OBSERVABILITY CONFERENCE
HELSINKI, 21.05.2026

# Observability Architecture

From tooling to strategic value creation

**Understand → Standardize → Scale**

Zoltan Pal

# Scale - From reactive to proactive / autonomous operations

## The maturity shift

**Reactive**

Manage incidents

**Proactive**

Detect patterns and impact earlier

**Autonomous**

Automate safe responses

Autonomous operations is hard to achieve when observability is fragmented and only reactive.

### ChatGPT

- *How many percentages of IT companies has IT strategy? …and how many percentage of IT strategies are successfully implemented?*
- *Most large organizations claim to have a digital or IT strategy — **around 90–94% in recent surveys** — but successful execution is much weaker. Research commonly suggests that around two-thirds of well-formulated strategies fail in execution, meaning only roughly **one-third are successfully implemented.***
  ***The gap is usually not strategy creation, but strategy translation, ownership, standardization, and execution.***

# Observability is losing its meaning because every team defines it differently.

## Symptom

- Different teams have built their own data models, definitions, tooling, dashboards.
- Ad-hoc custom solutions for demands
- No overall picture about FinOps and Tooling

## Result

- Data becomes fragmented >> **hard to correlate**
- Interpretation becomes subjective >> **no processes**
- Nobody has a shared view of what is happening >> **no strategy**
- Maintenance and configuration **cannot be centralized**

## Risk

- The organization will be optimized for **reactive actions** instead of proactive operations.
- Local tools will result **not an Enterprise-grade solution.**

# Understand - The current pattern: fragmented understanding



## Same incident. Different views.

**Infra team**

Resource signals

Proactive alerts for CPU, Disk, RAM etc.

**App team**

Technical symptoms

Correlation of logs, metrics, events

**Service desk / ITSM**

ITIL process status

SLA/SLO breaches, MTTR

**Business**

Impact analyses

Customer/business/brand impact in numbers

**No single interpretation layer**

Without a shared architecture, observability produces more data but not necessarily more understanding.

# Define the interpretation layer: what the organizations want to understand

- Service health and performance
- Operational risks, resilience and dependencies
- Incident impact analyses >> Technical vs Business impact
- Customer experience and behavior
- FinOps

**No shared definition means no shared understanding.**

# Standardize - Why tooling is not the real issue



**The question is not:** “Which tool?”

**The question is:** “What do we want to understand, and what decisions should improve?”

## Key points:

- Which data to collect
- Which analytics to set up
- Which process will use the collected data & analytics as an input
- Which strategic value will be improved by the process as an output

**Tooling answers how data is collected.**

**Strategy answers why data is collected.**

**Architecture connects the two**

# From technical telemetry to business-impact insight

## Which data to collect

- Must be based on the Application / Service priorities
- For wich KPIs ans processes
- Avoid duplication (common definitions)

## Which analytics to set up

- Performance KPIs (e.g.: RED)
- DevOps KPIs (e.g.: DORA)
- IT Service Management (SLA/SLO/SLI)
- User Experience Measures

## Process for strategic value

- SMART Capex
- NPS-Performance correlation
- Financial decisions

**Events** → **Interpretation** → **Business decisions**

Observability becomes valuable when it changes decisions, priorities, and outcomes.

# Observability cannot remain defined team by team

Architecture standardizes how signals are interpreted before tools are scaled.

**Signals**

logs, metrics, traces, events

**Model**

services, ownership, dependencies

**Analyses**

correlation, severity, impact

**Insight**

actionable, role-specific output

&

**Right stakeholder**

Operations, service owners, business teams, leadership

**Right process**

Incident, change, problem, service review, resilience

**Right decision**

Prioritize, communicate, mitigate, automate, improve

# Scale insight into roles, processes, and decisions

**Right stakeholder**

Operations, service owners, business teams, leadership

**Right process**

Incident, change, problem, service review, resilience

**Right decision**

Prioritize, communicate, mitigate, automate, improve

## Scale means insight is trusted and embedded into how the organization works.

- Look beyond baseline requirements to address the organization's immediate pain points
- Strategy defines tomorrow; pain points drive today's priorities
  ...and attention is anchored in the present
- Deliver value on today's challenges to build credibility and secure alignment for tomorrow's strategy

# Closing message

***Short Story...***

*Sometimes I feel that Dan Ariely words (posted in 2013) about 'Big data' are true for the Observability today:*

*"Big data is like teenage sex: everyone talks about it, nobody really knows how to do it, everyone thinks everyone else is doing it, so everyone claims they are doing it."*

**Tools should follow the observability architecture — not define it.**

**Understand what matters. Standardize how it is analyzed. Scale insights into decisions and operations.**

UNDERSTAND | STANDARDIZE | SCALE

**Thank you**