Eckerson Group | White Paper

The Ultimate Buyer's Guide to Data Observability

Five Criteria to Evaluate Products

KEVIN PETRIE

MARCH 2022

RESEARCH SPONSORED BY

acceldata

About the Author

For 25 years **Kevin Petrie** has deciphered what technology means to practitioners, as an industry analyst, writer, instructor, marketer and services leader. Kevin launched, built and led a profitable data services team for EMC Pivotal in the Americas and EMEA, and ran field training at the data integration software provider Attunity (now part of Qlik). A frequent public speaker and author of two books on data streaming, Kevin also is a data management instructor at eLearningCurve.

About Eckerson Group

Eckerson Group is a global research, consulting, and advisory firm that helps organizations get more value from data. Our experts think critically, write clearly, and present persuasively about data analytics. They specialize in data strategy, data architecture, self-service analytics, master data management, data governance, and data science. Organizations rely on us to demystify data and analytics and develop business-driven strategies that harness the power of data. **Learn what Eckerson Group can do for you!**

About This Report

Research for this report comes from numerous briefings with industry experts. This report is sponsored by Acceldata, who has exclusive permission to syndicate its content.

Table of Contents

Executive Summary

Real-time business intelligence (BI) projects, self-service analytics, data science projects, and the rise of data-driven applications make enterprise data architectures more complex. Data engineers struggle to manage data pipelines that deliver high volumes of accurate, timely data across heterogeneous hybrid and multi-cloud environments.

Data observability products help overcome these challenges. These products monitor and optimize analytics and AI/ML workloads, oftentimes operational and real-time, across application, data, and infrastructure layers. They can automate and observe data quality by assessing the accuracy, completeness, consistency, and timeliness of data. They also observe the performance of pipelines, including components such as Apache Spark processing and Kafka streaming, and data platforms such as Snowflake Data Cloud. Selected and implemented well, a data observability product can support use cases for data pipeline governance, data engineering, and IT planning. Together these use cases support BI, data science, and data-driven applications; as well as migrations of data and workloads to the cloud.

Data engineering teams and their managers should evaluate data observability products using five criteria: breadth of capabilities, data pipeline governance, heterogeneous integration, ease of use, and scale and performance. Consider these guiding principles as you create your short list of candidate products.

> ***Define your baseline requirements.*** *Assemble those "must-have" capabilities in each of the five evaluation criteria. Eliminate data observability products that fail to address these requirements.*

> ***Stress-test candidate products.*** *During your proof of concept, test the ability of short-listed products to address your stretch goals, especially as they relate to heterogeneous integration, scale, and performance.*

> ***Listen to your full team.*** *Both entry-level users and power users need to become more productive with your data observability product.*

The Need to Observe Data

Modern enterprise analytics has exploded into a world of complexity over the last 15 years.

Periodic business intelligence (BI) reports morphed into real-time dashboards and self-service reports. Data scientists entered the scene to build advanced algorithms that derive insights from multi-structured data. Data analysts, data scientists, and developers now collaborate to build data-driven applications that embed analytics features. To feed these new initiatives, data teams tap into an exploding supply of multi-structured data from internal and external sources.

Ambitious initiatives like these force enterprises to change their architectures rapidly. Many enterprises migrated data workloads to the cloud. But things got tricky because they moved to two clouds, left some data on-premises, or both. Now they need to keep all those datasets in sync. They build microservices that rely on modular and distributed applications. Because legacy systems never quite go away, these initiatives lead to a complex set of old and new architectural components.

Challenges

This complexity creates challenges. Data quality suffers thanks to duplicate and contradictory datasets. Data engineers struggle to cleanse and integrate high volumes and varieties of data. Without new levels of automation, they cannot scale pipelines to meet stringent service level agreements (SLAs). Amidst proliferating technology components, data teams lack a common platform and language on which they can collaborate to solve problems. All these challenges prevent them from delivering timely, high-quality data to voracious business owners. Because many data-driven applications are now customer facing, these challenges can have embarrassing and costly consequences.

Data engineering teams struggle to deliver timely, high-quality data to voracious business owners and customer-facing applications.

There is hope, in the form of data observability.

What is Data Observability?

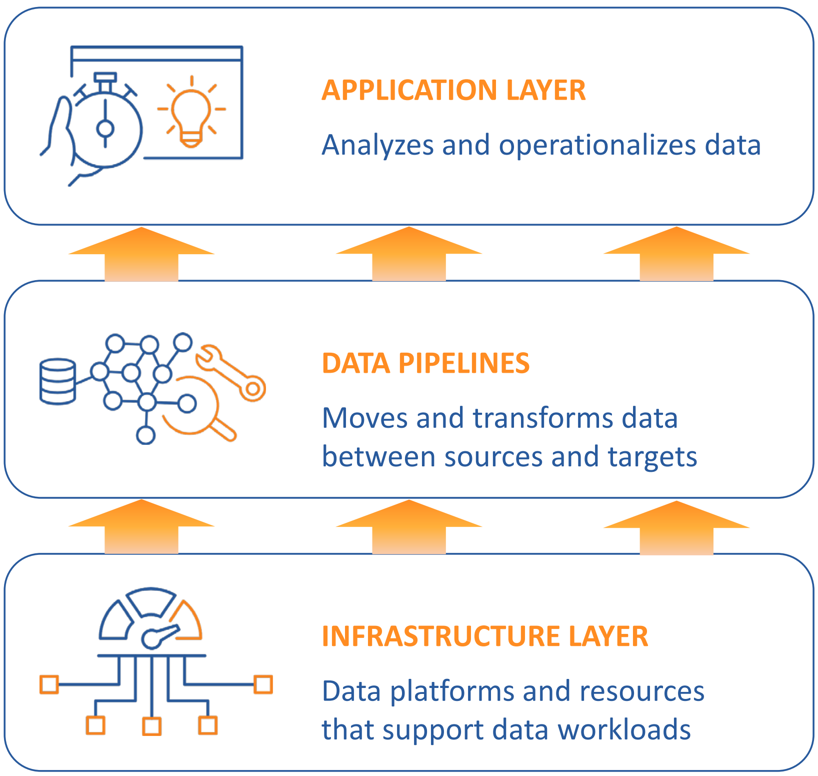
The emerging discipline of data observability offers tools and processes that help ensure the reliable, accurate, and timely delivery of data for modern analytics. Data observability monitors and optimizes analytics workloads across three layers of the data stack:

> **The application layer,** which include the tools, technologies, orchestration frameworks and applications that analyze and operationalize data.

- **The data pipeline layer**, which extracts, transforms, and loads data between sources and targets, for example using **Apache Spark** for processing or **Apache Kafka** for streaming.
- **The infrastructure layer**, which includes the data platforms, such as databases, and data warehouses, and storage and compute resources that support data workloads in hybrid environments.

Figure 1 illustrates the layers that data observability tracks.

Figure 1. Data Observability Observes Three Layers



This report defines data observability and its common use cases, then prescribes criteria that data teams can use to evaluate data observability products.

What is a Data Observability Product?

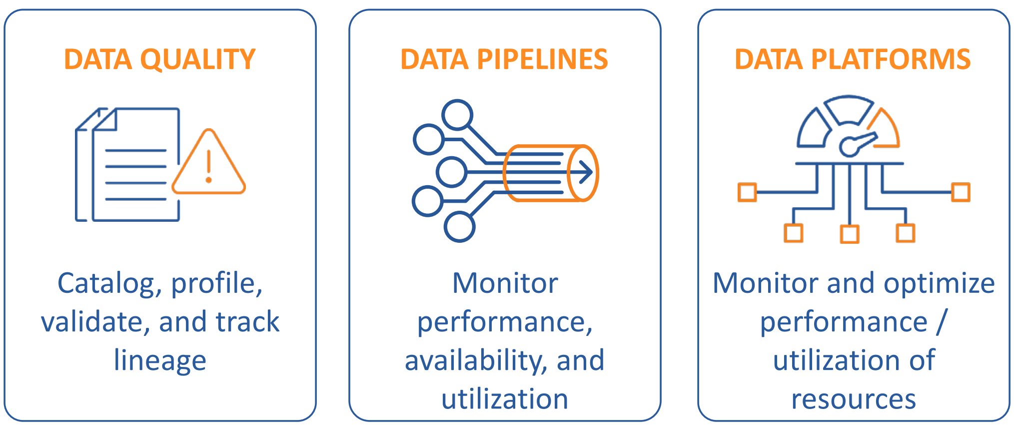
A data observability product is a tool that data teams, especially data engineers, use to ensure the quality and performance of data delivery to the application layer. It monitors, alerts, correlates, and analyzes logs, traces, and metrics across the application, data, and infrastructure layers. A data observability product uses automation and machine learning (ML) or other types of artificial intelligence (AI) to do the following:

- **Observe data quality** by cataloging, profiling, validating, and tracking the lineage of various datasets. This helps identify and remediate data quality issues. Vendors such as **BigEye** and **Anomalo** offer data observability products that focus on data quality.
- **Observe data pipelines** by monitoring the performance of data flows to help identify, remediate, and prevent performance and reliability issues.
- **Observe data platforms** by monitoring performance, availability, and utilization of data platforms that process data, as well as their linkages to supporting infrastructure. Vendors such as **Unravel** and **PepperData** offer data observability products that focus on infrastructure supporting data platforms.

Acceldata offers data observability software that addresses data quality, data pipelines, and data platforms.

Figure 2 illustrates the functionality of data observability products.

Figure 2. Data Observability Products



Use Cases

Before we dig into product evaluation criteria, let's explore common use cases for data observability products. The use cases fall into categories of data pipeline governance, data engineering, and IT planning.

Data Pipeline Governance

Data engineers work with governance officers and ML engineers to govern data quality and data drift.

- **Data quality**. They create and enforce rules to validate that target data is accurate, complete, and consistent. This means comparing source and target tables to identify null values, duplicate records, missing records, or altered schemas; then inspecting and fixing those issues. It also means cataloging, profiling, and tracking the lineage of data assets to assess their suitability for analytics.

- **Data drift**. They monitor data to detect two types of "data drift." First, they identify structural drift—changes to source structures such as schemas—and ensure those changes flow through to the target. Second, they identify ML model drift—degradations in the accuracy of model predictions due to changing business conditions—so they can re-train models. Data engineers help identify model drift by spotting changes to data inputs and features that make models less accurate.

Data Engineering

Data engineers, with the help of data architects, manage data pipeline performance and improve pipeline design.

- **Pipeline performance**. Pipelines must meet latency, throughput, and reliability SLAs. Data engineers correlate events, identify anomalies, and troubleshoot issues across the stack that affect pipeline performance. For example, they might spot an application slowdown, and trace the root cause to an overutilized Spark cluster. Then they allocate more CPU or memory to remove the bottleneck. Performance monitoring also helps tune pipelines and infrastructure as needed.
- **Pipeline design**. They also step back from daily firefights to design better pipelines and architectures. They use findings about performance and utilization to predict and prevent future bottlenecks. They select and configure pipeline components based on observed workload behavior, scenario modeling, and impact analysis.

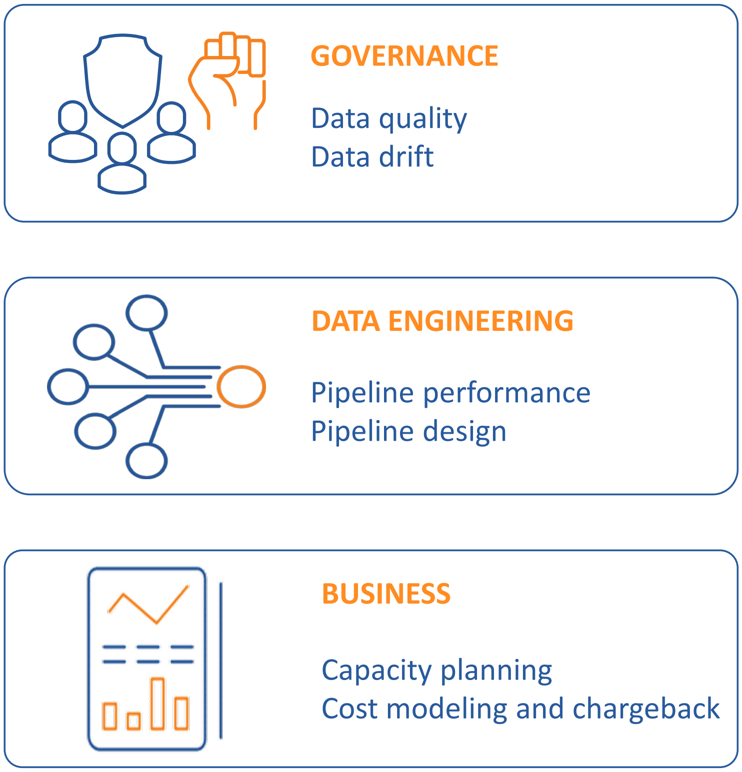
IT Planning

Data engineers work with platform engineers and site reliability engineers to assist capacity planning, and cost modeling and chargeback; then deliver their findings to business owners.

- **Capacity planning**. They measure and forecast the resources they need to meet SLAs for the business. They make these calculations based on available capacity, necessary buffer, and expected workload growth. They also forecast when starved resources or runaway workloads will create performance redlines.
- **Cost modeling and chargeback**. They model the implied operational cost of their SLAs, based on estimates such as aggregate compute, storage, memory, and data transfer requirements. They slice these estimates by time period, user group, geography, etc., to guide budgeting and chargeback decisions.

Figure 3 summarizes the primary use cases for data observability products.

Figure 3. Use Cases



Together, these use cases enable strategic data initiatives. They support BI, data science, and data-driven applications, as well as migrations of data and workloads to the cloud.

Five Product Evaluation Criteria

Now we dig into product evaluation criteria. Data engineering teams and their managers should evaluate data observability products according to five criteria: breadth of capabilities, data pipeline governance, heterogeneous integration, ease of use, and scale and performance.

Criterion 1. Breadth of Capabilities

A data observability product should support a wide range of use cases and capabilities to ensure data engineers can address the many moving parts in a modern enterprise environment. Ask the following questions to gauge the ability of candidate products to make this happen.

Does this Product Address the Data Pipeline Governance, Data Engineering, and Business Use Cases Outlined Above?

A data observability product should address all major enterprise use cases in these categories. Successful BI and data science projects depend on high-quality data that adapts to structural drift and model drift. They also depend on high-performance and well-designed data pipelines that give the business confidence in their SLAs. And business owners need confidence that data pipelines have the right capacity, meet cost targets, and align with chargeback requirements. Build an inventory of your specific use cases and assess whether candidate products address them all. It will need to provide comprehensive support, because data engineering teams don't have time to learn multiple data observability products or manage multiple products when troubleshooting issues.

A data observability product should address all major enterprise use cases for data pipeline governance, data engineering, and IT planning.

Does It Use AI/ML to Classify, Predict, and Prevent Issues/Events?

A data observability product should use native AI/ML models to proactively ensure data quality and timeliness. For example, a product might classify the risk of infrastructure bottlenecks based on utilization patterns and anomalies. In this scenario, it predicts performance by identifying "clusters" of similar historical outcomes according to various key performance indicators (KPIs). It also defines "if/then" relationships by associating different KPIs with one another. It might find that if CPU utilization for a given compute cluster and user group increases by 40 percentage points in one hour, then response times for those users are likely to exceed SLAs in the next two hours, which demands action.

ML techniques like these enable data observability products to predict issues, then recommend or even trigger preventive actions.

Does This Product Observe Data Workloads and Quality on Both a Real-Time and Historical Basis?

Data engineers and other stakeholders need both real-time and historical data observability. While the definition of real-time varies, in this context it means analysis within seconds or minutes—whatever the business requires for a given use case. The use cases of managing data quality, detecting drift, and managing pipeline performance all require real-time analysis of issues. Those use cases also require historical views and analysis of trends, such as those related to user behavior, data source accuracy, or compute utilization. The longer-range use cases of pipeline design, capacity planning, and cost modeling and chargeback, meanwhile, primarily depend on historical analytics.

Does It Analyze Logs, Metrics, and Traces?

A data observability product should cast a wide net to understand data quality and data workloads. It should analyze logs that describe events such as user logins, table read operations, and compute errors across the application, data, and infrastructure layers. It also should analyze metrics that quantify component characteristics such as CPU utilization levels, available storage capacity, and application throughput. It can extract metrics from logs, or derive them by calculating trends across logs. Finally, a data observability product should analyze traces, which describe sequences of events across components to support application tasks such as a BI tool query. Together logs, metrics, and traces provide granular views of pipeline and data health, and places them in the context of their surrounding architecture.

Does It Provide Granular Monitoring and Alerting?

Data engineers need to extract clear signals from a lot of noise. For example, they might automatically filter logs from containers, servers, or cloud compute nodes, so that their data observability product's dashboard displays just periodic metrics. But they might also configure automated alerts so that they know when a metric such as server utilization level hits a redline. And they might collaborate with governance officers to configure alerts of data quality issues based on scheduled comparisons of source and target tables for a given pipeline. A data observability product should support requirements like these with both default and custom options.

Does It Take Automated Action to Minimize Risks and Prevent or Resolve Issues?

Some use cases and workloads require automated action so data engineers can stay on top of issues. If a Spark processor slows down and maxes out its server CPUs for two minutes, data engineers might just want to shut that down automatically . A data observability product should help them configure automated actions such as this, using either default or custom options. It also should recommend ways data engineers can troubleshoot issues and integrate with incident management applications such as **Zendesk** to automatically trigger tickets for support desk personnel.

Criterion 2. Data Pipeline Governance

A data observability product needs to help govern data and reduce compliance risk. Ask the following questions to assess how well candidate products do this.

Does It Catalog Datasets?

Native data catalogs help data observability products ensure data quality. They enable data engineers and governance officers to discover, index, and search through metadata, then assess data quality and apply quality rules. This catalog feature also should have plans to integrate and share metadata with standalone enterprise data catalogs from vendors such as **Informatica Enterprise Data Catalog** and **Alation**, which cast a wider net across data environments.

How Does It Ensure Data Quality?

Data engineers and governance officers need to configure and apply consistent quality rules across data sets. They need a product that scans data in multiple locations on a scheduled basis to identify duplicate or contradictory datasets. It should detect anomalies such as null values, unexpected row counts, or high deviations from the mean, then help remediate those issues. Data engineers also need to automatically reconstruct and visualize the lineage of datasets. This helps them predict the impact of issues as they flow downstream to applications and workflow tools.

How Does It Detect and Respond to Data Drift?

Data engineers, and especially ML engineers, need to detect data drift to ensure the accuracy of analytical outputs. A data observability product should help them configure and activate drift detection rules, using either default or custom methods. For model drift, this might include basic checks for sudden value changes—for example, new minimum or maximum values in the range. The ML engineer can select the drift detection method, set the threshold, and configure an alert. They might also configure a notification to a workflow tool such as Airflow so it can halt downstream workflows when things go sideways.

Does This Product Protect Personally Identifiable Information?

Personally identifiable information (PII) can creep into the logs that data observability products analyze, and raise the risk of non-compliance with privacy regulations such as the **General Data Protection Regulation** (GDPR) and the **California Consumer Privacy Act** (CCPA). For example, application logs might include user passwords, credit card numbers, or encryption keys. Data engineers should be able to set filters within their data observability products to ensure their views exclude PII.

How Does It Authenticate and Authorize User Actions?

As with any analytics initiative, only authenticated users should be able to access a given data set, and they should only perform authorized actions on that data set. Look for data observability products that control actions by user, data source, and pipeline. Your data observability product should also integrate with third-party identity management tools such as **Okta**, **Onelogin**, **Azure Active Directory**, and **AWS Identity and Access Management (IAM)**, which provide one interface to authenticate users for many enterprise systems.

Can It Track User Actions to Assist Compliance Efforts?

Regulations such as GDPR, CCPA, and the **Health Insurance Portability and Accountability Act (HIPAA)** require enterprises to audit and document the actions they take with customer data. A data observability product should monitor and document any actions that users take with customer-related data and alert compliance officers of suspicious activity.

Criterion 3. Heterogeneous Integration

Data observability must ensure timely, accurate data delivery across heterogeneous environments. Ask the following questions to gauge the ability of candidate products to achieve this.

Does This Product Support a Broad Ecosystem of IT Components?

A data observability product should provide visibility into the data and pipelines that interact with all the popular components of the application, data, and infrastructure layers. Evaluate products based on their support for the key elements that touch your pipelines, ideally during a proof of concept. These elements might include the following:

- **Application layer**. This layer comprises BI tools, SQL client tools, data science platforms, notebooks, workflow tools, incident managers, and APM tools.
  - **BI tools** such as **Tableau**, **Looker**, and **Qlik** query data and help visualize the results.
  - **SQL client tools** such as **DbVisualizer** and **DataGrip** help write and manage structured query language (SQL) commands.
  - **Data science platforms** such as **Amazon SageMaker**, **Databricks**, and **Dataiku** help build, train, and deploy AI/ML models.
  - **Notebooks** such as **Jupyter** or **Apache Zeppelin** help develop and train AI/ML models.
  - **Workflow tools** such as **Airflow**, **Kubeflow**, and **Luigi** orchestrate sequences of application tasks based on query or model outputs.
  - **Incident managers** such as **Slack**, **PagerDuty**, and Zendesk handle notifications and remediation tasks based on data pipeline issues.
  - **APM tools** such as **New Relic** or **Dynatrace** also handle notifications and remediation tasks based on issues that data pipeline observability products identify.
- **Data pipeline layer**. This layer comprises data processors, pipeline tools, file formats, and data catalogs.
  - **Processors** such as Apache Kafka, Apache Spark, **ksqlDB**, and **Apache Flink** transform and analyze data.

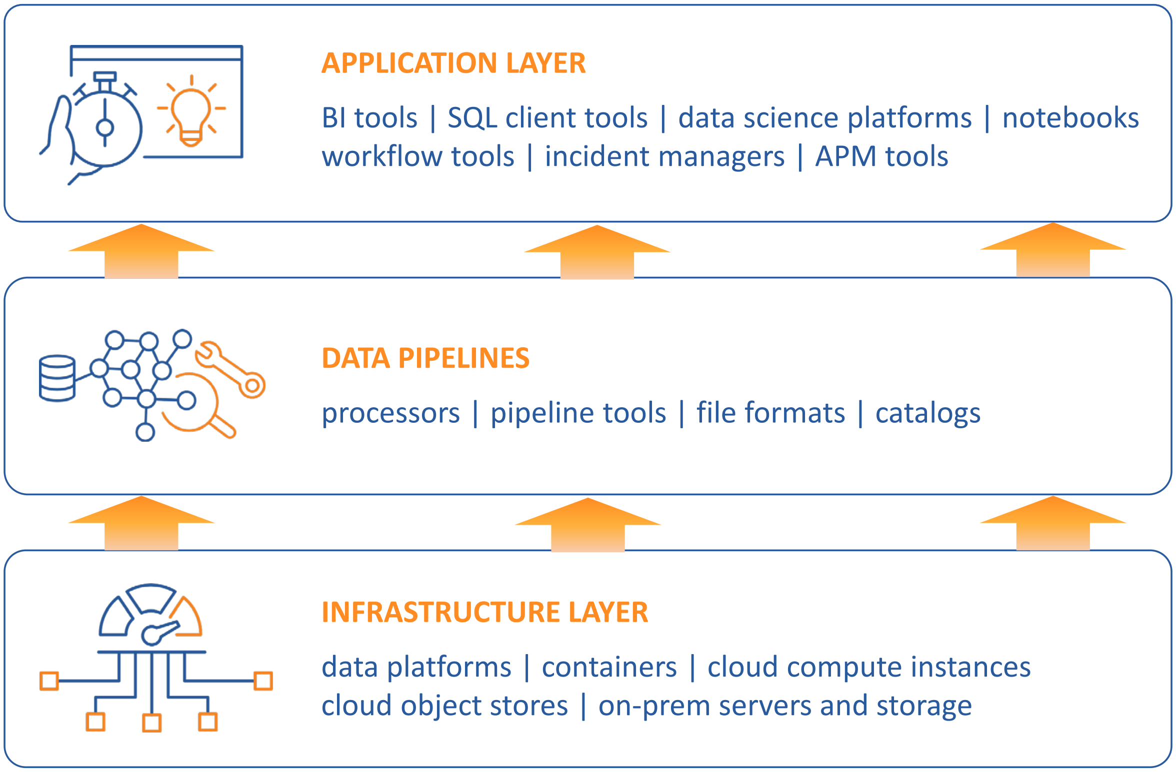
- **Pipeline tools** such as **dbt**, **Fivetran**, and **Qlik** help configure and manage data pipelines.
- **File formats** such as comma-separated values (CSV), **Apache optimized row columnar (ORC)** (for Apache Hive data), **Apache Parquet**, **Apache Avro**, and JavaScript object notation (**JSON**) organize data for storage, retrieval, and analysis.
- **Catalogs** such as Informatica Enterprise Data Catalog, Alation, and **Collibra** offer searchable inventories of metadata for data assets, as described earlier.

> **Infrastructure layer**. Data observability products must closely observe data platforms, as they have the highest impact on the performance and reliability of data workloads. They also observe any relevant interactions with supporting components such as containers, cloud compute, cloud storage, and on-premises servers and storage systems.

- **Data platforms** such as Snowflake and **Databricks** store, process, and present data to analytics tools and applications for consumption. They query data to support business intelligence and model it to support data science.
- **Containers** such as **Docker** and container orchestration tools such as **Kubernetes** bundle applications with system tools, libraries, and configuration files. This makes applications more modular, portable, and efficient as they consume infrastructure resources.
- **Cloud compute instances** such as Amazon Elastic Compute Cloud (**Amazon EC2**), **Azure Virtual Machines**, and **Google Cloud Compute Engine** provide virtualized servers that contain CPUs or GPUs, and memory resources to process data throughout the pipeline.
- **Cloud object stores** such as Amazon Simple Storage Service (**Amazon S3**), **Azure Blob Storage**, and **Google Cloud Storage** provide elastic capacity to store structured, semi-structured, and unstructured data objects.
- **On premises servers and storage systems** from vendors such as **Dell Technologies** and **HPE** provide compute and storage resources for enterprise data centers. They also combine, containerize, and virtualize these resources to provide cloud-like elasticity on premises.

Figure 4 illustrates the application, data, and infrastructure layers of the data observability ecosystem.

Figure 4. The Data Observability Ecosystem



How Does the Product Interact with the Application Layer?

The data observability product should visualize and share findings with various components of the application layer. For example, a data engineer might use Airflow to configure and view a data pipeline with RDBMS sources and Snowflake as a target. Then they toggle over to their observability product GUI to drill into that pipeline, including task status, execution time, and scores related to data quality or data drift. If some users prefer to interface with Slack or New Relic when troubleshooting issues, they should have the observability product send notifications to those tools.

Criterion 4. Ease of Use

The value of a data observability product hinges on the impact it has on the productivity of data engineering teams. The following questions help assess this value.

What Level of Effort is Required to Implement, Configure, and Monitor This Product in Your Environment?

Many skilled data engineers need scripting to create and manage specialized commands. But those same engineers—and a growing number of self-serve analysts and data stewards—can benefit from automating the simple and repetitive stuff. They need intuitive popup windows and dropdown menus that guide users through the necessary setup and configuration steps. The product should require one week or less of implementation and training for users to become proficient.

How Does This Product Present and Visualize Data Pipelines and Their Metadata?

This product should offer an intuitive, centralized view of all the data pipelines in an enterprise environment. Data engineers should be able to select one pipeline within that view, then pull up a summary of all its relevant metadata. This might include a list of its tables or other data elements, their lineage, the status of tasks such as merges and joins, and performance KPIs. That view also should summarize any relevant measures of data quality and drift. Then they should be able to drill into any of this metadata and learn more with a few clicks. A data observability product should make logical sequences like these natural and intuitive for users.

For example, a data engineer might scan a diagram of a pipeline—with blocks and arrows depicting the extraction, merging, and loading of key tables—then scroll down to see the status of pipeline tasks. They pull up windows containing a table of pipeline errors and their causes or graphical profiles of data elements, including value ranges and anomalies that might flag quality issues. They click on various tabs to inspect the quality of data elements, relationships between elements, their lineage, and the evolution of source schemas. Views like these give the data engineer confidence in the accuracy and timeliness of the data they deliver to the business.

How Does This Product Affect the Productivity of Your Data Teams, Data Engineers in Particular?

This product should make data engineers' jobs easier by reducing the time and effort required to meet business SLAs. It should offer a single source of truth that helps them collaborate with various colleagues—business owners, data analysts, data scientists, ML engineers, platform engineers, and site reliability engineers—to solve problems. Test the ability of candidate products to meet these requirements in a proof of concept. Devise realistic scenarios for your high priority use cases, and measure the execution and resolution times.

Is It Available as a Managed Service?

Seek out a data observability product that the data engineering team can use as a managed service if they so choose, either now or in the near future. By outsourcing the responsibility for software maintenance and upgrades, the team can focus on its core task of managing data quality and timeliness.

Criterion 5. Scale and Performance

A data observability product must scale to support large and growing data workloads, and it must perform by generating rapid insights about those workloads. Ask the following questions to evaluate whether candidate products meet these requirements.

Does This Product Support the Necessary Volume, Variety, and Velocity of Data?

A data observability product needs to collect logs, metrics, and traces from the myriad components that support data pipelines across the environment. The more sources, targets, pipelines, and other components in a given environment—and the faster those pipelines deliver data—the harder it is to observe everything. Put candidate products through a rigorous proof of concept that stress-tests their ability to observe big, complex, and fast environments at scale. Be sure to measure the impact on operational costs, especially cloud compute costs. Also ask vendors for referenceable clients for large data observability implementations, e.g., more than a petabyte of data or more than 100 compute nodes.

Can It Support Workload Spikes?

When data volumes surge—for example, during peak periods such as Black Friday, Cyber Monday, or quarter-end sales processing—that's when data engineers need data observability the most. During a proof of concept, test whether candidate products can keep pace with the logs, metrics, and traces that erupt at these times. These products should also monitor and control compute cycles as they scale to prevent unexpected compute bills from the cloud provider.

Does This Product Enable Fast Analysis and Response?

As described earlier, data engineers need real-time analytics, then automated alerts and actions to accelerate their response to critical issues. If the data observability product sees a spike in online customer transactions on Cyber Monday, the data engineer should identify this on their real-time dashboard so they can profile the workload and provision additional compute clusters. If it detects ML model drift for fraud detection, ML engineers should receive a real-time alert so they can intervene immediately to ensure risky transactions are not automatically approved. During a proof of concept, measure the time it takes to identify and resolve time-sensitive issues such as these. Was it fast enough to minimize damage to the business?

What is the Impact of This Product on the Enterprise Environment?

Data observability products should interoperate with their ecosystem in a non-intrusive way. Look for a product that uses open application programming interfaces (APIs) and enables users to change out components in a modular fashion as they scale its coverage of the environment. Agents and plug-ins, if required, should consume minimal resources and impose minimal processing burden on data sources or other components. Also assess the resource requirements and processing burden of any crawlers the product uses to collect metadata.

Becoming Observant

Your data observability product should empower your data engineers and their colleagues to improve the accuracy and timeliness of data delivery for your most strategic analytics initiatives. Selected and implemented well, it can enable your enterprise to derive higher value from your data assets while improving team productivity. But the devil lies in the details of product selection. Data engineers and their managers should evaluate how data observability products align with their specific use cases and requirements. Consider these guiding principles as you review your short list of products.

> **Define your baseline requirements**. Assemble those "must-have" capabilities in each of the five evaluation criteria: breadth of capabilities, data pipeline governance, heterogeneous integration, ease of use, and scale and performance. Filter out the data observability products that fail to address these requirements.

> **Stress-test candidate products**. During your proof of concept, test the ability of short-listed products to address your stretch goals, especially as they relate to heterogeneous integration, scale, and performance. Your environment will grow more complex, and your data volumes will rise in coming years.

> **Listen to your full team**. Both entry-level users and power users need to become more productive with your data observability product. Give them both a vote in the product selection process, including the proof of concept.

Explosive data supply and demand mean that modern enterprise analytics will continue to grow more complex. Choosing and implementing the right data observability product can help simplify things and put your most strategic analytics initiatives on the right path.

About Eckerson Group

Wayne Eckerson, a globally-known author, speaker, and consultant, formed **Eckerson Group** to help organizations get more value from data and analytics. His goal is to provide organizations with expert guidance during every step of their data and analytics journey.

Eckerson Group helps organizations in three ways:

> **Our thought leaders** publish practical, compelling content that keeps data analytics leaders abreast of the latest trends, techniques, and tools in the field.

> **Our consultants** listen carefully, think deeply, and craft tailored solutions that translate business requirements into compelling strategies and solutions.

> **Our advisors** provide competitive intelligence and market positioning guidance to software vendors to improve their go-to-market strategies.

Eckerson Group is a global research, consulting, and advisory firm that focuses solely on data and analytics. Our experts specialize in data governance, self-service analytics, data architecture, data science, data management, and business intelligence.

Our clients say we are hard-working, insightful, and humble. It all stems from our love of data and our desire to help organizations turn insights into action. We are a family of continuous learners, interpreting the world of data and analytics for you.

Get more value from your data. Put an expert on your side. **Learn what Eckerson Group can do for you!**

Eckerson Group

RESEARCH CONSULTING ADVISORY

GET • MORE • VALUE • FROM • YOUR • DATA

About the Sponsor

Acceldata is the industry's first Unified Data Observability platform for analytics and AI systems, enabling enterprises globally to thrive in a data-driven world. With the ability to observe, optimize, and scale complex data pipelines, enterprises leverage Acceldata's Observability platform to deliver seamless analytics and AI success in the cloud and on-premises. GE, True Digital, Walmart PhonePe, Michelin, PubMatic, DBS, and many other global enterprises are Acceldata customers that optimize their data success. We invite you to visit us at **Acceldata.io** and follow us on **LinkedIn.**