While there’s widespread consensus that data quality is very important, data observability is a relatively new field. Even when organizations prioritize their data pipelines, they are likely still newcomers to data observability and all of its terms and context. That’s why we’ve compiled this dictionary based on common terms we encounter in our day-to-day work. Hopefully it can help you understand some key definitions, differentiations, and components of data observability.

Analyst

An analyst or data analyst examines data to help make business decisions. The meaningful results they pull from the raw data help their company and customers make important decisions by identifying important facts and trends.

Anomaly detection

Detecting data points, events, and/or information that falls outside of a dataset’s normal behavior. [Anomaly detection](https://www.bigeye.com/blog/anomaly-detection-part-1-the-key-to-effective-data-observability) helps companies flag areas that might have issues in their data pipelines.

Business intelligence

Business intelligence is the practice of using data and analytics to understand what has happened, what is happening now, and why it is happening. It focuses on data from current and past events in order to help you predict the future.

CI/CD

Continuous integration (CI): Continuous Integration is a software development practice where members of a team integrate their work frequently, with each integration verified by an automated build (including test) to detect integration errors as quickly as possible.

Continuous Deployment (CD): Continuous deployment is a software practice where after each code commit, the project is automatically deployed.

Circuit breaker

Circuit breaker is a generic term that refers to a mechanism that stops a system when it receives a certain signal. In the data context, it’s been used to refer to stopping data pipelines when certain data quality tests fail. This prevents low-quality data from percolating into downstream processes.

Dashboard

A dashboard is an information management tool that visually tracks, analyzes, and displays metrics and key data points to monitor the health of a business, department, or specific process.

Database

A database is an organized form of data for easy access, storage, retrieval and management of data.

Popular databases include MySQL, Postgres (Relational) and MongoDB (NoSQL).

Data freshness

Data freshness refers to how up-to-date data is, e.g. the amount of time since a data table was last refreshed.

Data governance

Data governance is a collection of processes, roles, policies, standards, and metrics that ensure the quality and security of the data used across a business or organization. Data governance defines who can take what action, upon what data, in what situations, and using what methods.

Data lake

A data lake is a central location that holds a large amount of data in its native, raw format. Compared to a hierarchical data warehouse, which stores data in files or folders, a data lake uses a flat architecture and object storage to store the data.‍

Popular data lakes include Databricks Data Lake and Azure Data Lake.

Data mart

A highly structured repository where data is stored and managed until it’s needed. It differs from a data warehouse in scope: data warehouses are capable of serving as the central store of data for an entire business. A data mart fulfills the request of a specific division or business function.

Data mesh

Data mesh is a strategic approach for managing your data. Data mesh is a response to traditional centralized data management methods, through data warehouses and data lakes. Instead, data mesh emphasizes decentralization, allocating data ownership to domain-specific groups that can serve, own, and manage data.

Data migration

Data migration refers to moving your data from one location or application to another – for example, if a company is moving from Firestore as a database to Postgres. This process generally involves a lot of preparation and post-migration activities including planning, creating backups, quality testing, and validation of results.

Data monitoring

Data monitoring refers to periodically querying data sources (i.e. tables in data warehouses) to determine their state (their freshness, volume, quality, etc), and then surfacing this state information in the form of dashboards or alerts.

Data observability

[Data observability](https://www.bigeye.com/blog/what-is-data-observability-heres-what-you-need-to-know) is a method for achieving data reliability. It is the ability of an organization to truly *see* the health of all of their data. Observability platforms give a comprehensive view into the state of data and data pipelines in real-time.

Data observability specifically encompasses:

* monitoring the operational health of the data to ensure it’s fresh and complete
* detecting and surfacing anomalies that could indicate data accuracy issues
* mapping data lineage from the source all the way through to downstream tables or applications to more quickly identify root cause issues and better understand their impacts

Data pipeline

A set of actions to ingest raw data from disparate sources and move the data to a destination for storage and analysis. A pipeline also may include filtering and features that provide resiliency against failure.

Data quality

The standard for whether data is fit for use in analytics, whether that is a dashboard, a scheduled export, or an ML model.

* Data quality is not a binary state. Good and poor data quality can vary depending on the intended use for the data.
* Data quality has three phases: operational quality, logical quality, and application quality.

Data reliability

Data reliability, a term inspired by [Google's Site Reliability Engineering](https://sre.google/), refers to the work of creating standards, process, alignment, and tooling to keep data applications—like dashboards and ML models—reliable. This includes data pipeline test automation, manual task automation, data pipeline SLIs/SLOs/SLAs, and data incident management.

Data Reliability Engineer

A professional who acts as a steward over the quality of data and reliability of the data process. Data reliability engineers are typically responsible for building data pipelines to bring together information from different source systems. They integrate, consolidate and [cleanse data](https://searchdatamanagement.techtarget.com/definition/data-scrubbing) and structure it for use in analytics applications. They aim to make data easily accessible and to optimize their organization's [big data](https://searchdatamanagement.techtarget.com/definition/big-data) ecosystem. Data Engineers contribute, generally, to Data Observability by testing all the known edges and capturing bugs before release of code. By doing so, Data Reliability Engineers can focus on true anomalies.

Data replication

Data replication refers to keeping multiple copies of your data and keeping all of them updated.

Data source

A data source is where the data being used is located. A data source is commonly referred to as a data source name (DSN), which is defined in the application so that it can find the location of the data.

Data testing

Data testing refers to a manual process by which someone who has knowledge of the data, expresses a specific condition about the data, and checks that it’s true. For example, that there are no nulls in a table in a database.

Data warehouse

A data warehouse is a system used for storing and reporting on data. The data typically originates in multiple systems (e.g. a production database), then it is moved into the data warehouse for long-term storage and analysis.

Popular data warehouses include Snowflake, Redshift, and BigQuery.

DevOps and DataOps

DevOps is the practice of versioning, reviewing, and automating the release of code. It provides clear and consistent pathways from the engineer’s machine to a production environment. Continuous Integration/Continuous Deployment is a pattern in the DevOps practice. Infrastructure as code is another DevOps pattern. The goal of DevOps is to increase the speed at which code can be developed and released; scale to meet growing teams who are working on the same code; ensure reliability of released code; and ensure the reliability, restartability, resilience, and repeatability of the infrastructure to which that code is deployed.

As with application products, data products are increasingly being versioned in central repositories and released through CI/CD pipelines – this is the foundation of the DataOps pattern. Distributed and cloud based data platforms also require more complex infrastructure strategies like IAC. DataOps further expands on the DevOps practice by adding other patterns, like Data Observability that are essential to the Data Lifecycle.

Distribution

A distribution is a statistical term that refers to a function that shows the possible values for a variable and how often they occur.

ETL (Extract Transform Load)

ETL is a type of data integration that refers to the three steps (extract, transform, load) used to blend data from multiple sources. It's often used to build a data warehouse. ETL pipelines are often set and require an understanding of how the data should look and be transformed before landing in the data warehouse.

ELT (Extract Load Transform)

ELT is the process of extracting data from one or multiple sources and loading it into a target data warehouse. Instead of transforming the data before it's written, ELT takes advantage of the target system to do the data transformation. More and more companies are turning to ELT and investing in tools like Snowflake, Fivetran, DBT, and Airflow. In the world of ELT, traditional data quality processes break down.

Ingestion

Data ingestion is the process of absorbing data from a multitude of sources, and then transferring it to a target site where it can be analyzed and deposited. Data ingestion is a broader term than ETL or ELT, which focuses on data warehouses.

Instrumentation

Instrumentation refers to the process of collecting summary data by which you’ll observe your data systems. The four core blocks are metadata/logs, metrics, lineage, and deltas. In our case Bigeye is the tool the customer would use to collect data.

Lineage

Data lineage is the path that data takes through your data system, from creation, through any databases and transformation jobs, all the way down to final destinations like analytics dashboards and feature stores. Data lineage is an important tool for data observability because it provides context - it tells you:

* For each data pipeline job, which dataset it’s reading from and which dataset it’s writing to.
* Whether problems are localized to just one dataset, or cascade down your pipeline.

Logs

Logs are records of what happened when, that are generated by data systems. Logs can be unstructured or structured.

Machine Learning Engineer

A machine learning engineer/ML engineer is a role focused on researching, building and designing artificial intelligence (AI) systems to feed and automate predictive models.

Metrics

In data observability, metrics are numeric results that come from directly querying a dataset. In the data observability context, they often take the form of a time series.

Toil

Toil in a data/software context refers to the kind of work tied to running a production data service that tends to be manual, repetitive, automatable and that scales linearly as a service grows.

Tracing

Distributed tracing is a method of observing requests as they propagate through distributed cloud environments. Distributed tracing follows an interaction by tagging it with a unique identifier. This identifier stays with the transaction as it interacts with microservices, containers, and infrastructure.

Schema

A database schema is an abstract design that represents the storage of your data in a database. It describes both the organization of data and the relationships between tables in a given database.

SLA (Service Level Agreement)

SLA is a feature in the Bigeye platform that enables data teams to create trust with stakeholders and improve data culture. SLA combines multiple data quality metrics into a single group that tracks the health of a key asset like a query, dashboard, or machine learning model to establish transparency and ensure clear communication.

* SLA is based on tried-and-true practices from SRE and DevOps. [Read here to learn more about SLAs for data observability.](https://medium.com/bigeye/defining-data-quality-with-slas-ea79e916ebc7)
* The data observability metrics collected in SLA are made up of SLIs and SLOs that are put together into an overall agreement on what constitutes data quality for a specific piece of analytics, whether that’s a dashboard or query output.
* For example, an SLA with a 99.9% uptime guarantee (or a 0.1% error budget) allows for [43 minutes and 50 seconds](https://www.atlassian.com/incident-management/kpis/error-budget) of downtime each month. By viewing the SLA in Bigeye, the end user has a concrete way to know whether the data meets their expectations, and what level of reliability they should expect week to week or month to month.

SLI (Service Level Indicator)

SLIs are an important part of an SLA. SLIs measure specific aspects of performance. SLIs might be “hours since dataset refreshed” or “percentage of values that match a UUID regex.” When building an SLA (see definition) for a specific use case, the SLIs should be chosen based on what data the use case relies on. If an ML model can tolerate some null IDs but not too many, the rate of null IDs is a great SLI to include in the SLA for that model.

SLO (Service Level Objective)

SLOs give each SLI a target range. For example, the relevant SLOs could be “less than 6 hours since the dataset was refreshed” or “at least 99.9% of values match a UUID regex.” As long as an SLI is within the range set by its SLO, it’s considered acceptable for the target use case, and that aspect of its parent SLA is being met.

Streaming data

Streaming data is the continuous flow of data generated by various sources. By using stream processing technology, data streams can be processed, stored, analyzed, and acted upon as it's generated in real-time.

Popular streaming data platforms include Kafka, Confluent, Google Cloud Pub/Sub, and AWS Kinesis.

Structured, semi-structured, and unstructured data

Structured data is data that most often follows a tabular convention and is most often stored in conventional databases. Parquet is an example of structured data that is file based and also supports hierarchical data. Structured data enforces schema. Schema changes for structured data require a change to the table in which the data is stored. In the case of Parquet, schema is stored in the metadata of the file and incompatible schema amongst files is handled by the consumer and the read layer. For structured data, the key goals for Data Reliability Engineers, when implementing their data observability strategy, is to monitor values to ensure that those values conform to business rules.

Semi-structured data does not follow a tabular convention and, while able to be stored in conventional databases, is usually stored in NoSQL databases or in object stores or file systems. Examples of semi-structured data include: CSV, JSON, YAML, AVRO, Semi-structured data conforms, generally, to schema in the sense that the application publishing the data is bound to an object definition. Changes to the schema are code changes and are generally versioned. Different applications and different versions of applications may be writing the same semi-structured data to the same object store. In addition to monitoring values, Data Reliability Engineers should pay strict attention to schema drift and have a schema registry and schema evolution standard in place when implementing their data observability strategy.

Unstructured data is a loose concept and is generally applied to data that is not readable by a computer. Examples include: word documents, emails, picture files, log dumps, etc. No schema exists for unstructured data and, therefore, unstructured data cannot be observed for anomaly in traditional ways. Often, Data Reliability Engineers target commonality of use case to determine if the unstructured data conforms. Sometimes, machine learning is used to determine fit for use. A good example is a filter of image data based on an image containing certain attributes – like an airplane.

Volume

Data volume is the amount of data that you have, or the amount that’s being created within a certain timeframe, e.g. the number of rows being inserted into a table per day.