StackState

Redefining Observability

A 3-Step Approach for Gaining Control in Fast-Moving IT Landscapes

A Guide for IT Service Managers, Infrastructure Leaders, and Operations Leaders

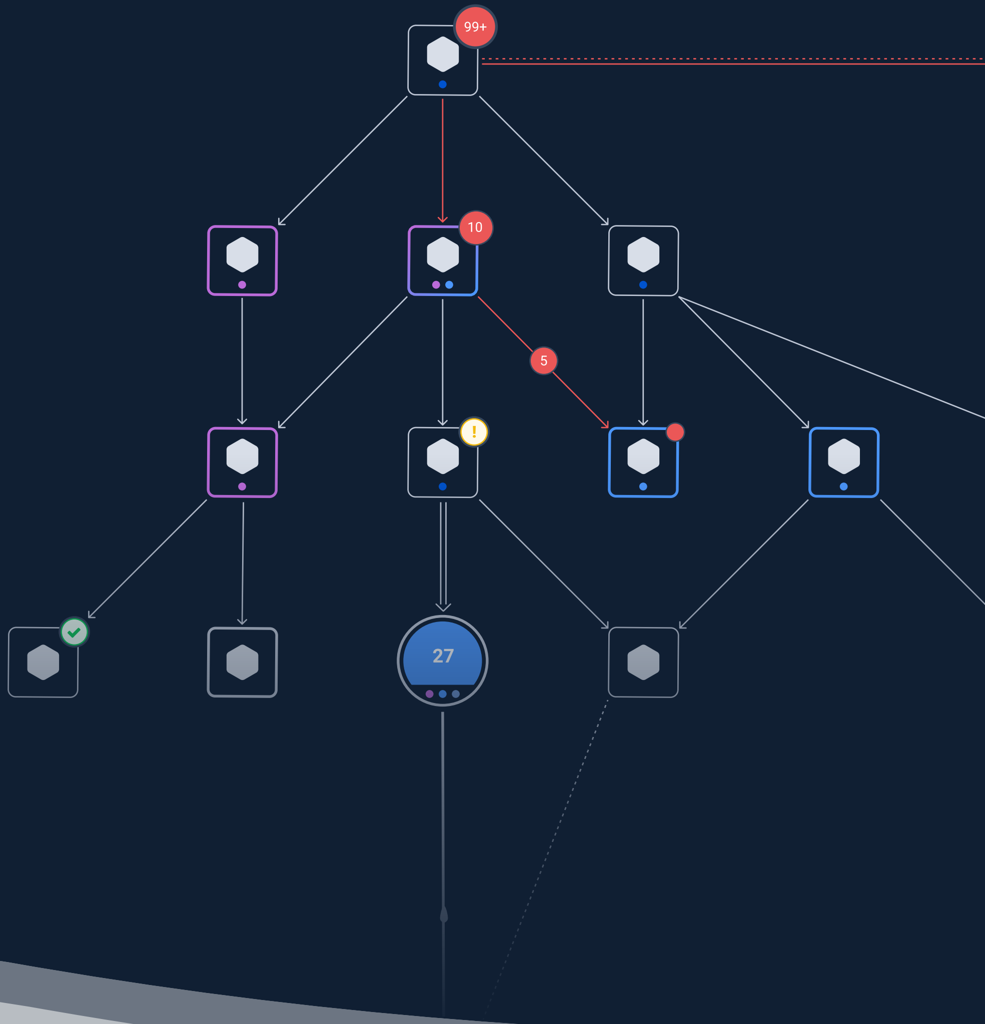


Table of Contents

Introduction

Market Challenges

For many organizations, the ever-increasing complexity of IT landscapes has diminished productivity. As companies move towards more dynamic architectures—like hybrid clouds, containers, and microservices—IT monitoring requirements change drastically. The onslaught of data these new infrastructures create, combined with a real shortage of skilled people and time, makes it increasingly difficult to maintain the pace. In response, many teams use 'observability' to gain some control. This whitepaper defines the conventional view of observability and explains why it's essential, but not enough. It also outlines a simple 3-step approach to dramatically improve observability to prevent outages, crush mean-time-to-repair (MTTR), and maximize productivity.

Observability

Current Market Definition

Current market definitions differ to some extent, but most IT experts concur: observability is about bringing different types of data together on a single pane. But which types of data? Most experts state that observability should correlate three: logs, metrics, and traces.

- *(Event) logs:* An event log is an immutable, time-stamped record of discrete events that happened over time. Event logs, in general, come in three forms but are fundamentally the same: a time-stamp and a payload in some context.

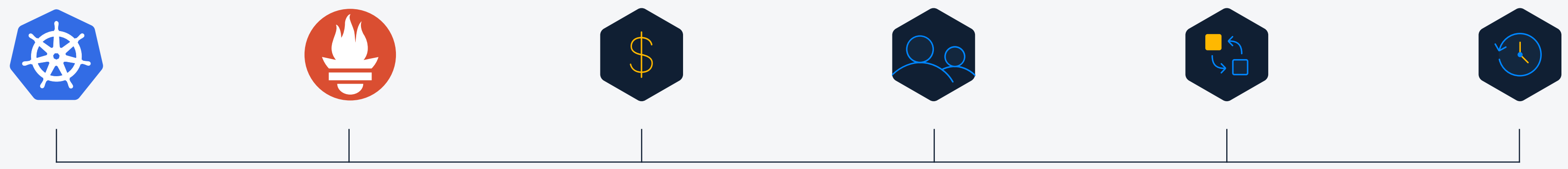
- *Metrics:* Metrics are a numeric representation of data measured over time. Metrics can harness the power of mathematical modeling and prediction to derive knowledge of the system's behavior over intervals in the present and future.

- *Traces:* A trace is a representation of a series of causally related distributed events that encode the end-to-end request flow through a distributed system. Traces are a representation of logs and the data structure looks similar. A single trace can provide visibility into both the path and structure of a request.

Many observability solutions available in today's market monitor logs, metrics, and traces. They ingest these three types of data to provide an overall picture of an IT landscape's health and performance. When an incident strikes, it automatically alerts a team to the problem by sending a notification. Here's an example of the business value of this kind of observability in action:

Example Case

An IT service manager at a major financial institution experiences an outage



Thursday

10:00 pm

The infrastructure team rolls out a new deployment with Kubernetes. The update was successful but wasn't announced across teams.

Friday

08:00 am

The order management team begins reporting problems. They detect a higher than average error rate. Thanks to Prometheus, they see the error is related to the payment service team.

08:15 am

The payment service team begins an investigation. Using its monitoring tool CloudWatch, it sees the time-out but doesn't know why it happened.

09:15 am

A crisis team is formed with people from different teams.

09:30 am

The crisis team relates the problem to the update from last night.

10:00 am

Infrastructure team rolls back the deployment. Everything works fine again.

Observability of logs, metrics, and traces was in place at the bank to assess the performance and health of the IT environment. However, since the cause and the impact needed to be correlated manually with different team members, precious time and resources were wasted.

This outage resulted in:

2 hours of downtime
$85k sales loss
and **-0.3%** net promoter loss

Still Too Siloed

In the past, different data types were kept in separate siloes, which caused extremely long MTTRs when incidents occurred. Removing the silos and bringing metrics, logs, and traces together was the birth, and very definition of, observability. Now, monitoring these three types of data allows you to reduce the damage of a significant outage. However, it’s not enough to cope with the increasingly faster and continuously changing IT landscapes of today.

Migrating workloads to the cloud and implementing continuous delivery to speed up software development is one solution, but it can strain IT teams. The fast, short release cycles and dynamic cloud and container environments make it difficult to keep track of the application and infrastructure landscape. A small change deep down in the infrastructure may seem innocuous but can have a significant impact over time. IT teams must understand how their infrastructure and applications are interrelated and how changes can affect the business.

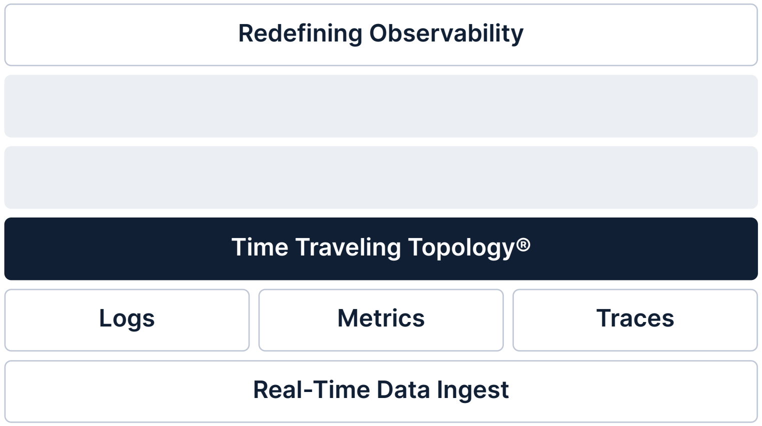
That’s why observability is still too siloed. Traditional observability tools do not integrate existing metrics, logs, and traces stored in other tools. And the landscape is too complex to bring everything into one place, especially for medium and large enterprises.

Redefining Observability

Three essential ingredients need to be added to observability to understand how applications relate to infrastructure: real-time dependencies, changes and artificial intelligence (AI).

Step 1: Adding Real-Time Topology

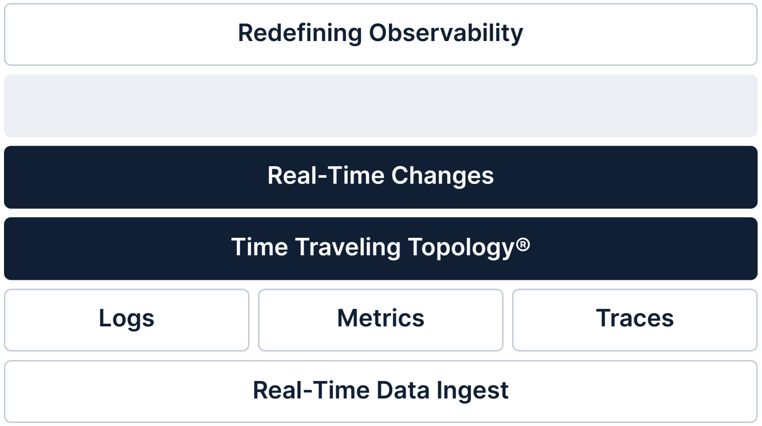
For the metrics, logs, and traces data to be more comprehensible and actionable, a context must be placed around all the data ingested—cloud, automation, service registries, CMDB, virtualization, networking, and deployment tooling. A topological overview that automatically merges the variety of existing data sources in real-time provides the perfect context and improves the quality and accuracy of the data collected.



Step 2: Tracking All Real-Time Changes

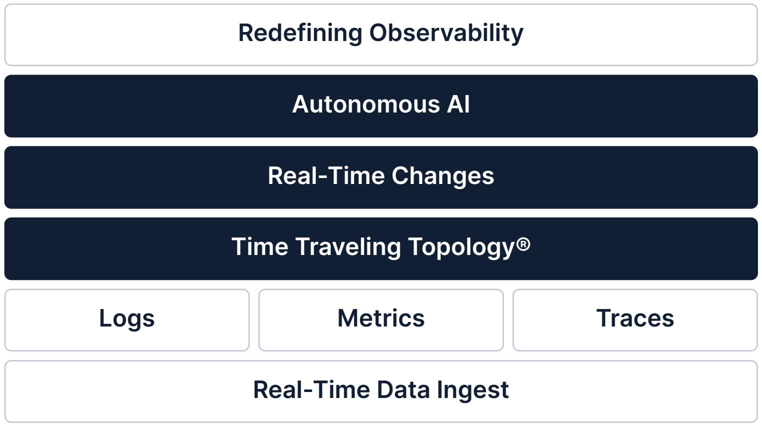
Placing the siloed data into context via real-time topology provides an understanding of real-time dependencies. Now it is essential to observe every change that occurs in the fast-moving IT landscape.

- *Health changes:* monitored IT components that change to a new health state, e.g., from healthy to critical.
- *Version changes:* deployed upgrades of service versions, e.g., upgrading the payment service from 3.0 to 4.0.
- *Topological changes:* New components that appear and disappear in the IT landscape and affect dependencies between existing running components.
- *Component property changes:* changing labels and tags of components, for example.



Step 3: Adding Autonomous AI

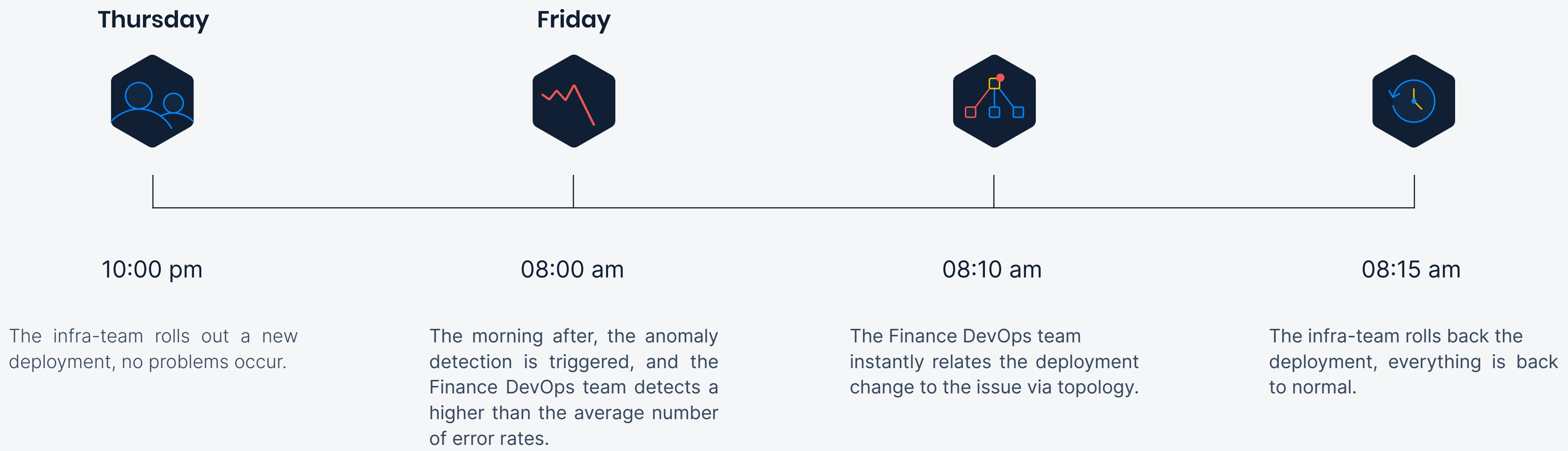
A natural evolution of IT operations analytics (ITOA) is the application of AI and machine learning (ML) techniques. Continuous machine learning allows you to detect anomalous behavior across your environments proactively. Data that is automatically put in context is the best fuel for AI and ML techniques, as the model does not need computation power to cut down the noise; it just knows. Early warning signals get your operations teams out in front of upcoming issues, enabling them to prevent the problems from impacting your business. At the very least, the time gained cuts down remediation time.



Let’s go back to the same example case as mentioned before of the IT service manager at a major financial institution experiencing an outage. But now the 3-steps of real-time topology, real-time changes and autonomous AI are added to the IT environment:

Example Case

An IT Service Manager at a major financial institution experiencing an outage uses the 3-step approach



Thursday

10:00 pm

The infra-team rolls out a new deployment, no problems occur.

Friday

08:00 am

The morning after, the anomaly detection is triggered, and the Finance DevOps team detects a higher than the average number of error rates.

08:10 am

The Finance DevOps team instantly relates the deployment change to the issue via topology.

08:15 am

The infra-team rolls back the deployment, everything is back to normal.

Because the cause and the impact of the outage don’t have to be correlated manually with different team members anymore, precious time and resources are saved.

Compared to the first example this outage resulted in:	**In stead of...**
15 minutes of downtime	~~**2 hours**~~ of downtime
$10k sales loss	~~**$85k**~~ sales loss
-0.08% net promoter loss	~~**-0.3%**~~ net promoter loss

Redefined Observability: Driving Actionable Insights

There is great value in knowing what, when, and why something happened, and its impact on other IT environments. The addition of topology and real-time changes to traditional observability provides instant visibility of the cause and effect of any change or failure across the silos. This knowledge allows you to accelerate your mean-time-to-repair (MTTR), quickly solve costly incidents, and avoid expensive meetings. The 3-step approach outlined in this paper enables:

Real-Time Observability:
Detect and alert incidents across your dynamic environment with automatic checks and self-driving anomaly detection.

Automatic Root Cause Analysis:
Speed up troubleshooting with automated discovery and dependency mapping to pinpoint the root cause across multiple teams.

Business Impact Analysis:
Instantly see the impact of new deployments on dynamic IT landscapes and business and automatically act.

Unified Insights:
Utilize current monitoring and IT investments by streaming your data into one scalable open observability platform.

Summary

The adoption of fast-moving IT landscapes has created new challenges for the market, and observability is the current response. However, limited to logs, metrics, and traces, traditional observability falls short. For one, it’s too siloed. DevOps teams across the organization must understand how their infrastructure and applications interrelate and how changes affect the business.

> ***“StackState makes enterprise-scale observability possible without relying only on a single tool.”***

Lodewijk Bogaards, CTO StackState

A 3-step approach to improve observability adds real-time topology, real-time tracking over time, and artificial intelligence:

1. Real-time topology places your logs, metrics, and traces from different teams in context.

2. Real-time tracking over time shows how deployments and other changes affect the IT environment’s structure and health.

3. AI detects anomalous behavior and can leverage the contextualized metrics, logs, traces, and changes over time.

About StackState

StackState provides real-time observability across all IT components and environments, enabling customers to autonomously detect anomalies, pinpoint the root cause, and assess business implications of new DevOps programs. A recognized “Cool Vendor” by Gartner, StackState has a successful pedigree of providing innovative solutions that reduce MTTR, maximize customer experience, and deliver cost-saving automation. StackState’s platform integrates with all data sources and monitoring tools to unify metrics, traces, logs, and events into a topological dashboard where AI-powered alerts enable operations teams to precisely and efficiently collaborate to resolve incidents.

StackState

Learn more

Unified Observability
for Hybrid IT

StackState HQ
1 Baltimore Place NW, Suite G100 Atlanta,
GA 30308
+31(0) 356 729 068

StackState The Netherlands
Stationsplein 32,
3511 ED, Utrecht, NL
+31(0) 356 729 068