unravel™

WHITE PAPER

DATA OBSERVABILITY: THE MISSING LINK FOR DATA TEAMS

DATA TEAMS ARE THE KEY TO UNLOCKING THE POWER OF DATA

Ten years ago, Marc Andreesen argued that "software is eating the world," and today software is indeed baked into every organization's operations, products, and services. But now data has taken center stage. Data is changing the world, delivering amazing outcomes—everything from addressing global problems (accelerating vaccine development, improving food availability, combating climate change) to fundamentally changing businesses' relationship with their customers. It's no exaggeration to say that every company has now become a data company.

Organizations recognize that advanced analytics is a battleground that separates success from failure. That's why they're all investing heavily in modernizing their data stack. But no matter what a company's data stack looks like, it's the people—data teams—who unlock the power of data, bring analytics to life, and deliver value to the business from this massive investment in the modern data stack. (See Figure 1.)

But data teams are overwhelmed. They struggle to keep pace with the increased volume, velocity, variety, complexity—and cost—of data applications and pipelines. Individual team members—data scientists and analysts, data engineers, operations teams, architects, product owners, and C-level analytics officers—are all facing their own particular challenges and obstacles. Collectively, they are frequently blocked from becoming the high-functioning, highly productive teams required to deliver timely, business-critical analytics and innovative insights that their companies so desperately need.

Put simply, the way data teams are doing things today isn't working.

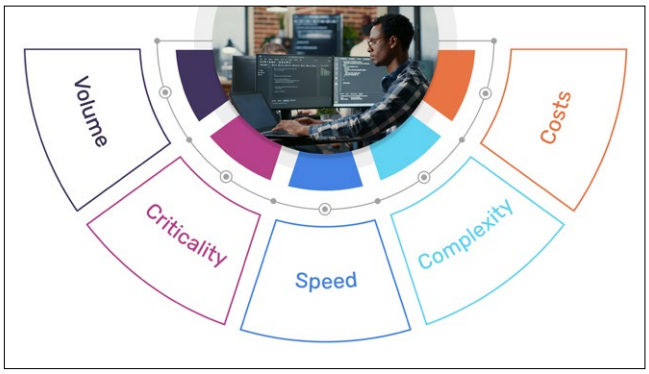

Figure 1. Data teams today are caught in a five-way crossfire

What Data Teams Can Learn from DevOps

Data teams today find themselves in much the same boat as software teams were (and some still are) 10+ years ago. They're facing many of the same challenges, at least generically—perpetual firefighting in production, time-consuming manual detective work, working in silos, finger-pointing, burying operations teams with trouble tickets, process bottlenecks, countless alerts, etc.

Software teams have successfully tackled these obstacles with a whole set of DevOps processes, best practices, and technologies that allow them to accelerate each stage of the software development lifecycle. As an industry, we're still in the early innings of figuring out exactly what such a DataOps playbook would look like, but what is undeniable is that **observability** is the foundation upon which the ability to accelerate the DataOps lifecycle is built. Data teams need the same kind of full-stack visibility, end-to-end contextual awareness, automation, and actionable intelligence that have empowered software teams.

Because right now, it's really hard for data teams to quickly and accurately get the information they need to do their jobs faster and better at each stage of the DataOps lifecycle. Different team members may have different questions they need answered, but there's no single source of truth for everyone to rally around to understand the problem. Without that unified shared visibility and intelligence, there's a lot of friction and frustration among team members. Different tools give different answers, the blame game begins, and it takes hours or days—sometimes even weeks—to get to the bottom of things. (See Figure 2.)

Traditional application performance monitoring (APM) observability tools like Datadog, Dynatrace, AppDynamics, and New Relic do a great job of solving this challenge for web applications. They automatically extract the pertinent details from the web application ecosystem, stitch them together into a meaningful context, and apply some analysis (topology mapping, baseline patterns, anomaly detection, root cause analysis) to tell you why something happened—all from a single view.

But APM observability falls short for data applications because it was never designed for the unique needs of data applications/pipelines. Data applications are a completely different animal, with a totally different class of problems, root causes, and remediations. Trying to use DevOps tools for DataOps doesn't work. If they did, data teams wouldn't be getting bogged down.

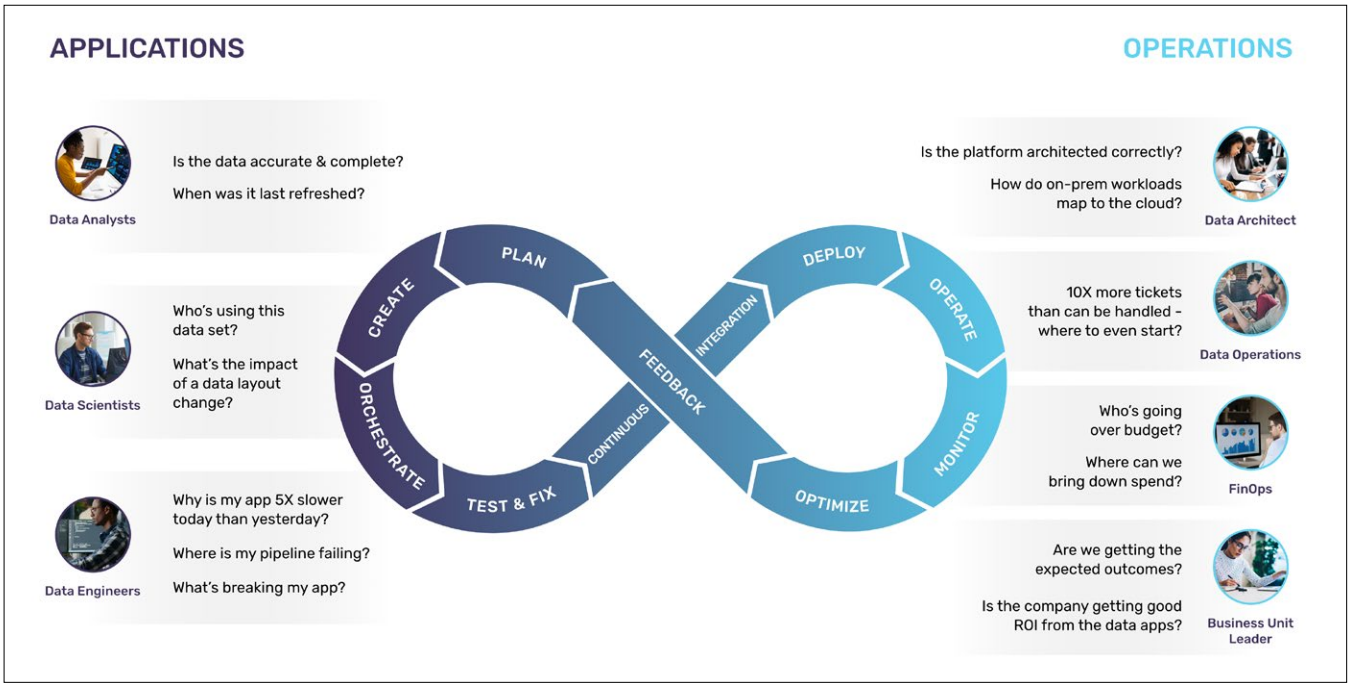

Figure 2. Different team members have different needs throughout the DataOps lifecycle

Why Data Teams Need Observability Designed for Data Teams

While observability for DevOps teams and observability for DataOps teams have generic similarities—capturing and correlating telemetry data in a meaningful context of situational awareness, applying sophisticated analysis to produce actionable intelligence—there are some fundamental distinctions that call for a purpose-built solution for DataOps.

Beyond data teams using different technologies, systems, and platforms (Spark, Kafka, Airflow, dbt, Databricks, Snowflake, BigQuery, Dataproc, Amazon EMR, et al.) and being composed of a different cast of characters with different skill sets, the very nature of data changes the paradigm. Web apps are request-response based, whereas data apps are parallel processing ones. If a web app has an issue, you are drilling down to understand which request was slow. If a data app has an issue, you are trying to understand problems such as degree of parallelism, load imbalance, skew, code execution, and data integrity, among others.

The different requirements for observing data applications/pipelines vs. observing software applications can be boiled down to two things:

- You need to capture very different telemetry data
- You need to do a completely different analysis to understand and fix issues

Different observability data

In addition to understanding the **performance** and reliability of a data pipeline or application, data teams must understand **data quality**, or the condition of the data as it flows through pipelines. And as more workloads move to the cloud, **cost** increasingly needs to be understood and governed at a granular level. These three dimensions are all interrelated, and DataOps teams must be able to "connect the dots" between them from a single interface.

Modern data stack applications don't run on a single system, but a "system of systems" comprising a dozen or more different components moving data from one node to another in a complex labyrinth of interdependencies. Data teams need information about performance, cost, and data quality both horizontally across all the various components/systems as well as vertically, from the application down to infrastructure and everything in between—and get a big-picture view to make sense of how everything works together as a whole.

Taking the data pipeline illustrated in Figure 3 as an example, you can see how interconnected and complex a single data application can become. Troubleshooting a failure, slowdown, or increased cost of this pipeline can be very challenging. The problem could be anywhere along the line—at the user, job, application, platform, or cluster level—for any one of a myriad different reasons. It may be that just a single job is failing due to a code, configuration, data layout, infrastructure, or orchestration issue. Yet that one job creates an upstream issue that proliferates downstream, bringing the entire application to its knees.

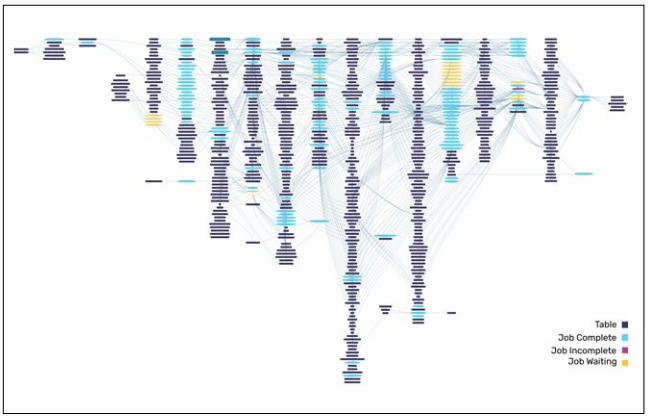

Figure 3. Data applications have complex dependencies—vertically and horizontally

Web observability APM technologies are intended to analyze a different class of performance problems, such as slow response rate, for a different type of application ecosystem, such as Java and PHP. It was never intended to absorb and correlate the kind of performance details buried within a modern data stack's components or "untangle the wires" among a data application's upstream/downstream dependencies.

DataOps teams need to understand how a data pipeline is performing in order to meet their service level agreements (SLAs), and the quality of the data as it flows through the pipeline. Web app observability tools simply were never designed to comprehend this duality. They do not capture, correlate, and analyze performance metrics from code, containers, execution configurations, data-quality metrics (freshness, duplication, accuracy), or volume of data, usage and access, data lineage—crucial information data teams need to either reactively detect/resolve or proactively prevent data application problems.

As more data workloads are migrated to the cloud, the cost of running data applications/pipelines can mushroom out of control all too easily and quickly. An organization with 100,000+ data jobs in the cloud has literally a million decisions—at the application, pipeline, and cluster level—to make about where, when, and how to run those jobs. And each decision carries a price tag. As organizations cede centralized control over infrastructure to a more democratized approach to spinning up resources, it becomes essential for both data engineers and FinOps to understand exactly where the money is going and identify opportunities to reduce/control costs.

Different kind of analysis

All these application workloads get broken down into multiple, smaller, often similar parts each processed concurrently, with the results re-combined upon completion—parallel processing. It's crucial that everything among these sub-parts—execution time, scheduling and orchestration, data partitioning, data lineage, and layout—be in sync and ordered, with late-arriving pieces handled accordingly. This is a very different computing paradigm than web applications, where what's most important is the response time of each service request and how that contributes to the overall response time of a user transaction. (See Figure 4.)

What's important with data applications is the degree of parallelism within each sub-part. Data teams need to see details at a highly granular job level—for each sub-task within each sub-part of each job—and in a job-specific "workload-aware" context that comprehends how the job fits in with the bigger picture at the application, pipeline, platform, and cluster levels.

How Data Teams Are Getting Observability Today

To get the fine-grained granular insight about performance, cost, and data quality, data teams are forced to cobble together information from a variety of different systems and tools. And as organizations scale their data stacks—picking and choosing various systems, platforms, and cloud services into a potpourri of Databricks, Snowflake, BigQuery, Amazon EMR, Dataproc, and more—the amount of information and number of sources make it extraordinarily difficult to paint a full picture of what's going on.

Most of the granular details needed are hidden in plain sight, already captured by one system or another. But they haven't been extracted, correlated, and put into meaningful context specifically for data teams. Each tool provides some of the information needed, but not all—and certainly not in the language of data applications/pipelines.

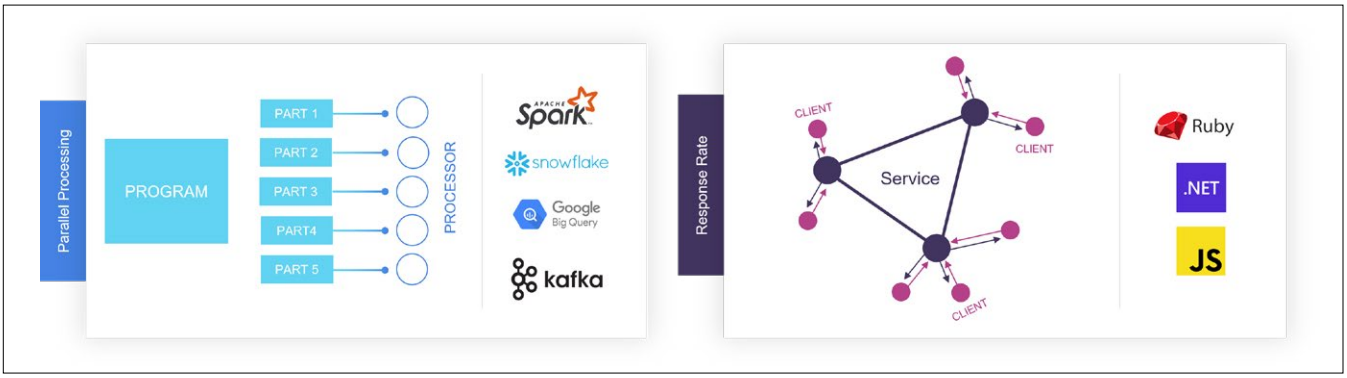

Figure 4. Modern data apps are fundamentally different than web apps

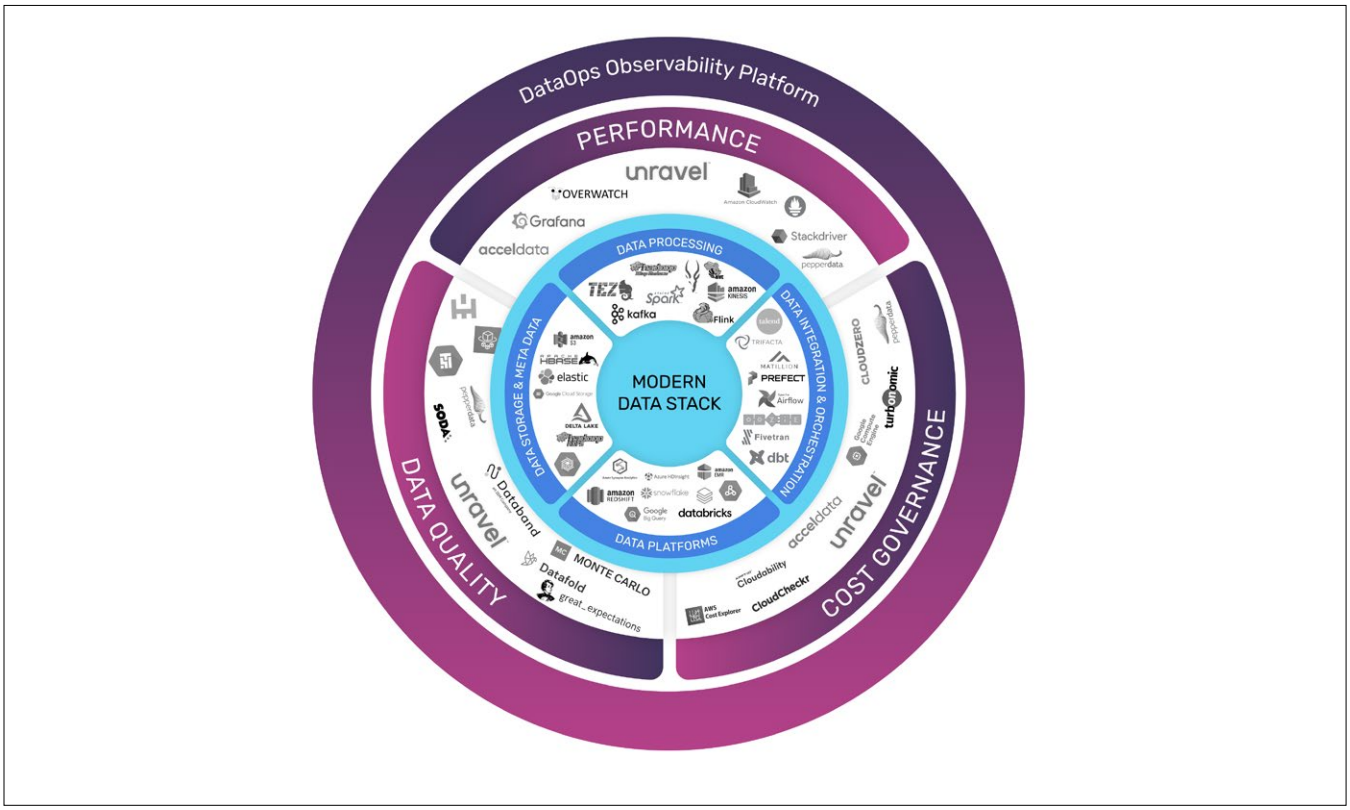

Figure 5. DataOps observability covers 3 interrelated aspects of the modern data stack

There are a slew of cloud, open-source, and proprietary data observability tools that provide metrics about one layer or system in isolation. For example, Spark UI has job-level details, but not infrastructure and configuration details and other information to connect together a full pipeline view—and of course it's limited to Spark. Platform-specific interfaces (Databricks, Amazon EMR, Dataproc, HDInsight, BigQuery, Snowflake) have the information about resource usage and the status of various services at the cluster level, but no details at the application or job level. Cost-management tools have aggregated information at a 10,00-foot level, but not granular details at the job, application, user, or team level. (See Figure 5.)

What data teams need is a single-source-of-truth platform that:

- Discovers and extracts telemetry details from every application, pipeline, system, and dataset—horizontally and vertically
- Correlates everything in a meaningful context
- Analyzes the details to provide actionable intelligence on why something happened
- Automates root cause analysis and remediation

Ideally, the solution can go "beyond observability" to not just show you where and why an issue exists but actually tell you what to do about it.

That's true DataOps observability.

What Is DataOps Observability?

Just like DevOps observability provides the foundational underpinnings to help improve the speed and reliability of the software development lifecycle, DataOps observability can do the same thing for the data application/pipeline lifecycle. But—and this is a big *but*—DataOps observability as a technology has to be designed from the ground up to meet the different needs of data teams.

DataOps observability cuts across multiple domains, or "flavors," of observability:

- **Data application/pipeline/model observability** ensures that data analytics applications/pipelines are running on time, every time, without errors. This is exactly the kind of observability that APM tools like Datadog, AppDynamics, Dynatrace, and New Relic provide to software teams for web apps but designed for the unique needs of data pipelines/apps.
- **Operations observability** enables data teams to understand how the entire platform is running end to end (vs. pinpointing problems with one application). It provides a unified view of how everything is working together, both horizontally and vertically, within the Databricks, Snowflake, Amazon EMR, BigQuery, Dataproc, etc., ecosystems.

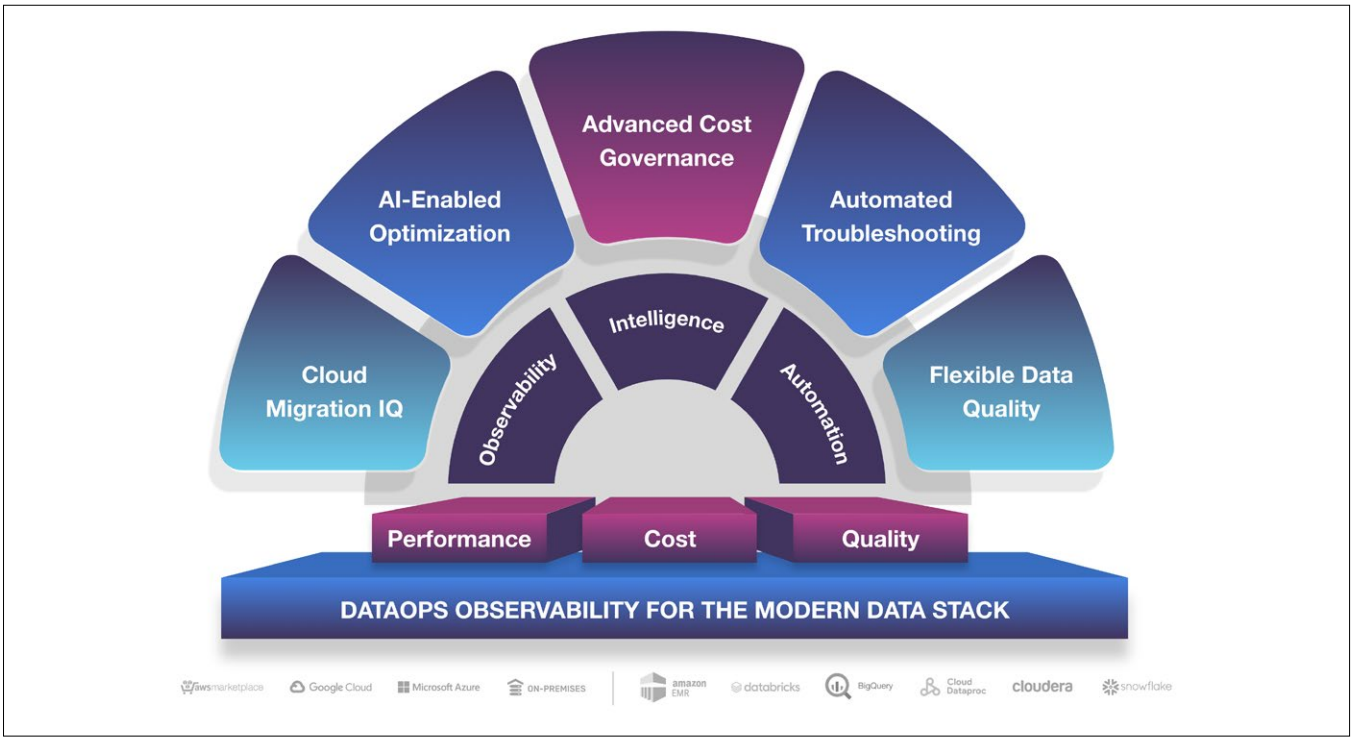

Figure 6. DataOps observability for the modern data stack

- **Business observability** has two parts: profit and cost. The first monitors and correlates the performance of data applications with business outcomes. This is really about ROI—are all these applications actually driving the kind of revenue or reducing costs? What is the business impact of failed applications, broken pipelines, incorrect or stale results? The second part is **FinOps observability**, where organizations are able to use real-time data to govern and control their cloud costs, understand where the money is going, set budget guardrails to prevent misuse and waste, and identify opportunities to optimize the environment to reduce costs (or at least do more for the same spend).
- **Data observability** looks at the datasets themselves, running quality checks to ensure correct results. It tracks lineage (where is the data coming from, and where is it going?), usage (where is it used, who can access it, and is it being stored in the most cost-efficient manner?), and the integrity and quality of data (is the data accurate, complete, reliable, fresh, and consistent at each stage of the pipeline from sources to output?).

Consequently, DataOps observability isn't just for data engineers or data operations teams, or data scientists and analysts, or data product owners and data architects, or chief analytics officers. Data teams can no longer afford to be myopically focused on one aspect or another, because problems in the modern data stack are all interrelated. Without a unified picture of everything that's going on, from different perspectives, the promise of DataOps—delivering reliable business-ready data products and services faster in an agile, efficient manner—will go unfulfilled. (See Figure 6.)

Extracting, correlating, and analyzing everything at a foundational layer in a data team-centric, "workload-aware" context combine together to deliver five capabilities that are the hallmark of DataOps observability—and empower different personas within data teams to do their jobs faster and better, no matter what or where they're tackling along the DataOps lifecycle:

- **End-to-end visibility** stitches together telemetry data and metadata from the multitude of components, assets, technologies, and processes across the full data stack—both horizontally and vertically—to give you a unified, in-depth understanding of the behavior, performance, cost, and health of your data and data workflows.
- **Situational awareness** puts all this aggregated information into a meaningful context for the task at hand. It's not enough simply to have a ton of granular details available in dashboards, charts, and graphs, they need to be correlated, visualized, and analyzed so you can make sense of it all immediately.
- **Actionable intelligence** tells you not just what's happening, but why. Sophisticated analytics applied to full-stack observability data automatically identify baseline patterns, detect anomalies, flag data issues, pinpoint root causes, and provide other insights into emerging or existing problems. Next-gen observability platforms go a step further and provide prescriptive AI-powered recommendations on what to do next.
- Everything either happens through or enables a **high degree of automation**. As much of the manual detective work as possible is automated—continuous and automatic discovery of

what's running in your data estate, capturing and correlating workload profiles, mapping dependencies and lineage, assessing data layout and data quality, troubleshooting, optimization for performance and cost, etc. Further, observability's actionable intelligence enables a higher degree of proactive measures, whether intelligent alerts or autonomous remediation actions. The best way to reduce firefighting is to avoid having a fire to put out in the first place.

- This proactive capability is **governance** in action. Governance is really just converting AI-powered recommendations and insights into impact. In other words, have the system apply the recommendations automatically. No human intervention is needed. Policy-based governance rules could be as benign as sending an alert to someone if some sort of data layout, performance, or cost threshold is violated or as aggressive as automatically requesting a configuration change for a container with more memory. This is true AIOps.

Summary

Observability for data applications/pipelines today is a bit like the parable about the blind men and an elephant, where each man feels a different part of the elephant's body—but only one part, such as the trunk or the tusk—then describes the elephant based on his limited experience, and their descriptions are all different.

Data teams have to stitch together details from a variety of disparate sources, jumping from screen to screen, in a labor-intensive effort that can take hours—even days—for a single problem. And still there's a high risk that it won't be completely accurate. Correlating all the details and putting them into context requires three things that are always in short supply: time, labor, and expertise. Even if you know what you're looking for, it's like looking for a needle in a stack of needles.

As more new and innovative technologies make their way into the modern data stack—and ever more workloads migrate to the cloud—it becomes increasingly necessary to have a unified DataOps observability platform with the flexibility to comprehend the growing complexity.

Effective DataOps observability starts with extracting the right kind of details at a granular level from every system in the modern data stack. Having complete and fine-grained details is the foundation upon which all other observability capabilities are built. Then everything can be contextualized and visualized in a "single pane of glass." Dashboards and reports should be able to slice and dice the information to enable anybody on the data team—experts and non-experts alike—to understand what's going on from different angles (resources, usage, job/application/pipeline/cluster performance, cost, data tables, alerts, optimization opportunities, cloud migration, etc.). This should be table stakes for any observability solution.

What takes DataOps observability from good to great is the level of automated analytics. At a minimum, AI capabilities should be able to identify root causes with a high degree of accuracy. Even better is workload-aware AI that doesn't just show you what went wrong and why, but tells you exactly what to do next—e.g., providing precise, prescriptive configuration changes to improve performance or reduce costs.

A DataOps observability platform should be fully extensible and allow integrations with a variety of systems, both on-premises and in the cloud. Most organizations have a combination of on-premises and cloud pipelines in their data estate, and this will probably be the reality for the foreseeable future. The modern data stack is such a complex, ever-changing environment, with newer platforms like Databricks and Snowflake ascending and older platforms like Cloudera fading, that any DataOps observability solution must be flexible enough to support myriad tools, technologies, and systems.

DataOps Observability Designed for Data Teams

Ready to take the next step on your modern data stack journey? Book a 30-minute demo to see what Unravel can do for you.

BOOK A DEMO

About Unravel Data

Unravel Data radically transforms the way businesses understand and optimize the performance and cost of their modern data applications — and the complex data pipelines that power those applications. Providing a unified view across the entire data stack, Unravel's market leading DataOps Observability platform leverages AI, machine learning, and advanced analytics to provide modern data teams with the actionable recommendations they need to turn data into insights. Some of the world's most recognized brands such as Adobe, 84.51° (a Kroger company), and Deutsche Bank rely on Unravel Data to unlock data driven insights and deliver new innovations to market. To learn more, visit www.unraveldata.com.