

## FactSet gains developer efficiencies, improves workflows, and reduces MTTR

For nearly 50 years, FactSet has provided the world's investment community with flexible, open data and software solutions to inform crucial financial decisions. By amalgamating data from global markets, public and private companies, and equity and fixed-income portfolios, FactSet aims to empower investor workflows, maximize fintech efficiency, and help financial professionals outperform the competition and the market.

With more 185,000 users worldwide, their solutions span the portfolio lifecycle, from research and portfolio management and analysis, to ESG investing, trade execution, and risk management.

Powered by AI and machine learning, their platform includes an open ecosystem of solutions that scale with third-party vendors and proprietary tools to eliminate silos and solve complex financial challenges. They offer out-of-the-box SaaS solutions, as well as APIs, data feeds, and engines to outfit developer communities.

# FACTSET®

Headquarters: Norwalk, CT

Industry Segment: Financial Technology

Employees: 10,000+

---

### Challenges

- Limited visibility into dependencies
- Manual process for finding and resolving errors
- Incomplete understanding of system health

### Business Results

- Complete system visibility
- Reduced MTTR from days to hours
- Real-time understanding of system health

## The Challenge

Fastset's complex tech estate includes hundreds of services, thousands of instances, and multiple languages. Like many teams managing distributed systems, they had little visibility into their entire system. When issues arose, they found it difficult and time consuming to conduct root cause analysis and to understand how databases, third-party dependencies, and managed services performance affected the application layers and the customer experience.

Without visibility into the dependency chain across the system, MTTR and productivity both suffered, especially if an issue was intermittent. To resolve intermittent issues in the lower-level infrastructure, Ed Thompson, a software engineer, had to manually comb through logs, determine which team was responsible, notify them, explain the issue, and then wait for a response. It was a process that often took several days.

## The Solution

FactSet was able to get complete system visibility—including dependencies— and real-time understanding of the health of their estate. With tracing, they can determine if there's a dependency that's not apparent by executing codes and they can rule out if an issue is hiding in their backend service or if it's down to the most fundamental layer—something that had been obscured previously. "There was an alert that went off for a service with a pool of hosts behind a load balancer. With Cloud Observability, we could immediately see which backend host was causing the issue," said Logan Capaldo, a principal software engineer. They can also see error spikes, even if they couldn't see the errors in a service itself. Errors weren't being logged where people were used to looking, and Cloud Observability helped resolve that.

**“This has helped change our behavior—to look at services before there are performance issues. Cloud Observability makes it easy, in both time and effort, to proactively address outliers and possible issues.”**

**Bhu Boopathy**

Software Engineer

**FACTSET**

## Outcomes

Unsurprisingly, mean time to resolution dropped from several days to roughly two hours and productivity is on the rise. With performance-based alerts, and availability metrics, the team is able to see in real-time the state of the system, the criticality of issues, and related dependencies. “We know, automatically, what impacts a service’s performance during and after deployment and can surface why it happened. We can also quickly identify a specific regression and share it with the team responsible,” said Capaldo. “It’s already improving how we deploy to production.”