

Stop Monitoring Servers. Start Monitoring Services.

Anton Kassimov



**All hosts are green. Can
users still pay?**

The kitchen is running. Dinner never arrives.

Component checks tell us
the kitchen is ready.

**Journey checks tell us
whether the meal arrived.**



The cost of missing service context

Internet banking

A payment fails or its confirmation never arrives.

Customers may retry or leave.
Support demand rises.
Trust is at risk.



Corporate email

Messages remain in an outbound queue.

Approvals and customer replies wait.
Work stalls while employees
still appear online.

A service view helps teams decide which customer outcome to restore first.

About me



Anton Kassimov

CEO at Gals Software

Monitoring practitioner

15+ years in monitoring

Zabbix Certified

Trainer

Expert

Contacts

a@galssoftware.com

antoniusfirst 



A service

An outcome a user consumes.

Applications, databases, infrastructure and integrations support that outcome.

**A completed payment.
A delivered message.**





Three business signals

**97.8
%**

Transaction success

Example: completed operations ÷ attempts

1.8 s

Response time

Example: p95 for the full user journey

**99.9
%**

Availability

Example: good service time ÷ expected time

Every service is a customer journey



1 Open the application

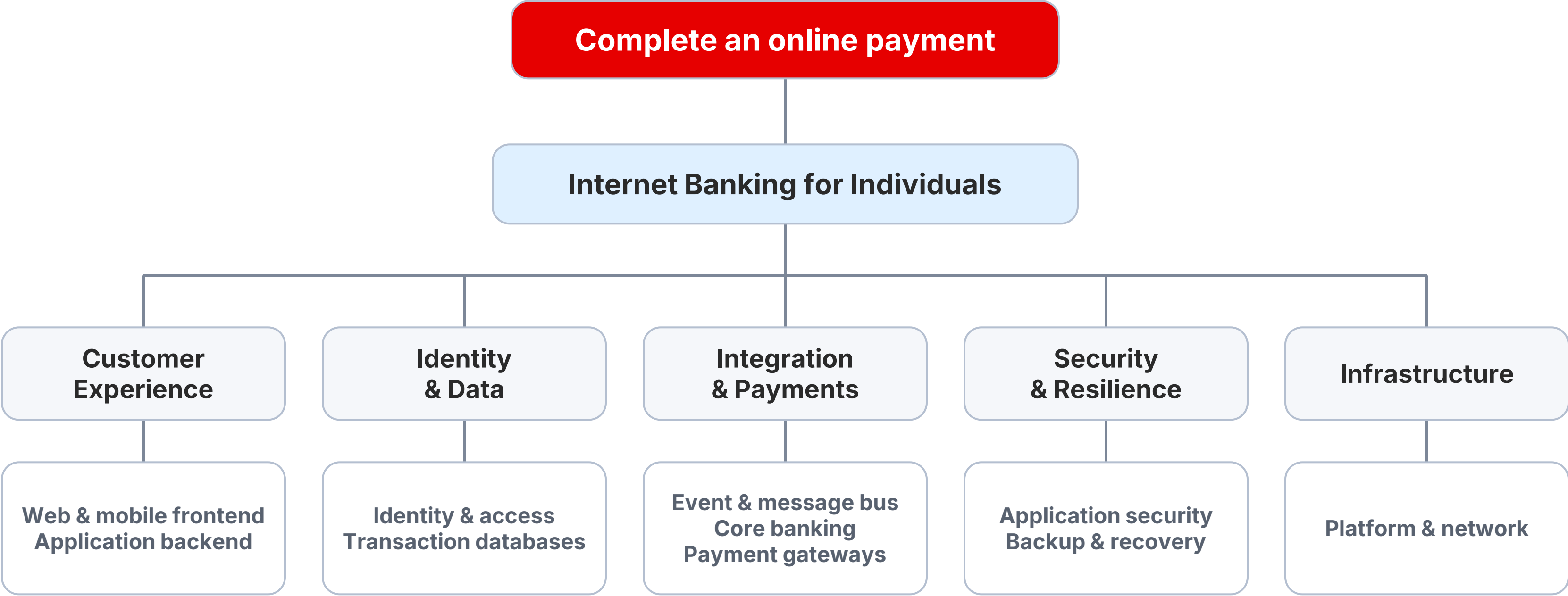
2 Authenticate

3 View the balance

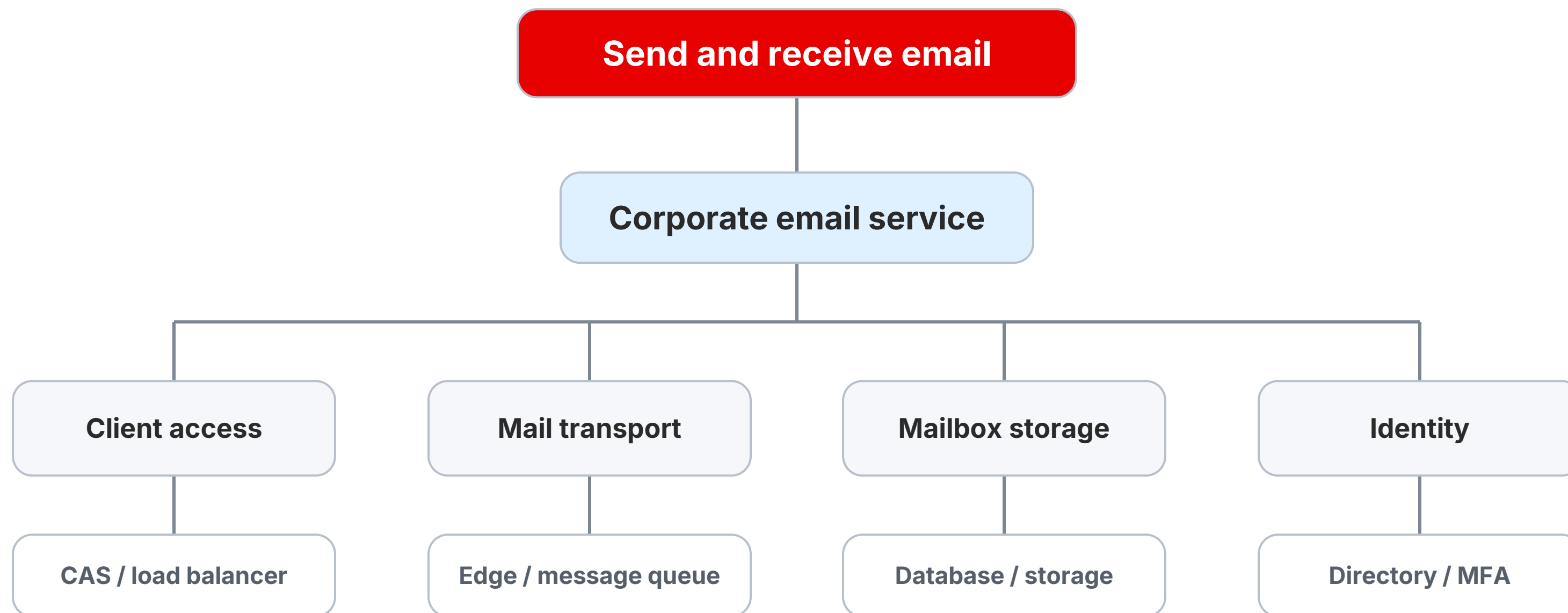
4 Submit the payment

5 Confirm the result

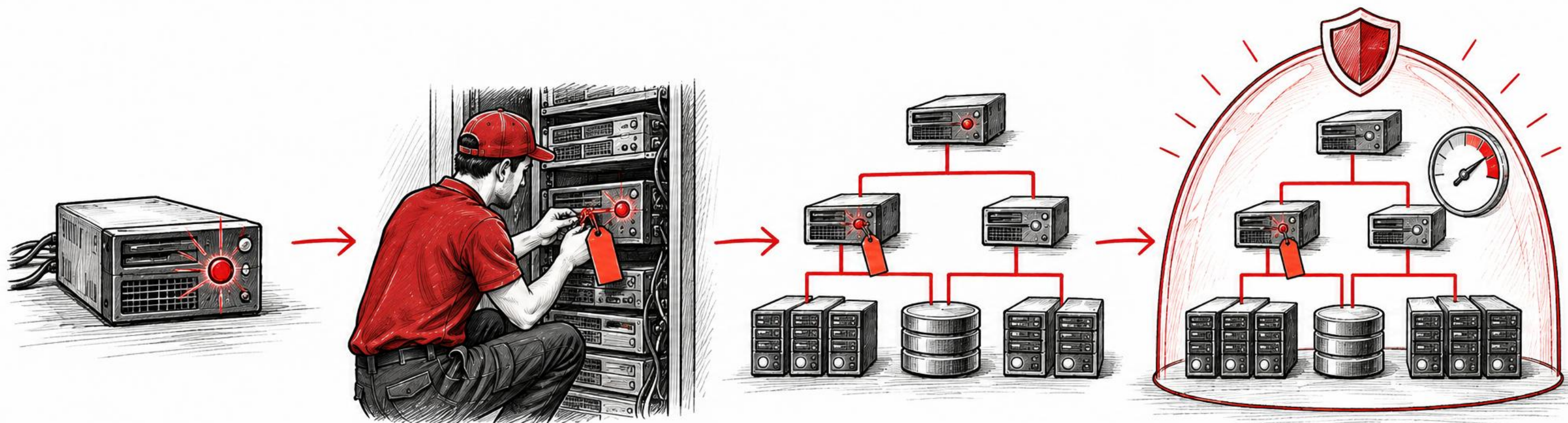
Internet banking dependencies



Corporate email dependencies



From leaf services to business impact



1. Define the leaf

A component with a clear failure meaning

2. Tag the problem

Use stable, semantic trigger tags

3. Build parents

Reflect the journey and redundancy

4. Add governance

Attach SLAs and service actions

Two tag systems, two different jobs

Problem tags

Connect a problem to a leaf service

bs_id = mail.edge

Edge Transport

All configured problem-tag conditions must match.



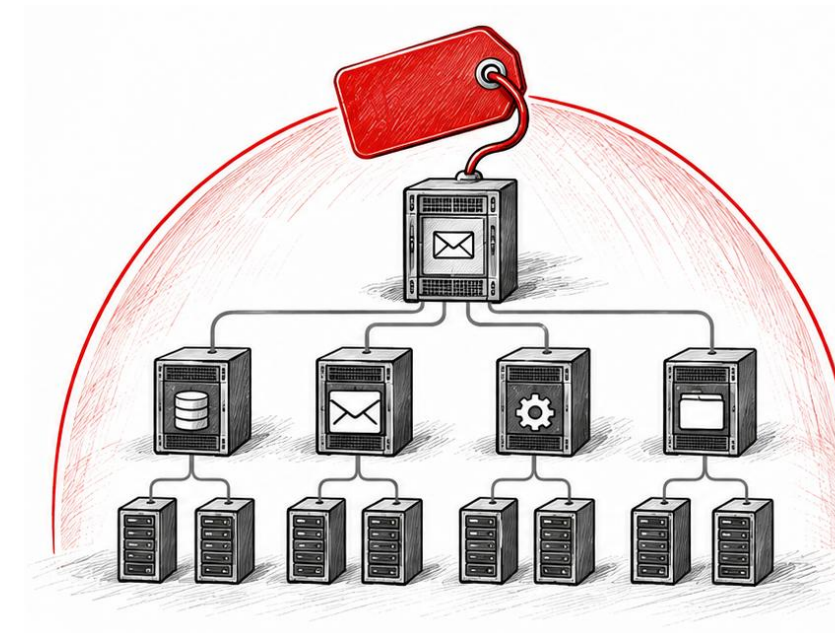
Service tags

Choose the scope of an SLA

business_service = corporate-email

Corporate Email SLA

Tag the relevant parent and child services.

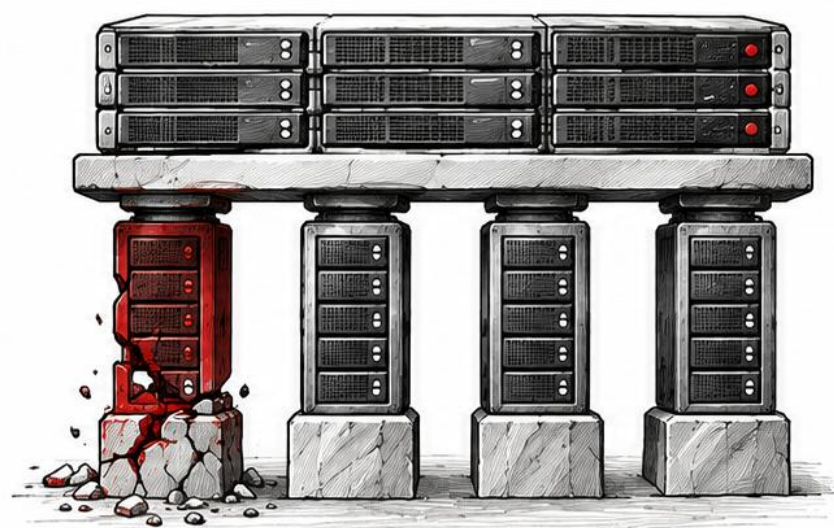


Status rules reflect dependency and redundancy



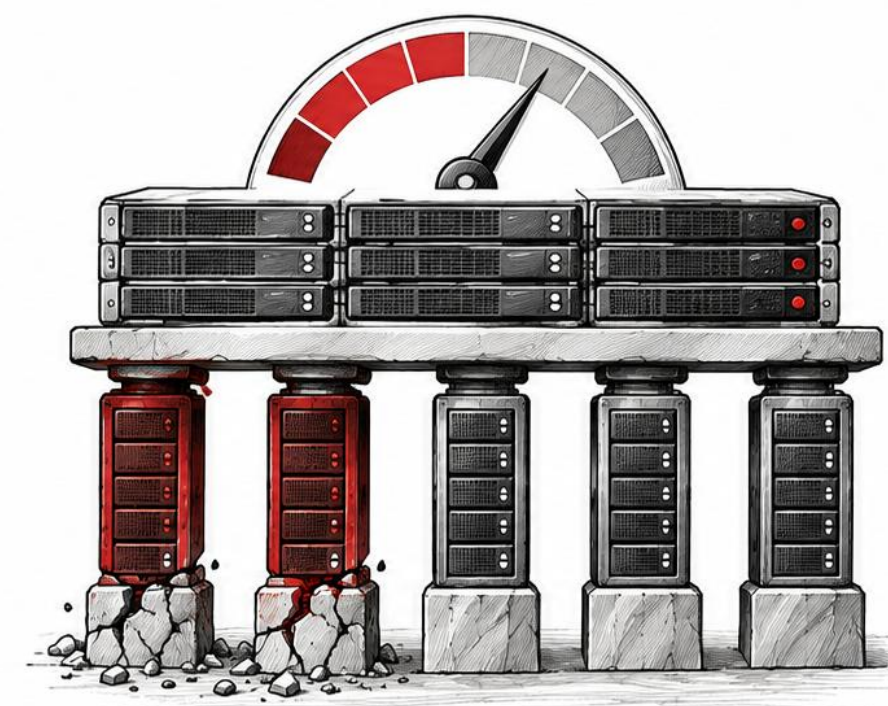
Most critical child

Use when one failed dependency can break the journey.



All children affected

Use when surviving components can keep the service working.



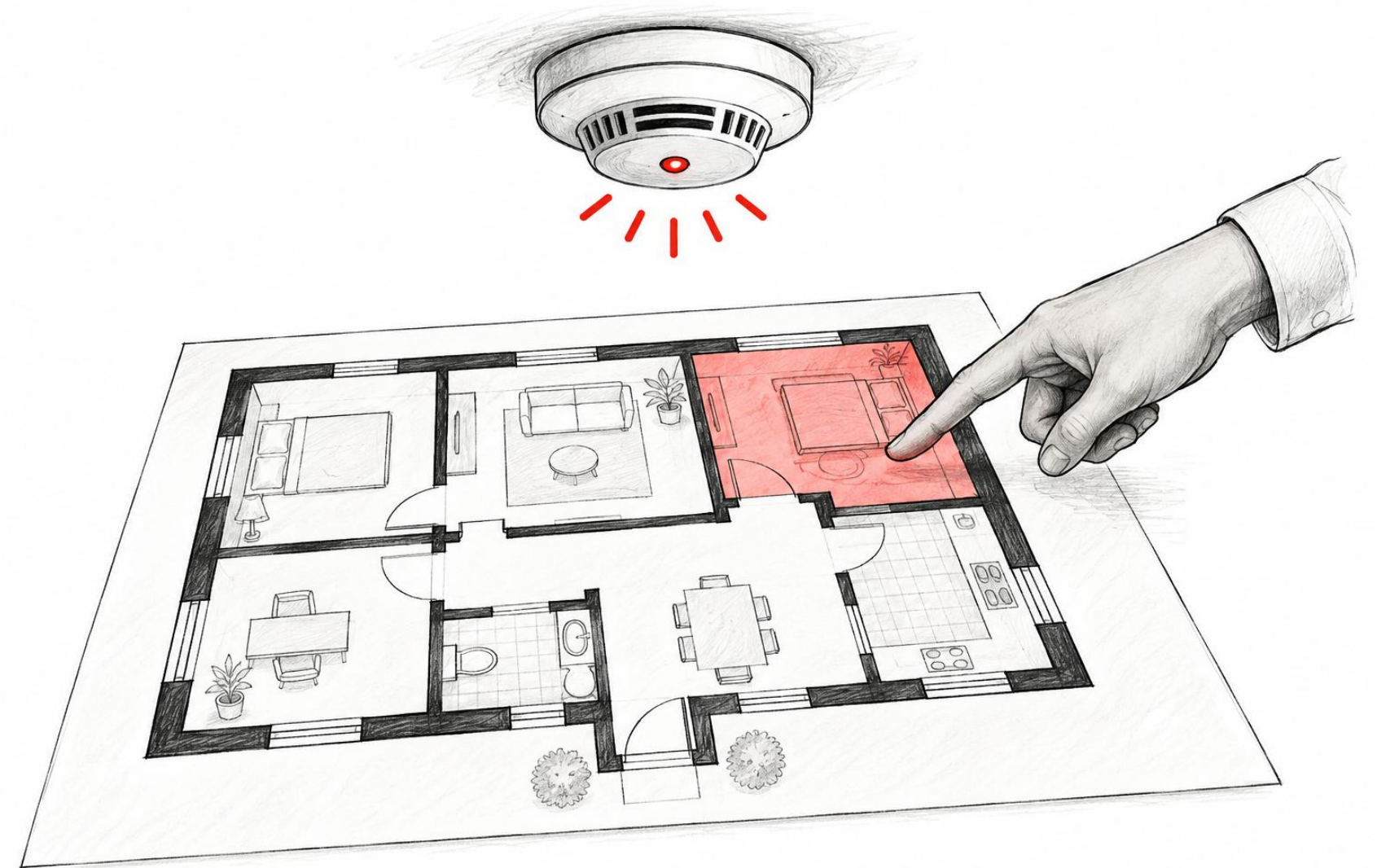
N of M / weighted rule

Use counts or weights for pools and partial capacity.

The alarm and the floor plan

**The alarm gets attention.
The plan guides the response.**

Problem events raise the alarm.
The service model connects it to
affected journeys and their
dependencies.



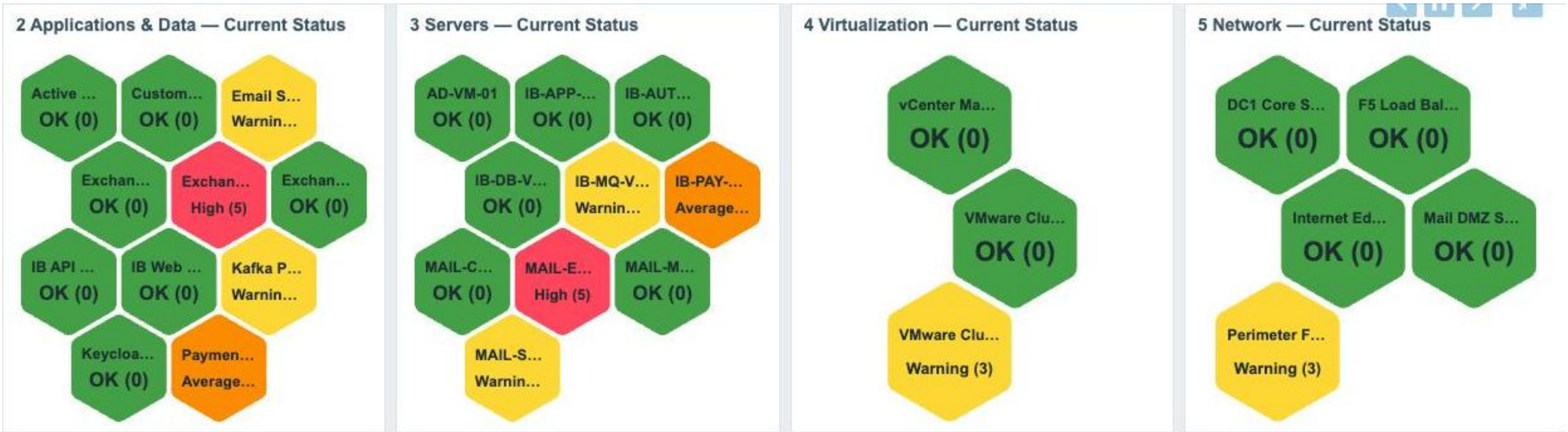
One entry point, five levels of investigation

Service → applications → servers → virtualization → network



Business impact first

Root tiles show native service status. Lower layers show each object's trigger state.



Internet banking: locate the affected branch

Native Services view • Business service → service layer

Internet Banking for Individuals

Parent services:

Status: Average

SLA: Internet Banking — Daily 99.9% (24x7): 0 ? Internet Banking — Monthly 99.9% (24x7): 0.0028 ?

Tags: business_service: internet-banking level: business_service portfolio: Business Service Demo

Name	Status	Root cause
Customer Experience 2	OK	
Identity & Data 2	OK	
Integration & Payments 3	Average	Payment Gateways: Payment success rate below target, Event & Message Bus: Message queue backlog above target
Security & Resilience 2	OK	
Infrastructure 1	OK	

Integration & Payments

Its Average state propagates to the business-service parent.

Payment success and message-queue problems are listed as contributing causes.

Corporate email: separate impact from noise

Native Services view • The same model, a different service

Corporate Email Service

Parent services:

Status: High

SLA: Corporate Email — Daily 99.9% (24x7): 0 Corporate Email — Monthly 99.9% (24x7): 0.0028

Tags: business_service: corporate-email level: business_service portfolio: Business Service Demo

Name	Status	Root cause
User Access 2	OK	
Messaging Core 2	High	Edge Transport: Outbound mail queue backlog above target
Identity & Traffic 2	OK	
Security & Compliance 2	Warning	Email Security: Email security processing latency above target
Resilience 2	OK	

Messaging Core

High severity is visible at the business-service level.

Edge Transport is the affected mail-routing component. Healthy branches remain visible.

From service impact to the responsible application

Corporate Email → Exchange Edge Transport → MAIL-EDGE-VM-01

The screenshot displays the Zabbix web interface. At the top, there are tabs for '1 Service Overview', '2 Applications', '3 Servers', '4 Virtualization', '5 Network', and 'SLA & Availability'. The '2 Applications' tab is active. On the left, a tree view shows the hierarchy: 'Corporate Email Service' (1), 'Application' (1), 'Exchange Client Access', 'Exchange Edge Transport' (1), 'Database', 'Directory Service', 'Security Gateway' (1), and 'Internet Banking for Individuals' (1, 1). The 'Exchange Edge Transport' application is selected and highlighted in yellow. The main panel shows the 'Selected Object — Current Status' with a large red box containing the text 'High (5)'. To the right, the 'Selected Object Context' panel shows details for 'Exchange Edge Transport' (1), including its description: 'Inbound and outbound message routing at the network boundary. Runs on: MAIL-EDGE-VM-01.' Below this, there are monitoring metrics: 'Monitoring' (0), 'Dashboards' (0), 'Latest data' (6), 'Graphs' (0), and 'Web' (0). At the bottom, there are tags: 'business_service: Corporate Email Service' and 'object_type: Application'.

Keep the context

Select the application.
Read its status. Identify
the hosting server.

Then inspect its queue,
CPU and memory metrics.

Current status is not the same as period availability

Corporate Email • Daily SLO 99.9% • 24x7

Service ▲	SLO	2026-09-08	2026-09-09	2026-09-10
Active Directory	99.9%	100	100	100
Backup & Recovery	99.9%	100	100	100
Client Access Services	99.9%	100	100	100
Corporate Email Service	99.9%	0.0062	0	0
Edge Transport	99.9%	0.0062	0	0
Email Security	99.9%	0.0062	0	0

Healthy components: 100% Affected branch and root: 0% today

Day	SLO	SLI	Uptime	Downtime	Error budget	Excluded downtimes
2026-09-10	99.9%	0	0	2h 20m 1s	-2h 20m 1s	
2026-09-09	99.9%	0	0	1d	-1d	
2026-09-08	99.9%	0.0062	5s	22h 19m 22s	-22h 19m 22s	

Views for each audience

Business

Outcome, trend, and customer impact

Service owner

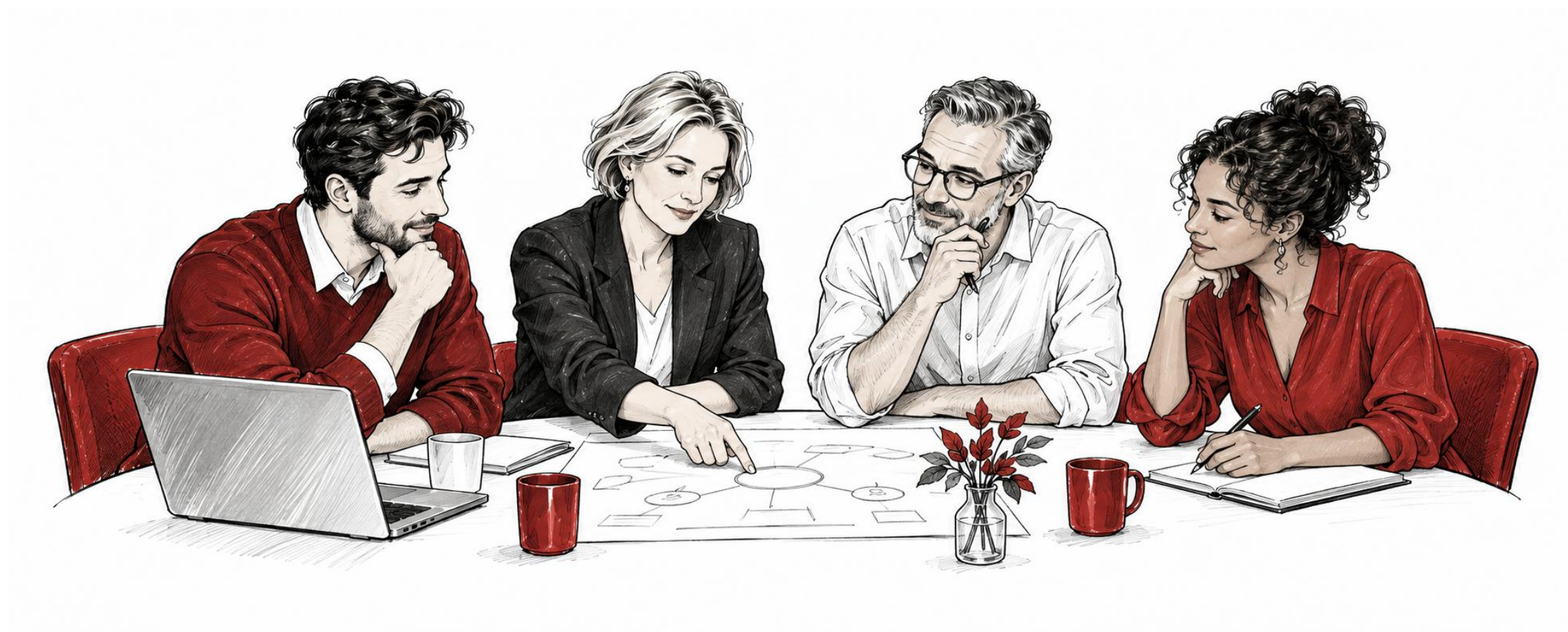
SLO, error budget, and dependencies

Operations

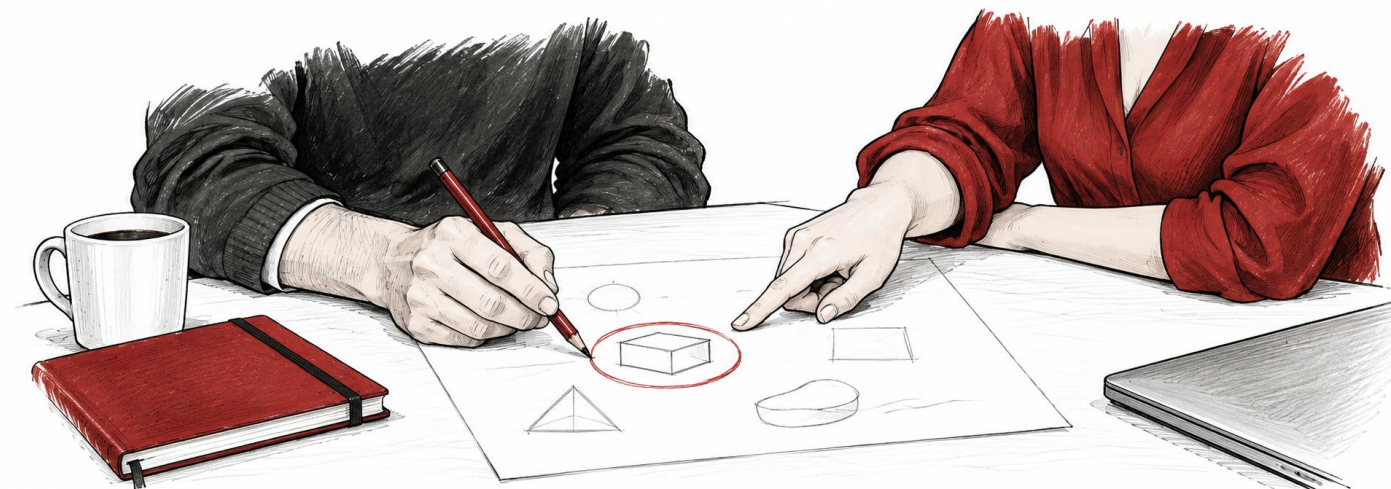
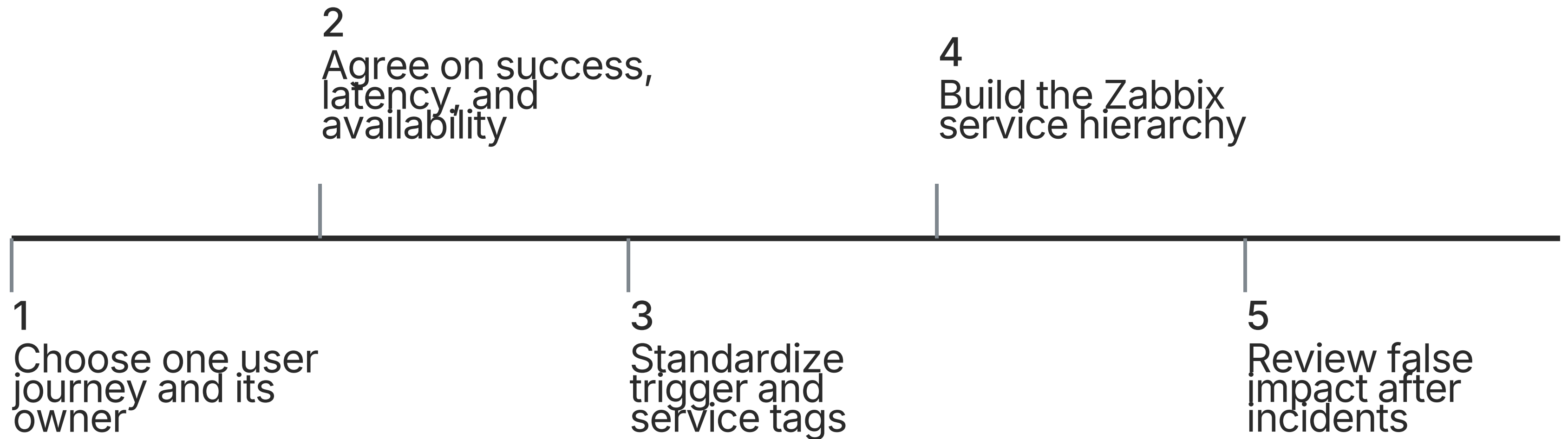
Impact, severity, and escalation path

Engineering

Root cause, evidence, and drill-down



A practical rollout in five steps



**Stop monitoring servers.
Start explaining service
impact.**



Questions?

What service will you model first?



Appendix: service modeling checklist

Answer these questions with the service owner before you create the first Zabbix service.

Outcome and owner

Who uses it?
What outcome do they expect?
Who owns the decision?

Dependencies and tags

Which leaf failures affect it?
Which semantic tags connect them?
Where is redundancy?

Rules and review

How should status roll up?
What SLO applies?
When will the model be reviewed?

Appendix: metric definitions

Availability (%) = uptime ÷ eligible service time × 100

Success rate (%) = successful transactions ÷ attempts × 100

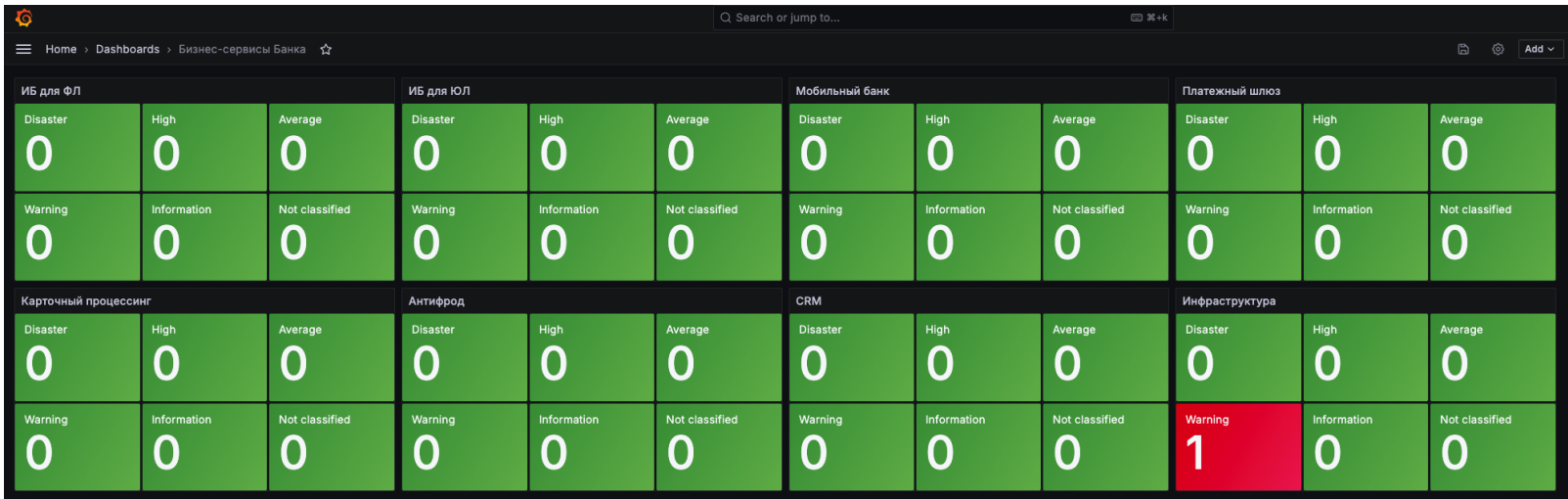
Latency = a journey percentile, such as p95 or p99

Allowed failure fraction = 1 – SLO (%) ÷ 100

At 99.9% SLO, the allowance is 0.1% of eligible time

Volume = transactions per time window; context, not health by itself

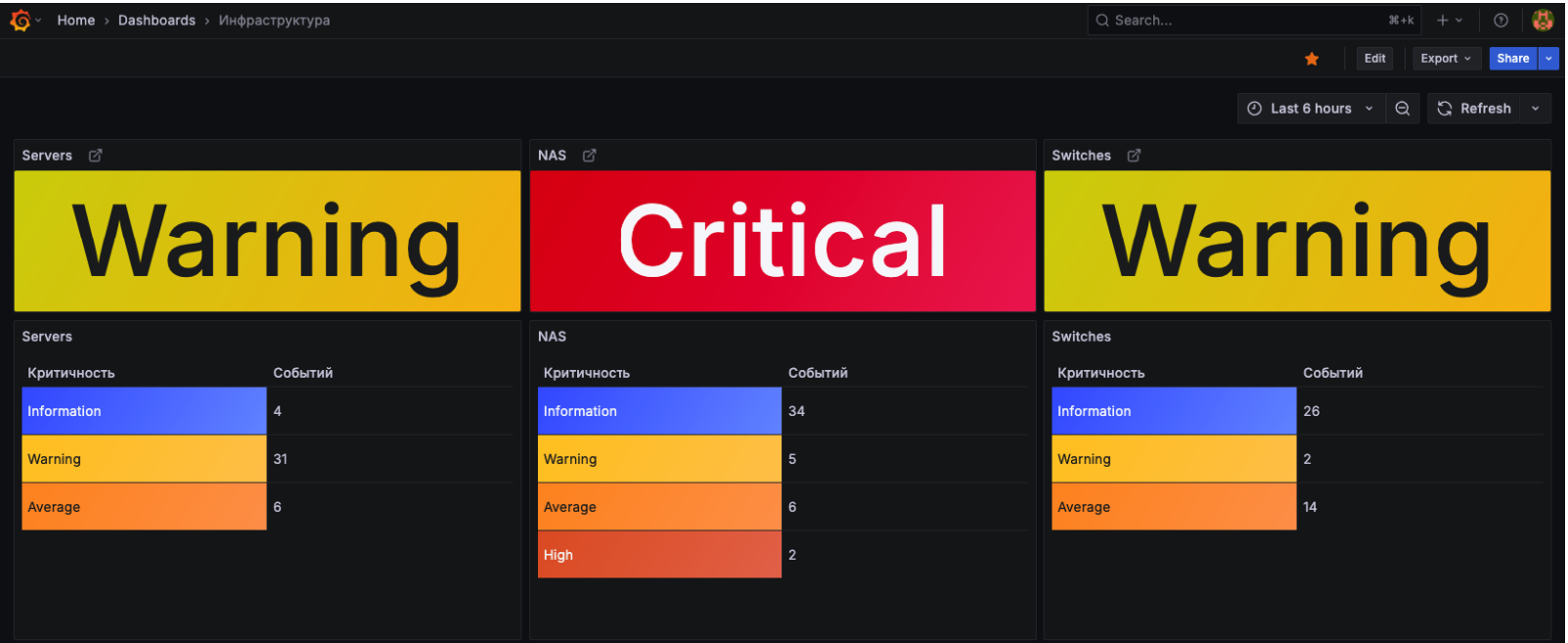
An optional Grafana view for the same audience



Portfolio view

Scan every service in seconds

Existing dashboard example • Separate from the Zabbix demo shown here



Incident view

Expand only the services that need attention