Benefits of data engineering on the lakehouse

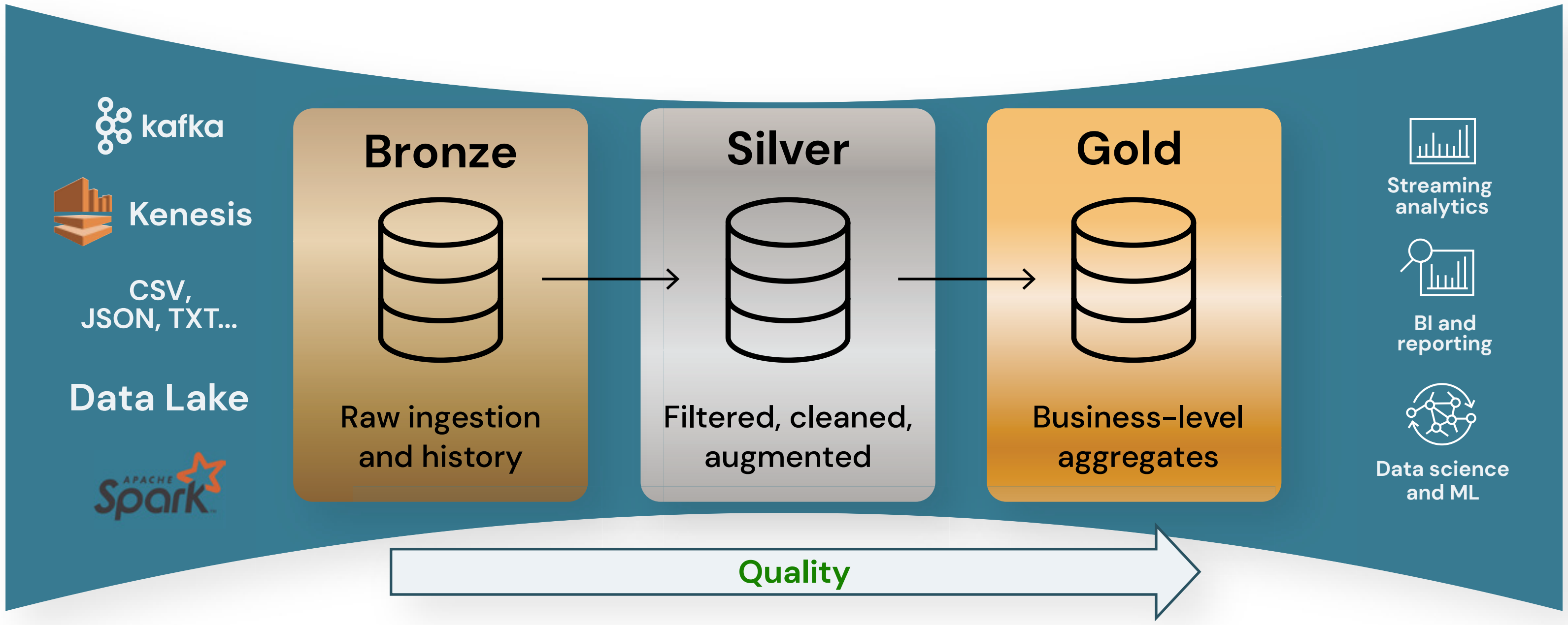
By simplifying and modernizing with the lakehouse architecture, data engineers gain an enterprise-grade and enterprise-ready approach to building data pipelines. The following are nine key differentiating capabilities that a data engineering solution team can enable with the Databricks Data Intelligence Platform:

- **Easy data ingestion:** With the ability to ingest petabytes of data, data engineers can enable fast, reliable, scalable and automatic data ingestion for analytics, data science or machine learning.
- **Automated ETL pipelines:** Data engineers can reduce development time and effort and focus on implementing business logic and data quality checks within the data pipeline using SQL or Python.
- **Data quality checks:** Improve data reliability throughout the data lakehouse so data teams can confidently trust the information for downstream initiatives with the ability to define data quality and automatically address errors.
- **Batch and streaming:** Allow data engineers to set tunable data latency with cost controls without having to know complex stream processing and implement recovery logic.
- **Automatic recovery:** Handle transient errors and use automatic recovery for most common error conditions that can occur during the operation of a pipeline with fast, scalable fault-tolerance.
- **Data pipeline observability:** Monitor overall data pipeline estate status from a dataflow graph dashboard and visually track end-to-end pipeline health for performance, quality, status and latency.
- **Simplified operations:** Ensure reliable and predictable delivery of data for analytics and machine learning use cases by enabling easy and automatic data pipeline deployments into production or roll back pipelines and minimize downtime.
- **Scheduling and orchestration:** Simple, clear and reliable orchestration of data processing tasks for data and machine learning pipelines with the ability to run multiple non-interactive tasks as a directed acyclic graph (DAG) on a Databricks compute cluster.
- **Natural Language Assistant:** Databricks assistant makes data engineers more productive with Text2SQL, document summary, code completion and generation, code explanation, and debugging.

Data engineering is all about data quality

The goal of modern data engineering is to distill data with a quality that is fit for downstream analytics and AI. Within the lakehouse, data quality is achieved on three different levels.

1. On a **technical level**, data quality is guaranteed by enforcing and evolving schemas for data storage and ingestion.
2. On an **architectural level**, data quality is often achieved by implementing the medallion architecture. A medallion architecture is a data design pattern used to logically organize data in a lakehouse with the goal of incrementally and progressively improving the structure and quality of data as it flows through each layer of the architecture, e.g., from Bronze to Silver to Gold layer tables.
3. The **Databricks Unity Catalog** comes with robust data quality management with built-in quality controls, testing, monitoring and enforcement to ensure accurate and useful data is available for downstream BI, analytics and machine learning workloads.



Data ingestion

With the Databricks Data Intelligence Platform, data engineers can build robust hyper-scale ingestion pipelines in streaming and batch mode. They can incrementally process new files as they land on cloud storage — with no need to manage state information — in scheduled or continuous jobs.

Data engineers can efficiently track new files (with the ability to scale to billions of files) without having to list them in a directory. Databricks automatically infers the schema from the source data and evolves it as the data loads into the Delta Lake lakehouse. Efforts continue with enhancing and supporting Auto Loader, our powerful data ingestion tool for the lakehouse.

What is Auto Loader?

Have you ever imagined that ingesting data could become as easy as dropping a file into a folder? Welcome to Databricks Auto Loader.

Auto Loader is an optimized data ingestion tool that incrementally and efficiently processes new data files as they arrive in the cloud storage built into the Databricks Data Intelligence Platform. Auto Loader can detect and enforce the schema of your data and, therefore, guarantee data quality. New files or files that have been changed since the last time new data was processed are identified automatically and ingested. Non-compliant data sets are quarantined into rescue data columns. You can use the [trigger once] option with Auto Loader to turn it into a job that turns itself off.

Ingestion for data analysts: COPY INTO

Ingestion also got much easier for data analysts and analytics engineers working with Databricks SQL. COPY INTO is a simple SQL command that follows the lake-first approach and loads data from a folder location into a Delta Lake table. COPY INTO can be scheduled and called by a job repeatedly. When run, only new files from the source location will be processed.

Data transformation

Turning SQL queries into production ETL pipelines typically involves a lot of tedious, complicated operational work. Even at a small scale, the majority of a data practitioner's time is spent on tooling and managing infrastructure.

Although the medallion architecture is an established and reliable pattern for improving data quality, the implementation of this pattern is challenging for many data engineering teams.

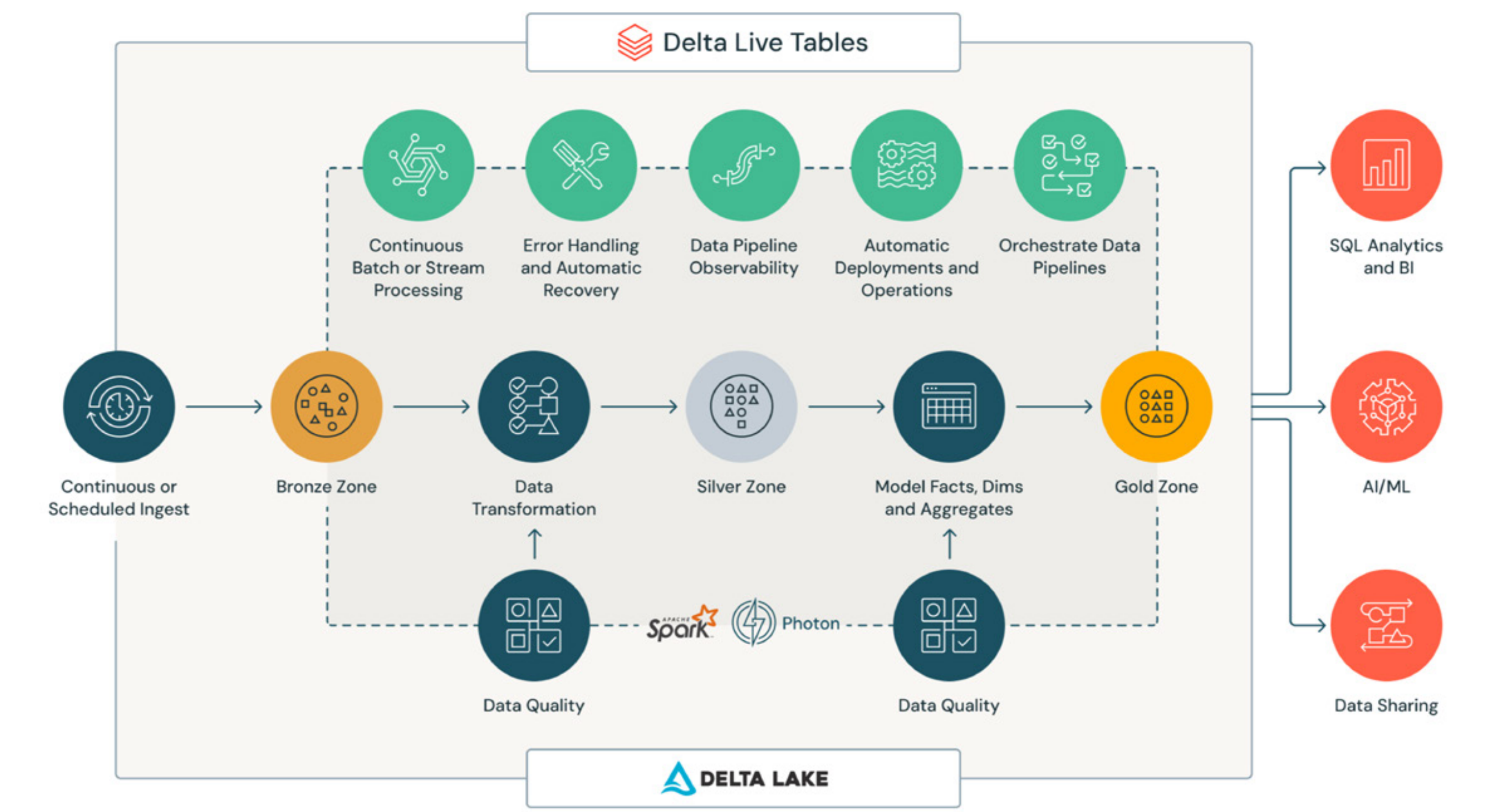
While hand-coding the medallion architecture was hard for data engineers, creating data pipelines was outright impossible for data analysts not being able to code with Spark Structured Streaming in Scala or Python.

Even at a small scale, most data engineering time is spent on tooling and managing infrastructure rather than transformation. Auto-scaling, observability and governance are difficult to implement and, as a result, often left out of the solution entirely.

What is Delta Live Tables?

Delta Live Tables (DLT) is a declarative ETL framework that helps data teams simplify and make ETL cost-effective in streaming and batch. Simply define the transformations you want to perform on your data and let DLT pipelines automatically handle task orchestration, cluster management, monitoring, and data quality and error management. Engineers are able to **treat their data as code** and apply modern software engineering best practices like testing, error-handling, monitoring and documentation to deploy reliable pipelines at scale. DLT fully supports both Python and SQL and is tailored to work with both streaming and batch workloads.

With DLT you write a Delta Live Table in a SQL notebook, create a pipeline under Workflows and simply click [Start].

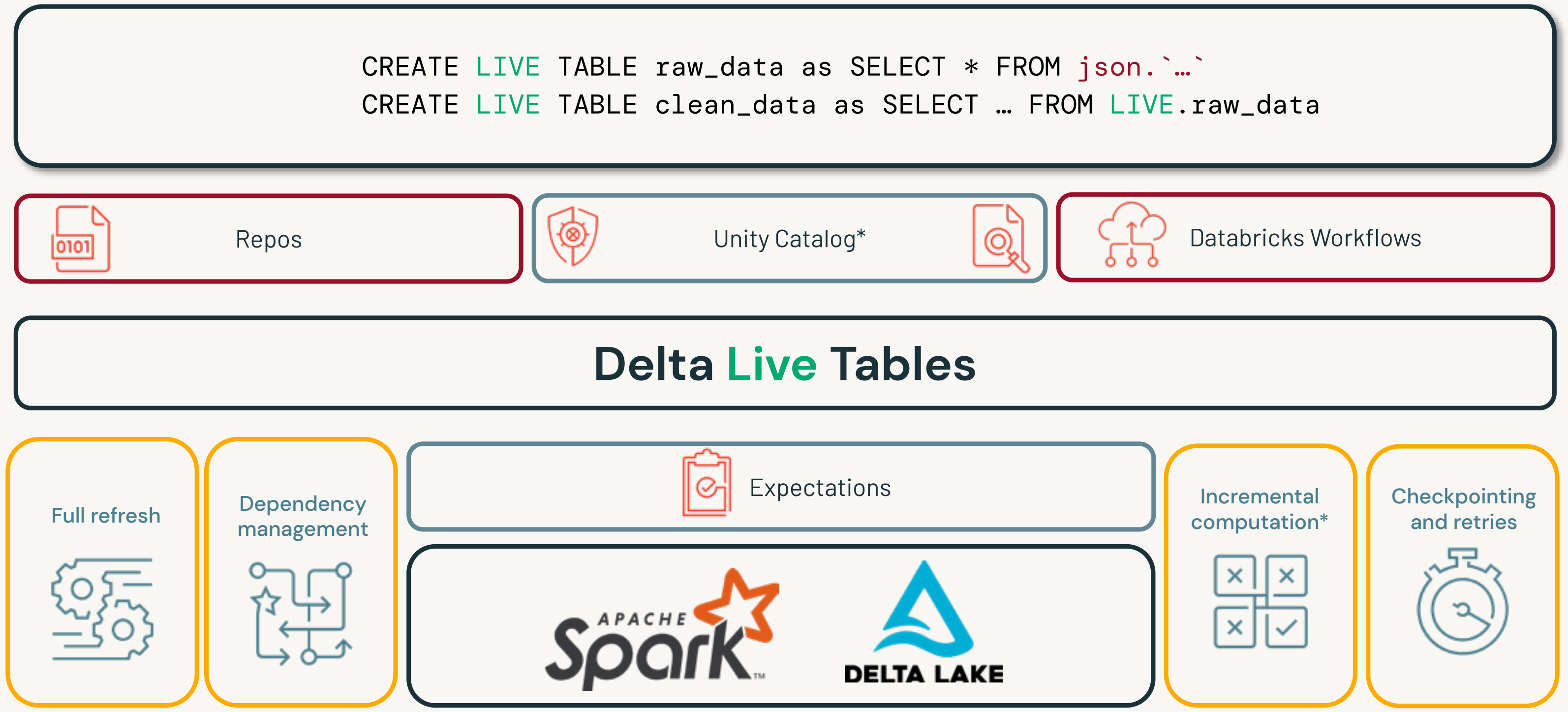

Delta Live Tables
Continuous Batch or Stream Processing
Error Handling and Automatic Recovery
Data Pipeline Observability
Automatic Deployments and Operations
Orchestrate Data Pipelines
Continuous or Scheduled Ingest
Bronze Zone
Data Transformation
Silver Zone
Model Facts, Dims and Aggregates
Gold Zone
Data Quality
Apache Spark
Photon
Data Quality
SQL Analytics and BI
AI/ML
Data Sharing
DELTA LAKE

DLT reduces the implementation time by accelerating development and automating complex operational tasks. Since DLT can use plain SQL, it also enables data analysts to create production pipelines and turns them into the often discussed "analytics engineer." At runtime, DLT speeds up pipeline execution applied with Photon.

Software engineering principles are applied for data engineering to foster the idea of treating your data as code. Your data is the sole source of truth for what is going on inside your business.

Beyond just the transformations, there are many things that should be included in the code that define your data. Declaratively express entire data flows in SQL or Python. Natively enable modern software engineering best practices like separate development and production environments, the ability to easily test before deploying, deploy and manage environments using parameterization, unit testing and documentation.

DLT also automatically scales compute, providing the option to set the minimum and maximum number of instances and let DLT size up the cluster according to cluster utilization. In addition, tasks like orchestration, error handling and recovery, and performance optimization are all handled automatically.



```
CREATE LIVE TABLE raw_data as SELECT * FROM json.`…`
CREATE LIVE TABLE clean_data as SELECT … FROM LIVE.raw_data
```

*Coming soon

Expectations in the code help prevent bad data from flowing into tables, track data quality over time, and provide tools to troubleshoot bad data with granular pipeline observability. This enables a high-fidelity lineage diagram of your pipeline to track dependencies and aggregate data quality metrics across all your pipelines.

Unlike other products that force you to deal with streaming and batch workloads separately, DLT supports any type of data workload with a single API so data engineers and analysts alike can build cloud-scale data pipelines faster without the need for advanced data engineering skills.

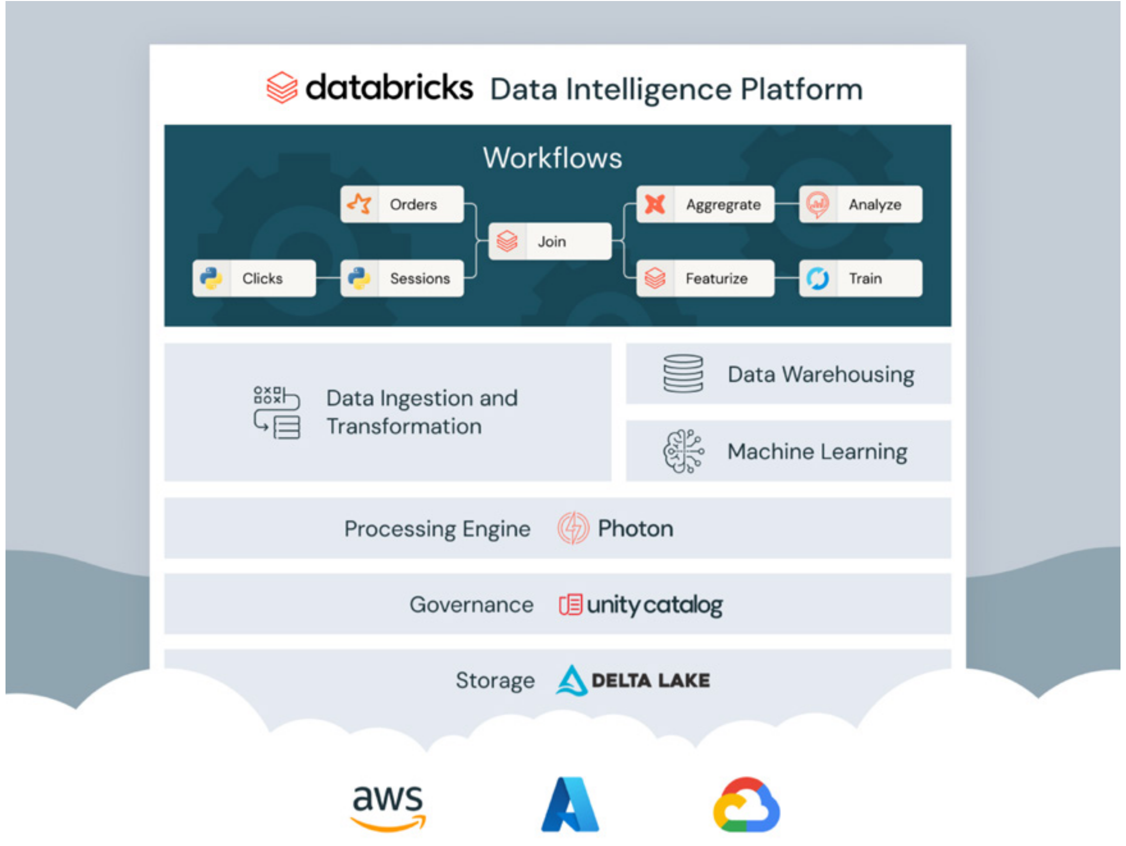
Data orchestration

The lakehouse makes it much easier for businesses to undertake ambitious data and machine learning (ML) initiatives. However, orchestrating and managing end-to-end production workflows remains a bottleneck for most organizations, relying on external tools or cloud-specific solutions that are not part of their lakehouse architecture. Tools that decouple task orchestration from the underlying data processing platform reduce the overall reliability of their production workloads, limit observability, and increase complexity for end users.

What is Databricks Workflows?

Databricks Workflows is the first fully managed and integrated lakehouse orchestration service that allows data teams to build reliable workflows on any cloud.

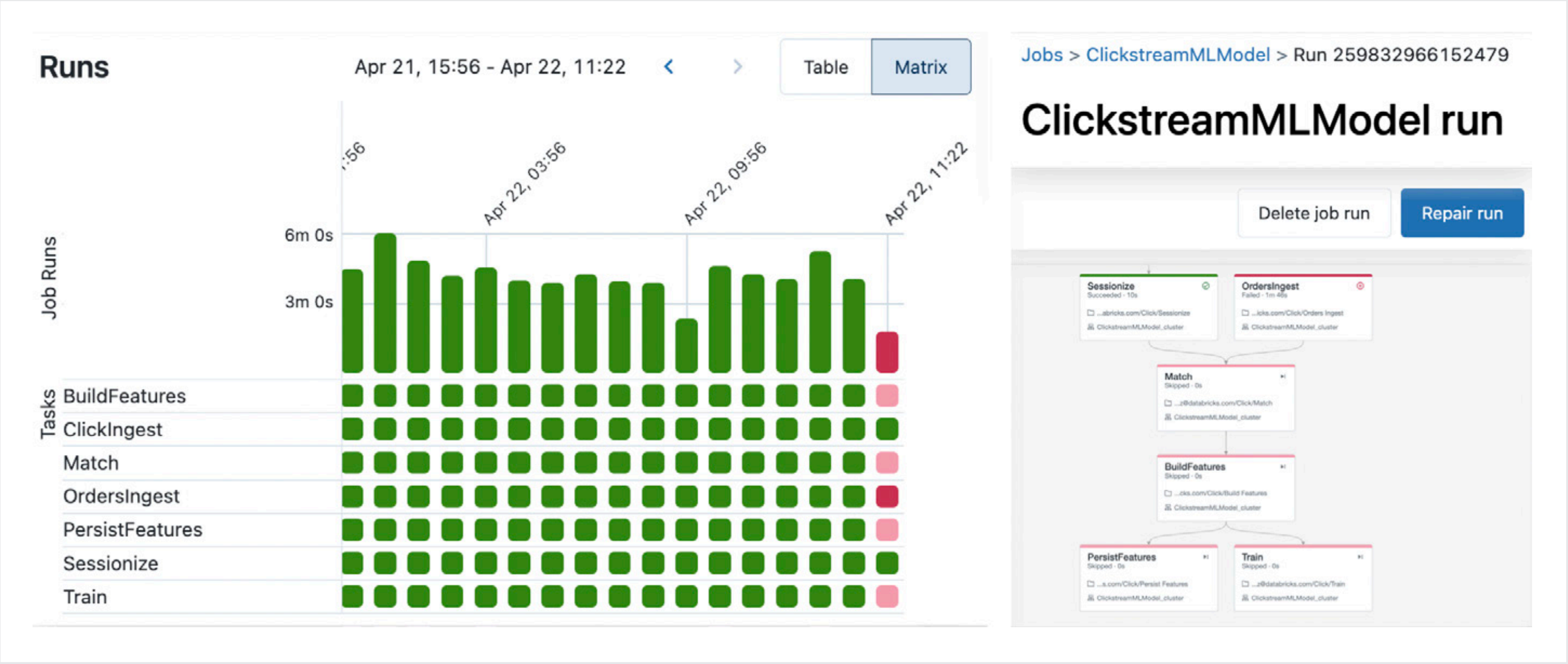
Workflows lets you orchestrate data flow pipelines (written in DLT or dbt), as well as machine learning pipelines, or any other tasks such as notebooks or Python wheels. Since Databricks Workflows has serverless compute, fully managed and reliable it eliminates operational overhead for data engineers, enabling them to focus on your workflows not on managing your infrastructure. It provides an easy point-and-click authoring experience for all your data teams, not just those with specialized skills. Deep integration with the underlying lakehouse architecture ensures you will create and run reliable production workloads on any cloud while providing deep and centralized monitoring with simplicity for end users.



Serverless Workflows provides a hands-off approach, fast start up times (<60s), automatic failover, and cost optimized mode.

Databricks Workflows' deep integration with the Databricks Data Intelligence Platform can best be seen with its monitoring and observability features. The matrix view in the following graphic shows a history of runs for a job. Failed tasks are marked in red. A failed job can be repaired and rerun with the click of a button. Rerunning a failed task detects and triggers the execution of all dependent tasks.

You can create workflows with the UI, but also through the Databricks Workflows API, or with external orchestrators such as Apache Airflow. Even if you are using an external orchestrator, Databricks Workflows' monitoring acts as a single pane of glass that includes externally triggered workflows.

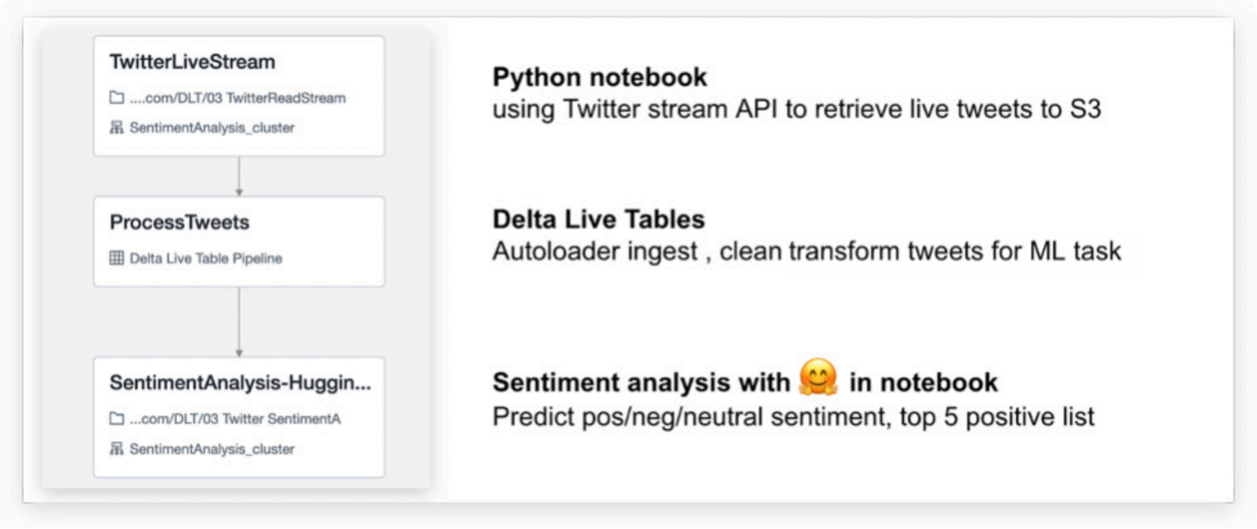

Runs
Apr 21, 15:56 - Apr 22, 11:22
Table
Matrix
:56
Apr 22, 03:56
Apr 22, 09:56
Apr 22, 11:22
Job Runs
6m 0s
3m 0s
Tasks
BuildFeatures
ClickIngest
Match
OrdersIngest
PersistFeatures
Sessionize
Train
Jobs > ClickstreamMLModel > Run 259832966152479
ClickstreamMLModel run
Delete job run
Repair run
Sessionize
Succeeded · 10s
...abricks.com/Click/Sessionize
ClickstreamMLModel_cluster
OrdersIngest
Failed · 1m 46s
...icks.com/Click/Orders Ingest
ClickstreamMLModel_cluster
Match
Skipped · 0s
...z@databricks.com/Click/Match
ClickstreamMLModel_cluster
BuildFeatures
Skipped · 0s
...cks.com/Click/Build Features
ClickstreamMLModel_cluster
PersistFeatures
Skipped · 0s
...s.com/Click/Persist Features
ClickstreamMLModel_cluster
Train
Skipped · 0s
...z@databricks.com/Click/Train
ClickstreamMLModel_cluster

Orchestrate anything

Remember that DLT is one of many task types for Databricks Workflows. This is where the managed data flow pipelines with DLT tie together with the easy point-and-click authoring experience of Databricks Workflows.

In the following example, you can see an end-to-end workflow built with customers in a workshop: Data is streamed from Twitter according to search terms, then ingested with Auto Loader using automatic schema detection and enforcement. In the next step, the data is cleaned and transformed with Delta Live table pipelines written in SQL, and finally run through a pre-trained BERT language model from Hugging Face for sentiment analysis of the tweets. Different task types for ingest, cleanse/transform and ML are combined in a single workflow.

Using Workflows, these tasks can be scheduled to provide a daily overview of social media coverage and customer sentiment for a business. After streaming tweets with filtering for keywords such as "data engineering," "lakehouse" and "Delta Lake," we curated a list of those tweets that were classified as positive with the highest probability score.


TwitterLiveStream
....com/DLT/03 TwitterReadStream
SentimentAnalysis_cluster
ProcessTweets
Delta Live Table Pipeline
SentimentAnalysis-Huggin...
....com/DLT/03 Twitter SentimentA
SentimentAnalysis_cluster

Python notebook
using Twitter stream API to retrieve live tweets to S3

Delta Live Tables
Autoloader ingest , clean transform tweets for ML task

Sentiment analysis with 🤗 in notebook
Predict pos/neg/neutral sentiment, top 5 positive list

Learn more

Data Engineering on the Databricks Data Intelligence Platform

Big Book of Data Engineering

Delta Live Tables

Databricks Workflows

CHAPTER

09

Data streaming?

There are two types of data processing: batch processing and streaming processing.

Batch processing refers to the discontinuous, periodic processing of data that has been stored for a period of time. For example, an organization may need to run weekly reports on a set of predictable transaction data. There is no need for this data to be streaming — it can be processed on a weekly basis.

Streaming processing, on the other hand, refers to unbounded processing of data as it arrives.

In a wide variety of cases, an organization might find it useful to leverage streaming data. Here are some common examples:

- **Retail:** Real-time inventory updates help support business activities, such as inventory and pricing optimization and optimization of the supply chain, logistics and just-in-time delivery.
- **Smart energy:** Smart meter monitoring in real time allows for smart electricity pricing models and connection with renewable energy sources to optimize power generation and distribution.
- **Preventative maintenance:** By reducing unplanned outages and unnecessary site and maintenance visits, real-time streaming analytics can lower operational and equipment costs.
- **Industrial automation:** Manufacturers can use streaming and predictive analytics to improve production processes and product quality, including setting up automated alerts.
- **Healthcare:** To optimize care recommendations, real-time data allows for the integration of various smart sensors to monitor patient condition, medication levels and even recovery speed.
- **Financial institutions:** Firms can conduct real-time analysis of transactions to detect fraudulent transactions and send alerts. They can use fraud analytics to identify patterns and feed data into machine learning algorithms.

Regardless of specific use cases, the central tenet of streaming data is that it gives organizations the opportunity to leverage the freshest possible insights for better decision-making and more optimized customer experiences.

Data Streaming Challenges

However, getting value from streaming data can be a tricky practice. While most data today can be considered streaming data, organizations are overwhelmed by the need to access, process and analyze the volume, speed and variety of this data moving through their platforms. To keep pace with innovation, they must quickly make sense of data streams decisively, consistently and in real time.

Three common technical challenges organizations experience with implementing real-time data streaming include:

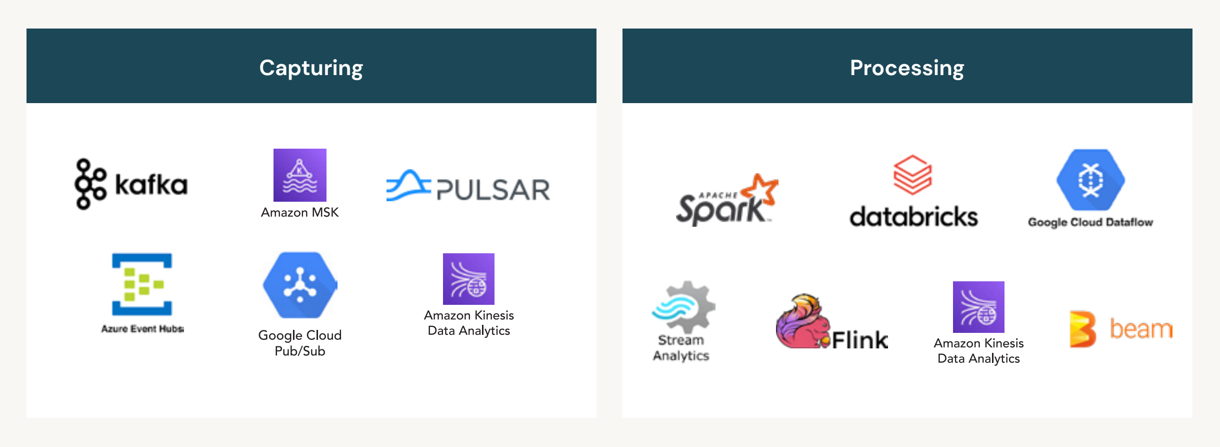
- **Specialized APIs and language skills:** Data practitioners encounter barriers to adopting streaming skillsets because there are new languages, APIs and tools to learn.
- **Operational complexity:** To implement data streaming at scale, data teams need to integrate and manage streaming-specific tools with their other cloud services. They also have to manually build complex operational tooling to help these systems recover from failure, restart workloads without reprocessing data, optimize performance, scale the underlying infrastructure, and so on.
- **Incompatible governance models:** Different governance and security models across real-time and historical data platforms makes it difficult to provide the right access to the right users, see the end-to-end data lineage, and/or meet compliance requirements.

Data streaming architecture

Before addressing these challenges head-on, it may help to take a step back and discuss the ingredients of a streaming data pipeline. Then, we will explain how the Databricks Data Intelligence Platform operates within this context to address the aforementioned challenges.

Every application of streaming data requires a pipeline that brings the data from its origin point — whether sensors, IoT devices or database transactions — to its final destination.

In building this pipeline, streaming architectures typically employ two layers. First, streaming capture systems **capture** and temporarily store streaming data for processing. Sometimes these systems are also called messaging systems or messaging buses. These systems are optimized for small payloads and high frequency inputs/outputs. Second, streaming **processing** systems continuously process data from streaming capture systems and other storage systems.



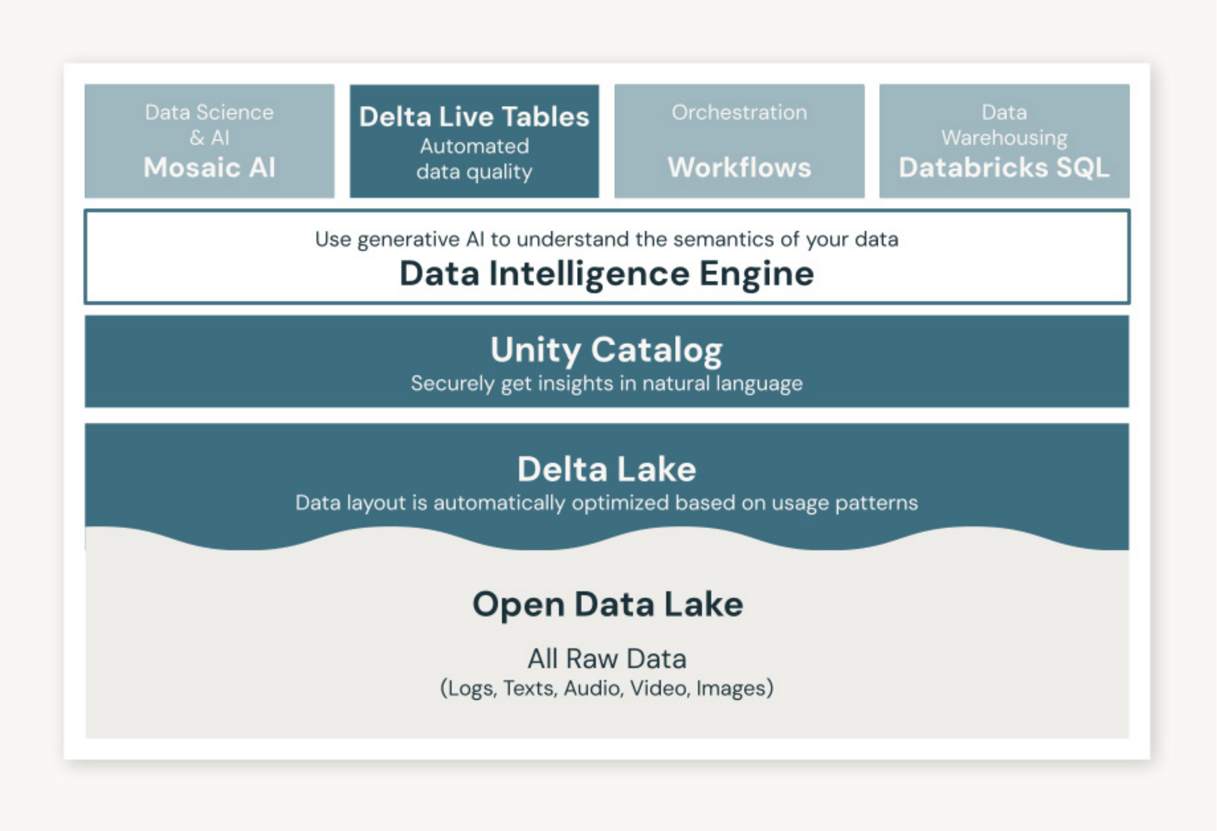
It may help to think of a simplified streaming pipeline according to the following seven phases:

1. Data is continuously generated at origin points
2. The generated data is captured from those origin points by a capture system like Apache Kafka (with limited retention)
3. **The captured data is extracted and incrementally ingested to a processing platform like Databricks; data is ingested exactly once and stored permanently, even if this step is rerun**
4. **The ingested data is converted into a workable format**
5. **The formatted data is cleansed, transformed and joined in a number of pipeline steps**
6. **The transformed data is processed downstream through analysis or ML modeling**
7. The resulting analysis or model is used for some sort of practical application, which may be anything from basic reporting to an event-driven software application

You will notice four of the steps in this list are in boldface. This is because the lakehouse architecture is specifically designed to optimize this part of the pipeline. Uniquely, the Databricks Data Intelligence Platform can ingest, transform, analyze and model on streaming data *alongside* batch-processed data. It can accommodate both structured *and* unstructured data. It is here that the value of unifying the best pieces of data lakes and data warehouses really shines for complex enterprise use cases.

Data Streaming on the Databricks Data Intelligence Platform

Now let's zoom in a bit and see how the Databricks Data Intelligence Platform addresses each part of the pipeline mentioned above.


Data Science & AI
Mosaic AI
Delta Live Tables
Automated data quality
Orchestration
Workflows
Data Warehousing
Databricks SQL
Use generative AI to understand the semantics of your data
Data Intelligence Engine
Unity Catalog
Securely get insights in natural language
Delta Lake
Data layout is automatically optimized based on usage patterns
Open Data Lake
All Raw Data
(Logs, Texts, Audio, Video, Images)

Streaming data ingestion and transformation begins with continuously and incrementally collecting raw data from streaming sources through a feature called Auto Loader. Once the data is ingested, it can be transformed from raw, messy data into clean, fresh, reliable data appropriate for downstream analytics, ML or applications. Delta Live Tables (DLT) makes it easy to build and manage these data pipelines while automatically taking care of infrastructure management and scaling, data quality, error testing and other administrative tasks. DLT is a high-level abstraction built on Spark Structured Streaming, a scalable and fault-tolerant stream processing engine.

Real-time analytics refers to the downstream analytical application of streaming data. With fresher data streaming into SQL analytics or BI reporting, more actionable insights can be achieved, resulting in better business outcomes.

Real-time ML involves deploying ML models in a streaming mode. This deployment is supported with structured streaming for continuous inference from a live data stream. Like real-time analytics, real-time ML is a downstream impact of streaming data, but for different business use cases (i.e., AI instead of BI). Real-time modeling has many benefits, including more accurate predictions about the future.

Real-time applications process data directly from streaming pipelines and trigger programmatic actions, such as displaying a relevant ad, updating the price on a pricing page, stopping a fraudulent transaction, etc. There typically is no human-in-the-loop for such applications.

Databricks Data Intelligence Platform differentiators

Understanding what the lakehouse architecture provides is one thing, but it is useful to understand how Databricks uniquely approaches the common challenges mentioned earlier around working with streaming data.

Databricks empowers unified data teams. Data engineers, data scientists and analysts can easily build streaming data workloads with the languages and tools they already know and the APIs they already use.

Databricks simplifies development and operations. Organizations can focus on getting value from data by reducing complexity and automating much of the production aspects associated with building and maintaining real-time data workloads.

Databricks is one platform for streaming and batch data. Organizations can eliminate data silos, centralize security and governance models, and provide complete support for all their real-time use cases under one roof — the roof of the lakehouse.

Finally — and perhaps most important — Delta Lake, the core of the Databricks Data Intelligence Platform, was built for streaming from the ground up. Delta Lake is deeply integrated with Spark Structured Streaming and overcomes many of the limitations typically associated with streaming systems and files.

In summary, the Databricks Data Intelligence Platform dramatically simplifies data streaming to deliver real-time analytics, machine learning and applications on one platform. And, that platform is built on a foundation with streaming at its core. This means organizations of all sizes can use their data in motion and make more informed decisions faster than ever.

See why customers love streaming on the Databricks Data Intelligence Platform with these resources.

Learn more

Data Streaming Webpage

Project Lightspeed: Faster and Simpler Stream Processing With Apache Spark

Structured Streaming Documentation

Streaming — Getting Started With Apache Spark on Databricks

CHAPTER

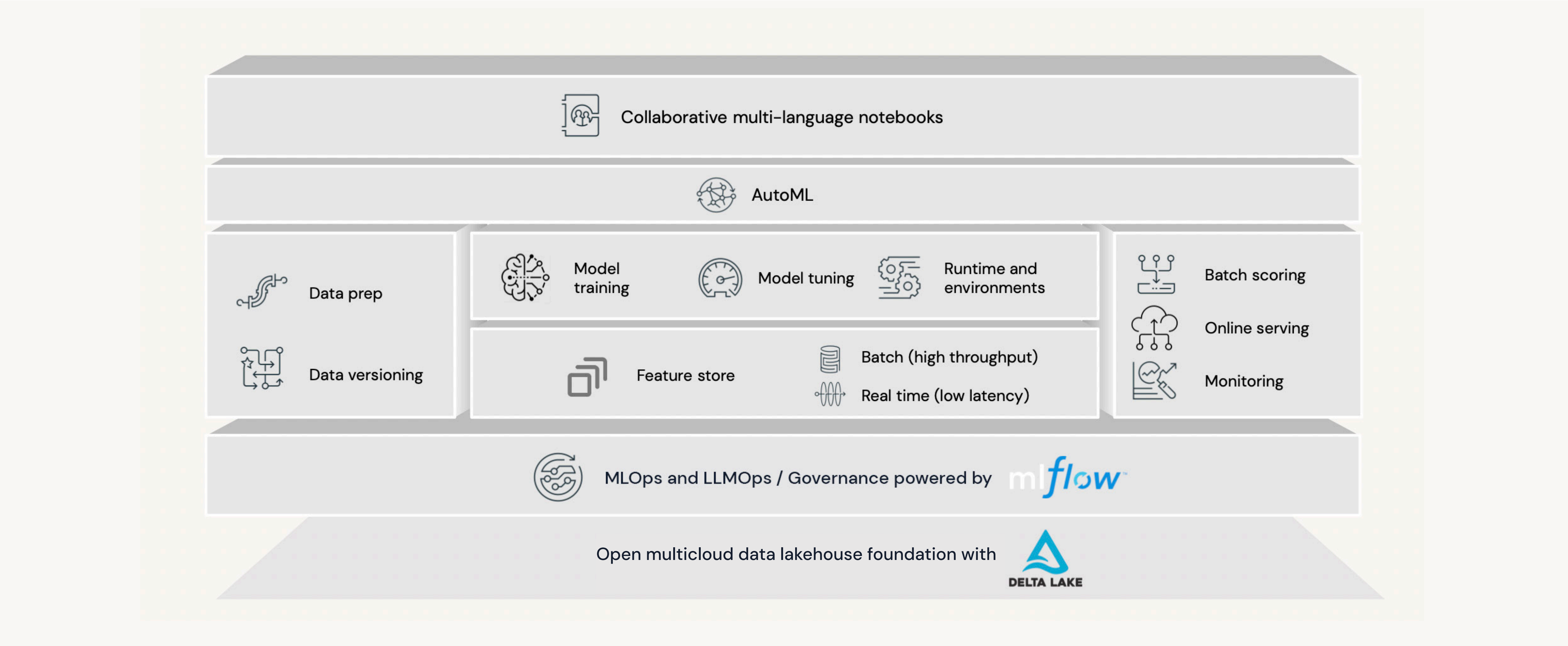
10

Data science, AI and machine learning

While most companies are aware of the potential benefits of applying machine learning and AI, realizing these potentials can often be quite challenging for those brave enough to take the leap. Some of the largest hurdles come from siloed/disparate data systems, complex experimentation environments, and getting models served in a production setting.

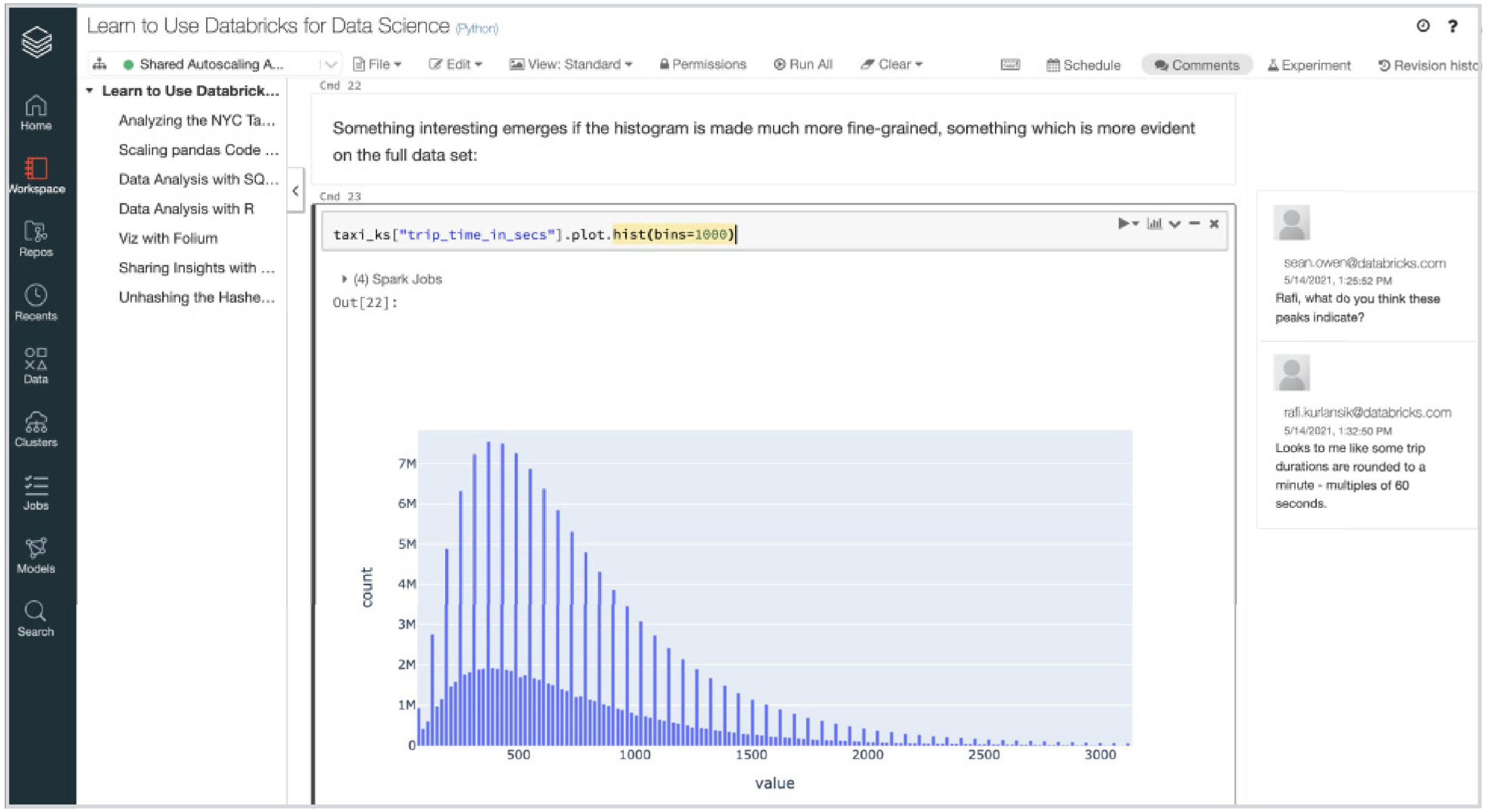
Fortunately, the Databricks Data Intelligence Platform provides a helping hand and lets you use data to derive innovative insights, build powerful predictive and generative AI models, and enable data scientists, ML engineers, and developers of all kinds to create within the space of machine learning and AI.

Databricks Machine Learning



Exploratory data analysis

With all the data in one place, data is easily explored and visualized from within the notebook-style experience that provides support for various languages (R, SQL, Python and Scala) as well as built-in visualizations and dashboards. Confidently and securely share code with co-authoring, commenting, automatic versioning, Git integrations and role-based access controls. The platform provides laptop-like simplicity at production-ready scale.



Something interesting emerges if the histogram is made much more fine-grained, something which is more evident on the full data set:

```
taxi_ks["trip_time_in_secs"].plot.hist(bins=1000)
```

```
Out[22]:
```

sean.owen@databricks.com
5/14/2021, 1:25:52 PM
Rafi, what do you think these peaks indicate?

rafi.kurlansik@databricks.com
5/14/2021, 1:32:50 PM
Looks to me like some trip durations are rounded to a minute - multiples of 60 seconds.

Model creation and management

From data ingestion to ML model and LLM training and tuning, all the way through to production model serving and versioning, Mosaic AI brings the tools needed to simplify those tasks.

Get right into experimenting with the Databricks ML runtimes, optimized and preconfigured to include most popular libraries like scikit-learn, XGBoost and more. Also, experiment with the foundation models on Databricks like DBRX, Llama 2, Mistral, or foundation models hosted elsewhere like GPT-4, Claude 3, Cohere, and Stable Diffusion. Massively scale thanks to built-in support for distributed training and hardware acceleration with GPUs.

From within the runtimes, you can track model training sessions, package and reuse models easily with MLflow, an open source machine learning platform created by Databricks and included as a managed service within Mosaic AI. It provides a centralized location from which to manage models and package code in an easily reusable way.

Training these models often involves the use of features housed in a centralized feature store. Fortunately, Databricks has a built-in feature store that allows you to create new features, explore and re-use existing features, select features for training and scoring machine learning models, and publish features to low-latency online stores for real-time inference.

If you are looking to get a head start, AutoML allows for low to no-code experimentation by pointing to your data set and automatically training models and tuning hyperparameters to save both novice and advanced users precious time in the machine learning process.

AutoML will also report back metrics related to the model training results as well as the code needed to repeat the training already custom-tailored to your data set. This glass box approach ensures that you are never trapped or suffer from vendor lock-in.

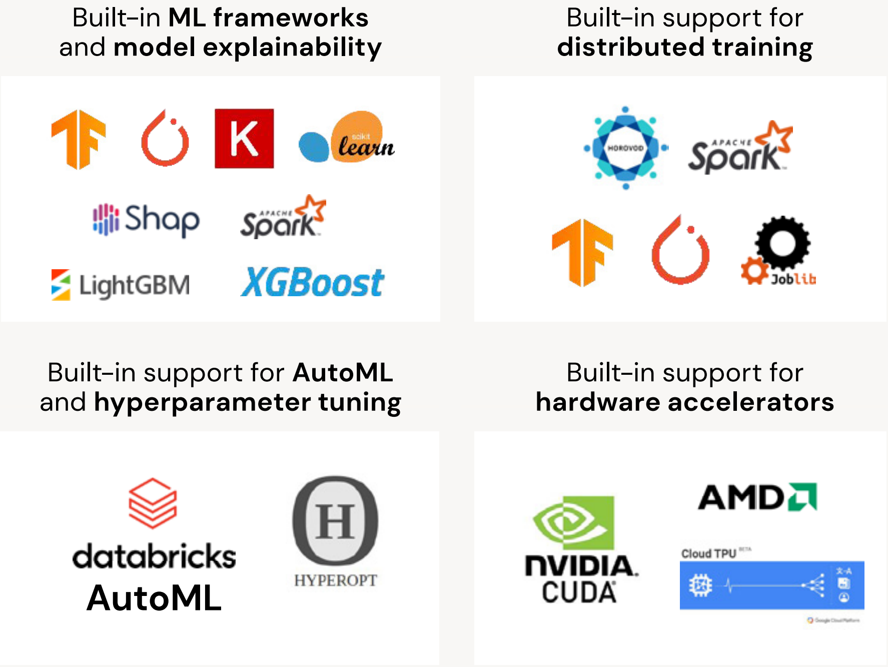
In that regard, Mosaic AI supports the industry's widest range of data tools, development environments, and a thriving ISV ecosystem so you can make your workspace your own and put out your best work.

Compute platform

Any ML workload optimized and accelerated

Databricks Machine Learning Runtime

- Optimized and preconfigured ML frameworks
- Turnkey distribution ML
- Built-in AutoML
- GPU support out of the box


Built-in **ML frameworks** and **model explainability**

Built-in support for **distributed training**

Built-in support for **AutoML** and **hyperparameter tuning**

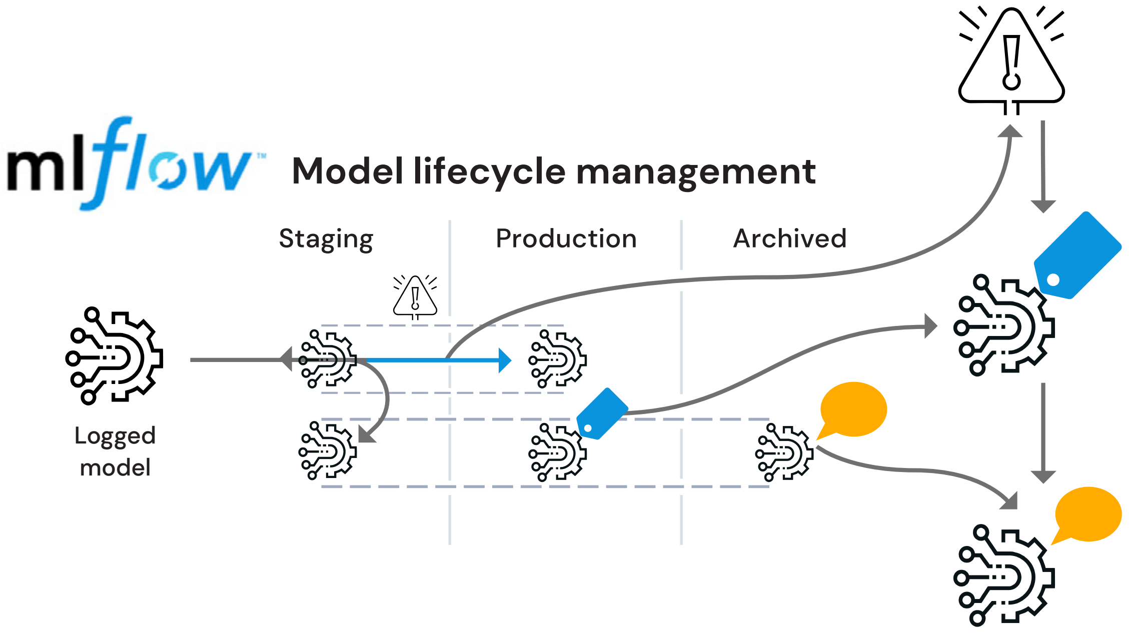
Built-in support for **hardware accelerators**

Deploy your models to production

Exploring and creating your machine learning and large language models typically represent only part of the task. Once the models exist and perform well, they must become part of a pipeline that keeps models updated, monitored and available for use by others.

Databricks can help here by providing a world-class experience for model versioning, monitoring and serving within the same platform that you can use to generate the models themselves. This means you can make all your ML and GenAI pipelines in the same place, monitor them for drift, retrain them with new data, and promote and serve them easily and at scale.

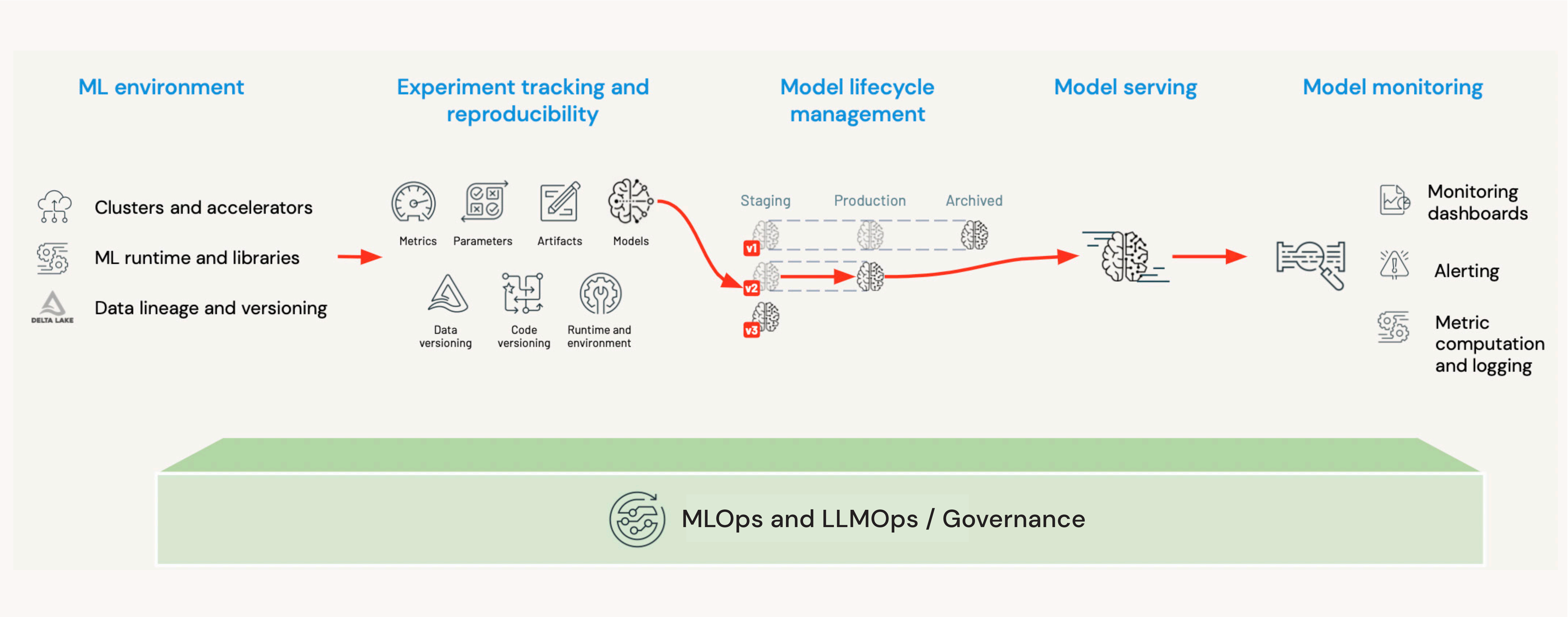
Throughout the ML lifecycle, rest assured knowing that lineage and governance are being tracked the entire way. This means regulatory compliance and security woes are significantly reduced, potentially saving costly issues down the road.



Webhooks allow registering of callbacks on events like stage transitions to integrate with CI/CD automation.

Tags allow storing deployment — specific metadata with model versions, e.g., whether the deployment was successful.

Comments allow communication and collaboration between teammates when reviewing model versions.



Learn more

Databricks Machine Learning

Databricks Data Science

Databricks ML Runtime Documentation

databricks