



APM is dead?

Observability is the new norm

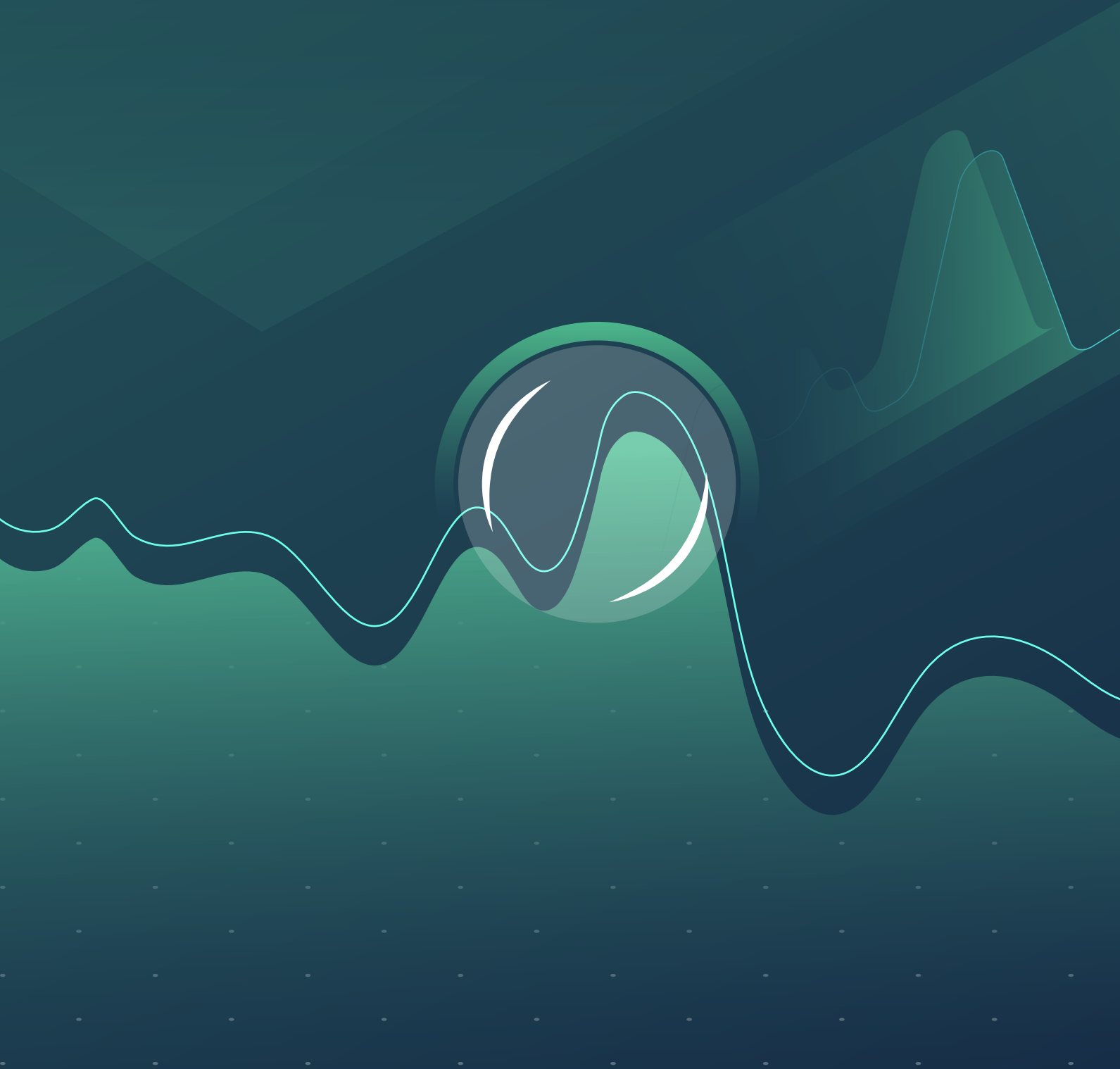


Table of contents



- 1 APM is dead, and Observability is the new norm02
- 2 What is APM?03
- 3 APM history and key challenges with APM04
- 4 What is Observability, and how this term arrived?07
- 5 Does Observability replace APM?08
- 6 Why did businesses think more about Observability?09
- 7 Should you migrate from APM to Observability? and why?10
- 8 How to choose the right Observability platform?11
- 9 How does Middleware Observability work?13
 - What differentiates Middleware from other legacy APM providers ?14
- 10 Conclusion15

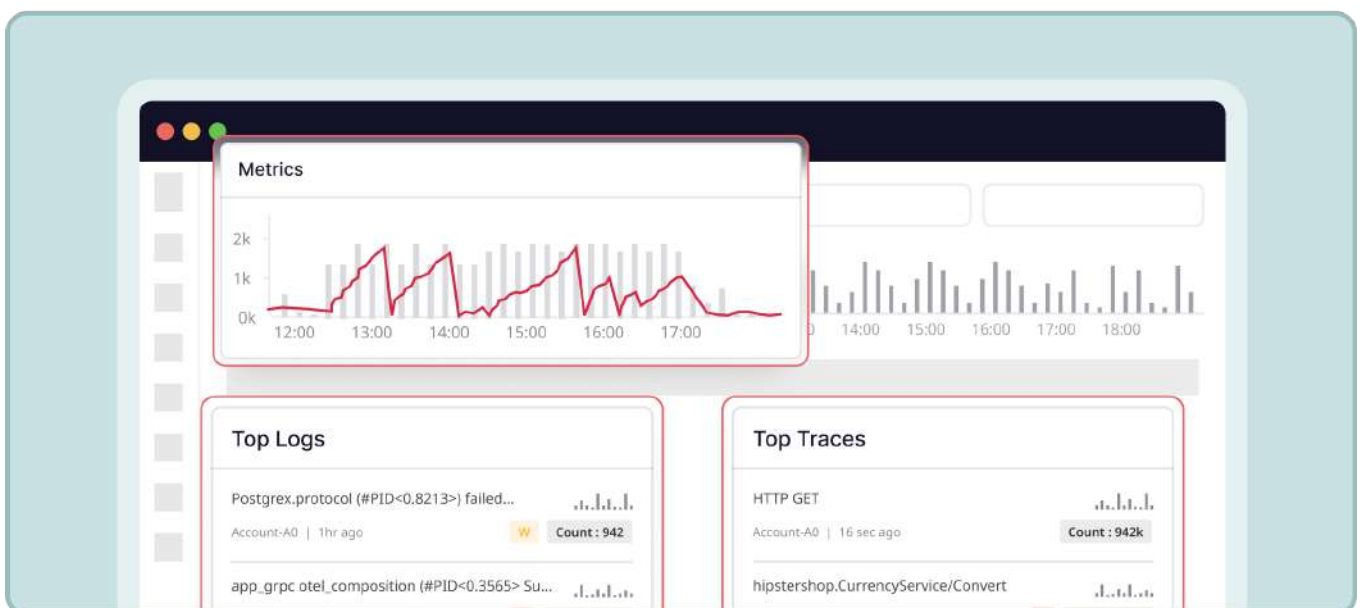
1 APM is dead, and Observability is the new norm

Organizations rely on complex distributed systems to power operations in today's digital age. Hence, the systems have to function at peak performance for optimum profit and productivity, but ensuring this has become more challenging than ever. Traditional Application Performance Management (APM) tools have long been used to monitor and manage the performance of applications. APM tools collect application performance metrics such as response time, throughput, and error rates. They are effective in monitoring and managing traditional monolithic applications.

However, because of the rampant complexity of modern systems, APM tools have been sidelined for Observability—a more holistic approach to application performance management. Observability transcends the collection of application performance metrics and involves collecting and analyzing various types of data, such as logs, traces, and events.

In this paper, we explore the limitations of traditional APM tools and the benefits of adopting the Observability approach. We also discuss how Observability can help organizations achieve their business and performance goals.

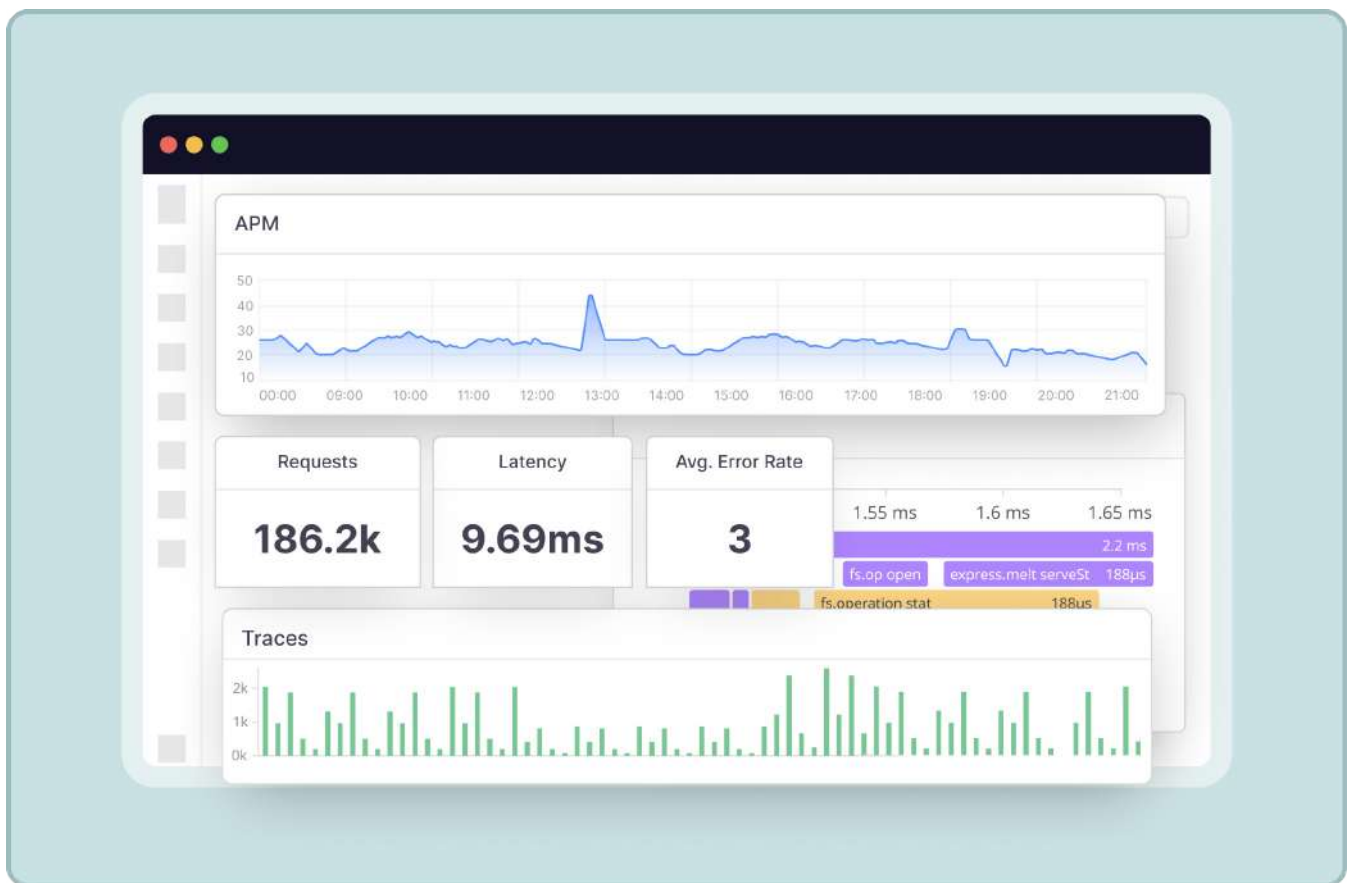
“The complexity of modern systems has led to a shift away from APM tools and towards observability, which provides a more holistic view of application performance”



2 What is APM?

Application performance management (APM) involves activities—monitoring application performance, identifying and diagnosing performance issues, optimizing application performance—involved in tracking and optimizing applications' performance and health.

APM tools use metrics like response time, throughput, errors, and resource utilization to gauge the health of your applications. They also offer transaction tracing, code-level diagnostics, and alerting features.



3 APM history and key challenges with APM

APM emerged in the late 1990s as organizations began to rely more heavily on web-based applications. At that time, it was focused on monitoring the performance of web applications and ensuring optimum end-user experience. But in 2008, when applications became more complex and distributed, APM solutions evolved and offered more advanced monitoring capabilities, such as deep code-level tracing, to enable the identification and troubleshooting of performance issues across multiple tiers of the application stack.

Generally, APM

- Gave end-to-end visibility into the performance of the entire application stack—including dependencies and infrastructure.
- Helped IT teams manage time for high-priority functions, enabling them to work more efficiently
- Aided in the optimization of resource utilization, reducing infrastructure and compute costs.

“In 2010 when cloud computing became a buzzword, APM’s functions were downsized because it could not match the complexities of microservices, multi-cloud containers, and serverless computing. Some key challenges that this era posed to APM are”





Some key challenges that this era posed to APM are

⬡ Complexities of root cause analysis

Cloud-based applications easily integrate with many other applications, which leads to dependencies and interdependencies. While APM can give insight into an application's performance, it cannot dissect dependency-level issues or uncover malignancies at the infra level. APM is, therefore, not a comprehensive solution in a multi-cloud environment. When issues arise from the dependent components and infrastructures such as virtual machines, hosts or storage. APM will only alert about the issues but won't be able to give reasons for the issue or where the issue arose from.

⬡ Inability to handle massive data

Containerized applications' data are often broken into smaller units, leading to a stream of information to deal with. In this situation, identifying issues gets tricky as IT teams get flooded with massive datasets caused by several incidents. APM tools cannot process these vast amounts of data.

⬡ The speed of modern application environment

The APM tools cannot catch up with the speed of the modern application environment, which is characterized by continuous integration and continuous deployment (CI/CD). There is no static reference point for monitoring as the environment deploys rapidly. Before the IT teams could implement the resolution of problems, the application environment might have changed.

⬡ Vendor-locking of APM tools

Due to numerous independent components, modern application environments often require a number of monitoring tools. These tools, however, come with vendor lock-in. So, when your application grows and you need to scale, you must pay extra. If you can't pay, you must use the tools with their restrictions. This fragments the tools for monitoring the entire application stack, presenting the licensing challenge.

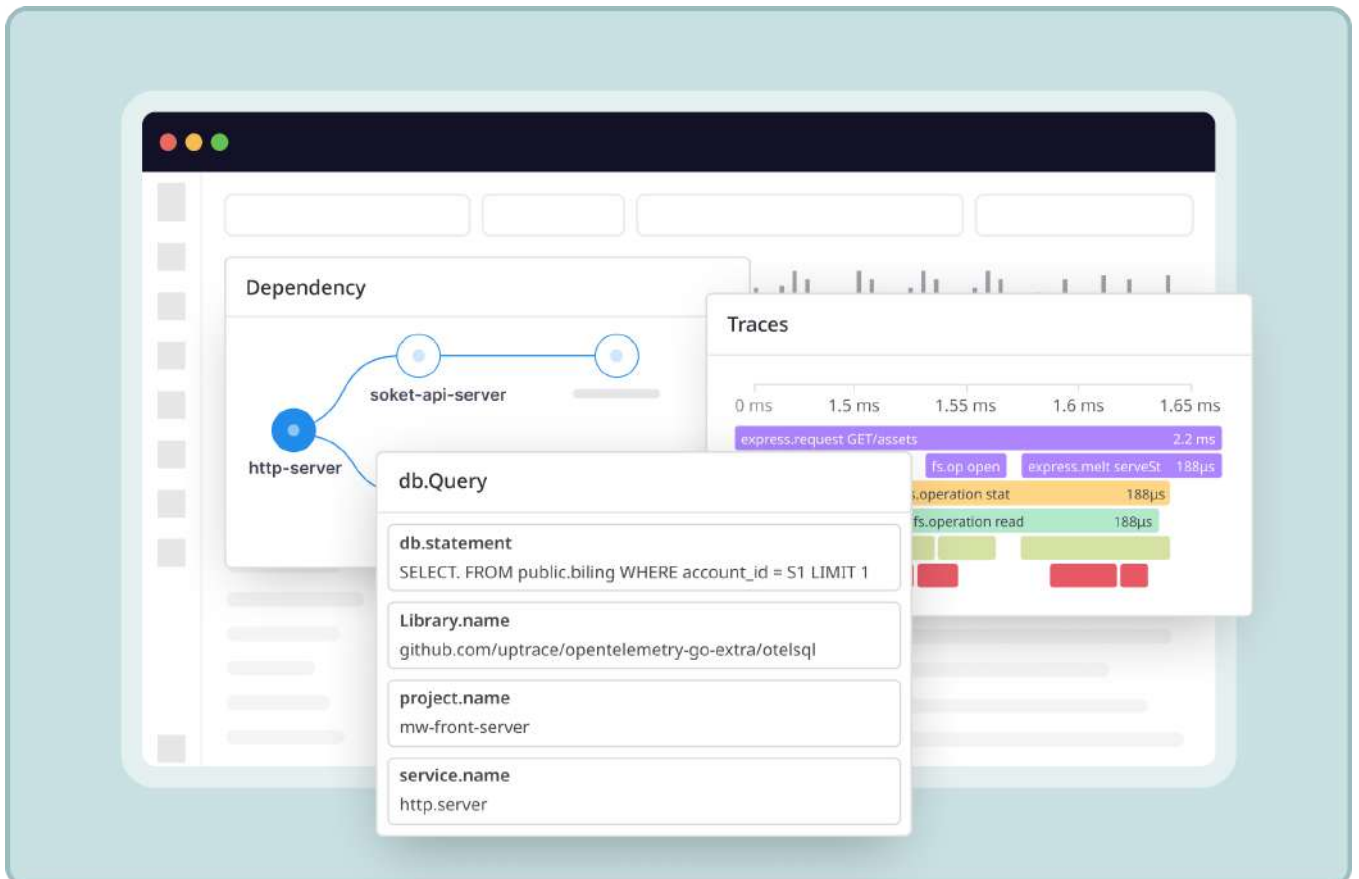
The traditional license cost of an APM solution has been in the area of \$2,000 per host per year, making it difficult for enterprises to deploy APM solutions on **every application** that is critical to their Digital Transformation initiatives. The sheer number of services and hosts also makes it operationally difficult to keep agents up to date or even roll out the necessary monitoring to deliver the value proposition of APM.

🛡️ Security risks during increased traffic

Traditional Application Performance Management tools cannot differentiate between traffic that rises due to seasonal festivals or Distributed Denial of Service (DDoS) attacks. This gap won't enable APM to discover threats or suspicious activities within the application. Thereby opening the application environment to cybersecurity concerns.

The afore-mentioned challenges make it very difficult to adopt APM. But leaving applications unmonitored is disastrous and counterproductive. What, then, is the way out?

Here comes Observability.



4 What is Observability?

Observability refers to the ability to understand the internal state of a complex system based on its external outputs. It is the ability to observe and analyze the behavior of a system and determine its internal state, performance, and health. This includes monitoring performance metrics and collecting and analyzing other types of data, such as logs, traces, and events. Observability is achieved using monitoring tools and techniques like logging, tracing, and metrics collection.

The term "observability" originates from the field of control theory, which is the study of how to control a system to achieve a desired behavior. Rudolf Emil Kalman coined the term in the 1960s. Although "observability" was in routine use amongst engineers working in process and aerospace industries, the term did not enter the lexicon of IT practitioners until some 30 years afterward.

Today, observability is a key concept in many fields, especially in software engineering, where 90% of the 1300 participants acknowledged that observability is critical to the success of their business and expect it to become the most important pillar of enterprise IT. It is 2023, and observability has become necessary for all and sundry in the software engineering industry.

"Observability becomes necessary for all and sundry in the software engineering industry in 2023."



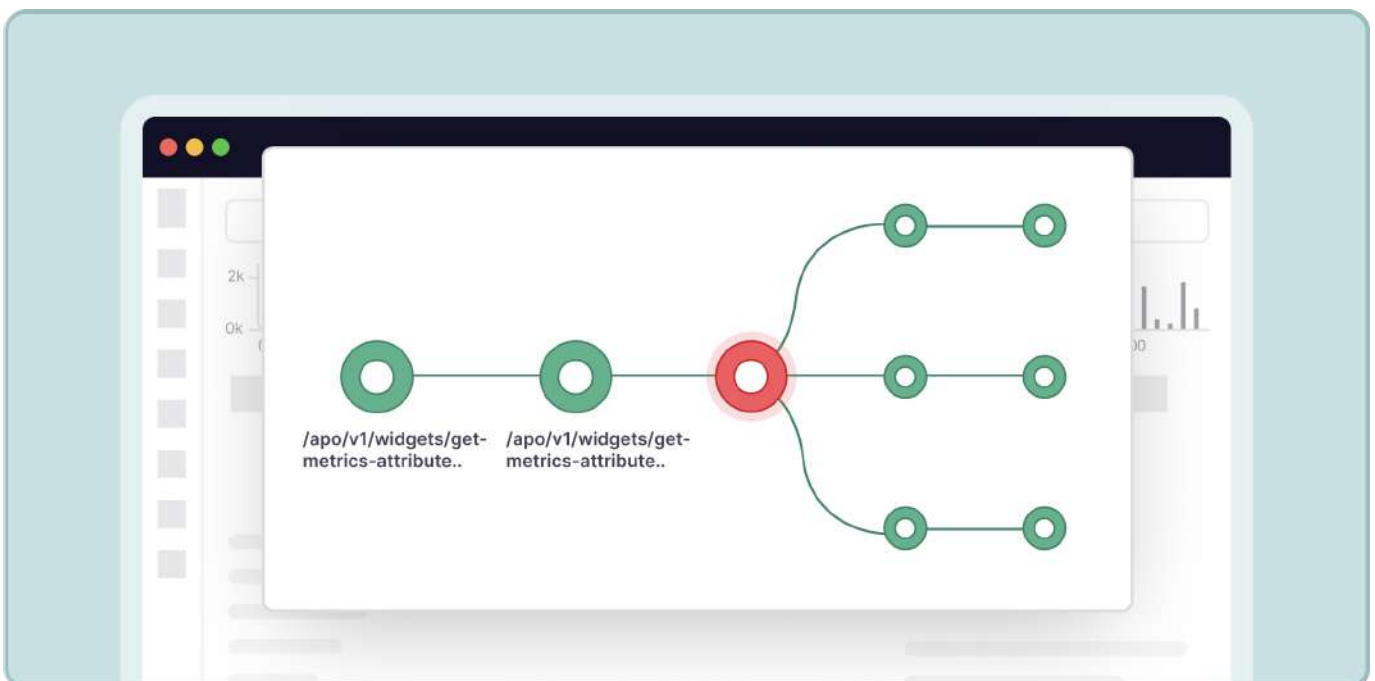
5 Does Observability replace APM?

Due to its sophisticated approach to performance management, it is wise to believe that Observability has replaced APM. It evolved from APM and expanded its capabilities. While APM is primarily concerned with monitoring and managing application performance, Observability focuses on understanding complex systems' overall behavior.

“Due to its sophisticated approach to performance management, it is wise to believe that Observability has replaced APM.”

Observability provides additional benefits beyond what APM can offer. For example, observability can help in debugging issues unrelated to performance, such as data consistency or security. It can also enable more advanced analysis techniques like anomaly detection or predictive analytics.

Overall, while APM remains an important tool for monitoring the performance of individual applications, observability has emerged as a more powerful and comprehensive approach to understanding complex systems. This makes it a more powerful solution for performance management in the future.



6 Why did businesses think more about Observability?

Asides from the fact that **Observability is more advanced than traditional APM**, some other reasons why businesses gravitated toward the concept are:

Holistic

A holistic tool can correlate the interactions and relationships between various components. It works on massive data collected from multiple sources, giving organizations root cause analysis of events. Sampling and statistical rollups are simply not good enough anymore. Observability ensures that the collected data is correct regarding the interaction, transaction or unit of work for which the data is collected.

Speed

An Observability tool could match the speed of today's modernized application and give a real-time and point-in-time metrics analysis. It can give granular details of code changes, thereby giving IT teams informed decisions and helping them improve applications' performance.

Comprehensive

Observability tools offer comprehensive monitoring and management of applications, collecting every relevant data in the cloud environment. It covers every facet of computing, including deployment, measurement and optimization, capturing metrics, logs, traces, dependencies, relationships and configuration state over time.

Automation

There is simply no time to run installation scripts to install monitoring agents manually. It is also difficult for support engineers to keep track of what is running in each container and to install the appropriate agents. Observability offers fully automated instrumentation that senses what is in the container and automatically injects the correct instrumentation as the container is launched.

7 Should you migrate from APM to Observability? and why?

Since traditional APM tools lack full-stack visibility into the state of the application, it is relevant to migrate to observability for an up-to-date monitoring service. Observability encompasses the functionalities of APM by giving IT teams visibility into the infrastructure, application, and whole development pipeline.

“Observability provides a holistic view of a system, allowing teams to quickly identify and resolve issues and gain insights into how the system behaves.”

However, if you are primarily concerned with monitoring the performance of a single application, then APM may be sufficient for your needs. But observability is key if you are looking for a complete understanding of your system's behavior!

Efficient observability requires two things:



1. A mindset and approach shift.



2. A good observability tool/platform.

8 How to choose the right Observability platform?

Following **best practices** when implementing observability is the holy grail to end-to-end visibility and monitoring of your application. Consider these factors when choosing an observability platform:

Automation

A good observability tool should automate every aspect of tracking and response. The tool should be able to automatically calculate the relationships and dependencies for each unit of the application stack.

Intelligence-based performance

Many technology tools are adopting Artificial intelligence to perform their various functions. A good Observability tool should be **AI-enabled** to provide intelligent insights that are adaptive and self-modifying.

Ease of use

The observability tool should be flexible and accessible. It should be easy to understand and use and provide IT teams with real-time and accurate information with its user-friendly interface.

Context

The insights given by a good observability tool should be contextual and clear. It should give crystal details of the relationship between various components of the application without compromising relevance and accuracy.

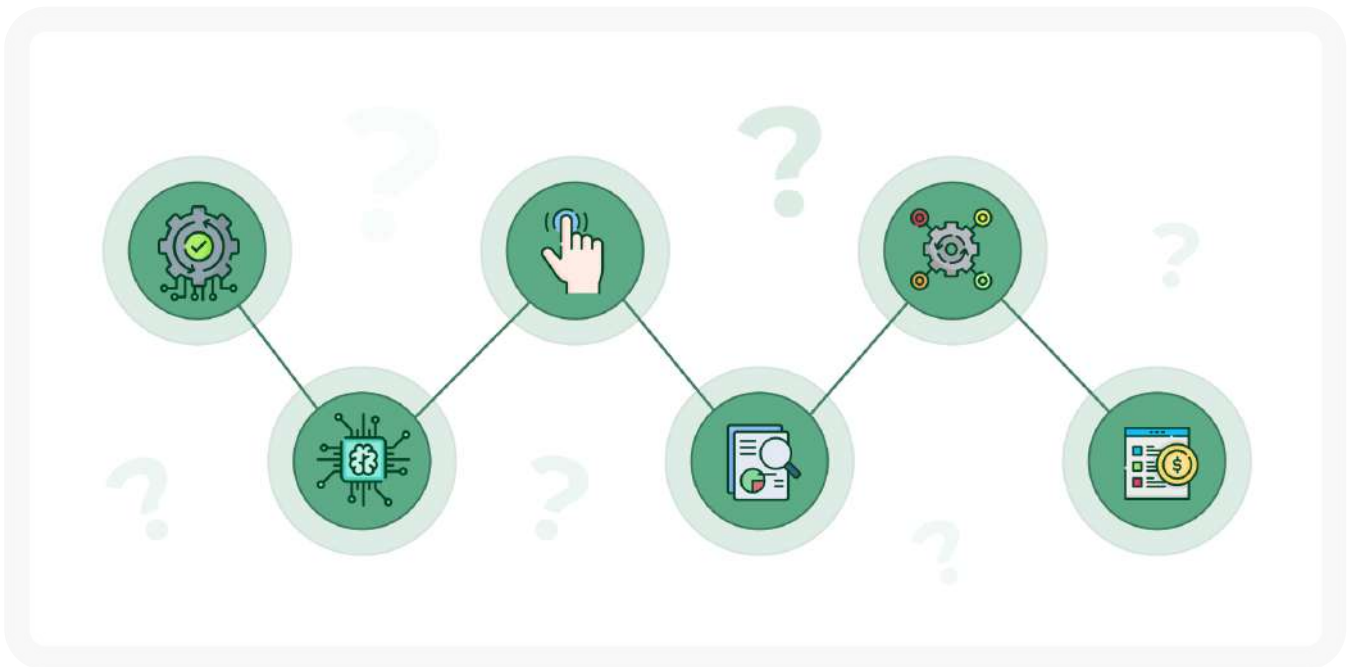
To enable this, the tool must use open instrumentation agents to gather telemetry data—metrics, events, logs, and traces—from open-source or vendor-specific entities—like services, hosts, applications, and containers.

📍 Action-oriented

A good observability tool must—alongside visualizing—help in debugging and improving your entire stack to optimize end users' experience. It should be able to analyze enormous amounts of raw telemetry data collected for correlations and context in high cardinality.

📍 Price

As it monitors and collects relevant OTel metrics, an efficient observability tool must also be cost-effective. It should operate a pay-as-you-go pricing model, conferring a gradual increase onto your observability expenditures as your infrastructure needs scale.



9 How does Middleware Observability work?

Middleware APM solution combines the functions of traditional APM with the depth of observability to meet the need of modernized applications. Synergizing observability tactics with APM tools, Middleware can move smoothly from one use case to the other and support all the different data types to provide input and analyze the full range of information available through observability.

A cloud-native enabled platform, Middleware uses Open telemetry lib to collect traces from the application. As a first step, the user needs to install the Middleware agent, which supports Node.js, Go, Python, and Java language. Users can use the installation script per their platform and install the agent.

“Middleware uses Open telemetry lib to collect traces from the application.”

Once the agent is installed, MiddleWare fetches traces from the application using open telemetry lib. On every trace generated in the user’s application, open telemetry sends those traces to Middleware. Middleware collects the raw data from that, processes the raw data, and converts the data into visualization format.

Middleware display services listing, trace listing, dependency graph, profiling, error logs, debug logs all these data (which is processed from the raw data collected by the agent).

To get more specific error-related data, users can run this command on their installation script:

```
{
  tracker.info('Info sample');
  tracker.warn('Warning sample');
  tracker.debug('Debugging Sample');
  tracker.error('Error Sample');
}
```

More details and customization for error tracking can be found in the help documentation.

» What differentiates Middleware from other legacy APM providers?

In the past, APM solutions were based only on service monitoring. However, Middleware provides APM with distributed traces, metrics and logs that users can relate to. MiddleWare stands out by the following qualities:

- 1 Easy to Install and operate:**
Middleware also offers a user-friendly interface so that non-expert IT teams can easily operate and interpret insights.
- 2 Middleware monitors what's important**
for your customers. It monitors users' response times as they interact with the application, enabling IT teams to predict customer behaviors and assess the business effect of a particular issue.
- 3 MiddleWare gives dynamic, AI-powered alerts and insights,**
which makes it self-adaptive and more intelligent than de facto observability tools. You can set multiple parameters and their thresholds and get alerts when the thresholds are passed.
- 4 MiddleWare offers distributed tracing**
to support the monitoring of complex systems. It is an ideal tool for transaction-oriented hybrid cloud environments where transactions must be traced through multiple layers.
- 5 It offers a unified timeline view,**
and its dashboard gives end-to-end visibility into the performance of your application and the state of its infrastructure. Unification of observability helps save time and money, and **83% of organizations** attested to that.





10 Conclusion

The rapid evolution of software architectures and the accompanying complexity have rendered traditional APM tools inadequate for providing comprehensive insights into system behavior. Observability shifted the focus from reactive monitoring to proactive detection and resolution, providing a holistic view of system behavior and enabling teams to identify and resolve issues more quickly and effectively.

As organizations continue to adopt cloud-native architectures and distributed systems, the need for observability will only continue to grow, and those who fail to embrace it risk being left behind in a world where software reliability and performance are critical to business success.

To sum up, APM may not necessarily be dead, but it has certainly been surpassed by observability in modern system monitoring and management. Middleware APM combines both the functionalities of traditional APM and today's observability for granular and comprehensive observability of your application.

Request a demo
or try **Middleware** out today.

[Book a Demo](#)