

Putting Zabbix Data to Work

-

AI-Based Anomaly Detection and Advanced Reporting



IntelliTrend GmbH



Contact: Wolfgang Alper

 www.intellitrend.de



ZABBIX
PREMIUM PARTNER

wolfgang.alper@intellitrend.de

Putting Zabbix Data to Work

Zabbix captures data that shows how your systems behave over time. That knowledge can feed external applications and help detect unexpected behavior.

- **Turn data into insight** - practical reports for technical teams, managers, and managed service providers.
- **Connect systems** - structured, machine-readable reports for automated processing and integration.
- **Recognize unusual behavior** - specialized AI models that learn from historical data and bring anomaly scores back into Zabbix.

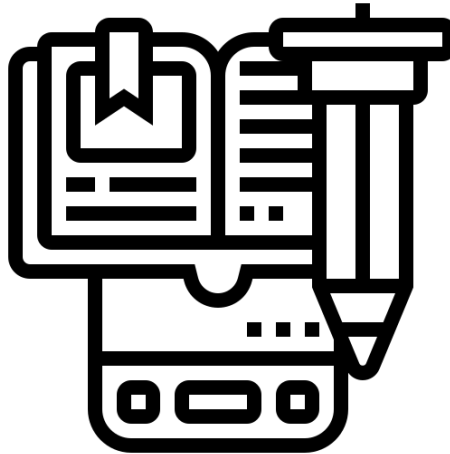


The following examples are based on IntelliTrend Prisma.
A free version is available on <https://www.intellitrend.de>

Putting Zabbix Data to Work

Turn data into insight

Technical and management reports



Reports are based on widgets

Problems, availability, response times, alerting, hosts and metrics, governance and your own context



Problems

what is broken, and for how long

- Open problems
- Problem history
- Problem overview
- Long problems
- Trigger state
- Busy triggers



Availability

SLA compliance per service

- SLA
- SLA overview

99.95 %
Webshop, SLO 99.9 %



Response times

how fast teams react

- MTTR
- MTTI

42 min · 7 min
MTTR · MTTI, db-01 high CPU



Alerting

did notifications work

- Alert log
- Failed alerts
- Redundant alerts

12 failed
notifications this month



Hosts & metrics

values, graphs, inventory

- Host numeric items
- Host text items
- Host graph
- Host inventory



Governance

who changed what

- Audit log

3 changes
to templates, by 2 users



Context

your own text and notes

- Markdown

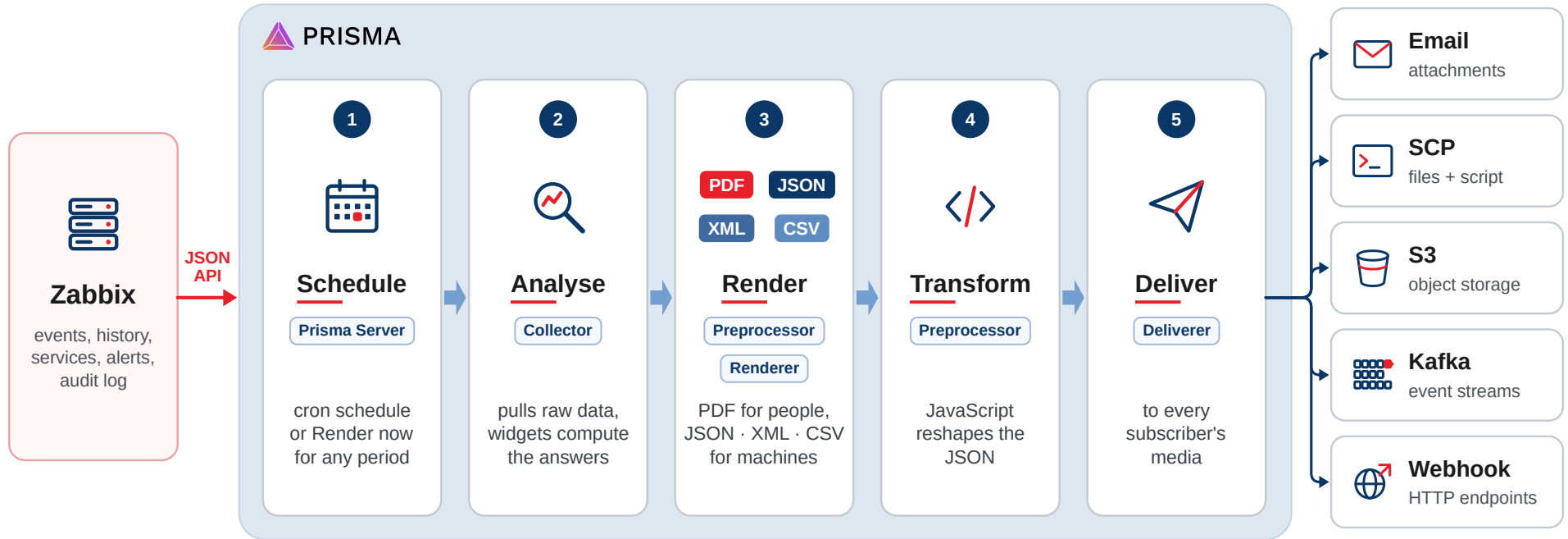
Summary
your own words, in Markdown

19
widgets, freely
combined in one report

every widget declares the
render types it supports

The path of a report

Five stages of a report between the Zabbix API and the subscribers' media



- Scheduled or on demand

- Every format is rendered once per run

- Each subscriber picks formats and media

Time to interaction and recovery: MTTI and MTTR

Three problems of one trigger: from raw events to two mean times and a ranking

1 RAW EVENTS FROM ZABBIX

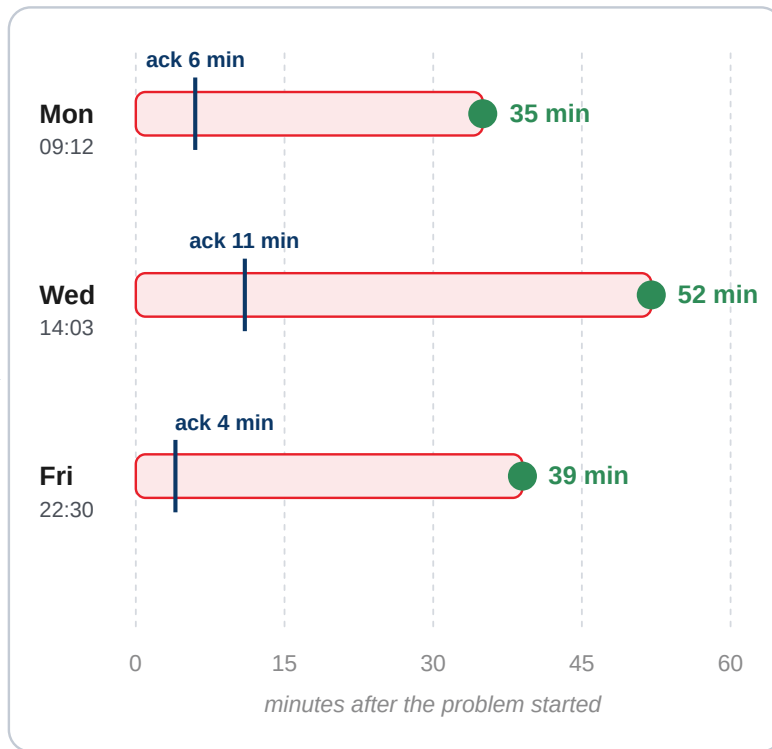
db-01: high CPU load

Mon 09:12 PROBLEM
Mon 09:18 ACK
Mon 09:47 OK
Wed 14:03 PROBLEM
Wed 14:14 ACK
Wed 14:55 OK
Fri 22:30 PROBLEM
Fri 22:34 ACK
Fri 23:09 OK

Zabbix keeps events; the widget pairs them into problems

- problem open
- acknowledged (ACK)
- recovered (OK)

2 ONE BAR PER PROBLEM



3 RESULT

MTTR mean time to recover

42 min

$(35 + 52 + 39) / 3$

MTTI mean time to interact

7 min

$(6 + 11 + 4) / 3$

MTTR widget · host group ERP

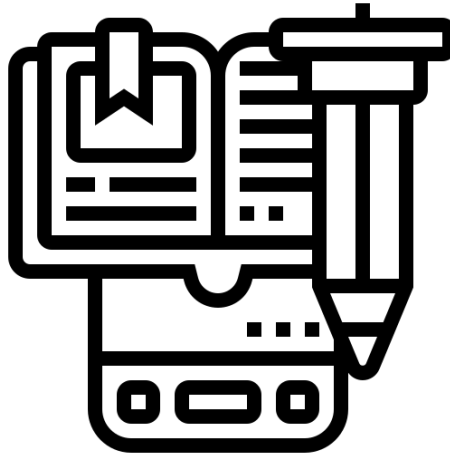
erp-app: queue length **3 h 10 min**

db-01: high CPU load **42 min**

web-02: HTTP 5xx **18 min**

Putting Zabbix Data to Work

Turn data into insight Examples



IT Operations Review - Q3 2026

Quarterly report · Q3 2026 · for the executive board

- The IT department's quarterly review for the **executive board**.
- **4 of 5 service levels met.** ERP missed its 99.9% target (99.86%) because of the storage incident in September.
- In July an **expired certificate** took the customer portal offline for 70 minutes; certificates now renew automatically.
- Customer portal orders grew **22%** over the quarter.
- The report ends with the open issues and known risks, and what IT invests in next quarter.



Quarterly report
July 1st, 2026, 00:00 — October 1st, 2026, 00:00

IT Operations Review — Q3 2026

Quarterly review of IT service availability and operational performance for the Orvella Group executive board.
Prepared by IT Infrastructure & Operations.

Table of contents

Widget	Page
Executive summary	2
Service levels at a glance	3
Business service availability	4

Executive summary

Markdown widget · IT Operations Review - Q3 2026



Executive summary

Markdown widget

Q3 at a glance

- **4 of 5 service level agreements met.** ERP missed its 99.9% target with **99.86%** after the storage incident on 23 September (INC-2291).
- **19 high-severity problems**, restored after **1h 51m** on average.
- **Customer portal orders up 22%** over the quarter, with no capacity problems.

What went wrong

INC-2291, 23 September: a storage firmware update slowed the ERP database for 3.5 hours at night. The vendor has acknowledged the defect; the corrected firmware will be installed together with their engineer on 6 October.

Customer portal certificate, 14 July: an expired TLS certificate took the portal offline for 70 minutes. The warning had been open for two weeks. **Certificates now renew automatically.**

Investments for Q4

1. **Second SMS provider** for on-call paging (approved)
2. **Storage firmware 7.5.1** with vendor support on site
3. **Monitoring hygiene:** 40% fewer notifications, so the on-call team can focus on real problems

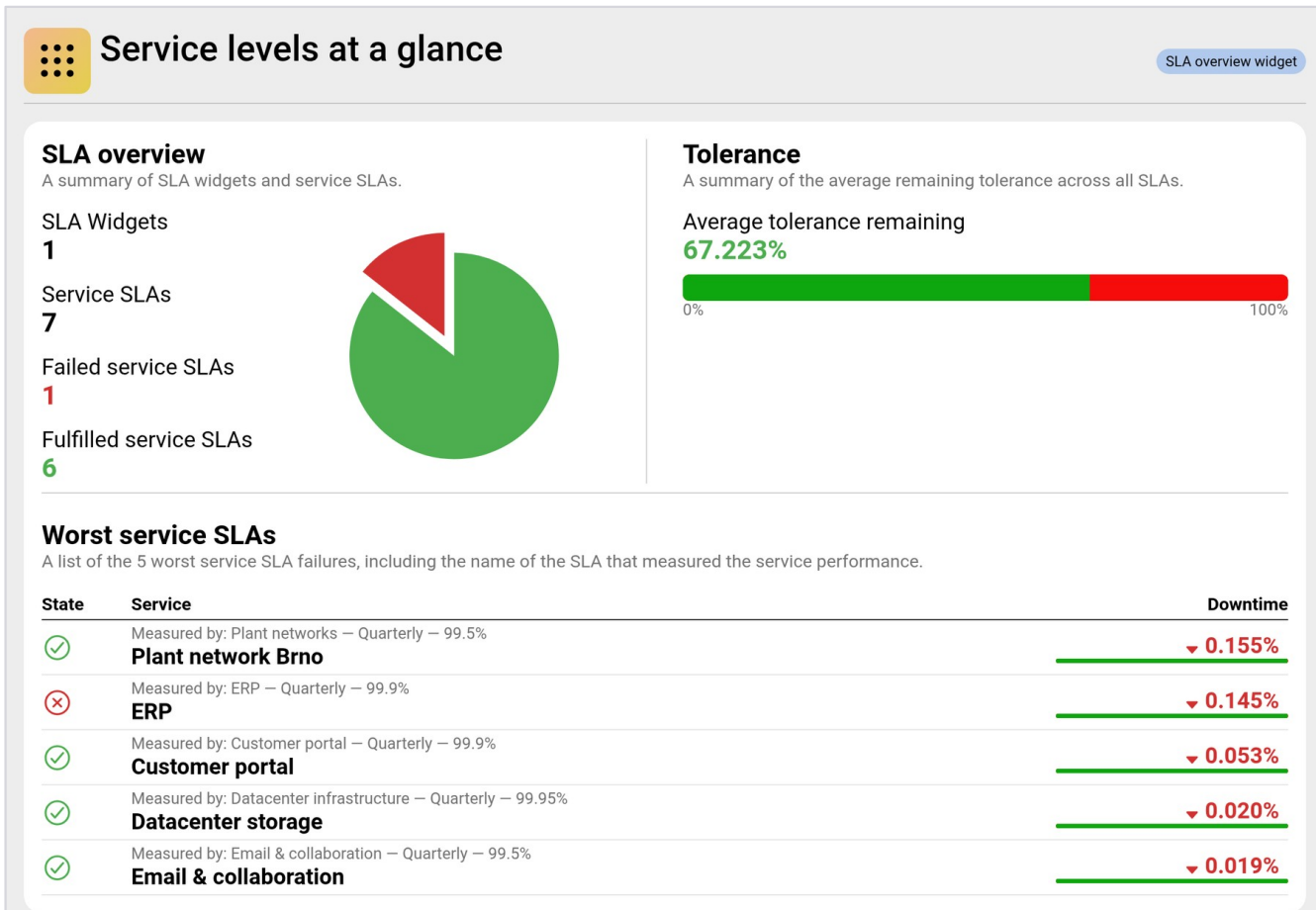
- Executive summary in plain language: results, what went wrong and the investments for next quarter.



ZABBIX
PREMIUM PARTNER

Service levels at a glance

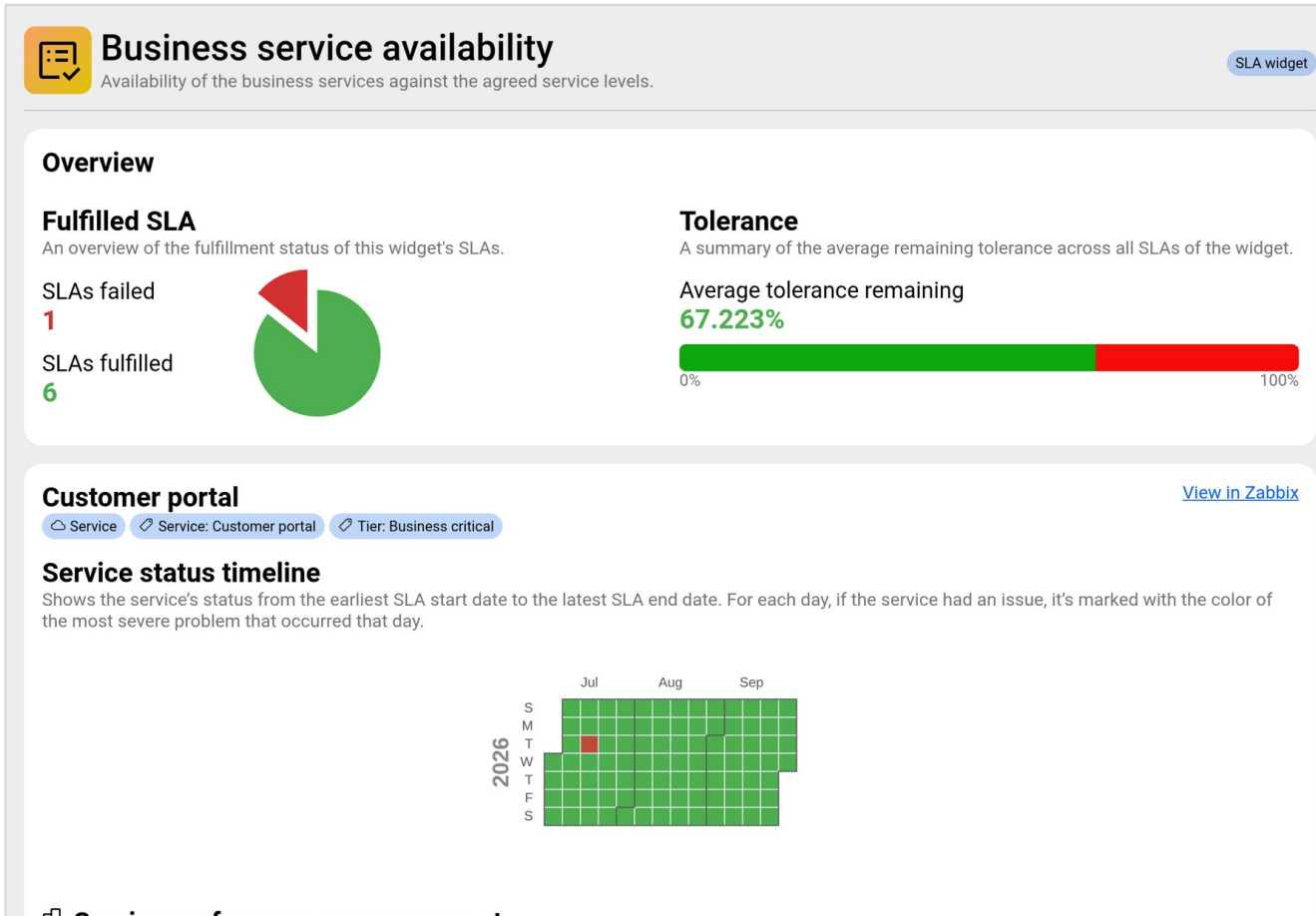
SLA overview widget · IT Operations Review - Q3 2026



- **SLA overview:** all SLA widgets of the report at a glance.
- One failed SLA and the average remaining tolerance; the worst services are listed with their downtime.

Business service availability

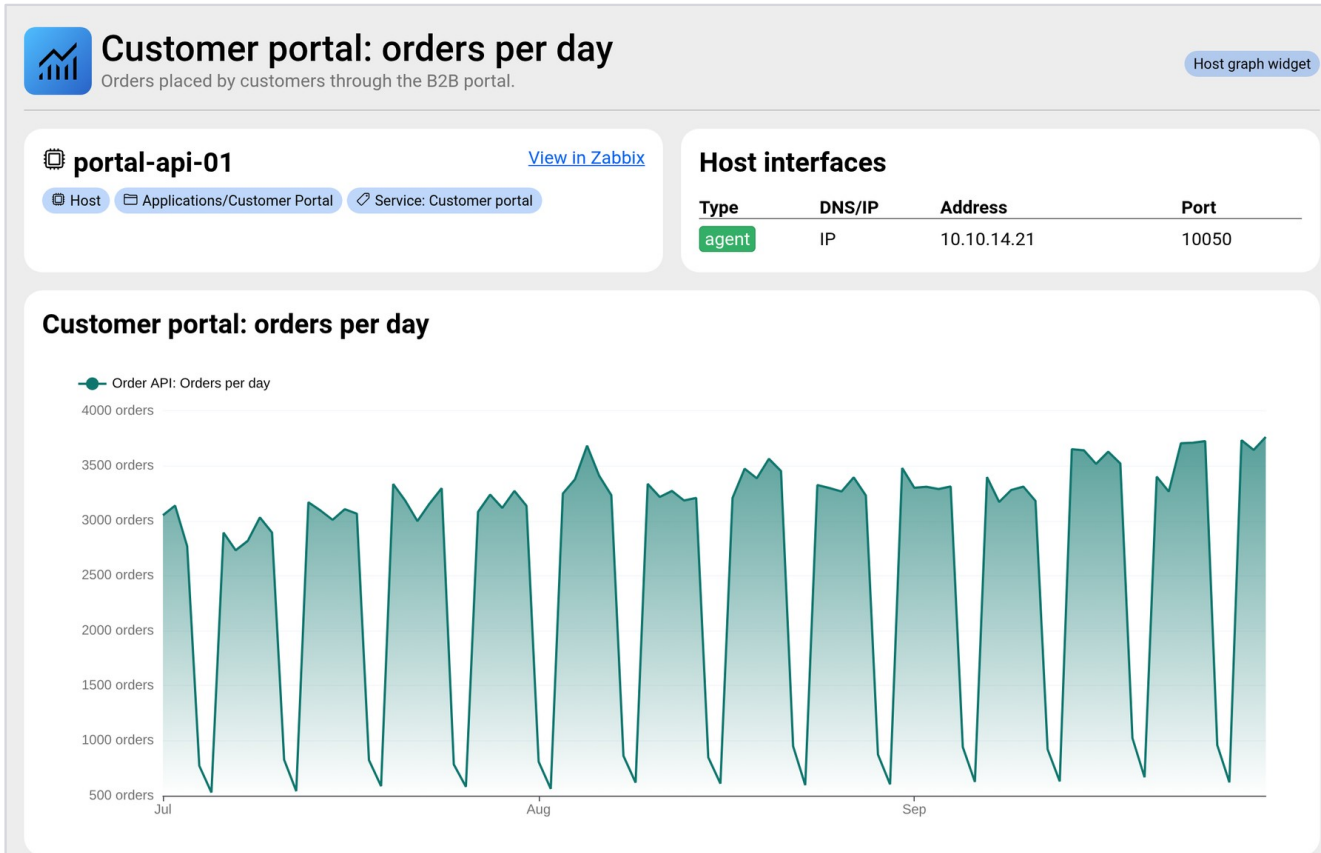
SLA widget · IT Operations Review - Q3 2026



- **SLA widget:** availability per business service against the agreed target.
- The calendar marks the days with outages; the bar shows how much of the tolerance was used.
- ERP missed its target, the customer portal stayed within it despite the July outage.

Customer portal: orders per day

Host graphs widget · IT Operations Review - Q3 2026



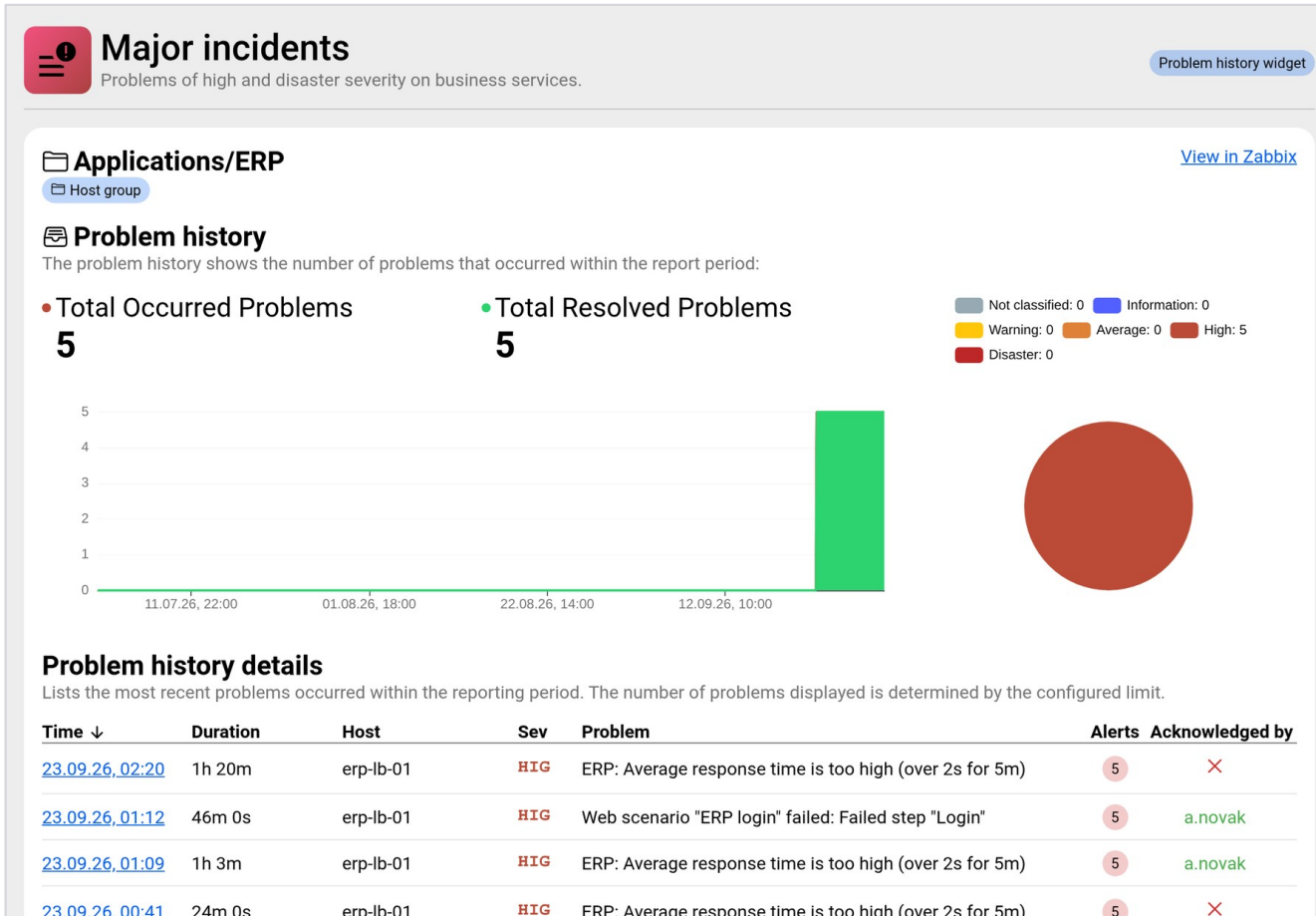
- A business metric on a host graph: orders per day through the customer portal.
- A clear weekly rhythm and **22% growth** over the quarter.



ZABBIX
PREMIUM PARTNER

Major incidents

Problem history widget · IT Operations Review - Q3 2026



- Only the major incidents: high and disaster problems on the business services during the quarter.

Mean time to interaction

Mean time to interaction (MTTI) widget · IT Operations Review - Q3 2026

 **Mean time to interaction**

Mean time to interaction widget

 **Datacenter/Storage**

[View in Zabbix](#)

 Host group

Mean time to interaction (MTTI)
A list of the triggers of a host group that have exceeded a specified minimum number of activations, sorted by MTTI. Lower MTTI times are better.

Min. activations: 1

Limit: 8

Host	Sev	Trigger name	MTTI ↓	Activations	Interaction	Manually closed	Acknowledged
san-a	HIG	SAN-A: Controller B: Write latency is too high (over 20 ms for 5m) 	33m 0s	4	50.00%	0.00%	50.00%
san-a	INF	SAN-A: Controller B: Firmware version has changed 	25m 0s	2	100.00%	100.00%	100.00%
san-a	AVG	SAN-A: Controller B: Read latency is too high (over 20 ms for 5m) 	6m 0s	2	50.00%	0.00%	50.00%
san-a	DIS	SAN-A: Controller B is in degraded state 	2m 0s	1	100.00%	0.00%	100.00%

NCL – Not classified

INF – Information

WRN – Warning

AVG – Average

HIG – High

DIS – Disaster

 **Datacenter/Database**

[View in Zabbix](#)

 Host group

Mean time to interaction (MTTI)
A list of the triggers of a host group that have exceeded a specified minimum number of activations, sorted by MTTI. Lower MTTI times are better.

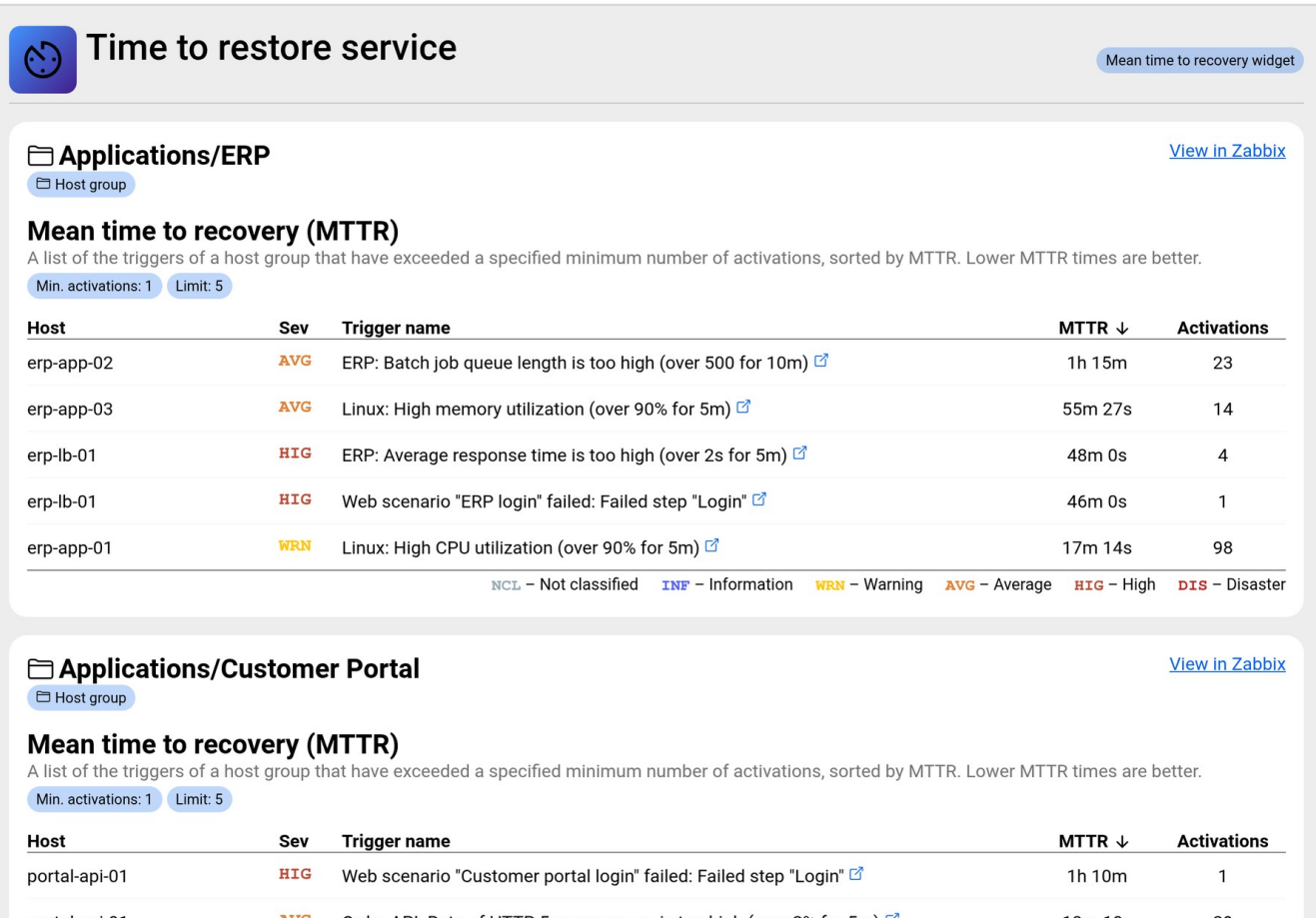
Min. activations: 1

Limit: 8

- **Mean time to interaction:** how quickly someone reacted to each trigger's problems.
- The degraded controller was acknowledged after 2 minutes, the write latency problems after 33 minutes on average.
- Interaction, manual-close and acknowledgement rates per trigger.

Time to restore service

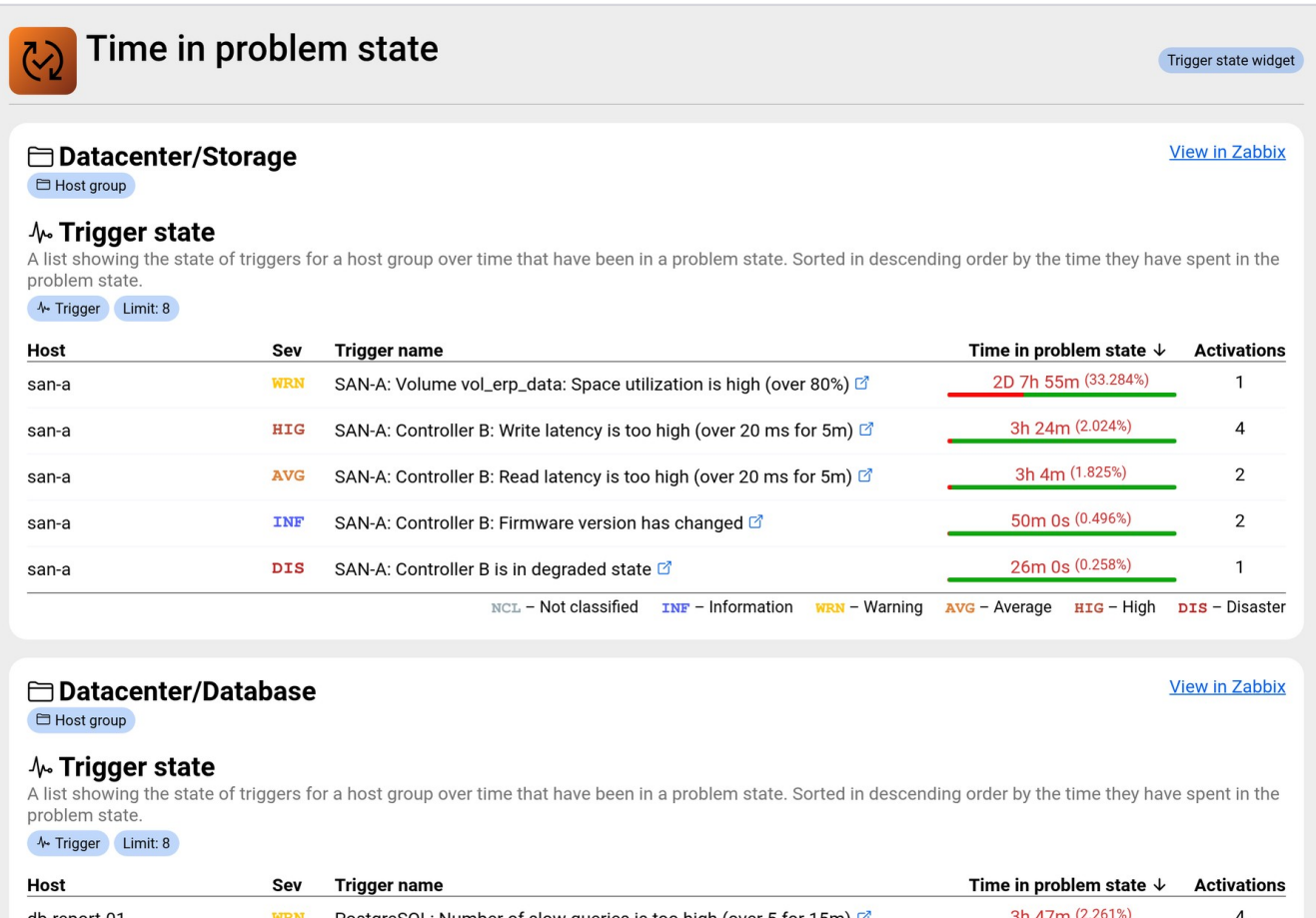
Mean time to recovery (MTTR) widget · IT Operations Review - Q3 2026



- Time to restore service per trigger for the most important services.
- The expired certificate took 1h 10m to fix.

Time in problem state

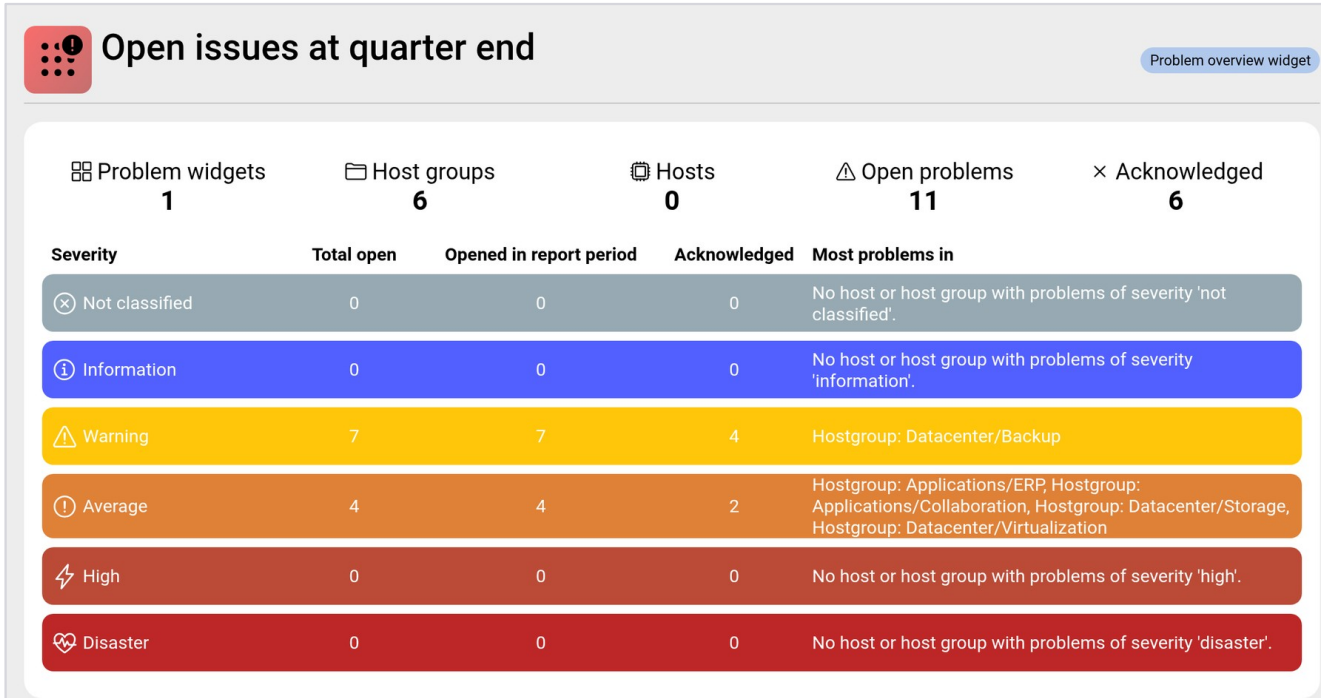
Trigger state widget · IT Operations Review - Q3 2026



- **Trigger state:** the share of the period each trigger spent in problem state.
- The ERP volume warning, open since Friday, tops the storage group with 33%.
- Overlapping problems of the same trigger are counted once.

Open issues at quarter end

Problem overview widget · IT Operations Review - Q3 2026



- Open issues at the end of the quarter by severity: no high or disaster problems.

Known risks

Long problems widget · IT Operations Review - Q3 2026



Known risks

Issues open for more than a week, with the owner's latest note.

Long problem widget

Datacenter/Backup

Host group

[View in Zabbix](#)

⚠ Long problems

A list of problems of a host group that have exceeded a specified duration.

Min. duration: 7D 0h 0m Limit: 5

Time	Duration ↓	Host	Sev	Problem	Acknowledged
09.09.26,04:12	21D 22h 48m	backup-01	WRN	Backup repository: Free space is low (less than 15%)	×
19.09.26,02:55	12D 0h 5m	backup-01	WRN	Backup job "Mailbox-Daily" finished with warnings	t.brandt
19.09.26, 07:55 – t.brandt acknowledged and added message: "Two locked PST files, users informed."					
20.09.26,03:10	10D 23h 50m	backup-01	WRN	Backup job "Fileserver-Weekly" finished with warnings	×

NCL – Not classified INF – Information WRN – Warning AVG – Average HIG – High DIS – Disaster

Datacenter/Database

Host group

[View in Zabbix](#)

⚠ Long problems

A list of problems of a host group that have exceeded a specified duration.

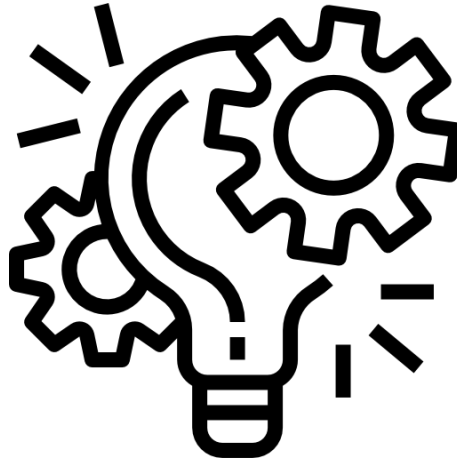
Min. duration: 7D 0h 0m Limit: 5

Time	Duration ↓	Host	Sev	Problem	Acknowledged
17.09.26,13:40	13D 13h 20m	db-report-01	WRN	Linux: /var/lib/postgresql: Disk space is low (used > 80%)	k.yilmaz
17.09.26, 16:40 – k.yilmaz acknowledged and added message: "Archiving old partitions next sprint."					

- **Known risks:** issues open for more than a week, such as backup capacity, a database disk and a hypervisor health warning.

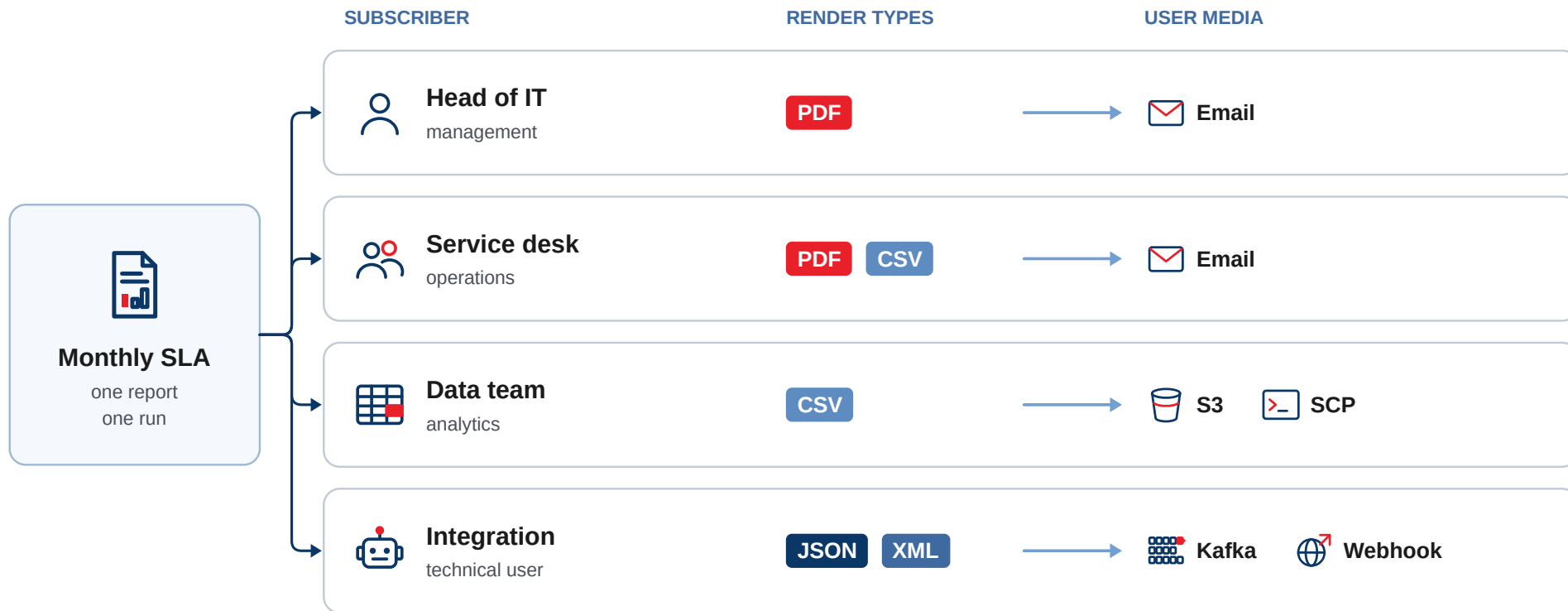
Putting Zabbix Data to Work

Reports under the hood



One report, different render types, many subscribers

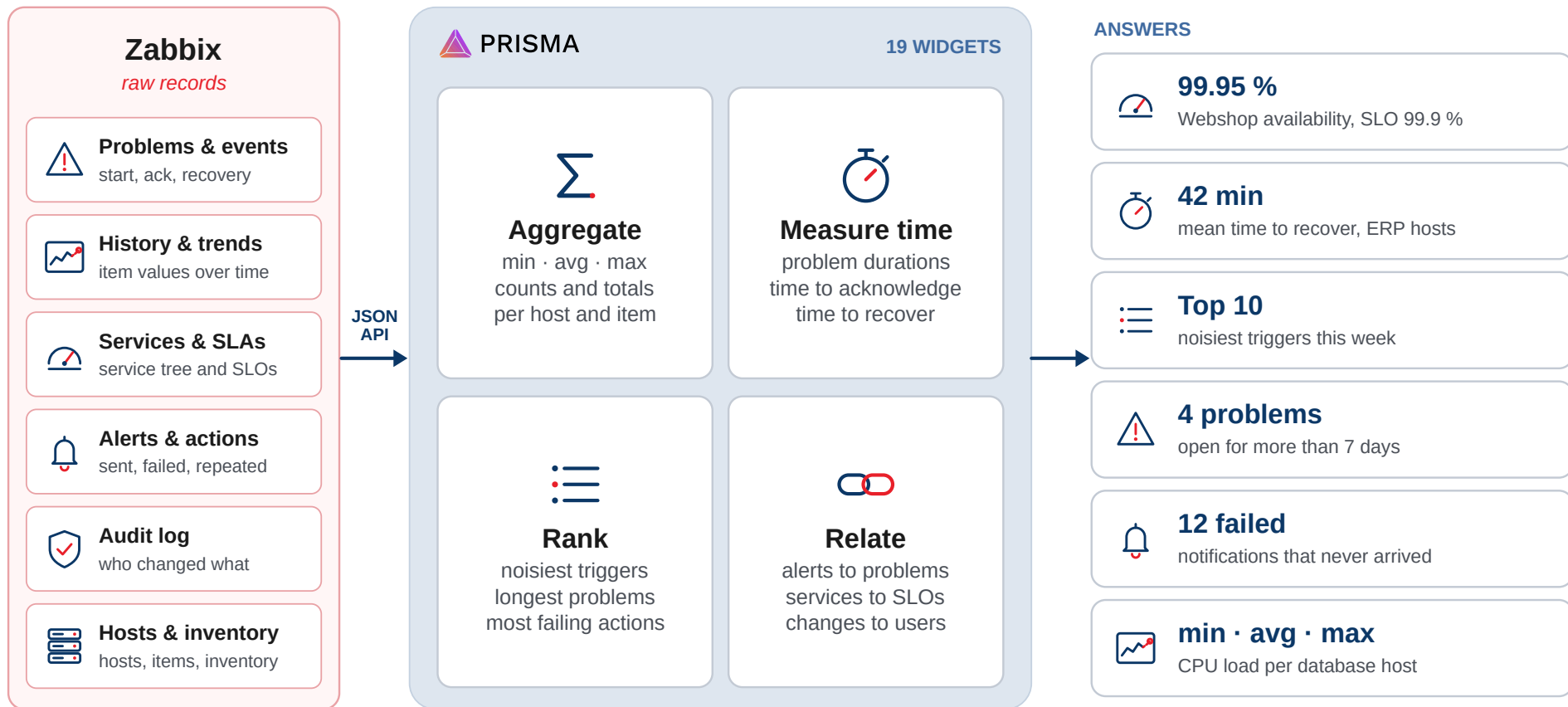
Each subscriber picks their own render types and user media



A widget that does not support a render type is simply left out of that format

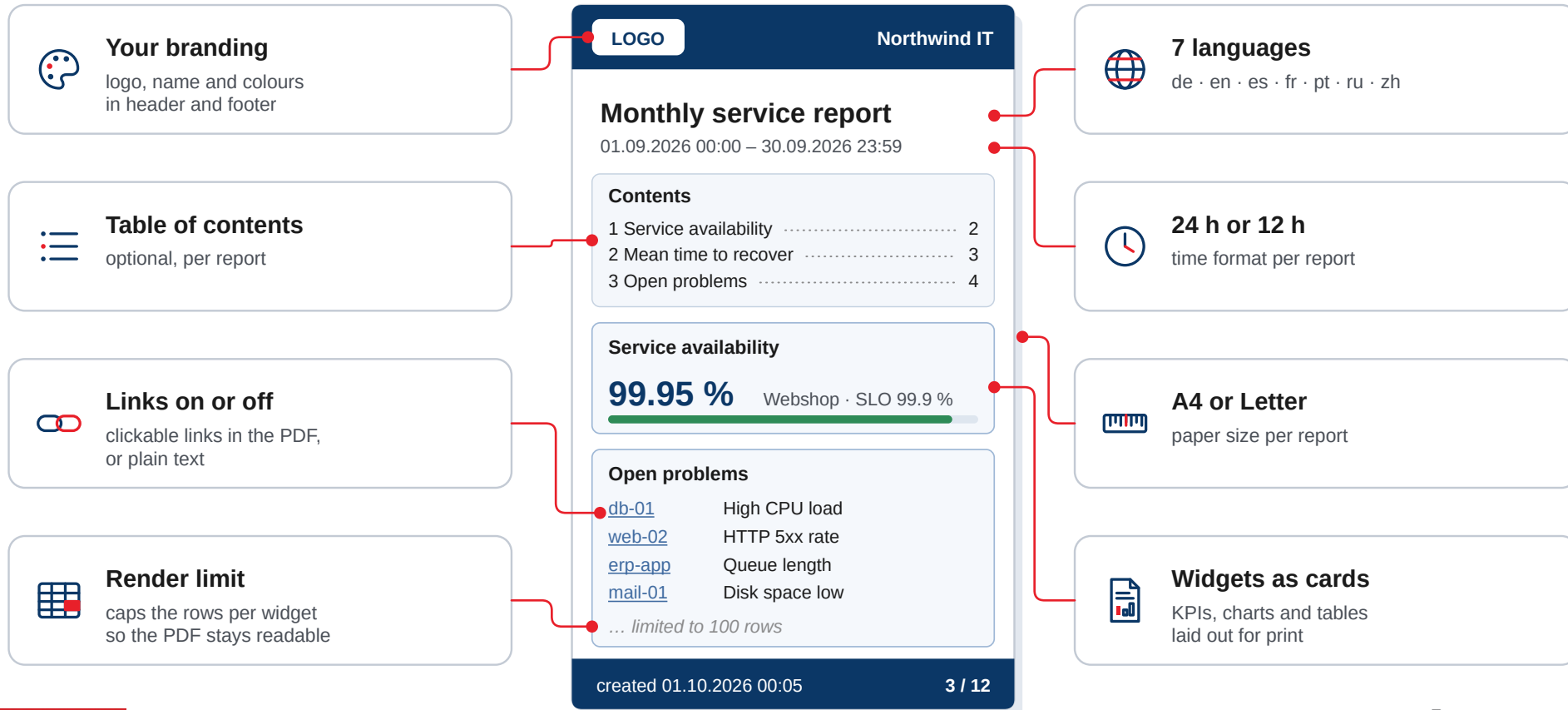
From raw records to answers

Widgets turn raw Zabbix records into the numbers people actually ask for



PDF: made for people

Branded, paginated and readable, with the options a printed report needs



JSON: the complete report

Report metadata plus one object per widget, the most complete format and the input for transformers

Monthly service report.json

```
{
  "name": "Monthly service report",
  "description": "Webshop and ERP",
  "zabbixServerName": "zbx-prod",
  "zabbixServerVersion": "7.0.12",
  "periodType": "monthly",
  "widgets": [
    {
      "name": "sla-widget",
      "displayName": "Service availability",
      "error": "",
      "services": [
        {
          "name": "Webshop",
          "slis": [
            {
              "goodSLA": 99.9,
              "totalUptime": 99.95
            }
          ]
        }
      ]
    },
    {
      "name": "mttr-widget",
      ...
    }
  ],
  "start": 1788220800,
  "end": 1790812800,
  "createdAt": 1790813105
}
```

Report metadata

which report, which Zabbix

widgets[]

one object per widget:
header, error, its data

Period and run

unix timestamps

WHERE THE JSON GOES



Transformers

reshape it in
JavaScript first



Kafka

the message value,
as is



Webhook

inside the envelope
as data.json



Files

email attachment,
SCP, S3

XML: the same report as elements

For XSLT, integration middleware and systems that only accept XML

Monthly service report.xml (abridged)

```
<Report>
  <Name>Monthly service report</Name>
  <ZabbixServerName>zbx-prod</ZabbixServerName>
  <PeriodType>monthly</PeriodType>
  <Widgets>
    <SLAWidget>
      <DisplayName>Service availability</DisplayName>
      <Services>
        <Service>
          <Name>Webshop</Name>
          <Slis>
            <GoodSLA>99.9</GoodSLA>
            <TotalUptime>99.95</TotalUptime>
          </Slis>
        </Service>
      </Services>
    </SLAWidget>
    ...
  </Widgets>
  <Start>1788220800</Start>
</Report>
```

SAME CONTENT AS THE JSON, AS ELEMENTS



XSLT

turn it into HTML, another XML dialect or a mail body



Integration middleware

route and map it with the XML tools you already run



Legacy imports

systems that only accept XML files



Widgets without XML support are left out of the XML file (Alert log); the others stay.

CSV: tables for spreadsheets and BI

One table per widget type in one ZIP, with an optional decimal comma



Monthly service report.csv.zip

one ZIP per report run



sla.csv



mttr_triggers.csv



host_problems.csv



numeric_item_history.csv



numeric_item_aggregates.csv



alerts.csv

One table per widget type: all SLA widgets of the report share sla.csv; numeric items add an aggregates table

sla.csv

header row + one row per SLI

ServiceName	GoodSLA	TotalUptime	Failed	...
Webshop	99,9	99,95	false	...
Payments	99,9	99,82	true	...
ERP	99,5	99,71	false	...
Mail	99	99,98	false	...
Storage	99,5	99,63	false	...
VPN	99	99,4	false	...
Backup	99	99,9	false	...



Decimal comma

99,95 instead of 99.95, for spreadsheet locales that expect it

OPENS IN



Spreadsheets

open and filter



BI tools

load as tables

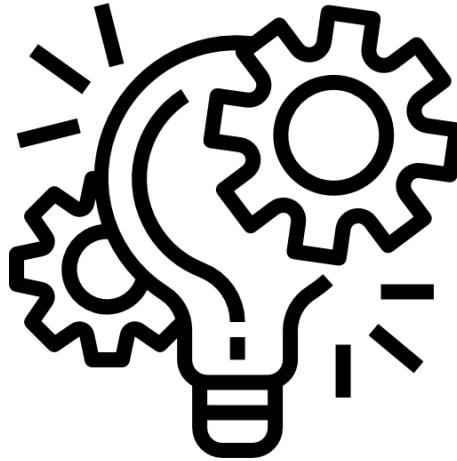


Databases

import, e.g. by an SCP post-script

Putting Zabbix Data to Work

Reports under the hood Javascript-Transformer



A transformer is one JavaScript function

It receives the report as JSON and returns whatever JSON the receiver needs

Report JSON

report.json

```
{
  "name": "Webshop SLA",
  "periodType": "monthly",
  "widgets": [ ... ],
  "id": 42
}
```

Your transformer

transformer.js

```
function transform(report) {
  // any JavaScript
  report.reportID = report.id;
  return report;
}
```

Any JSON value

delivered JSON

```
{
  "id": 42,
  "name": "Webshop SLA",
  "periodType": "monthly",
  "reportID": 42,
  "widgets": [ ... ]
}
```

an object, an array or a whole new schema

ONE REPORT RUN

PDF, XML and CSV never pass the transformers



What a transformer can do

Anything JavaScript can do with a JSON object



Add fields

stamp IDs, tenants or sources

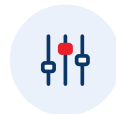
```
report.reportID = report.id;
```



Remove fields

drop what the receiver must not see

```
delete report.zabbixServerAPIURL;
```



Modify values

convert units, rename, reformat

```
report.start *= 1000; // ms
```



Compute KPIs

counts, averages, SLO checks

```
met = sli.totalUptime >= sli.goodSLA
```



Flatten

nested widgets into flat rows

```
report.widgets.flatMap(w => ...)
```



Map to a schema

exactly what the receiver expects

```
return { records: rows };
```

Example: one row per SLA

Nested SLA widget results become flat rows with a computed 'met' flag

SLA widget in the report

report.json

```
"widgets": [{
  "name": "sla-widget",
  "services": [{
    "name": "Webshop",
    "slis": [{
      "name": "Gold",
      "goodSLA": 99.9,
      "totalUptime": 99.95
    }]
  }, {
    "name": "Payments",
    "slis": [ ... ]
  }
], ... ]
```



Transformer

sla-rows.js

```
function transform(report) {
  const rows = [];
  for (const w of report.widgets) {
    if (w.name !== "sla-widget") continue;
    for (const s of w.services)
      for (const sli of s.slis)
        rows.push({
          service: s.name, sla: sli.name,
          slo: sli.goodSLA,
          sli: sli.totalUptime,
          met: sli.totalUptime >= sli.goodSLA,
        });
  }
  return { report: report.name, rows };
}
```



One row per SLA

delivered JSON

```
{
  "report": "Webshop SLA",
  "rows": [{
    "met": true,
    "service": "Webshop",
    "sla": "Gold",
    "sli": 99.95,
    "slo": 99.9
  }, {
    "met": false,
    "service": "Payments",
    ...
  }
]
```

- Nested services and SLIs become flat rows
- met is computed in the transformer: $SLI \geq SLO$

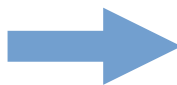
- A shape any database, BI tool or stream can ingest
- Field names as in the SLA widget's JSON export

Use a transformer to support OpenTelemetry

The report JSON on the left, the OTLP request the OpenTelemetry Collector receives on the right

report.json · excerpt

```
{
  "name": "Service SLAs",
  "periodType": "monthly",
  "start": 1785542400,
  "end": 1788220800, ..... 1
  "widgets": [{
    "name": "sla-widget",
    "displayName": "Customer SLAs",
    "services": [{
      "name": "Webshop", ..... 2
      "slis": [{
        "name": "Webshop 99.9", ..... 3
        "goodSLA": 99.9,
        "totalUptime": 99.96, ..... 4
        "failed": false
      }]
    }, ...]
  }, ...]
}
```



transform()

after the transformer · OTLP JSON

```
{ "resourceMetrics": [{
  "resource": { "attributes": [ ... ] },
  "scopeMetrics": [{
    "scope": { "name": "prisma.report" },
    "metrics": [{
      "name": "zabbix.sla.sli",
      "unit": "%",
      "gauge": { "dataPoints": [{
        "timeUnixNano": "1788220800000000000", ..... 1
        "asDouble": 99.96, ..... 4
        "attributes": [
          { "key": "service",
            "value": { "stringValue": "Webshop" } }, ..... 2
          { "key": "sla",
            "value": { "stringValue": "Webshop 99.9" } } ..... 3
        ] }, ... ] }
      }, { "name": "zabbix.sla.slo", ... } ]
    }
  ]
}]}
```

1 report.end

→ timeUnixNano

2 services[].name

→ attribute service

3 slis[].name

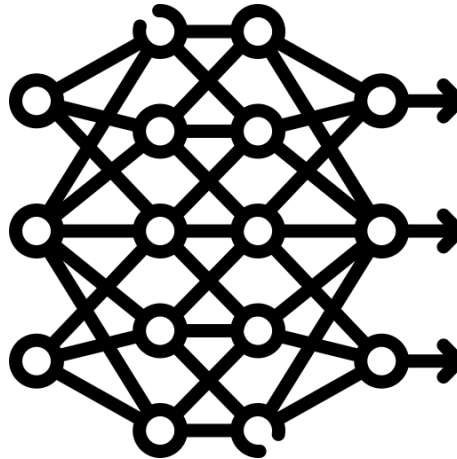
→ attribute sla

4 slis[].totalUptime

→ asDouble (%)

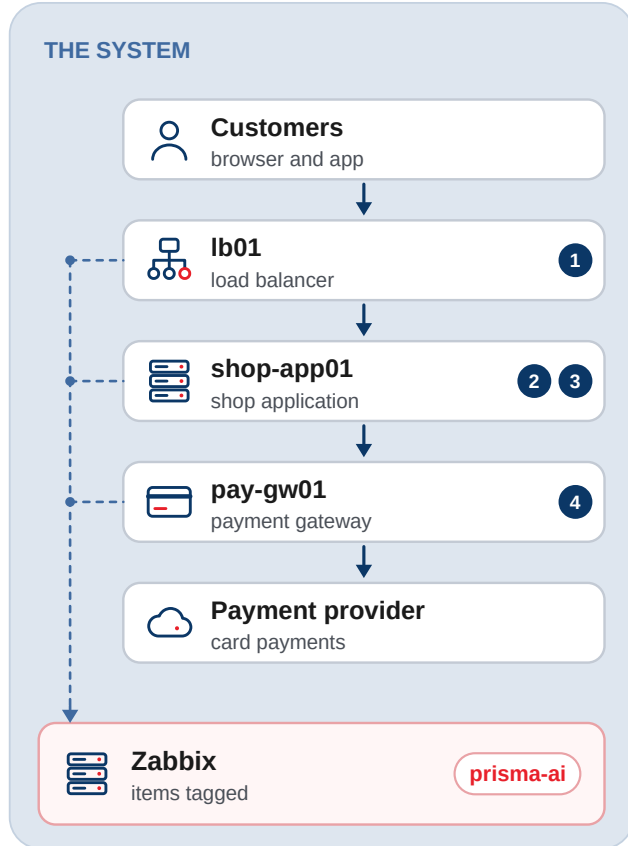
Putting Zabbix Data to Work

Using external AI-Triggers for Time-Series Data - no ChatGPT ;-)



Webshop: what the model sees

Four items from three hosts: demand, intent to buy, the outcome and the payment dependency



WHAT THE MODEL LEARNS FROM

4 items, one model

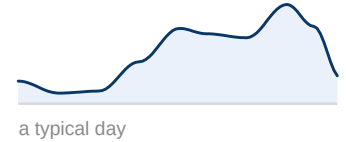
1



Sessions per minute

lb01 · shop.sessions

How many customers are in the shop right now:
the demand everything else depends on



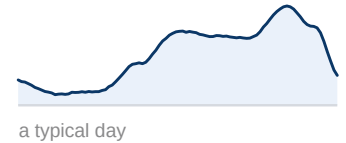
2



Checkouts started per minute

shop-app01 · shop.checkout.started

Customers who press 'Buy': the intent to pay



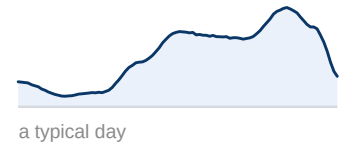
3



Orders completed per minute

shop-app01 · shop.orders.completed

Paid orders: the business outcome, and the
item nobody can set a sensible static limit on



4



Payment API latency p95, ms

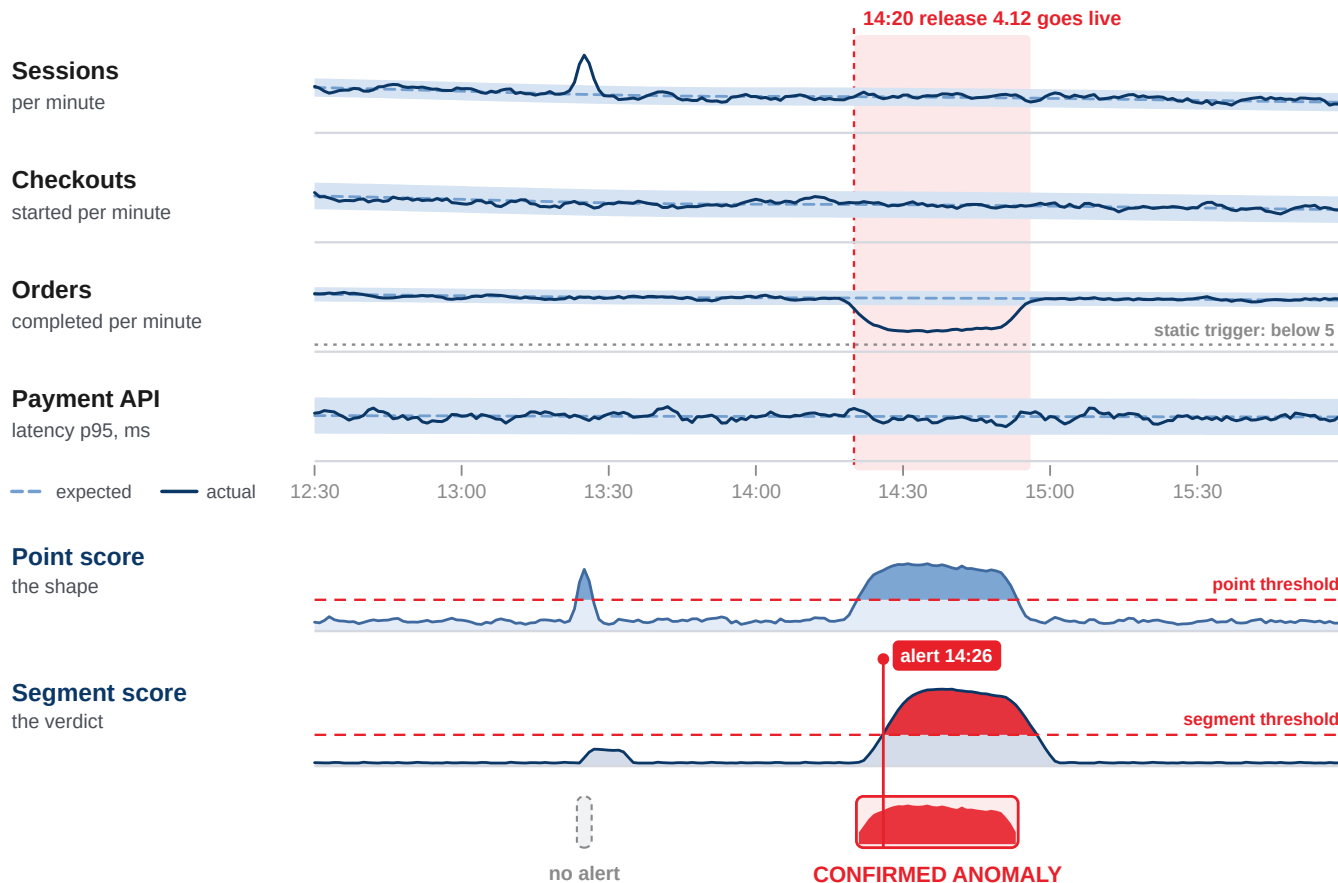
pay-gw01 · pay.api.p95

Response time of the external payment
provider, as seen from the gateway



Orders drop, traffic does not

A release breaks card payments: every value stays plausible, but orders no longer follow checkouts



ZABBIX PROBLEM

Average

14:26

[PRISMA AI] segment anomaly
detected by model 'Webshop
checkout'

prisma-ai

STATIC TRIGGERS

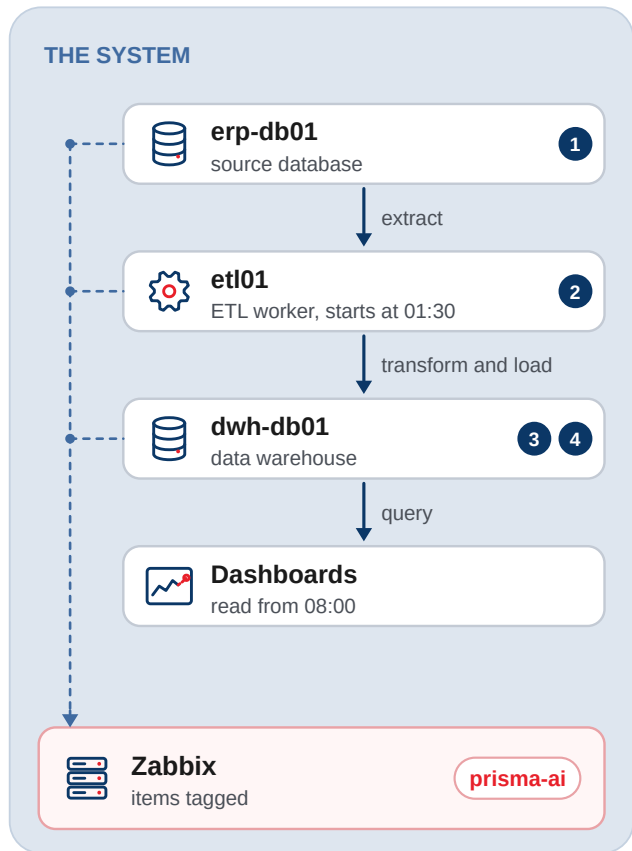
Orders below 5 per minute	<i>silent</i>
HTTP 5xx above 2 %	<i>silent</i>
Payment p95 above 2 s	<i>silent</i>

WHAT HAPPENED

- 13:25** A bot burst on the sessions: a point blip, no alert
- 14:20** Release 4.12 breaks the card form in some browsers
- 14:26** The segment trigger fires and pages the on-call engineer
- 14:52** Release rolled back, orders recover

Nightly housekeeper: what the model sees

Four items from three hosts: the housekeeper to extract, transform and load data into the warehouse



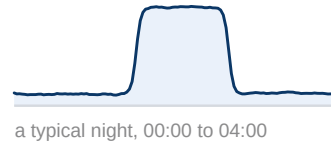
WHAT THE MODEL LEARNS FROM

4 items, one model

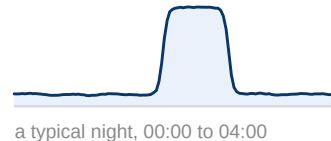
1 **Source DB reads** MB/s
`erp-db01 · db.read.throughput`
What the load extracts from the ERP system, on top of a little background traffic



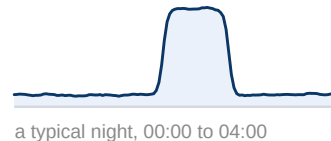
2 **ETL worker CPU** %
`etl01 · system.cpu.util`
Busy only while the load runs: the transform step between source and warehouse



3 **Rows inserted** per second
`dwh-db01 · dwh.rows.inserted`
What actually lands in the warehouse

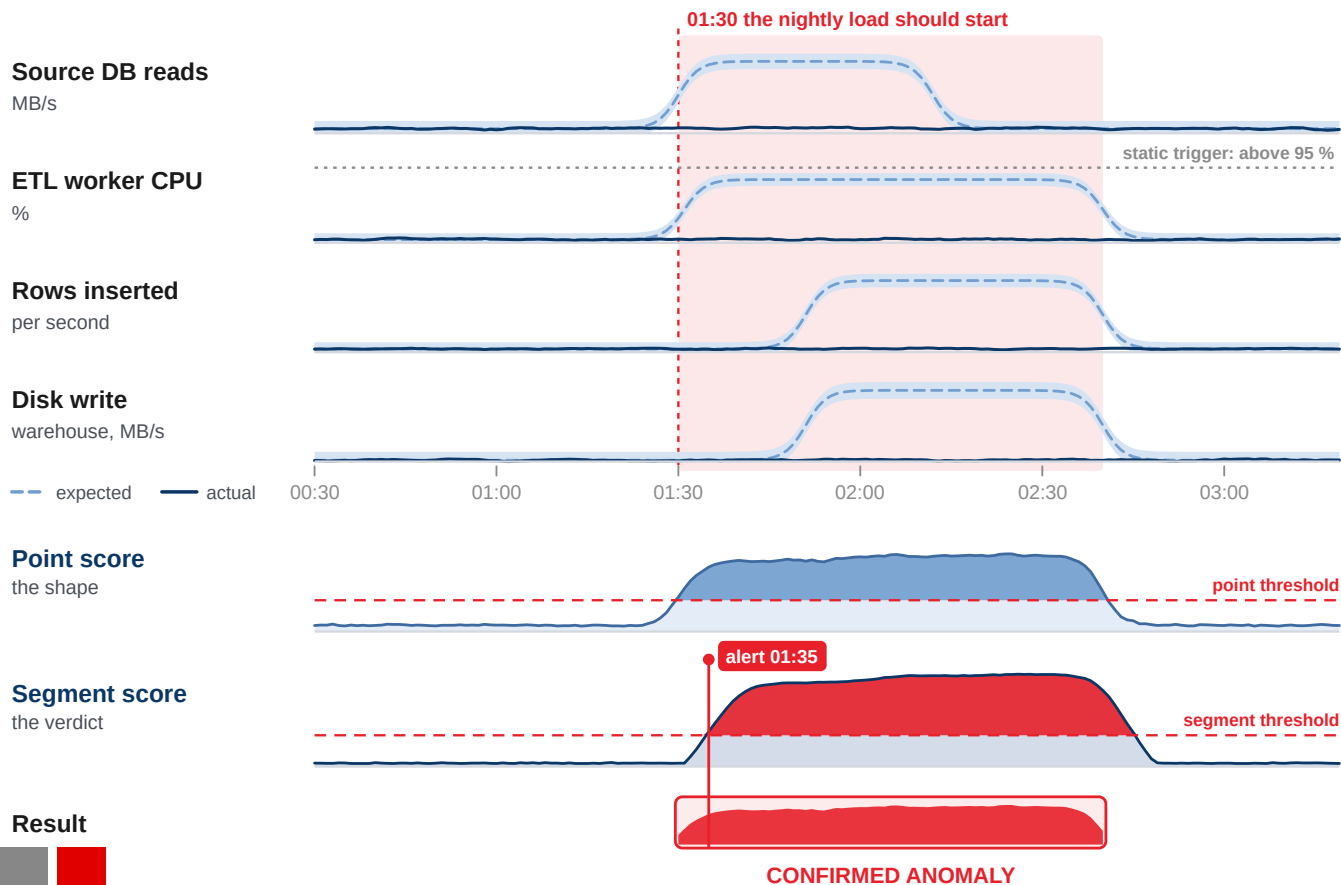


4 **Warehouse disk write** MB/s
`dwh-db01 · dwh.disk.write`
The storage side of the same load



The nightly load that never ran

Nothing happens at 01:30: no value is too high, but the load the model expects is missing



ZABBIX PROBLEM

Average

01:35

[PRISMA AI] segment anomaly detected by model 'Data warehouse load'

prisma-ai

STATIC TRIGGERS

ETL CPU above 95 %	silent
Source reads above 300 MB/s	silent
Disk write above 400 MB/s	silent

WHAT HAPPENED

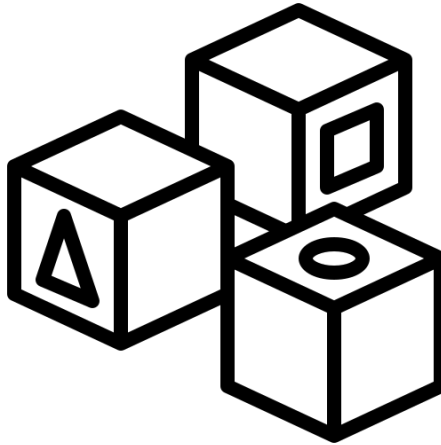
- 01:30** The load does not start: the scheduler cannot log in after a password rotation
- 01:35** The segment trigger fires and pages the data engineer on call
- 03:20** Credentials fixed, the load is started by hand
- 04:35** Load finished, long before office hours. Without the alert: empty dashboards at 08:00



ZABBIX
PREMIUM PARTNER

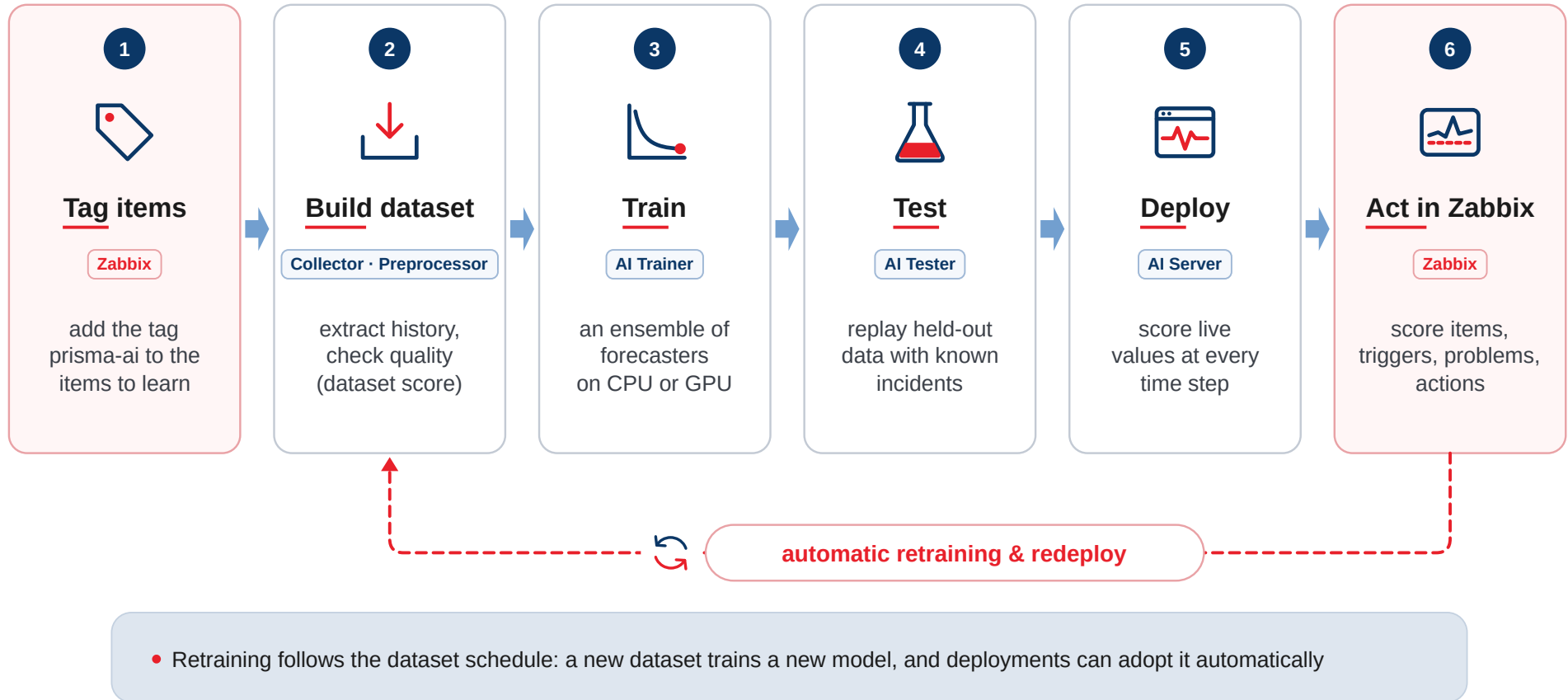
Putting Zabbix Data to Work

AI-Triggers under the hood



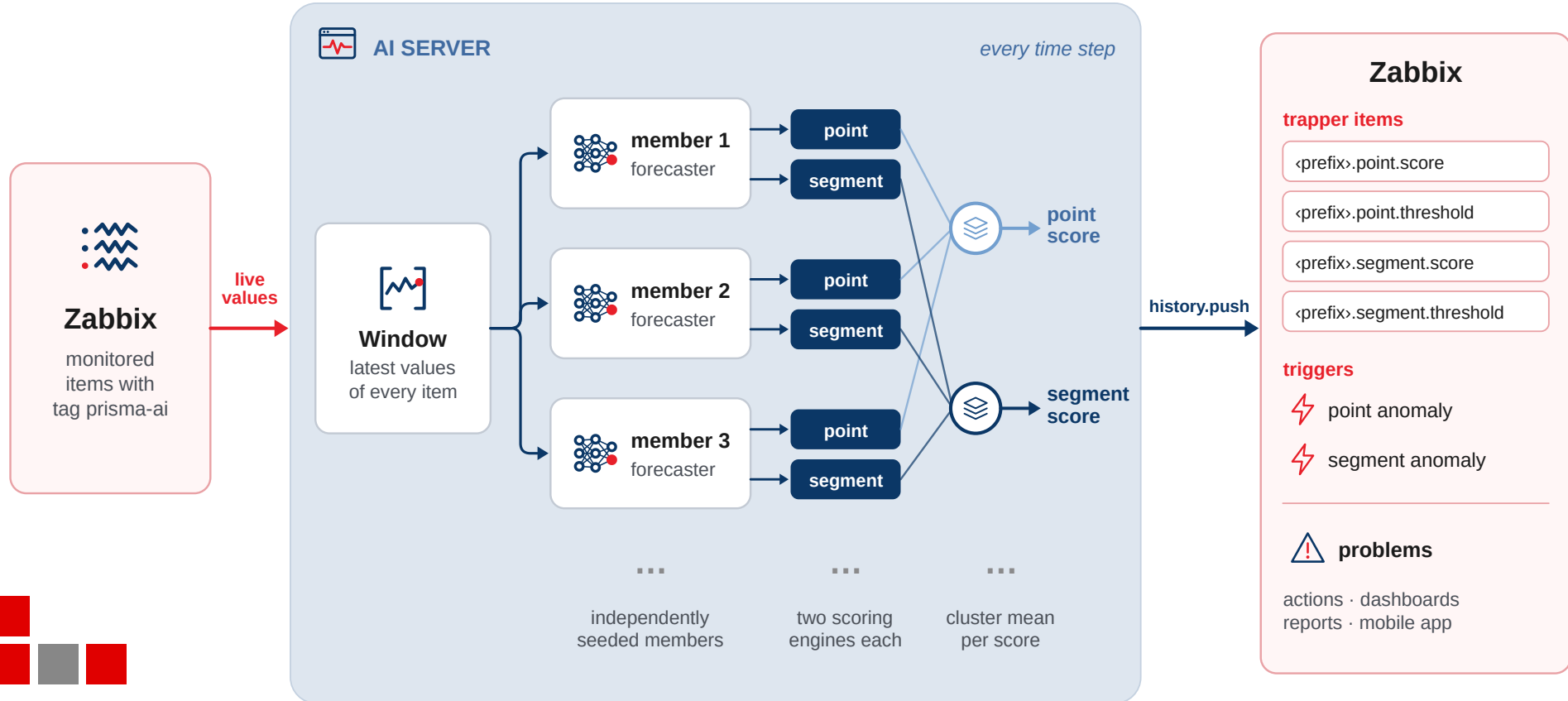
The life of an AI model

From a tagged Zabbix item to scores that behave like native Zabbix data



Live scoring, step by step

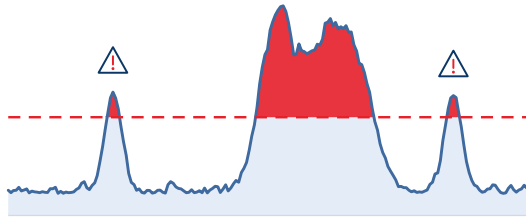
At every time step: window, ensemble, two scores, back to Zabbix



Why point and segment detection

No need to choose between a precise alert and a descriptive one

Point detection alone



every crossing alerts, blips included

Shape of the anomaly

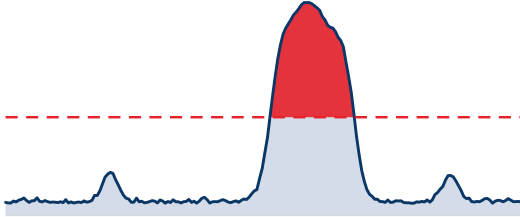


Precision (few false alarms)



**Rich shape for analysis,
but noisy as an alert**

Segment detection alone



one clear verdict, little contour

Shape of the anomaly

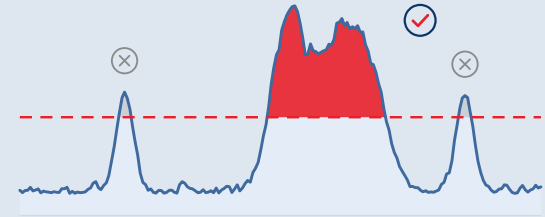


Precision (few false alarms)



**Precise alerts,
but little to analyse**

Combined in Prisma



blips dropped, the event keeps its shape

Shape of the anomaly



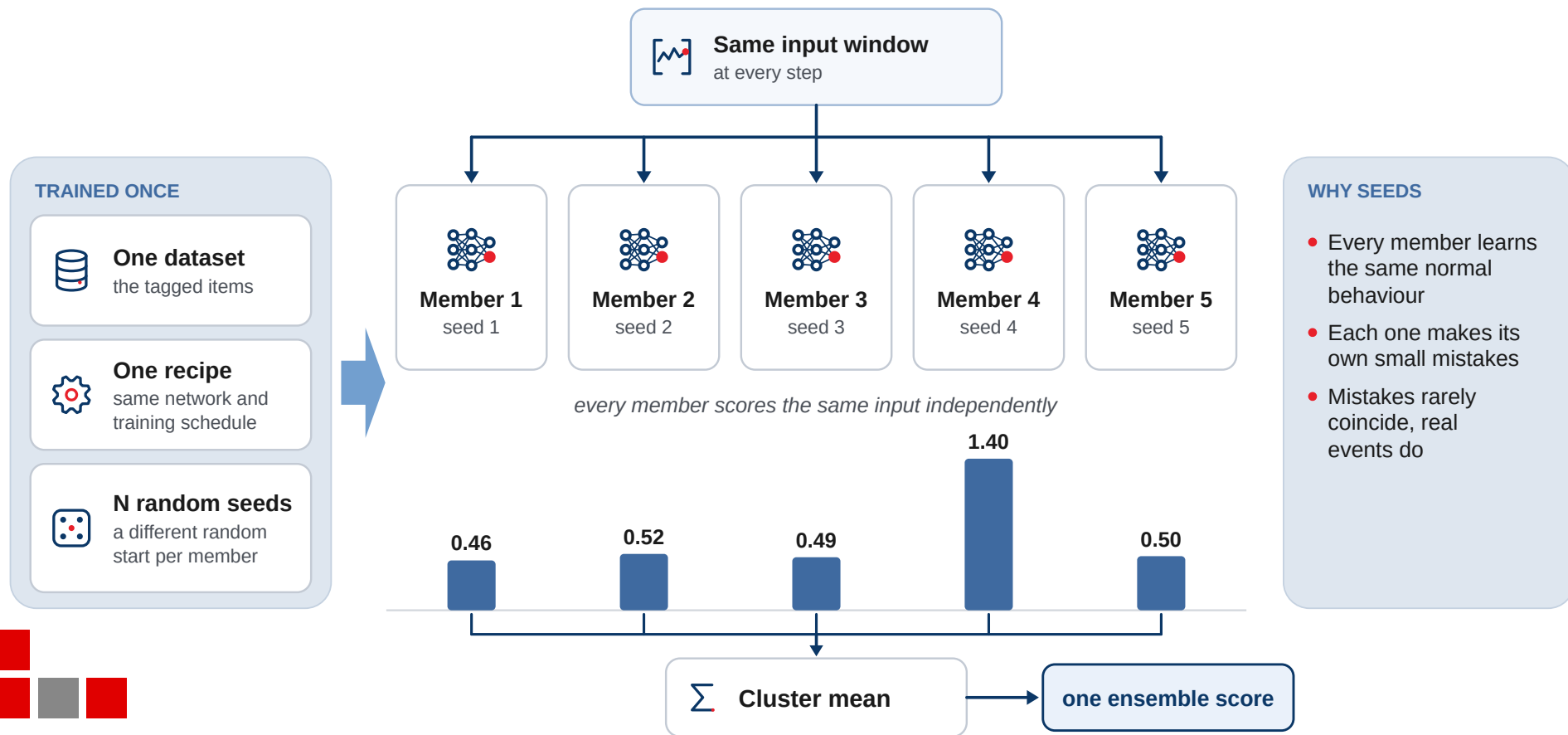
Precision (few false alarms)



**Alert with segment precision,
analyse with the point contour**

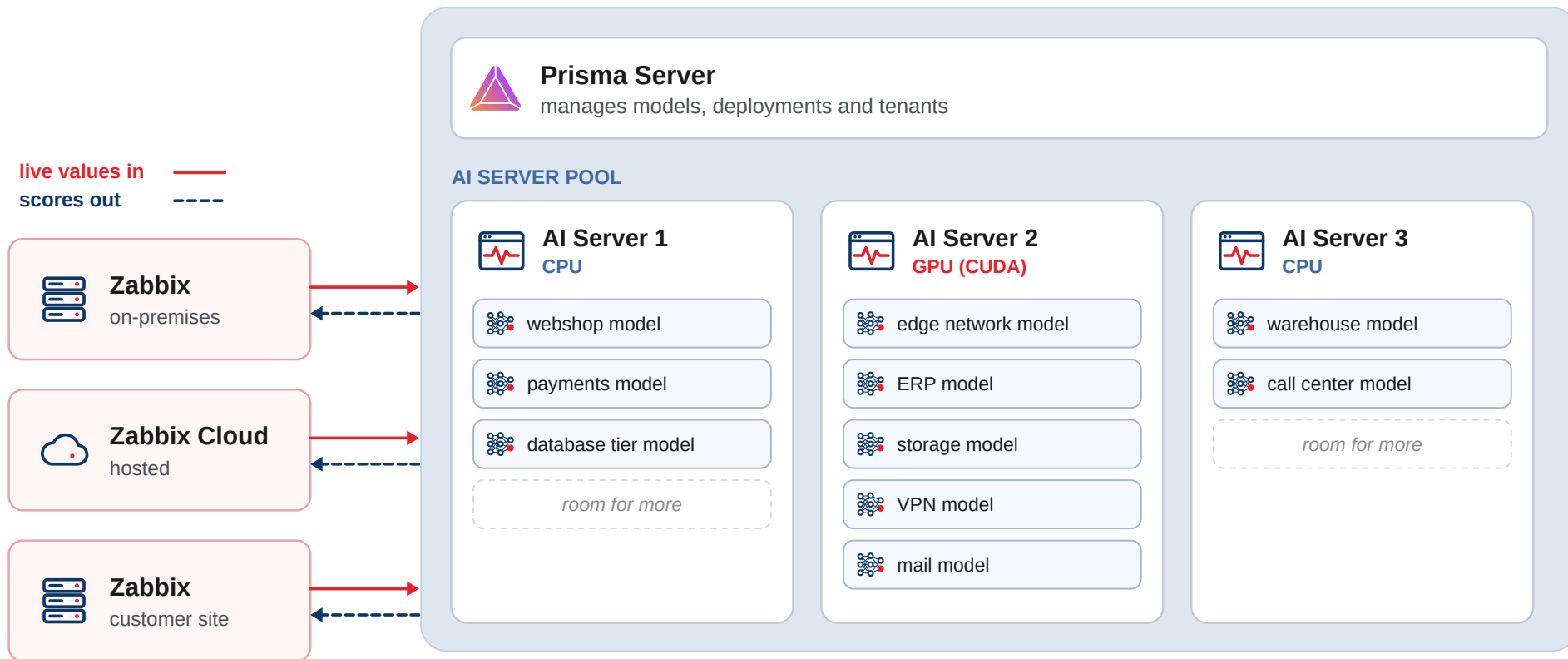
An ensemble of independently seeded forecasters

Same data, same recipe, different random seeds: every member scores the same input at every step



Scaling out

One Prisma, several Zabbix environments, a pool of AI Servers

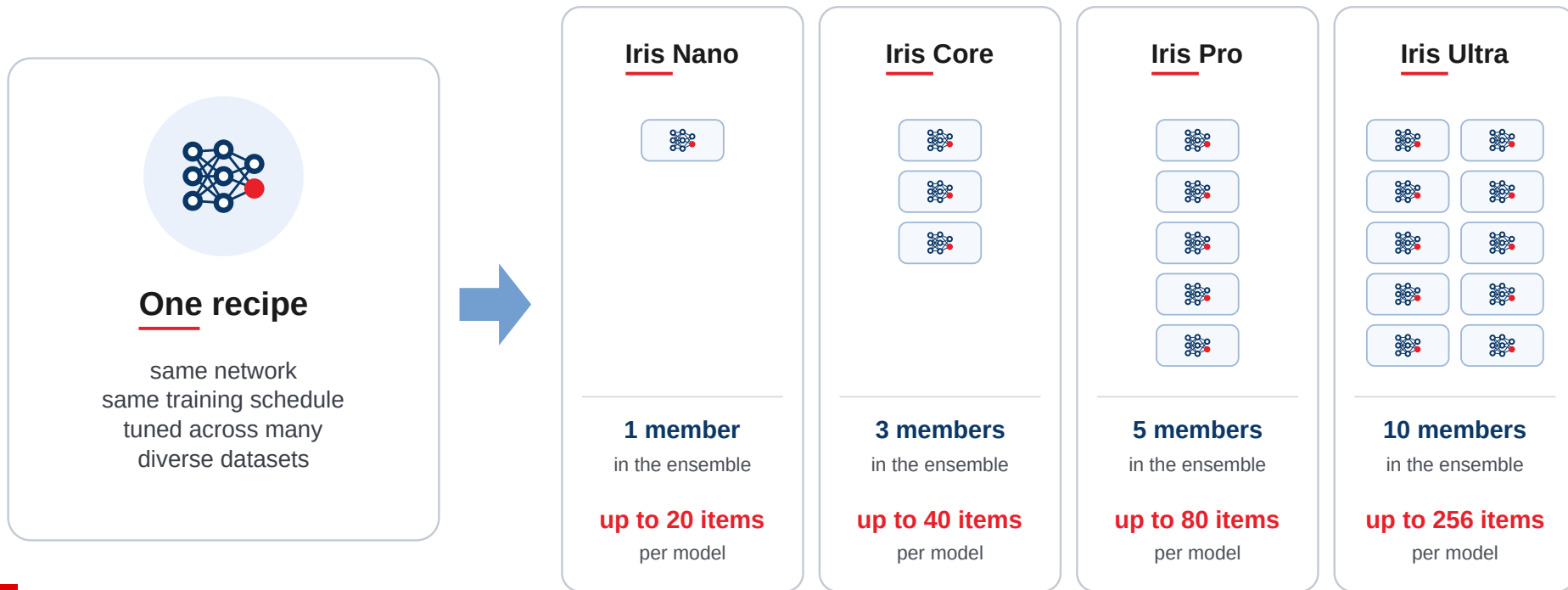


- Scale out like Zabbix proxies: add AI Servers when you add models

- Many models per AI Server · CPU or CUDA GPU

One recipe, four presets

Presets scale the ensemble and the item capacity, not the model



A larger preset buys more independently seeded members and more items, not a different model

What lands in Zabbix

Four trapper items and two triggers per deployment: to Zabbix it is just more data

LATEST DATA · HOST WEBSHOP

Name	Key	Last value
Point detection anomaly score	webshop.ai.point.score	0.34
Point detection threshold	webshop.ai.point.threshold	1.00
Segment detection anomaly score	webshop.ai.segment.score	0.12
Segment detection threshold	webshop.ai.segment.threshold	1.00

Zabbix trapper · numeric (float)

tag: prisma-output

one value per inference step

TRIGGERS

Average

[PRISMA AI] segment anomaly detected by model 'Webshop'

the alert to act on

```
last(/Webshop/webshop.ai.segment.score)>last(/Webshop/webshop.ai.segment.threshold)
```

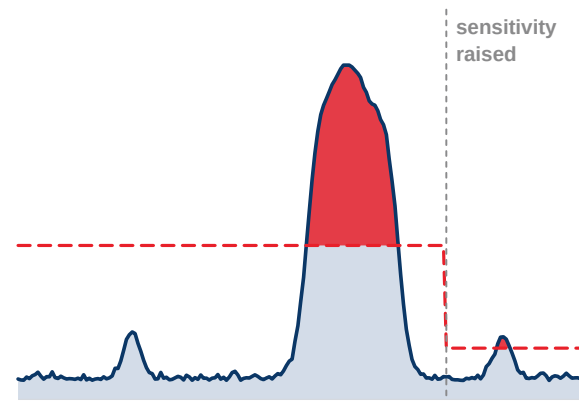
Average

[PRISMA AI] point anomaly detected by model 'Webshop'

context around it

```
last(/Webshop/webshop.ai.point.score)>last(/Webshop/webshop.ai.point.threshold)
```

Score and threshold in one graph



— segment.score - - - segment.threshold

- Thresholds are items, not constants
- The current sensitivity is visible next to the scores
- Changes show up in every graph, dashboard and report

Your Zabbix Data Can Do More - Takeaways

Reports for people

Turn monitoring history into clear reports for technical teams, managers and customers.

Understand performance

Data for systems

Deliver structured data and transform JSON to fit external applications and workflows.

Automate further processing

AI for monitoring

Learn normal behavior across related metrics and return anomaly scores to Zabbix.

Complement existing triggers

Start your journey with IntelliTrend Prisma

Explore Reporting and AI with your own Zabbix data.

 www.intellitrend.de/prisma



ZABBIX
PREMIUM PARTNER



Putting Zabbix Data to Work

-

AI-Based Anomaly Detection and

Advanced Reporting

Thank You!



IntelliTrend GmbH

Contact: Wolfgang Alper

 www.intellitrend.de



ZABBIX
PREMIUM PARTNER

wolfgang.alper@intellitrend.de