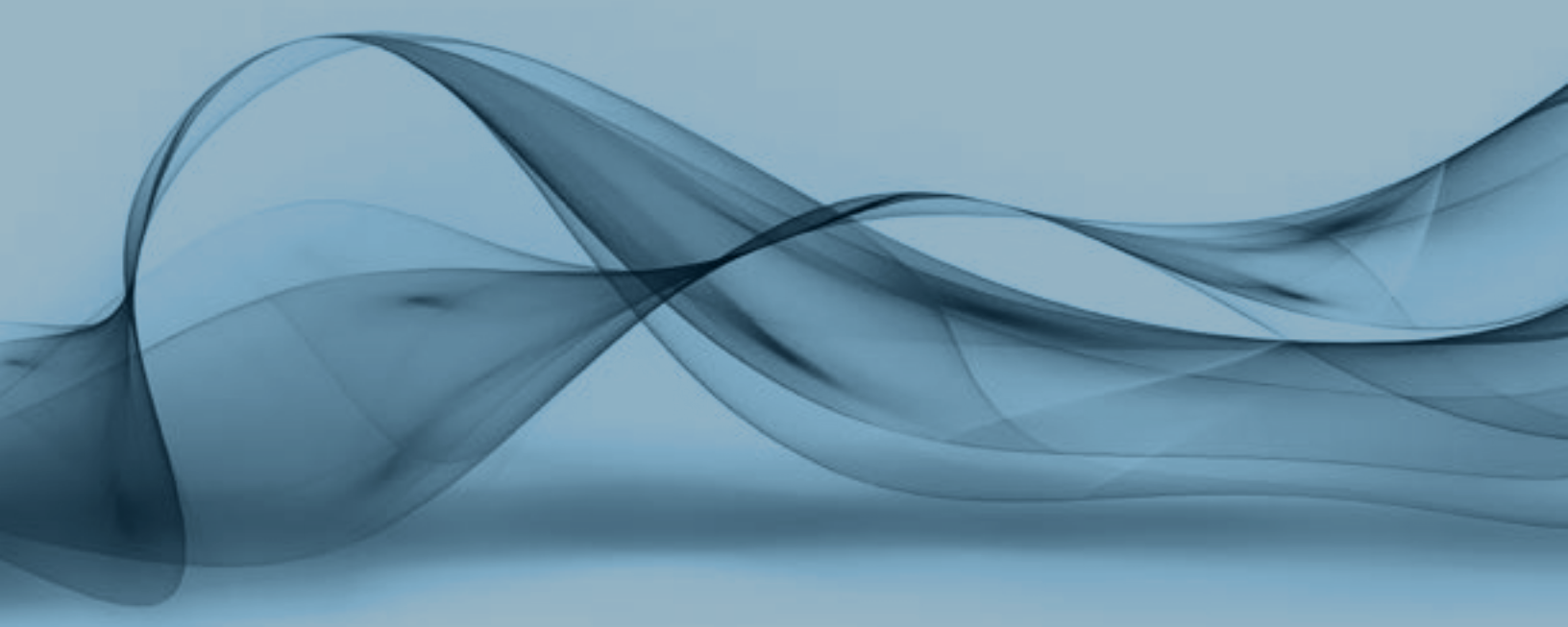
Cribl

WHITE PAPER

What is Observability?



WHITE PAPER

What Is Observability?

Observability allows you to understand the behavior of applications and infrastructure from the data they produce.

THE CHALLENGE

Monitoring helps teams answer known questions about their infrastructure and environment, but it doesn't give insight into new signals or opportunities.

THE SOLUTION

On the other hand, observability allows staff to understand new or unexpected behaviors in applications and infrastructure from the data they produce.

THE BENEFITS

- Get insights into application and infrastructure behaviors you didn't know existed
- Discover unknown events and develop a plan of action
- Complement your existing monitoring practices for full insight into your environment

Over the last two years, the increasing complexity of modern distributed systems and application architectures has highlighted the limits of legacy monitoring approaches. Legacy monitoring remains fixated on collecting and reporting errors, restricting its effectiveness in today's dynamic and ephemeral environments. Observability takes a new approach, allowing teams to interrogate system behavior without the limits imposed by legacy methods and products.

Defining Observability

There are as many definitions of observability as there are vendors in the market today. The definition we like at Cribl comes from Gartner:

> *Observability is the characteristic of software and systems that allows them to be "seen" and allows questions about their behavior to be answered.*
> — *From* **"Innovation Insight for Observability"**

That seems straightforward enough, but this simple definition has created confusion among site reliability engineers (SREs), SecOps, and ITOps teams. Many conflate observability with its older cousin, monitoring. This misses the point. Observability isn't about alerts and dashboards. Those are the things you already know. Observability is about discovering things you didn't know.

Another way to think about the differences between observability and monitoring is with a data infrastructure analogy. A monitoring platform is like a data warehouse. Well-known, well-understood data is ingested into your platform of choice and shaped to answer the questions you know you want to ask. That's what a monitoring platform is good at: answering the known questions.

Observability is more like an exploration environment, like a data lake. Data lakes collect data from across the organization and skilled users explore stored data for new signals or opportunities. When a promising opportunity is found, the data is refined and optimized, then used in a more consumable platform, like a data warehouse.

We need both exploration and optimized delivery. That's why we need both monitoring and observability.

YOU NEED BOTH OBSERVABILITY AND MONITORING FOR FULL INSIGHT INTO YOUR ENVIRONMENT, INCLUDING APPLICATIONS AND INFRASTRUCTURE.

The Need for Observability And Monitoring

The upswell of interest in observability continues driving confusion in the monitoring space. IT teams question if they need observability and how existing monitoring tools will work in this new concept. Some of the questions we hear from our customers are:

- *Does observability replace monitoring?*
- *How do observability and monitoring work together?*
- *What new tools do I need to take advantage of observability?*

The reality is you need both observability and monitoring for full insight into your environment, including applications and infrastructure. The easiest way to think about these two concepts is the popular OODA Loop. The OODA Loop has been used for years in domains as diverse as military strategy to IT operations. If you're unfamiliar with OODA, it stands for Observe, Orient, Decide, and Act. It's a good way to think about how observability and monitoring are complementary to each other.

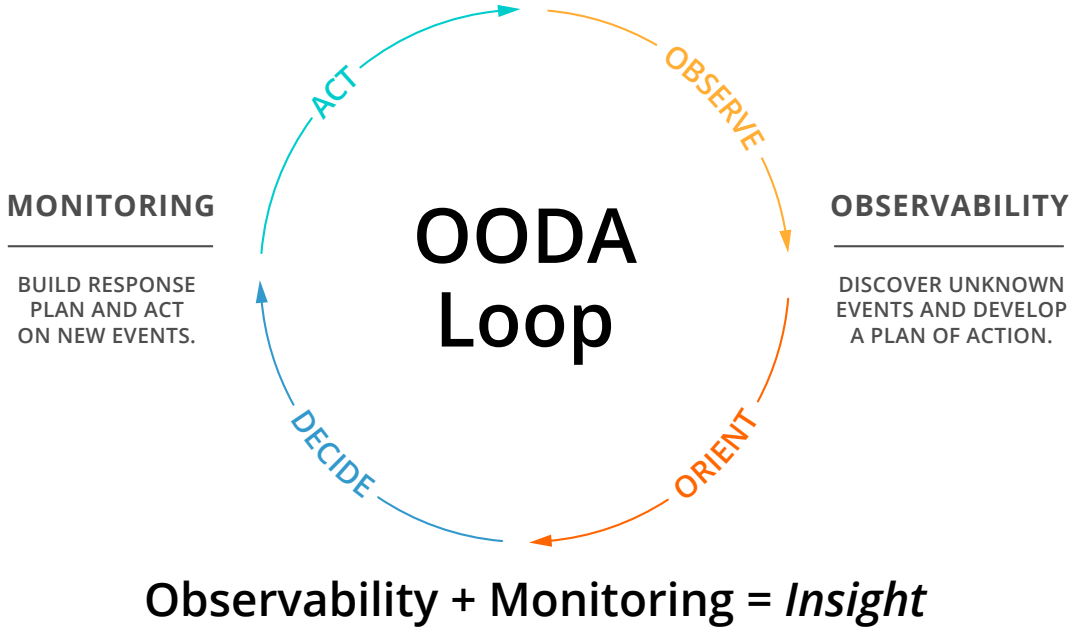


Figure 1. OODA Loop

On the right side of this Figure 1, you have Observe and Orient. This is where the observability side of the equation comes in. The left side, Decide and Act, is the monitoring part of the equation. Think about the two halves of the equation like this: Observability lets you discover new or unexpected signals in your environment. These are the things you're not monitoring for yet – you don't even know they exist! This is what observability gives you: The ability to discover and understand behavior of applications and infrastructure from the data these things emit. Observability is about discovery.

Monitoring, the second half of the equation, is about doing. This is where Decide and Act come in. Once you know the data indicates a performance challenge or a potential security breach, you can configure your monitoring systems for those signals and automate the types of responses that are most suitable for these new conditions.

While the concepts of observability and monitoring are closely related, there are differences. The biggest difference starts with data.

WITH AN INCLUSIVE APPROACH TO ALL SOURCES OF DATA, OBSERVABILITY TOOLS CAN DELIVER BETTER VALUE FOR DISCOVERY.

The Diversity of Observability Data

Traditional monitoring platforms take an exclusive approach to the data they use, giving them a limited view of the environment they monitor. These platforms primarily use data from dedicated software agents deployed across applications and infrastructure. Agents only send data to one place: the target monitoring platform. Most enterprises have twenty to thirty monitoring tools, each with their own dedicated agents, resulting in disconnected silos of information. This limits the effectiveness of monitoring solutions.

Observability takes the opposite approach. With an inclusive approach to all sources of data, observability tools can deliver better value for discovery. Today, the focus is on four distinct data types: events, logs, metrics, and traces, commonly shortened to MELT.

In the near future, Cribl believes observability platforms will consume a much more diverse array of data sources, including configuration management data, dependency maps, and data from data marts, lakes, and warehouses. This end-to-end business visibility is crucial to understanding the factors impacting business outcomes.

As the need for more data becomes apparent, another challenge must be overcome, and that's the need to instrument everything, from distributed applications to containers to traditional infrastructure.

The Need for Pervasive Instrumentation

While we just talked about the need for more data, this remains a critical gap in monitoring and observability strategies. For observability to deliver on its promises, everything must be instrumented, from legacy, monolithic C++ applications to distributed microservice architectures running on containers. It sounds simple, right? Just sprinkle in some instrumentation and off you go. The reality is quite a bit different.

The expectation is DevOps teams embed instrumentation into their code as part of the development process. While a nice idea, there are four reasons this falls short. First, the quality of instrumentation varies. Many log statements are terse and only understandable by the developer who wrote them. The message "In function xyx123!" isn't helpful to an SRE digging into a performance problem that cropped up in the latest release.

Second, instrumentation libraries vary by implementation, giving inconsistent results across language bindings. OpenTelemetry is trying to improve this, but its progress is slow and still requires developers to do more work that, if we're honest, doesn't benefit them. It benefits SREs and SecOps. (We've written about the challenges of mismatched incentives across teams here.)

The third problem with instrumentation is completeness. While you can instrument the applications you're building today, you must also account for the range of legacy and third-party applications your business relies on. You may not have the source code for critical applications, or the person who wrote it retired or left the company. Third-party applications may not have any instrumentation, or it suffers from the aforementioned problems.

Cribl's view is instrumentation must be not only pervasive, but consumable for the operations teams using it. To meet these challenges head on, we've built an instrumentation utility called AppScope. AppScope provides black-box instrumentation capabilities for any Linux binary without the need to change code, or anything else in your environment.

INSTEAD OF RELYING ON SILOED POINT-TO-POINT CONNECTIONS, AN OBSERVABILITY PIPELINE CENTRALIZES ALL OF YOUR OBSERVABILITY DATA PROCESSING, GIVING YOUR TEAMS FULL CONTROL OVER EVERY ASPECT OF YOUR DATA.

However, this introduces the fourth challenge: massive data volumes. Fully instrumented applications and infrastructure can produce terabytes of data each day. That data must be delivered to a range of destinations, like time series databases, logging analytics platforms, SIEMs, and APMs. Each destination has unique data formatting and optimization requirements, and all of the data may not be valuable to each destination, or data might only be valuable based on other signals.

Connecting these sources to destinations in an intelligent way is the crux of the observability challenge. This is where the observability pipeline comes in.

The Observability Pipeline

An observability pipeline is a strategic control layer positioned between the various sources of data, like networks, servers, applications, and software agents, and the multiple destinations in today's IT and SecOps environments. Instead of relying on siloed point-to-point connections, an observability pipeline centralizes all of your observability data processing, giving your teams full control over every aspect of your data.

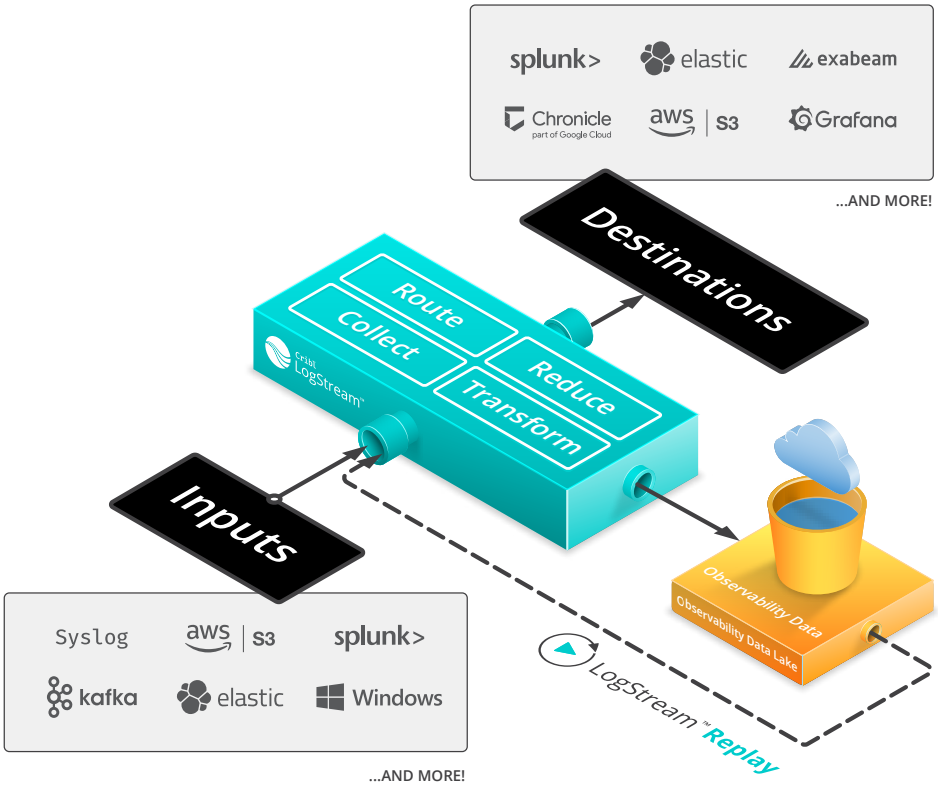


Figure 2. Cribl Marketecture

From a data management perspective, observability pipelines allow you to:

- *Filter out redundant data to improve performance and lower the cost of ingest-based destination platforms.*
- *Enrich data with additional context, like GeoIP, for improved downstream analysis and compliance with various data privacy laws.*
- *Redact sensitive data to comply with data governance requirements.*
- *Route data from one source to multiple destinations, eliminating the need to deploy new agents alongside deploying a new service.*
- *Send full-fidelity data to low-cost storage and replay it when needed.*

OBSERVABILITY GIVES YOU THE OPPORTUNITY TO DISCOVER AND UNDERSTAND YOUR DYNAMIC ENVIRONMENTS IN NEAR REAL TIME.

Abstracting the sources and destinations of observability data offers massive benefits to IT and SecOps teams, including:

- *Providing a single point for governing data and applying consistent rules for data redaction, access control, and sharing.*
- *Reducing the amount of redundant data flowing into downstream systems like logging analytics, SIEM, and SOAR platforms.*
- *Accelerating onboarding new tools by sharing data from one source with multiple destinations.*

Conclusion

Complexity increases uncertainty. Today's teams grapple with dozens of different technologies spread across on-prem and cloud environments, making it impossible to know ahead of time what's important. Observability gives you the opportunity to discover and understand your dynamic environments in near real time.

ABOUT CRIBL

Cribl is a company built to solve customer data challenges and enable customer choice. Our solutions deliver innovative and customizable controls to route security and observability data where it has the most value. We call this an observability pipeline, and it helps slash costs, improve performance, and get the right data, to the right destinations, in the right formats, at the right time. Join the dozens of early adopters, including market leaders such as TransUnion and Autodesk, to take control and shape your data. Founded in 2017, Cribl is headquartered in San Francisco, CA. For more information, visit **www.cribl.io** or our **LinkedIn**, **Twitter**, or **Slack** community.