

# OBSERVABILITY VS EVENT MANAGEMENT: WHAT'S THE DIFFERENCE?



When it comes to alerts and alarms, there is only one [movie](#) scene that comes straight to mind: “Houston, we have a problem!” The explosion of one oxygen tank on the Apollo 13 lunar mission's set off a colorful array of buttons on the rocket as well as at the mission control, leading to hectic but heroic efforts to bring the astronauts back to earth.

Similar scenes are replayed daily in IT departments globally as [monitoring systems](#) bring notice of failed components or degraded performance. That's why having an approach to detect such events and alerts—and respond appropriately—is a [critical capability](#) for any service management organization.

While event management has been a mainstay of service management for years, observability has recently risen to prominence as the go to approach for modern tech organizations. But are they really very different? This article breaks down the two approaches.

## Observability basics

The term observability is not new but was first established under control systems engineering. Here, observability is defined as the ability to measure the internal states of a system by examining its [outputs](#). When trying to understand observability, we look at it as a characteristic rather than an activity. In other words, the more observable a system is, the better placed we are to pinpoint the reason for things going wrong.

Where monitoring refers to that activity of looking at the change of state in a system to determine if it is working well or not, observability is more about the actual capability of the system and its management to effectively convey the reason for the change of state. So, we can actually look at observability from two perspectives:

- The design of the system
- The capability of monitoring tools

(Compare [observability & monitoring](#).)

## The design of the system

The [Google Cloud Architecture Guides](#) on DevOps indicate that systems are instrumented with code or components that expose the inner state. A well instrumented system will aid observability as the system itself provides quality outputs that reveal its true internal state.

For instance, during development or deployment, code is added to the software system to keep track of connection pool information such as unused connections, failed connections etc., which can be exposed through:

- Observability & event management (OEM) tools
- Scripts
- A third-party monitoring solution

## The capability of monitoring tools

Observability has been primarily marketed (rather 'overhyped') as a hallmark of modern monitoring tools, particularly [application performance monitoring](#) (APM) solutions. These solutions include features that can collect, analyze, and correlate external state data from a variety of [telemetry sources](#) such as logs, metrics, distributed traces, and user sessions.

The use of such tools, leveraging [artificial intelligence](#) (AI) and cloud centric capabilities such as [CI/CD](#), provide the means of keeping up with visibility requirements for the ever-evolving landscape of modern technology systems like:

- Microservices
- Containers
- Serverless functions

## Observability in ITSM

In the world of technology service management, observability is a critical differentiator in faster detection and resolution of [incidents](#) and problems that would plague our applications or infrastructure, resulting in bad customer experience as well as lost business outcomes.

According to [Ubuntu](#), the degree of observability in a system depends on the quality of telemetry information collected and the way it is processed, which enables one to know and investigate in a timely fashion how the system is performing, what issues are occurring and what their impact is.

This is especially important when trying to address service quality, especially when one considers benefits such as reduction in [Mean Time to Restore Service](#) which is a key customer experience

indicator.

## Event management overview

Event management is the practice that acts on monitored changes of state of services and their associated components, by determining their significance, and identifying and initiating the correct response to them.

(Read our [event management explainer](#).)

According to the ITIL® 4 practice guide related to this topic, information about events is also recorded, stored, and provided to relevant parties. Events materialize when a set threshold is passed (could be a warning or an exception) which triggers a pre-defined response such as:

- Creating an alert or other notification
- Creating an incident
- Changing a status of a previously recorded alert or notification
- Initiating a reactive action towards the respective component or service

From a process perspective, the event handling process relies on inputs from system notifications and monitoring tools outputs which are then taken through the following activities as guided by a monitoring plan:

- Event detection
- Event logging (for significant events)
- Event filtering and correlation check (might be iterative)
- Event classification (critical, major, medium, minor)
- Event response selected
- Notifications sent, response procedure carried out

These activities can be manual or automated depending on the service provider organization's capabilities, and result in appropriate responses including event analysis, incident management and stakeholder engagement. It is clear that event management is not simply the action of responding to system alerts, but rather an all-encompassing capability that requires people (roles), information and technology, processes and where required partners and suppliers for success.

(Learn about the [people, process, technology & partners paradigm](#).)

## Drawing the line between observability & event management

The evolution from ITIL v3 to ITIL 4 saw a change of name for this key practice (previously process) from "Event Management" to "Monitoring and Event Management". The rationale behind this decision was informed by the fact that monitoring is a trigger for event management, but not all monitoring results in the detection of an event.

So, can we say that observability is only related to monitoring? Not quite, as can be seen that the value of observability spans across the design and development lifecycle of systems.

Some of the benefits of observability, as identified by [IBM](#), include having systems that are easier to understand, monitor, update, and repair, leading to higher quality and ultimately meeting business and customer needs. But from the understanding of the activities of event management, it is obvious

that the value of observability can only be fully achieved when a mature and improving event management practice is in place.

## Related reading

- [BMC Service Management Blog](#)
- [BMC IT Operations Blog](#)
- [BMC DevOps Bog](#)
- [What is Security Information and Event Management \(SIEM\)?](#)
- Risk Management Practices in ITIL® 4 Environments
- [Monitoring Microservices with Spring Boot Actuator & AspectJ](#)