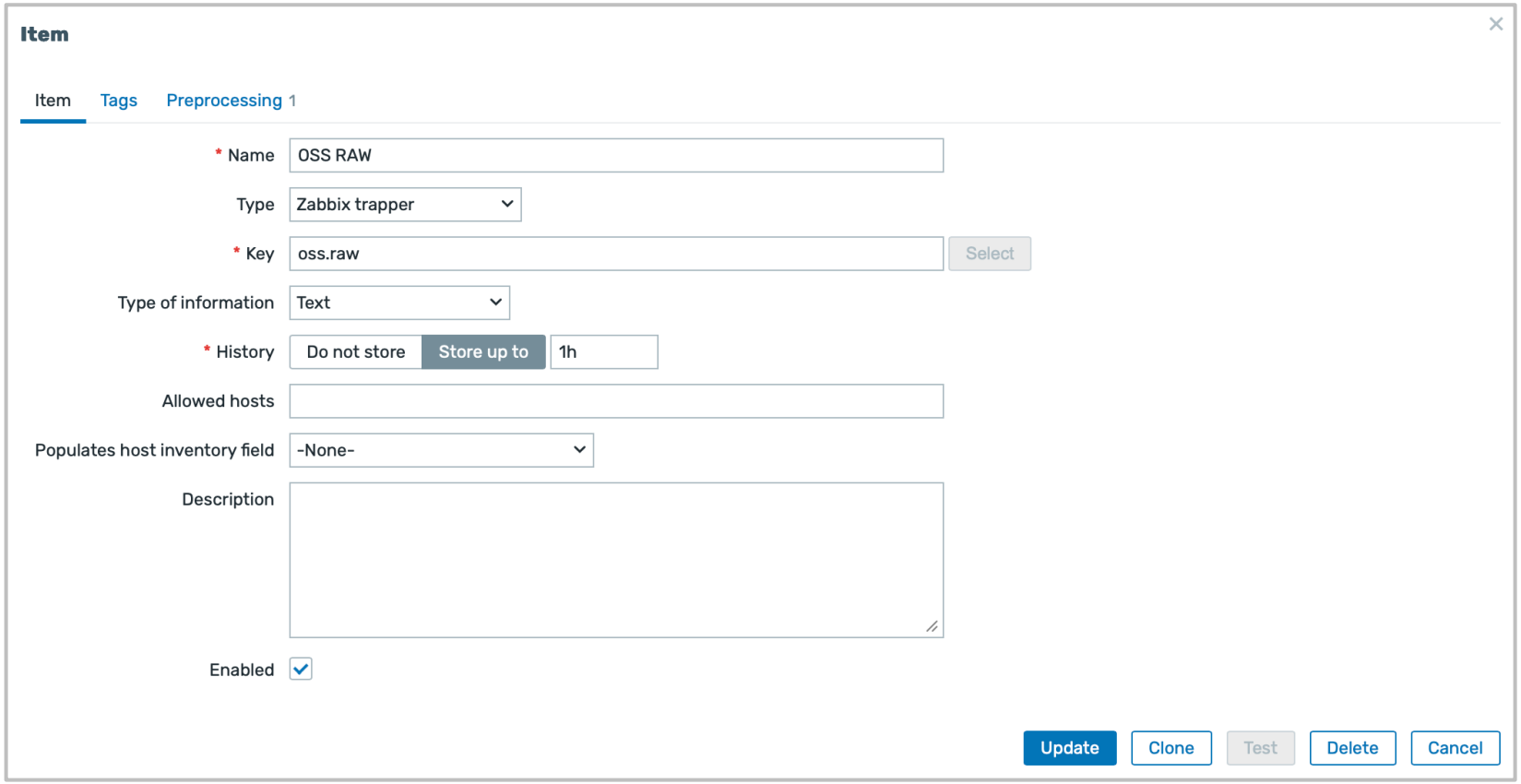


initMAX

Technical Solution in Zabbix

Zabbix Architecture - item

- Preprocessing is intentionally minimal: JSON validation only.



initMAX

Technical Solution in Zabbix

Zabbix Architecture - triggers

- “Not classified” covers cases where enrichment fields are not found during regex parsing.

Severity	Name ▲	Operational data	Expression	Status	Tags
Information	{{ITEM.VALUE1}.regsub("\"zs_summary\":\"(.*?)\",",\1)}		Problem: **find(**/O2-ITEMHOST-OSS-V2/oss.raw,,"like","\"zs_perceivedSeverity\":\"CLEAR\"")<>1 and **find(**/O2-ITEMHOST-OSS-V2/oss.raw,,"like","\"zs_perceivedSeverity\":\"Indeterminate\"")=1 Recovery: **find(**/O2-ITEMHOST-OSS-V2/oss.raw,,"like","\"zs_stateChangeDefinition\":\"CLEAR\"")=1	Enabled	AlarmId: {{ITEM.VALUE1}.regsub("\"zs_alarmId\":\"(.*?)\",",\1)} CaseId: {{ITEM.VALUE1}.regsub("\"zs_caseid\":\"(.*?)\",",\1)} Dom1: {{ITEM.VALUE1}.regsub("\"zs_agent\":\"(.*?)\",",\1)} •••
Disaster	{{ITEM.VALUE1}.regsub("\"zs_summary\":\"(.*?)\",",\1)}		Problem: **find(**/O2-ITEMHOST-OSS-V2/oss.raw,,"like","\"zs_stateChangeDefinition\":\"CLEAR\"")<>1 and **find(**/O2-ITEMHOST-OSS-V2/oss.raw,,"like","\"zs_perceivedSeverity\":\"Critical\"")=1 Recovery: **find(**/O2-ITEMHOST-OSS-V2/oss.raw,,"like","\"zs_stateChangeDefinition\":\"CLEAR\"")=1	Enabled	AlarmId: {{ITEM.VALUE1}.regsub("\"zs_alarmId\":\"(.*?)\",",\1)} CaseId: {{ITEM.VALUE1}.regsub("\"zs_caseid\":\"(.*?)\",",\1)} Dom1: {{ITEM.VALUE1}.regsub("\"zs_agent\":\"(.*?)\",",\1)} •••
High	{{ITEM.VALUE1}.regsub("\"zs_summary\":\"(.*?)\",",\1)}		Problem: **find(**/O2-ITEMHOST-OSS-V2/oss.raw,,"like","\"zs_stateChangeDefinition\":\"CLEAR\"")<>1 and **find(**/O2-ITEMHOST-OSS-V2/oss.raw,,"like","\"zs_perceivedSeverity\":\"Major\"")=1 Recovery: **find(**/O2-ITEMHOST-OSS-V2/oss.raw,,"like","\"zs_stateChangeDefinition\":\"CLEAR\"")=1	Enabled	AlarmId: {{ITEM.VALUE1}.regsub("\"zs_alarmId\":\"(.*?)\",",\1)} CaseId: {{ITEM.VALUE1}.regsub("\"zs_caseid\":\"(.*?)\",",\1)} Dom1: {{ITEM.VALUE1}.regsub("\"zs_agent\":\"(.*?)\",",\1)} •••
Average	{{ITEM.VALUE1}.regsub("\"zs_summary\":\"(.*?)\",",\1)}		Problem: **find(**/O2-ITEMHOST-OSS-V2/oss.raw,,"like","\"zs_stateChangeDefinition\":\"CLEAR\"")<>1 and **find(**/O2-ITEMHOST-OSS-V2/oss.raw,,"like","\"zs_perceivedSeverity\":\"Minor\"")=1 Recovery: **find(**/O2-ITEMHOST-OSS-V2/oss.raw,,"like","\"zs_stateChangeDefinition\":\"CLEAR\"")=1	Enabled	AlarmId: {{ITEM.VALUE1}.regsub("\"zs_alarmId\":\"(.*?)\",",\1)} CaseId: {{ITEM.VALUE1}.regsub("\"zs_caseid\":\"(.*?)\",",\1)} Dom1: {{ITEM.VALUE1}.regsub("\"zs_agent\":\"(.*?)\",",\1)} •••
Warning	{{ITEM.VALUE1}.regsub("\"zs_summary\":\"(.*?)\",",\1)}		Problem: **find(**/O2-ITEMHOST-OSS-V2/oss.raw,,"like","\"zs_stateChangeDefinition\":\"CLEAR\"")<>1 and **find(**/O2-ITEMHOST-OSS-V2/oss.raw,,"like","\"zs_perceivedSeverity\":\"Warning\"")=1 Recovery: **find(**/O2-ITEMHOST-OSS-V2/oss.raw,,"like","\"zs_stateChangeDefinition\":\"CLEAR\"")=1	Enabled	AlarmId: {{ITEM.VALUE1}.regsub("\"zs_alarmId\":\"(.*?)\",",\1)} CaseId: {{ITEM.VALUE1}.regsub("\"zs_caseid\":\"(.*?)\",",\1)} Dom1: {{ITEM.VALUE1}.regsub("\"zs_agent\":\"(.*?)\",",\1)} •••
Not classified	{{ITEM.VALUE1}.regsub("\"zs_summary\":\"(.*?)\",",\1)} FALLBACK!		Problem: **find(**/O2-ITEMHOST-OSS-V2/oss.raw,,"like","\"zs_stateChangeDefinition\":\"CLEAR\"")<>1 and **find(**/O2-ITEMHOST-OSS-V2/oss.raw,,"like","\"zs_perceivedSeverity\":\"Indeterminate\"")=0 and **find(**/O2-ITEMHOST-OSS-V2/oss.raw,,"like","\"zs_perceivedSeverity\":\"Warning\"")=0 and **find(**/O2-ITEMHOST-OSS-V2/oss.raw,,"like","\"zs_perceivedSeverity\":\"Minor\"")=0 and **find(**/O2-ITEMHOST-OSS-V2/oss.raw,,"like","\"zs_perceivedSeverity\":\"Major\"")=0 and **find(**/O2-ITEMHOST-OSS-V2/oss.raw,,"like","\"zs_perceivedSeverity\":\"Critical\"")=0 Recovery: **find(**/O2-ITEMHOST-OSS-V2/oss.raw,,"like","\"zs_stateChangeDefinition\":\"CLEAR\"")=1	Enabled	AlarmId: {{ITEM.VALUE1}.regsub("\"zs_alarmId\":\"(.*?)\",",\1)} CaseId: {{ITEM.VALUE1}.regsub("\"zs_caseid\":\"(.*?)\",",\1)} Dom1: {{ITEM.VALUE1}.regsub("\"zs_agent\":\"(.*?)\",",\1)} •••

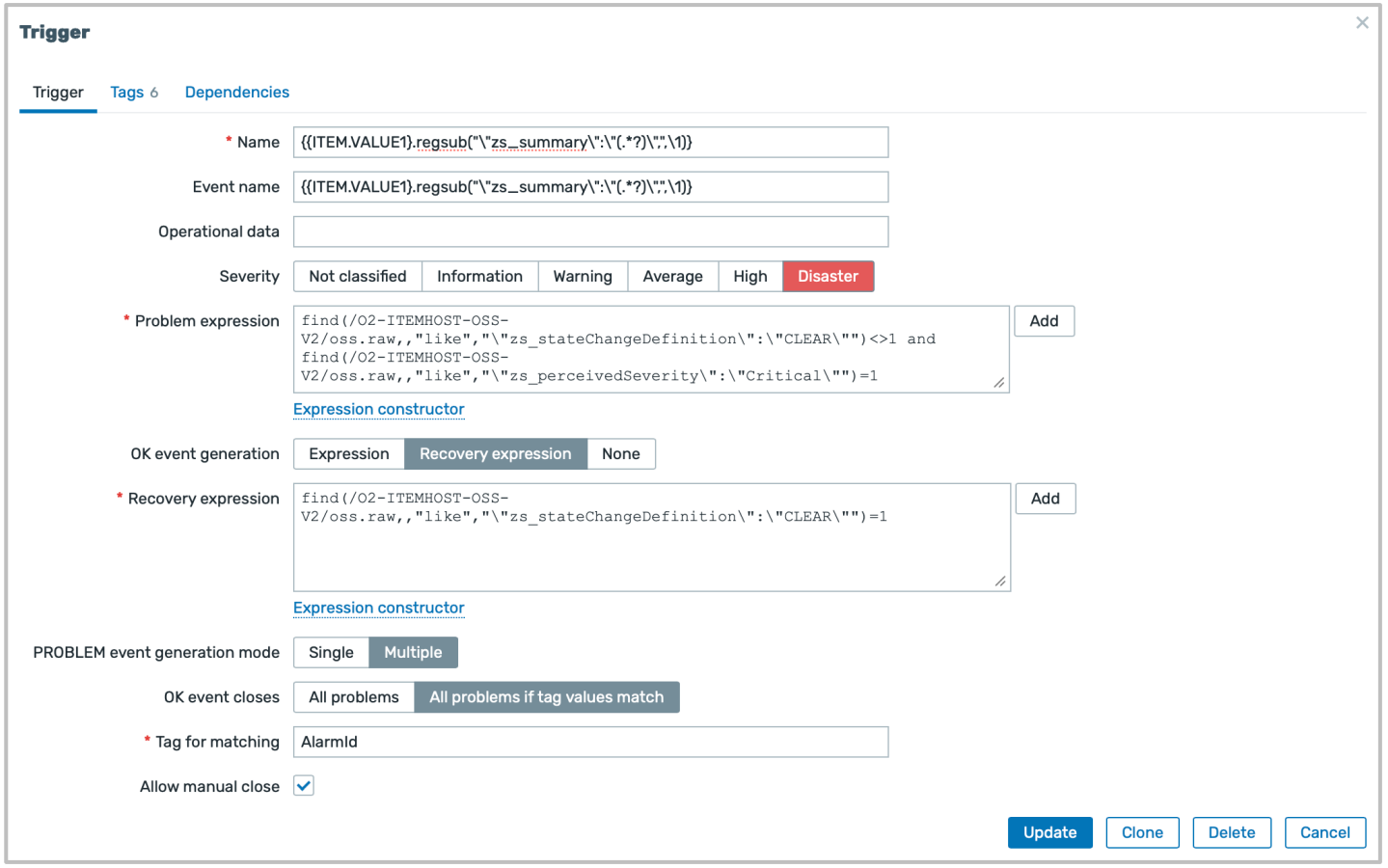
Displaying 6 of 6 found

initMAX

Technical Solution in Zabbix

Zabbix Architecture – trigger detail

- The **event name** is generated using the **regsub()** function, and the trigger runs in **multiple event generation** mode.
- **Problem closing** is performed only when a **matching recovery/close event** arrives with the **same AlarmId** (correlation key).



initMAX

Technical Solution in Zabbix

Zabbix Architecture – trigger detail

- In this example, we add multiple additional **tags** later, which are then used for **event filtering** and routing to a specific area/team.

Trigger

Trigger | Tags 6 | Dependencies

Trigger tags | Inherited and trigger tags

Tags

Name	Value	
AlarmId	{{ITEM.VALUE1}.regsub("\"zs_alarmId\":\"(.*?)\",",\1)}	Remove
CaseId	{{ITEM.VALUE1}.regsub("\"zs_caseid\":\"(.*?)\",",\1)}	Remove
Dom1	{{ITEM.VALUE1}.regsub("\"zs_agent\":\"(.*?)\",",\1)}	Remove
Dom2	{{ITEM.VALUE1}.regsub("\"zs_domain_l2\":\"(.*?)\",",\1)}	Remove
EventType	{{ITEM.VALUE1}.regsub("\"zs_eventType\":\"(.*?)\",",\1)}	Remove
Identifier	{{ITEM.VALUE1}.regsub("\"zs_identifier\":\"(.*?)\",",\1)}	Remove

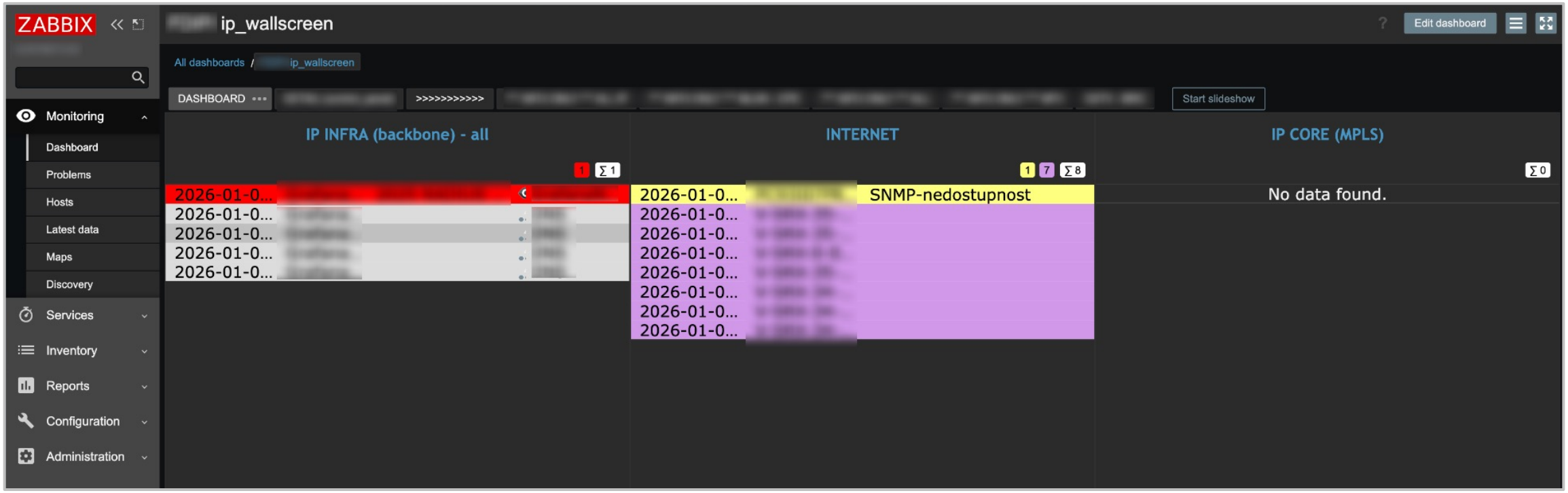
Add

Update | Clone | Delete | Cancel

initMAX

Technical Solution in Zabbix

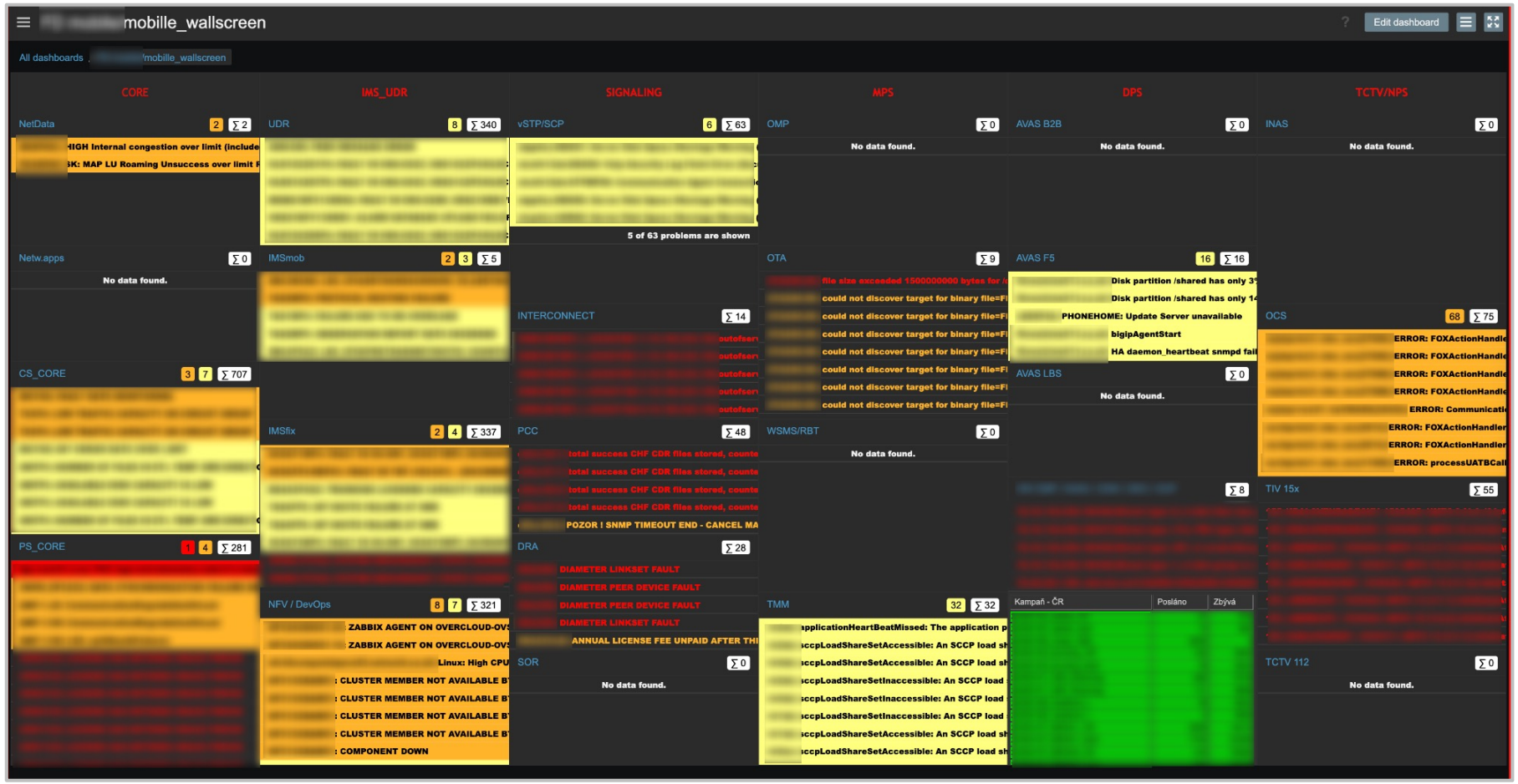
Zabbix Architecture – Custom widget



Migrating Enterprise Monitoring: From IBM Netcool to Zabbix

Technical Solution in Zabbix

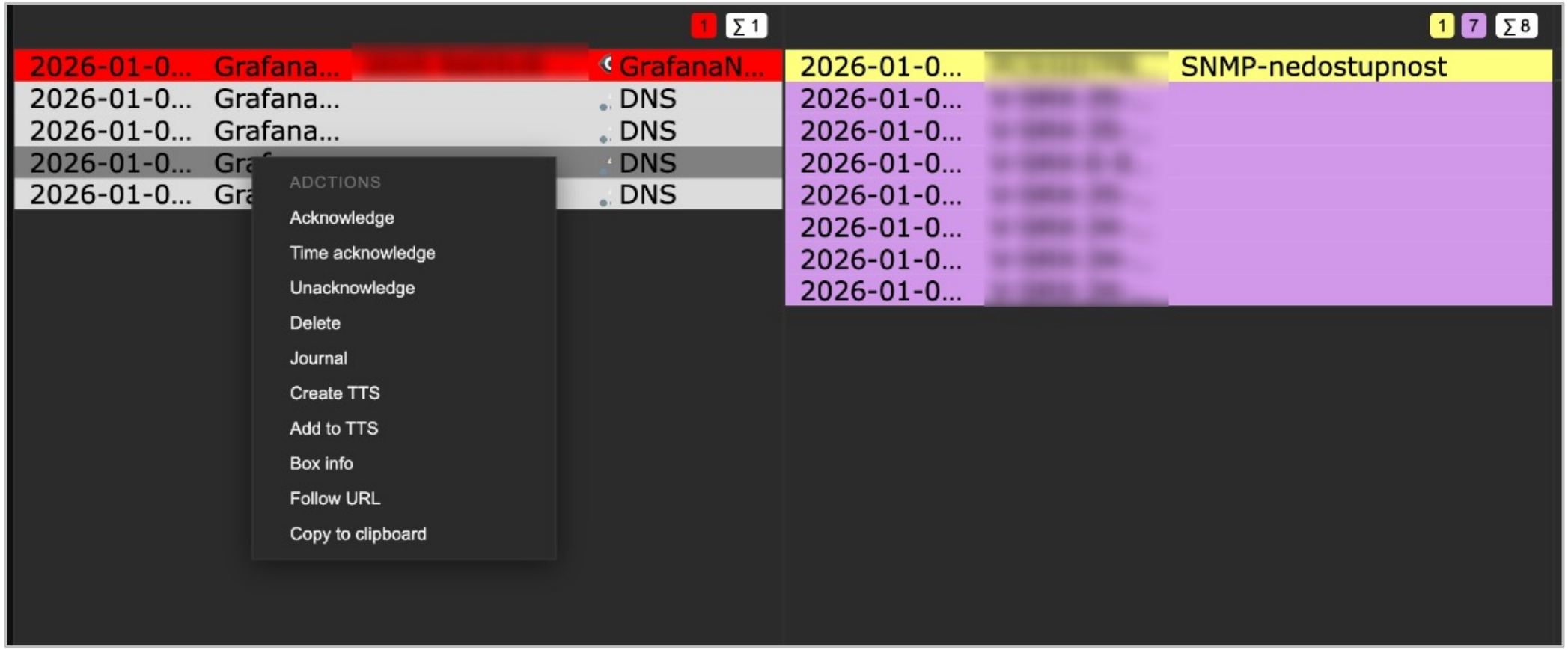
Zabbix Architecture – Custom widget



initMAX

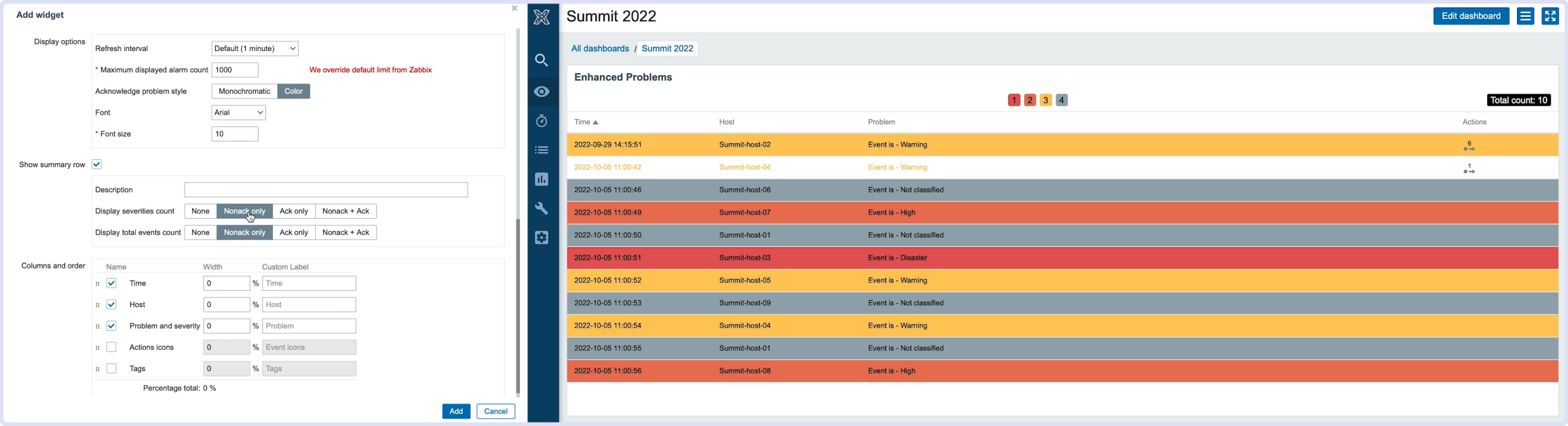
Technical Solution in Zabbix

Zabbix Architecture – Custom scripts



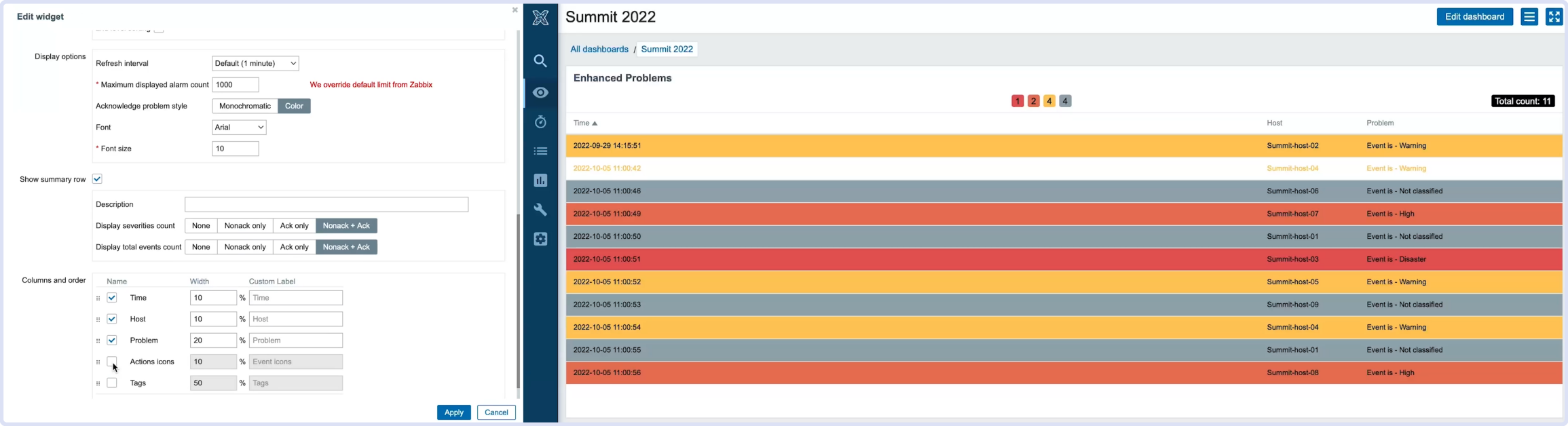
Technical Solution in Zabbix

- Based on built-in Problem widget
- New feature counter events by severities and total events with variations of Ack and Nonack options



Technical Solution in Zabbix

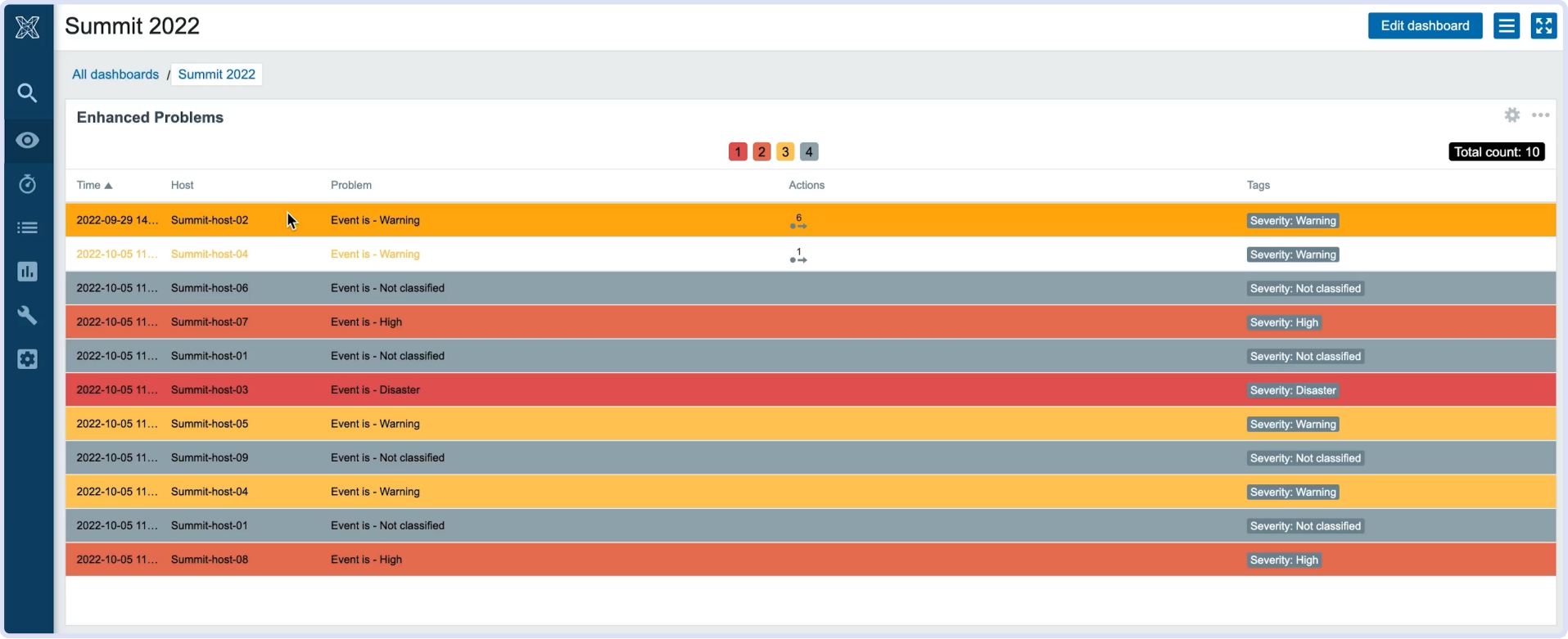
- Based on built-in Problem widget
- New feature choose desired columns you need with custom order, column width and label



initMAX

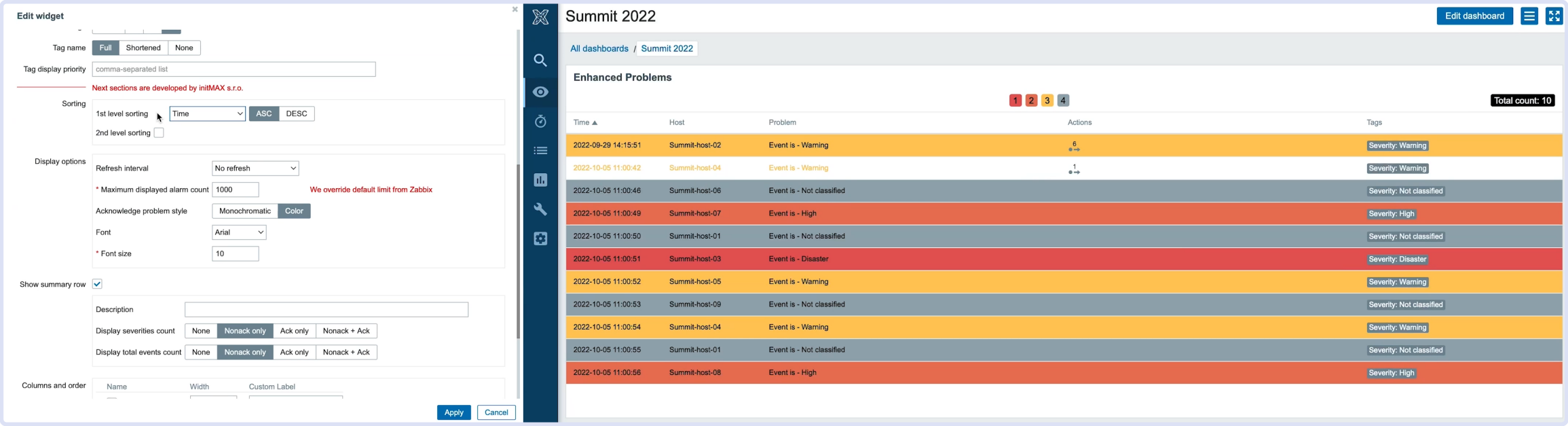
Technical Solution in Zabbix

- Provides modern user comfort - operator can select multiple events by combination of mouse dragging and mouse clicking within pressed CTRL or SHIFT keys
- Context menu is available on right-click by mouse and operator can easily make bulk operations



Technical Solution in Zabbix

- Multi-level sorting of events for every columns
- Ability to change ascending or descending direction for each level of sorting



initMAX

Example

- Every our example is freely available in our GIT repository
- https://git.initmax.com/initMAX-Public
- Source code is commented and provides examples

- Widget Header https://git.initmax.com/initMAX-Public/Zabbix-UI-Modules-Header-Widget
- Zabbix Compact Dashboard https://git.initmax.com/initMAX-Public/Zabbix-UI-Modules-Compact-Dashboard
- Enhanced Problems https://git.initmax.cz/initMAX-Public/Zabbix-UI-Widgets-Enhanced-Problems

5

Customer Satisfaction & Outcomes

initMAX

Customer Satisfaction & Outcomes

Customer Satisfaction & Outcomes

- Small internal team fully operates the umbrella monitoring stack.
- Full in-house control: config changes, outages, upgrades — no vendor lock-in.
- NiFi logic enables new data sources/device onboarding with zero downtime.
- Upgrade in progress: **Zabbix 6.0 → 7.0**, incl. **history.push** + PostgreSQL improvements.
- Multiple department Zabbix instances feed the umbrella Zabbix (central monitoring hub).
- **Cost savings**: license-free platform, lower ops overhead, no unnecessary vendor spend.
- As initMAX: support, upgrades, performance tuning, plus training and custom integrations/widgets.
- **~80% lower TCO:** Zabbix + initMAX support vs. previous licenses + support.

Customer Satisfaction & Outcomes

Customer Satisfaction & Outcomes — Case Study

- We're in the final stage of publishing an **O2 video case study** with more project details — it will be available soon on our **LinkedIn** and **website.**
- If you'd like more webinars and updates, follow us on **LinkedIn.**

https://www.linkedin.com/company/initmax/

initMAX

Contact us:

Phone:	+420 800 244 442
Web:	https://www.initmax.com
Email:	tomas.hermanek@initmax.com
LinkedIn:	https://www.linkedin.com/company/initmax
Twitter:	https://twitter.com/initmax
Tomáš Heřmánek:	+420 732 447 184