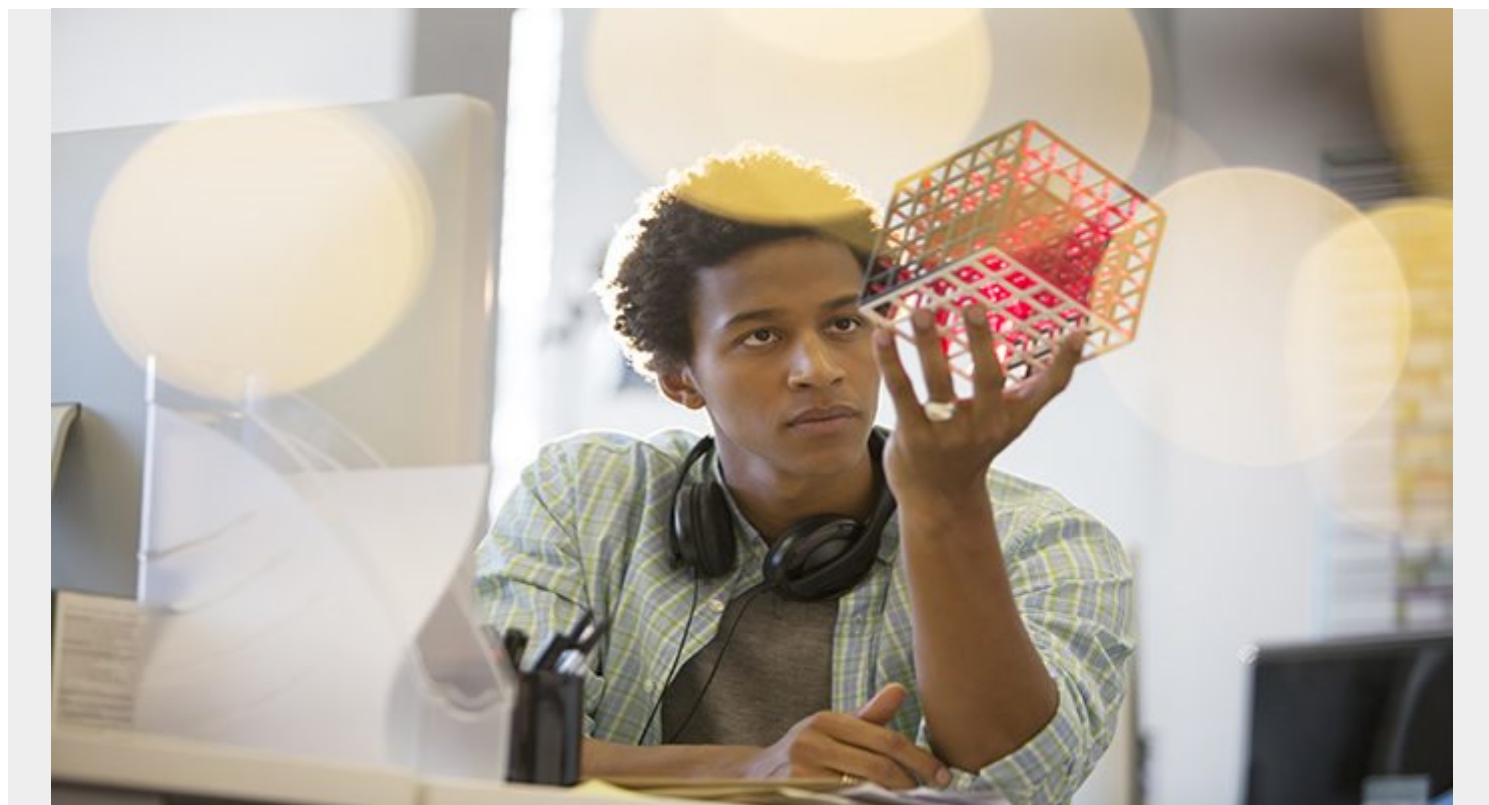


OBSERVABILITY VS MONITORING: WHAT'S THE DIFFERENCE?



To aid with our understanding of Observability vs Monitoring let's look at the evolution of the Enterprise IT world. Enterprise IT, application and business service development are increasingly complex. The interdependencies within the underlying architecture has become more fragmented resulting in difficulty visualizing the full IT Stack.

The internet delivers IT infrastructure services from hyperscale data centers at distant geographic locations. Companies are moving towards cloud-native delivery, resulting in modern distributed applications creating a perfect storm of complexity with constantly emerging technologies, hybrid-cloud infrastructures, and businesses expecting delivery of more features faster.

Companies are consuming these services – like microservices and containers - as distributed functions across layers of infrastructure and platform services. Consumers expect regular, continuous feature improvements through new releases.

To meet these requirements, IT service providers and enterprises must aggressively manage business service performance, improve stability, and predict & prevent performance degradation and outages—all in the context of the rapidly changing and evolving IT landscape. This requires closely observing and monitoring metrics and datasets related to service performance to optimize system availability, particularly during upgrades and code launches.

Observability seems like the hot new topic in the IT world, but the reality is it has been with us for a long time. Only recently, however, has it entered the IT realm, combining with monitoring to offer a more powerful approach to business service performance management. System observability and

monitoring play critical roles in achieving system dependability — they may be interdependent but they're not the same thing. Let's understand the differences between monitoring and observability, and how they are both critical for enhanced end to end visibility.

What is monitoring?

In Enterprise IT, monitoring is the process of instrumenting specific components of infrastructure and applications to collect data – usually metrics, events, logs, and traces – and interpreting that data against thresholds, known patterns, and error conditions to turn the data into meaningful and actionable insights.

Monitoring is focused on the external behavior of a system, specifically those data targeted for collection. Monitoring is most effective in relatively stable environments, where key performance data and normal vs abnormal behavior is known. When enterprise IT was predominantly run in an organization's own data center, monitoring was an appropriate way to approach managing the environment.

The introduction of public and private clouds, the adoption of DevOps, the emergence of new technologies and the massive scale of data brought on by digital transformation, the proliferation of mobile devices and IoT has created a situation where monitoring is no longer an effective approach for IT Operations.

What is observability?

The concept of Observability was introduced by R. Kalman in 1960 in the context of control systems theory. In control systems theory, observability is a measure of how well internal states of a system can be inferred from knowledge of its external outputs. In essence, it's a method for learning about what you don't know from what you do know. The relationship between the known and the unknown can be represented mathematically.

So given enough known, external data and time to do the mathematical calculations, the internal, unknown state of the system can be determined. This approach is well suited for modern, Enterprise IT, as distributed infrastructure components operate through multiple abstraction layers. This makes it impractical and challenging to understand the health of complex services by selecting specific components to instrument for telemetry and looking for threshold breaches, events, etc.

The challenge to implementing observability in IT has been the volume, variety, and velocity of external data, combined with having the computational power and domain knowledge needed to analyze and make sense of it in real-time. Effective IT Operations teams now need observability platforms that can consume vast quantities of data from a variety of sources and submit that data to immediate intensive computational analysis. Fortunately, such platforms, like BMC Helix Operations Management, are now available.

Comparing Observability and Monitoring

For simple systems, traditional monitoring is effective and can provide some measure of insight into a system's health. Consider a single server machine. It can be easily monitored using metrics and parameters such as hardware energy consumption, temperature, data transfer rates, and processing speed. These parameters are known to be highly correlated with the health of internal system

components.

Now consider a large, complex, business service. It is made up of multiple applications that span public and private clouds, a diversity of distributed infrastructure, and maybe even a mainframe. There are too many systems, some not directly accessible, that if monitored without knowledge of the key performance data, systems and error conditions, will generate too much uncontextualized data and, in turn, unnecessary alerts, data, and false flags.

In the second case, an observability and AIOps approach is needed. Rather than selecting the data to monitor and examine the behavior of that data relative to trends, known errors, etc., all available data from all systems should be consumed. Aggregated into a high-performance data store, it should be combined with a comprehensive topology of all assets, systems, and applications that builds a comprehensive model of relationships and dependencies.

On this foundational observability layer, high-performance, domain-informed AI and ML algorithms can be applied to determine which externally observable data are correlated with which services and infer the health of those services from their behavior. This is the power of an observability and AIOps approach, such as that used by BMC Helix Operations Management.

Coda: Observability in DevOps

The concept of observability is prominent in [DevOps software development lifecycle \(SDLC\) methodologies](#). In earlier waterfall and agile frameworks, developers built new features and product lines while separate testing and operations teams tested for software dependability. This siloed approach meant that infrastructure operations and monitoring activities were beyond development's scope. Projects were developed for success and not for failure: debuggability of the code was rarely a primary consideration. Infrastructure dependencies and application semantics were not adequately understood by the developers. Therefore, apps and services were built with low inherent dependability. Monitoring failed to yield sufficient information about the known-unknowns, let alone the unknown-unknowns, of distributed infrastructure systems.

The prevalence of DevOps has transformed [SDLC](#). Monitoring goals are no longer limited to collecting and processing log data, metrics, and distributed event traces; monitoring is now used to make the system more observable. The scope of observability therefore encompasses the development segment and is facilitated by people, processes, and technologies operating across the SDLC pipeline.

Collaboration among cross-functional Devs, [ITOps](#), Site Reliability Engineers (SRE) and QA personnel is critical when designing a highly performant and resilient system. Communication and feedback between developers and operations teams is necessary to achieve observability targets of the system that will help QA yield correct and insightful monitoring during the testing phase. As a result, DevOps teams can test systems and solutions for true real-world performance. Continuous iteration based on performance feedback can further enhance the ability to identify potential issues in the systems before the impact reaches end-users.

Observability offers actionable intelligence for optimizing performance, giving DevOps, SREs, and IT Operations increased agility by staying ahead of any potential service degradation or outages. Observability is not limited to technologies but also covers the approach, organizational culture, and priorities in reaching appropriate observability targets, and hence, value of monitoring initiatives.