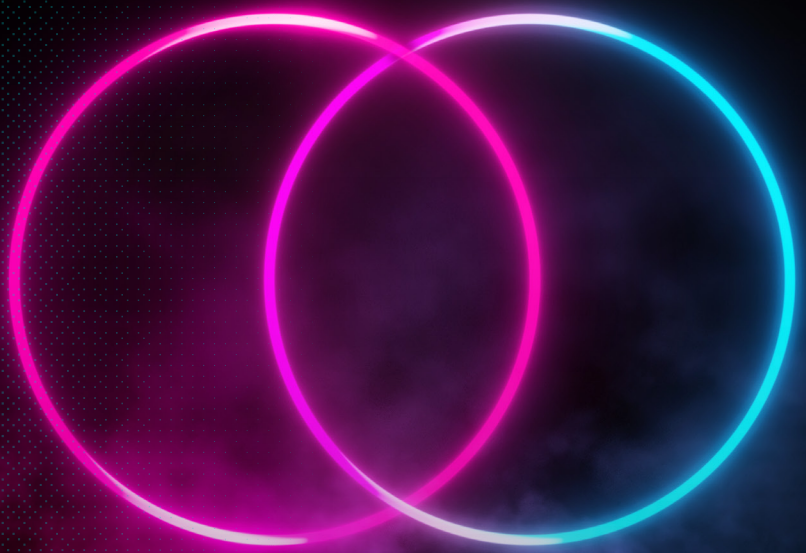
Cribl

>WHITE PAPER_

Organizational independence:

controlling your own path to observability.



in·de·pend·ence

/ˌindəˈpendəns/ **NOUN**

The fact or state of being free from outside control; not depending on another's authority.

ob·serv·a·bil·i·ty

/əbˈzərv əbilə dē/ **NOUN**

The practice of interrogating your environment without knowing in advance the questions you need to ask.

Cribl

>WHITE PAPER_

Organizational independence:

controlling your own path to observability.

The value of independence.

While the term “observability” has gained momentum over the past few years, elements of the practice of observability have been around for decades. Today, there are several tools for observability, including APM tools, log analytics, time series databases, SIEMs, Data Lakes, and UEBAs. These tools help development, operations, and security teams gain insights into the health of their infrastructure.

Independence with respect to observability means being able to choose the best tools and approaches. Challenges to this independence can come both externally and internally. Vendors, other teams, and budget concerns can restrict your choices and seek to control how you observe your systems. At the same time, different team goals within an organization have evolved to become more specific, making the need for independence more important.

Vendors, other teams, and budget concerns can restrict your choices and seek to control how you observe your systems.

These teams negotiate how to get data, whether they support specialized SIEM and UEBA tools, and other elements of control over how to analyze data. Many organizations find themselves moving away from independence as they try to standardize on certain analytics tools and look to manage costs. So how do you regain control over your own path?

Independence from vendor lock-in.

Most observability journeys start with a single analytics tool. While organizations in these early days may hope that one tool can meet all of their objectives, they soon learn that seeking specific answers requires specialized tooling. Though it may be in the best interest of vendors for their customers to be locked into their platform, true independence requires having the flexibility to choose from a wide range of fit-for-purpose tooling.

One of the more interesting and valuable developments in introducing flexibility to observability projects is the concept of separating your systems of analysis from your systems of retention. Many tools, especially indexed logging tools, prioritize fast searches. The tradeoff with this is that they require larger, more performant storage footprints for retaining data — adding more compute and storage capacity quickly becomes very expensive.

Smart organizations are taking advantage of low-cost destinations like S3-compatible storage, file systems, and data lakes for longer-term retention. This allows them to analyze high-priority data in real time, but drop lower-priority data from their analytics tools and opt to retain this data in object storage.

Adding new tools to your observability toolbox can give you the flexibility and control you need to succeed. A challenge that comes from this, however, is that new tools often require you to deploy new agents or collectors to get data into the right format for analysis. Observability professionals need the flexibility to choose a wide range of tools without having to add agents to every application, server, and endpoint. We'll share strategies later to address this.

Independence for security teams.

Let's face it, Security Operations is an observability challenge. Some of the tooling is specialized for security-specific investigations and observations, but these tools are still looking at logs, metrics, and traces to learn more about their environment. Because observability practices often started in the IT Operations department, that team often chooses which tools are being used, and may also control how data is collected and routed.

Observability professionals need the flexibility to choose a wide range of tools without having to add agents to every application, server, and endpoint.

For truly independent observability, Security teams should have control over their own data and be able to choose the tools that best fit their needs. UEBA or SIEM tooling build on the foundation of log analytics tools, but are purpose-built to solve security operations challenges. Security teams often also need to retain data for longer periods of time, in case they need to investigate a data breach — many of which can occur several months to years before they are discovered. The decision about how long to retain security data should be owned by the Security team.

The Security team is also responsible for protecting sensitive personal data like social security numbers or credit card information. Security professionals need to be in control over how this information is handled, who can access it, and whether or not it should be masked. Because of these and other security-specific use cases, security professionals must have independence in making decisions about their own execution of observability projects.

Independence for IT operations teams.

IT Ops teams may have made some of the earlier decisions about how to conduct observability, but there are opportunities for them to increase flexibility. The answers you seek from your environment, and therefore the goals of observability, are constantly evolving. IT Ops needs to be able to add new data sources to solve new use cases, without impacting other teams and existing projects by changing the way data is collected and routed to various analytics tools. They should also be empowered to add specialized tools, or change tools.

What other teams may view as a control issue, IT Ops may think of as a dependency problem. If other teams rely on IT Ops to own observability infrastructure within the organization, IT Ops may be spending more time re-instruments and supporting these projects than they do optimizing the overall environment. This problem worsens if they are spending a lot of time deploying new agents or rewriting log formats for other teams. Independence is put at risk if other teams are viewed as co-dependent.

Independence for data architecture and logging teams.

In some organizations, with mature observability programs, a Data Architecture or Logging team controls a lot of the observability motions. This team is tasked with looking for the best ways to get data, in all of the right formats, to teams for analysis. The same team may also be responsible for optimizing how data is stored for compliance and long-term analysis. Team members work with various internal customers on new use cases, and solve some of the trickier challenges of observability.

These teams face some of the same co-dependency problems we discussed with the IT Ops teams. When the security team needs to perform an investigation relating to a data breach, the Logging team may be responsible for finding the "old" data and preparing it for analysis. These teams may feel like they have become data jockeys — running laps to get data to various teams instead of focusing on improving the way their organization can use data most effectively. They may also have to decide between infrastructure budgets and the optimization of observability. These choices restrict independence.

The trade-offs of independence.

Seeking answers to new questions that sprout up on the journey to observability requires new data sources, tools, and approaches. Often, adding a new tool means deploying new agents and collectors. This effort is far from trivial. Large organizations may have tens of thousands of applications, servers, and other endpoints that need to be observed. Most tools have their own data formats that require data to be collected to match. This may force the collection of the same data in different formats for multiple tools and destinations. The added flexibility of having all of the right tools triggers a host of challenges in collecting new data.

Most tools have their own data formats that require data to be collected to match. This may force the collection of the same data in different formats for multiple tools and destinations.

Analyzing and retaining new data, and keeping it over longer time frames, can be very expensive. For many organizations, infrastructure costs — both storage and compute — can be more expensive than commercial-software license costs. The Security team demands data be kept on board for multiple years so they are never caught without the data they need to investigate past breaches. The Data Architecture team, on the other hand, may push back (citing budget concerns), or be forced to perform gymnastic workarounds to keep everything in balance.

Giving more control to different teams can introduce security exposure. While one of the goals of creating more independence is sharing the responsibility of data ownership, this can be risky. Organizations need to find balance between data retention, budget, tool flexibility, and security.

An observability pipeline for independent observability.

An observability pipeline gives you independence without the sting of negative tradeoffs. Observability pipelines can help empower teams to control their own observability projects without depending on IT Ops or Logging teams' being on standby to provide support.

An observability pipeline helps you parse, filter, enrich, and secure data for all of your teams. It can take data as it is today and shape it into formats that various current and future tools require. It allows you to choose the right tools for any observability challenge. It can also help you break free from vendors that no longer meet your needs, or add new products to supplement your analytics efforts.

An observability pipeline lets you collect more data, keep it longer, all while spending a lot less money. It helps you separate data destinations used for analyzing observability data from lower-cost destinations that retain it for longer periods of time. It also allows you to collect data once and shape it for multiple teams, tools, and data stores.

An observability pipeline frees your teams to do their jobs. It allows each to focus on making improvements, and tackling new projects, rather than supporting existing commitments and putting out fires.

There are several ways to implement an observability pipeline, including building your own, using open-source resources to cobble together some of the functionality we've discussed here, or buying an out-of-the-box solution. We recommend choosing an enterprise-ready solution purpose-built to solve the challenges we've detailed throughout this paper. Cribl Stream is an excellent choice to make independence a priority in your observability journey.

Cribl Stream lets you process observability data, eliminate noise, enrich the data with third-party data, and deliver it to any tool before you pay to analyze it.

Cribl stream: freedom to pursue observability.

Cribl Stream lets you process observability data, eliminate noise, enrich the data with third-party data, and deliver it to any tool before you pay to analyze it. Stream helps you get the right data where you want, in the formats you need.

Stream helps you route data where it has the most value. Send data to the most effective destinations, including low-cost object storage for long-term retention. Route data to the best tool for the job — or all the tools for the job — by translating and formatting data into any tooling schema you require. Let different departments choose different analytics environments without having to deploy new agents or forwarders.

Stream also helps you reduce data volume, to help control costs and slash infrastructure budgets. It can reduce as much as 50% of ingested data volume, reducing costs and improving system performance. Eliminate duplicate fields, null values, and any elements that provide little analytical value. Filter and screen events for dynamic sampling, or aggregate log data into metrics for massive volume reduction. Do all this without worrying about missing data later — you can keep a full-fidelity copy in a low-cost destination, and replay it through Stream if needed.

Stream is also the best way to get multiple data formats into the tools you trust for your Security and IT efforts. Use Stream as a universal receiver to collect from any data source - and even to schedule batch collection from REST APIs, Kinesis Firehose, Raw HTTP, and Microsoft Office 365 APIs. In addition, recall data from object storage to replay to analytics tools for later investigations, using ad-hoc data collection. This can balance the competing goals of keeping more of the right data longer, while cutting what you spend on storage.

Stream lets you take data in whatever format it comes, and shape it into the data you need. No matter how messy the data you have today, Stream can make it beautifully functional to feed all of your tools. Process your observability data before you pay to analyze it. Translate and transform, enrich, parse, and structure data to focus on signal and not noise. Enrich logs with third-party data such as geoIP or known threat databases. Secure log data with encryption, masking, and role-based access controls.

In summary.

Stream helps every team achieve a more independent approach to observability. Teams can make their own decisions without burdening the work of others. They can customize their approach to finding answers in the data their environment generates. By using an observability pipeline like Stream, you can improve observability, cut costs, and eliminate a lot of the risks and negative tradeoffs associated with changing your infrastructure.

- Observability has evolved over time.
- New tools and approaches can better help you achieve your goals.
- You must have the flexibility to choose new tools, ask new questions.
- Organizations need independence from vendor lock-in.
- Teams should pursue observability projects independently, and make decisions optimized for their individual use cases.
- Artificially constrained choices have negative consequences.
- Observability pipelines give you independence without negative trade-offs.
- Cribl Stream is an enterprise-ready observability pipeline that gives your teams the independence to choose the right data, for the best tools, in the formats you need, all while cutting costs.