



# Big Data Ecosystem

Technical Deep Dive – Day4

**Sashank Pappu**  
Azure Data & AI Specialist

**Shiva Priya**  
Azure Data Engineer



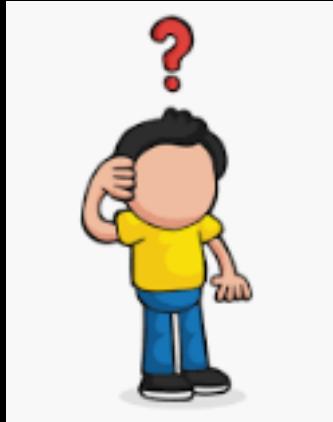
# What shall we learn today ?

- Azure Data Bricks Understanding & Scalability
- Data Bricks Delta
- Data Lake Integration with Data Bricks
- Snow Flake with Data Bricks
- Streaming Data using Event Hubs & Data Bricks

# More Detailed Agenda

- 9-10 AM : Understanding on Databricks
- 10-11:30 : Read & Write Data Bricks
- 11:30-11:45 : BREAK !!!
- 11:45-1PM : Databricks Delta
- 1-2PM : LUNCH
- 2-2:30 : Delta Part 2
- 2:30-2:50 : BREAK!!!
- 3:15-5 : Streaming with Databricks

# Distributed storage PaaS on Azure



Errr....  
What are all these?  
Why so many?  
When do I use these?  
Wait, now you have this new thing called \*Delta\*?  
I am dizzy and ready to pass out!

## MPP database



Azure SQL Datawarehouse

## Object store

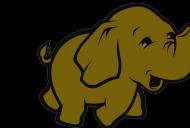


Azure  
Data Lake  
Store



Azure  
Blob Storage

## No-SQL



Azure HDInsight-HBase



Azure Cosmos DB  
SQL API  
Cassandra API  
MongoDB API

# Scalable OLTP, IoT store, time series database

Fit for purpose

## MPP database



Azure SQL Datawarehouse

## Object store



Azure  
Data Lake  
Store



Azure  
Blob Storage

## No-SQL



Azure HDInsight-HBase



Azure Cosmos DB  
SQL API  
Cassandra API  
MongoDB API

Optimized for low latency - point reads,  
random CRUD, small range queries/scans,  
transaction guarantees

# Datawarehousing

Fit for purpose 

## MPP database



Azure SQL Datawarehouse

Optimized for parallel processing of structured data with T-SQL, server side programmability, in-memory processing and massive scale. Complement with an RDBMS based reporting mart for supporting direct query for BI – for concurrency beyond Azure SQL Data warehouse concurrency limits.  
Alternately complement with Azure Analysis Services (AAS) for BI

## No-SQL



Azure HDInsight-HBase



Azure Cosmos DB  
SQL API  
Cassandra API  
MongoDB API

## Object store



Azure  
Data Lake  
Store



Azure  
Blob Storage

# Data Lake

## Fit for purpose

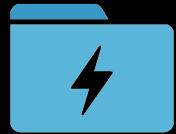
### MPP database



Azure SQL Datawarehouse

Optimized for structured and unstructured data storage, massively scalable; RBAC.

### Object store



Azure  
Data Lake  
Store



Azure  
Blob Storage

### No-SQL



Azure HDInsight-HBase

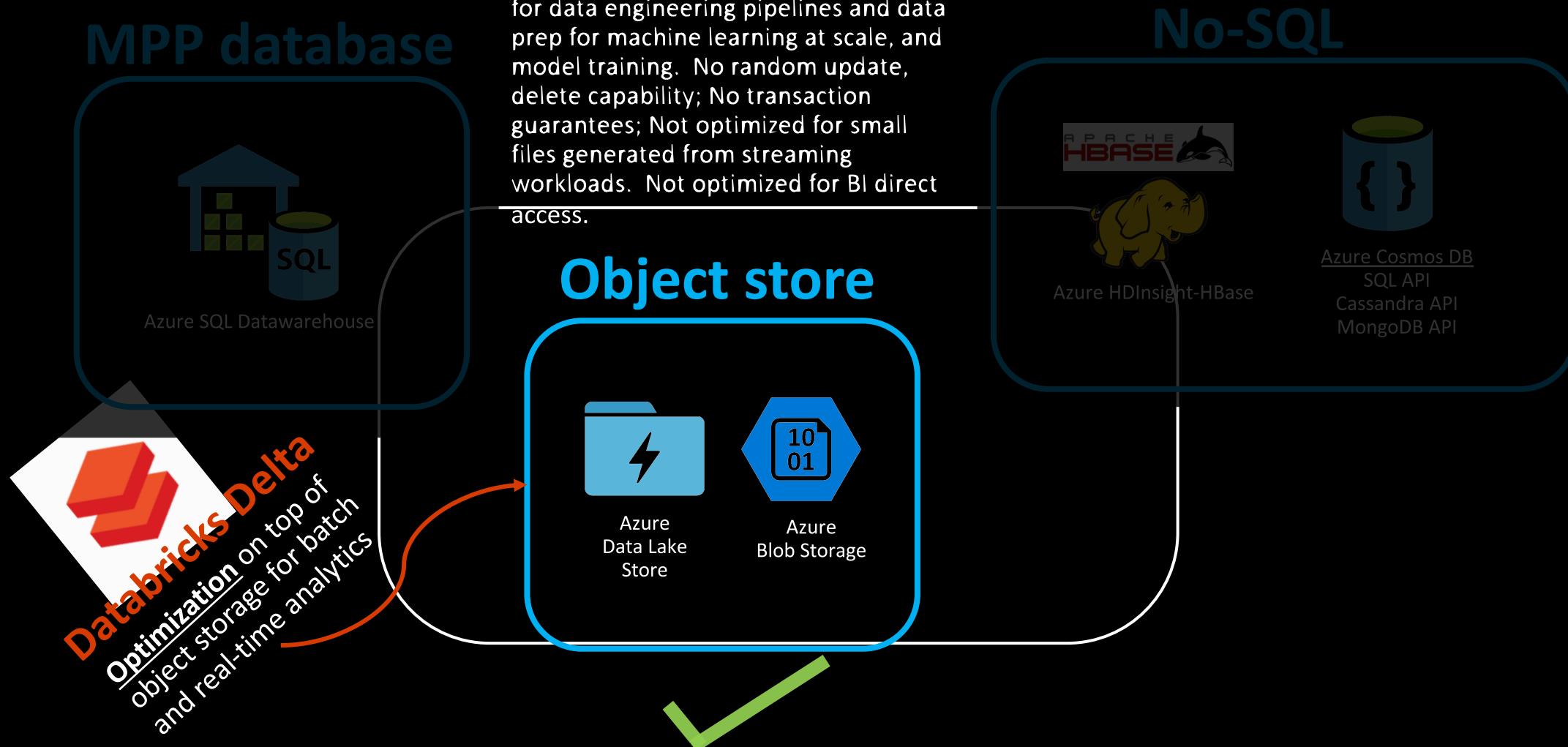


Azure Cosmos DB  
SQL API  
Cassandra API  
MongoDB API



# Analytics store – structured, unstructured

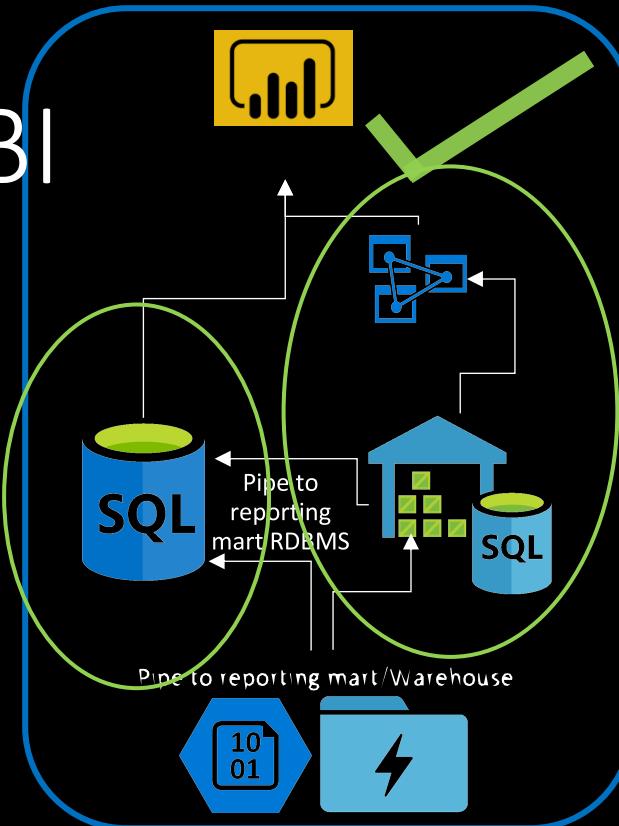
## Fit for purpose



# Reporting & BI

Fit for purpose

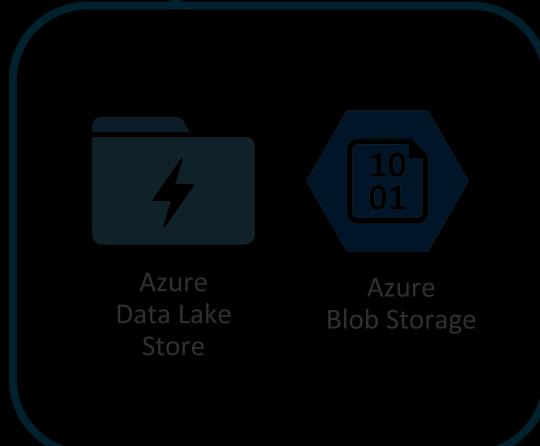
## MPP database



## No-SQL



## Object store



# Real-time analytics – persistence and serve layer

Fit for purpose

## MPP database



Azure SQL Datawarehouse

Not as optimized for either real time performant ingest or for serving, but supports sinking from stream processing pipelines in an append mode with some nuances.

## Object store

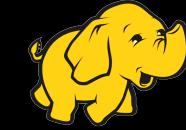


Azure  
Data Lake  
Store



Azure  
Blob Storage

## No-SQL



Azure HDInsight-HBase



Azure Cosmos DB  
SQL API  
Cassandra API  
MongoDB API



# THE MODERN DATA ESTATE



LOB



CRM



Graph



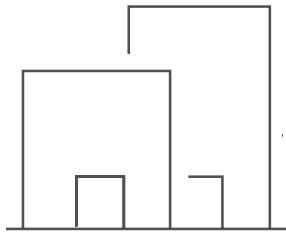
Image



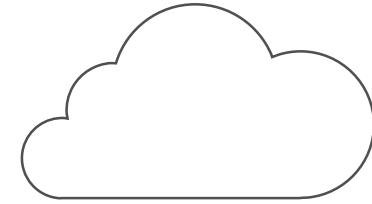
Social



IoT



Operational databases  
Data warehouses  
Data Lakes



Operational databases  
Data warehouses  
Data Lakes

Hybrid

Reason over any data, anywhere

Flexibility of choice

Security and performance

# THE MICROSOFT OFFERING



LOB



CRM



Graph



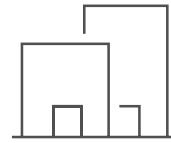
Image



Social



IoT



SQL Server



← Hybrid →

Azure Data Services

Easiest lift and shift  
with no code changes

**Industry leader 2 years in a row** Operational databases

Operational databases **70% faster than Aurora**

**#1 TPC-H performance** Data warehouses

Data warehouses **2x global reach than Redshift**

**T-SQL query over any data** Data lakes

Data lakes **No Limits Analytics with 99.9% SLA**

AI built-in | Most secure | Lowest TCO

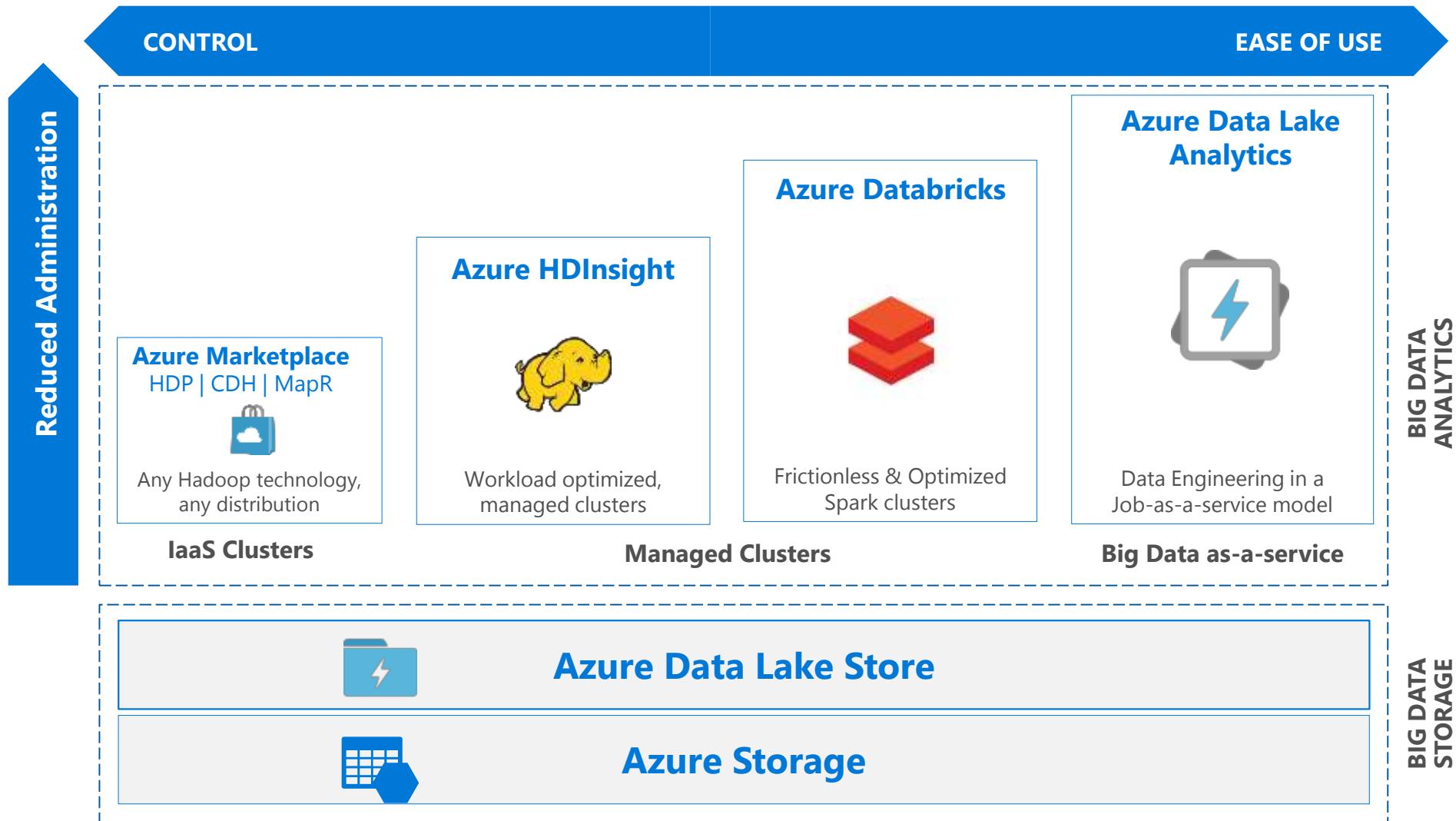
Reason over any data, anywhere

Flexibility of choice

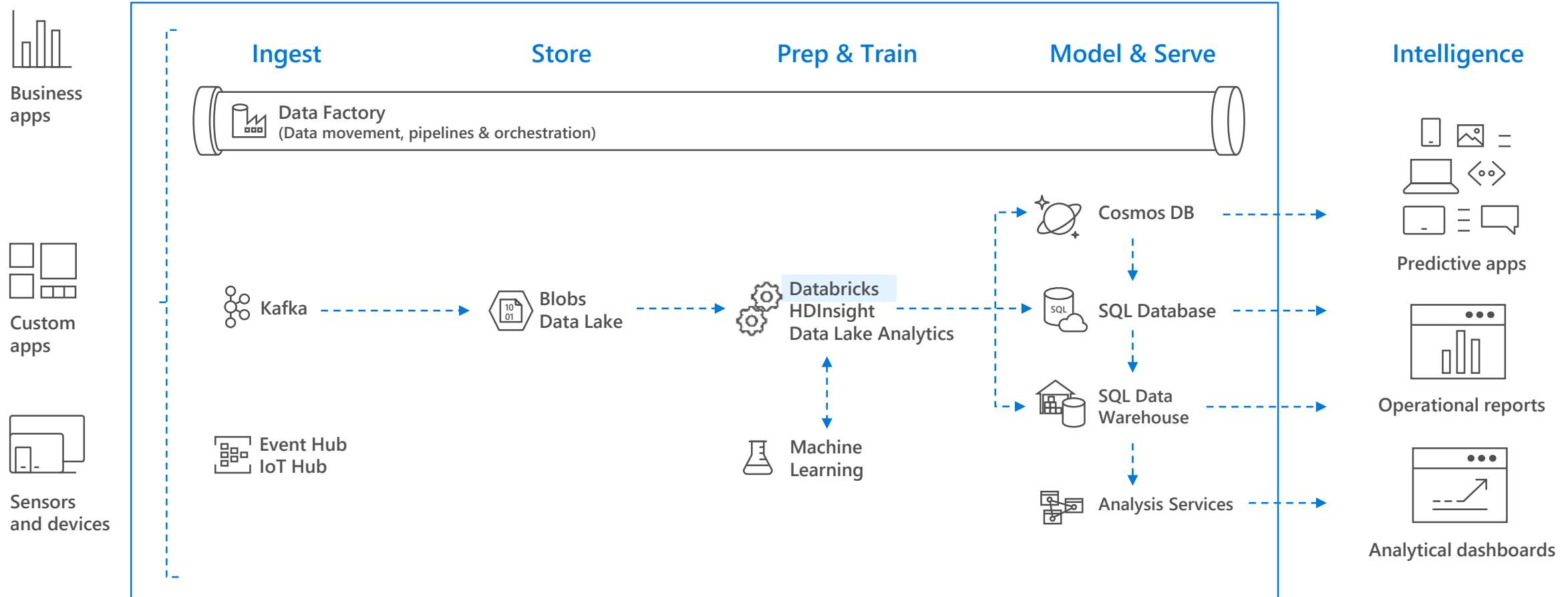
Security and performance

# Big Data & Advanced Analytics in Azure

# KNOWING THE VARIOUS BIG DATA SOLUTIONS



# BIG DATA & ADVANCED ANALYTICS AT A GLANCE



Azure Databricks  
Powered by Apache Spark

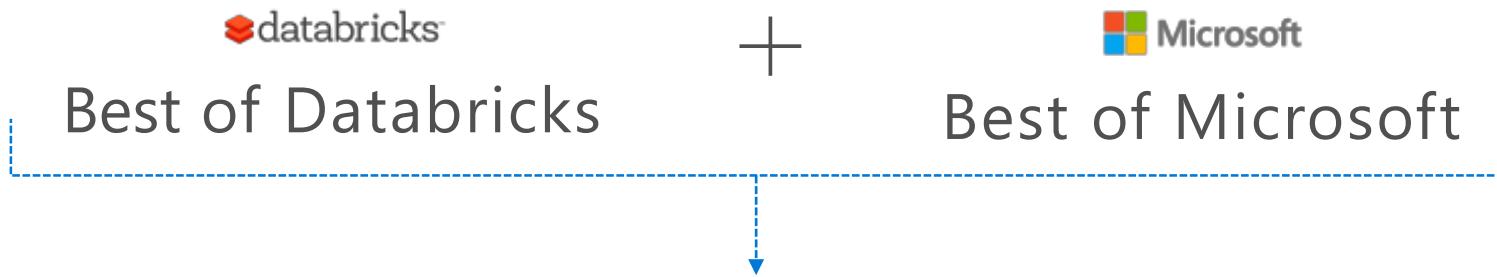


# Why Spark?

- Open-source data processing engine built around **speed, ease of use, and sophisticated analytics**
- In memory engine that is up to **100 times faster than Hadoop**
- **Largest open-source data project** with 1000+ contributors
- **Highly extensible** with support for Scala, Java and Python alongside Spark SQL, GraphX, Streaming and Machine Learning Library (MLlib)

# What is Azure Databricks?

A fast, easy and collaborative Apache® Spark™ based analytics platform optimized for Azure



 Designed in collaboration with the founders of Apache Spark



One-click set up; streamlined workflows



Interactive workspace that enables collaboration between data scientists, data engineers, and business analysts.



Native integration with Azure services (Power BI, SQL DW, Cosmos DB, Blob Storage)



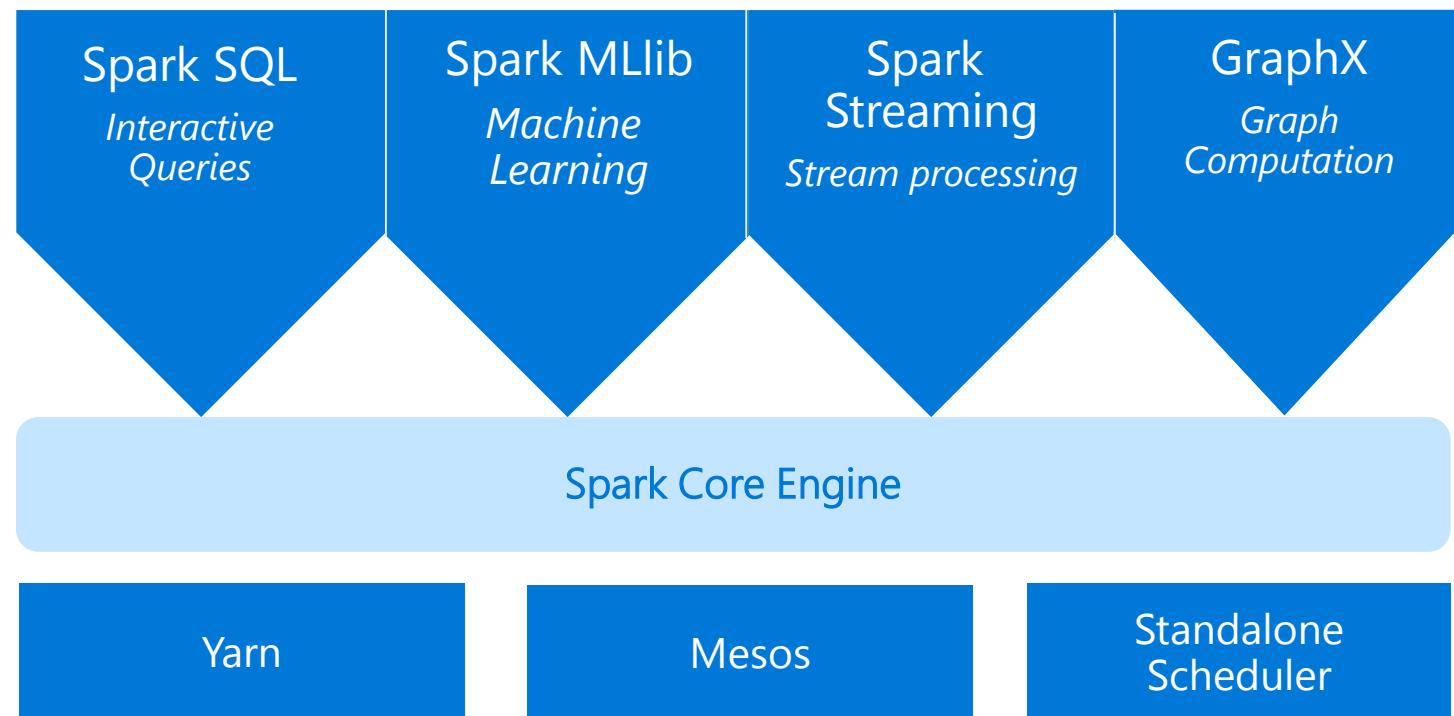
Enterprise-grade Azure security (Active Directory integration, compliance, enterprise-grade SLAs)

# A P A C H E S P A R K

An unified, open source, parallel, data processing framework for Big Data Analytics

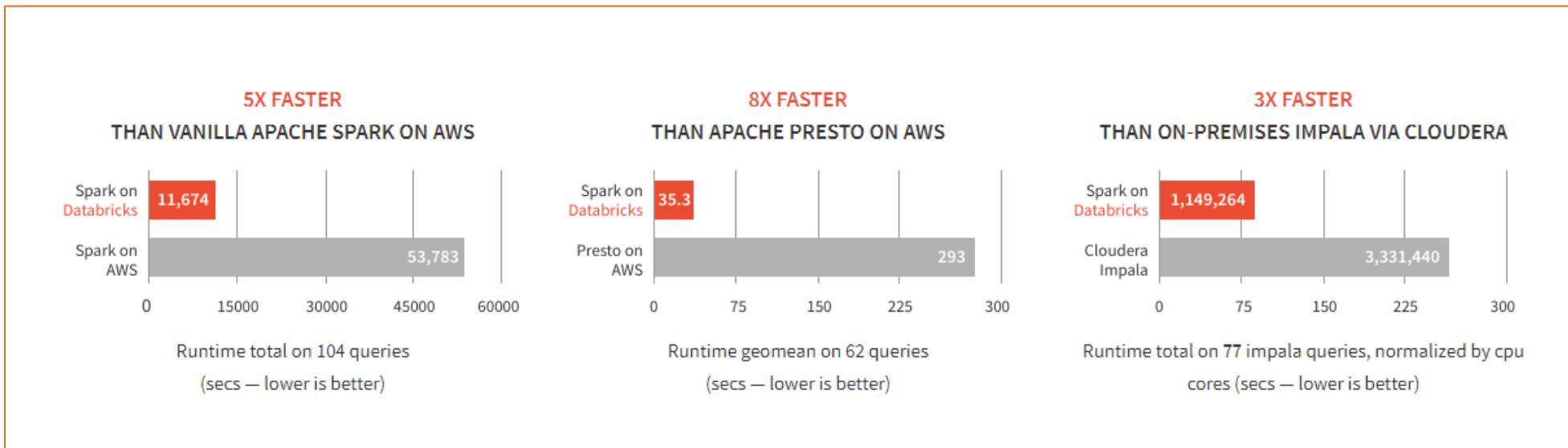
Spark Unifies:

- Batch Processing
- Interactive SQL
- Real-time processing
- Machine Learning
- Deep Learning
- Graph Processing



# DATA BRICKS SPARK IS FAST

Benchmarks have shown Databricks to often have better performance than alternatives



**SOURCE:** [Benchmarking Big Data SQL Platforms in the Cloud](#)

# Differentiated experience on Azure

## ENHANCE PRODUCTIVITY

**Get started quickly** by launching your new Spark environment with one click.

**Share your insights in powerful ways** through rich integration with Power BI.

**Improve collaboration** amongst your analytics team through a unified workspace.

**Innovate faster** with native integration with rest of Azure platform

## BUILD ON THE MOST COMPLIANT CLOUD

**Simplify security and identity control** with built-in integration with Active Directory.

**Regulate access** with fine-grained user permissions to Azure Databricks' notebooks, clusters, jobs and data.

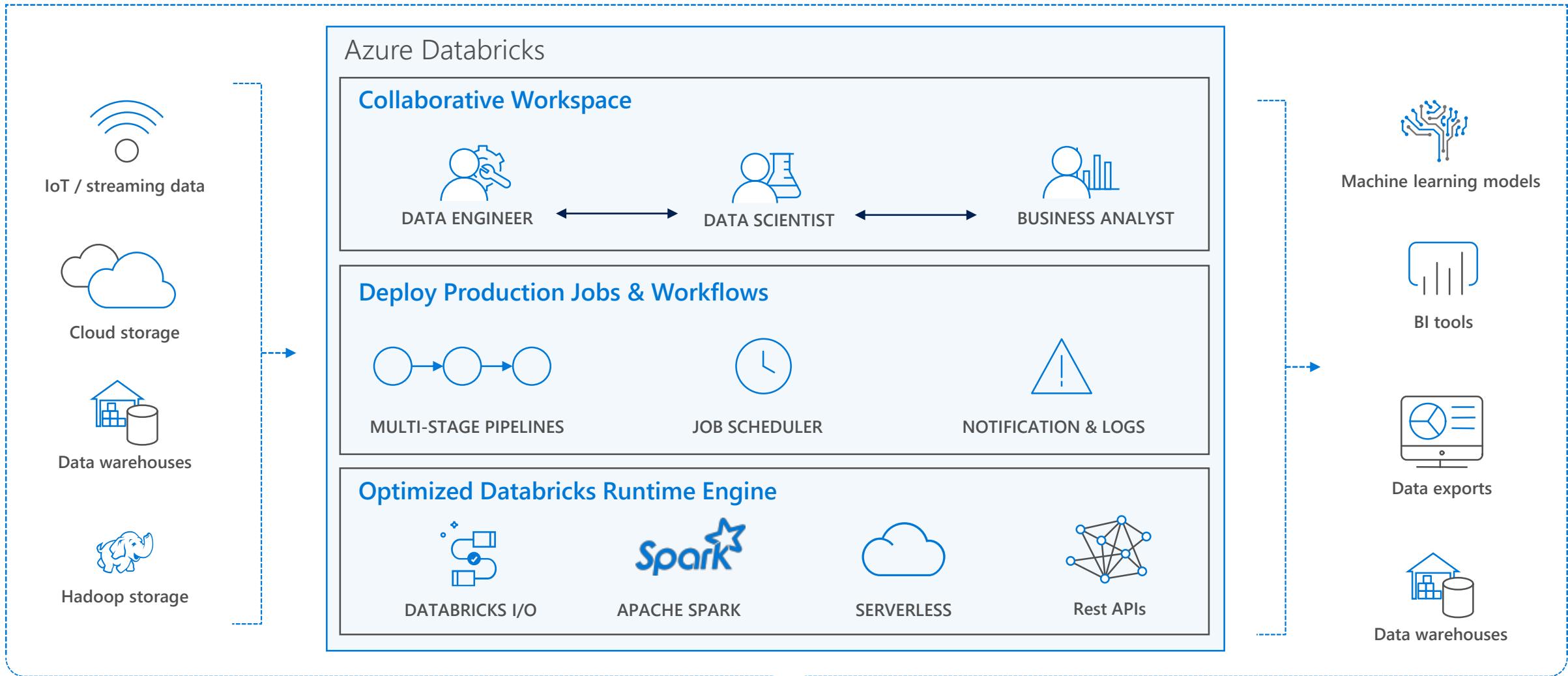
**Build with confidence on the trusted cloud** backed by unmatched support, compliance and SLAs.

## SCALE WITHOUT LIMITS

**Operate at massive scale** without limits globally.

**Accelerate data processing** with the fastest Spark engine.

# Azure Databricks



Enhance Productivity

Build on secure & trusted cloud

Scale without limits

# Collaborative Workspace

## GET STARTED IN SECONDS

Single click to launch your new Spark environment

## INTERACTIVE EXPLORATION

Explore data using interactive notebooks with support for multiple programming languages including R, Python, Scala, and SQL

## COLLABORATION

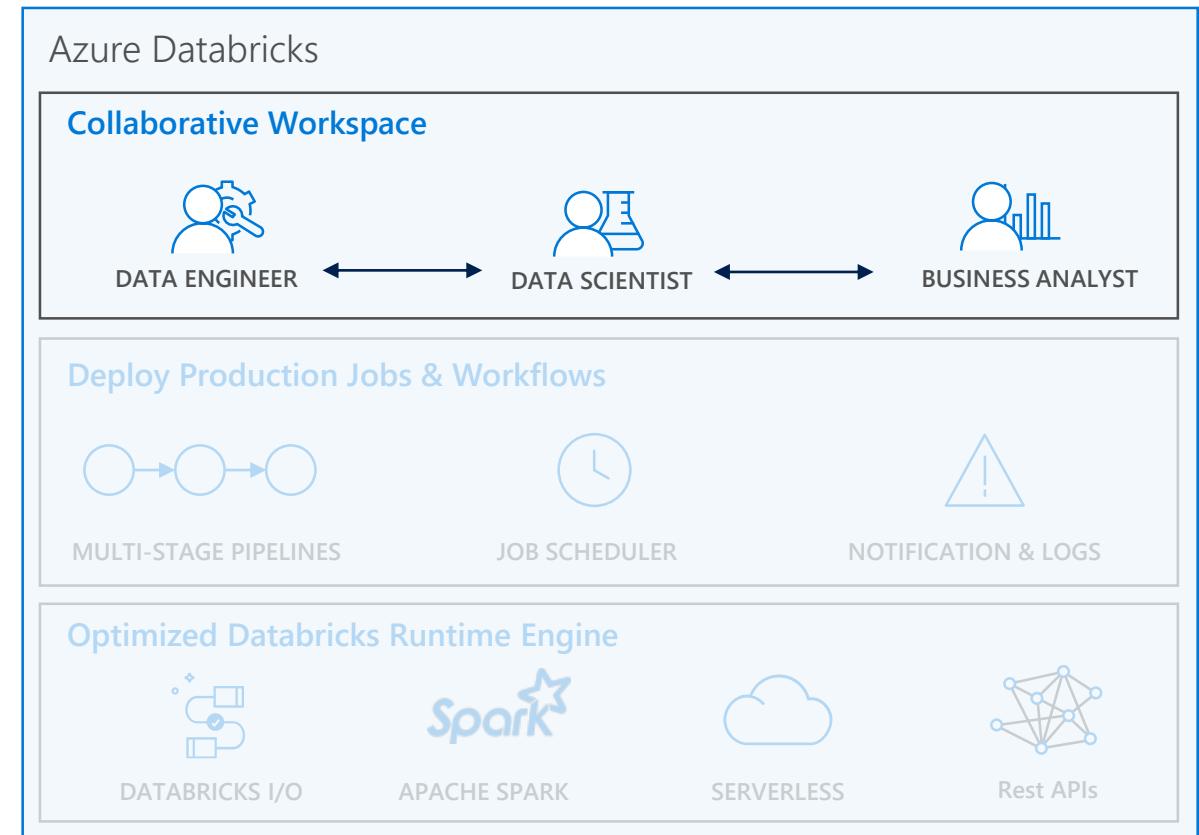
Work on the same notebook in real-time while tracking changes with detailed revision history, GitHub, or Bitbucket

## VISUALIZATIONS

Visualize insights through a wide assortment of point-and-click visualizations. Or use powerful scriptable options like matplotlib, ggplot, and D3

## DASHBOARDS

Rich integration with PowerBI to discover and share your insights in powerful new ways



# Deploy Production Jobs & Workflows

## JOB SCHEDULER

Execute jobs for production pipelines on a specific schedule

## NOTEBOOK WORKFLOWS

Create multi-stage pipelines with the control structures of the source programming language

## RUN NOTEBOOKS AS JOBS

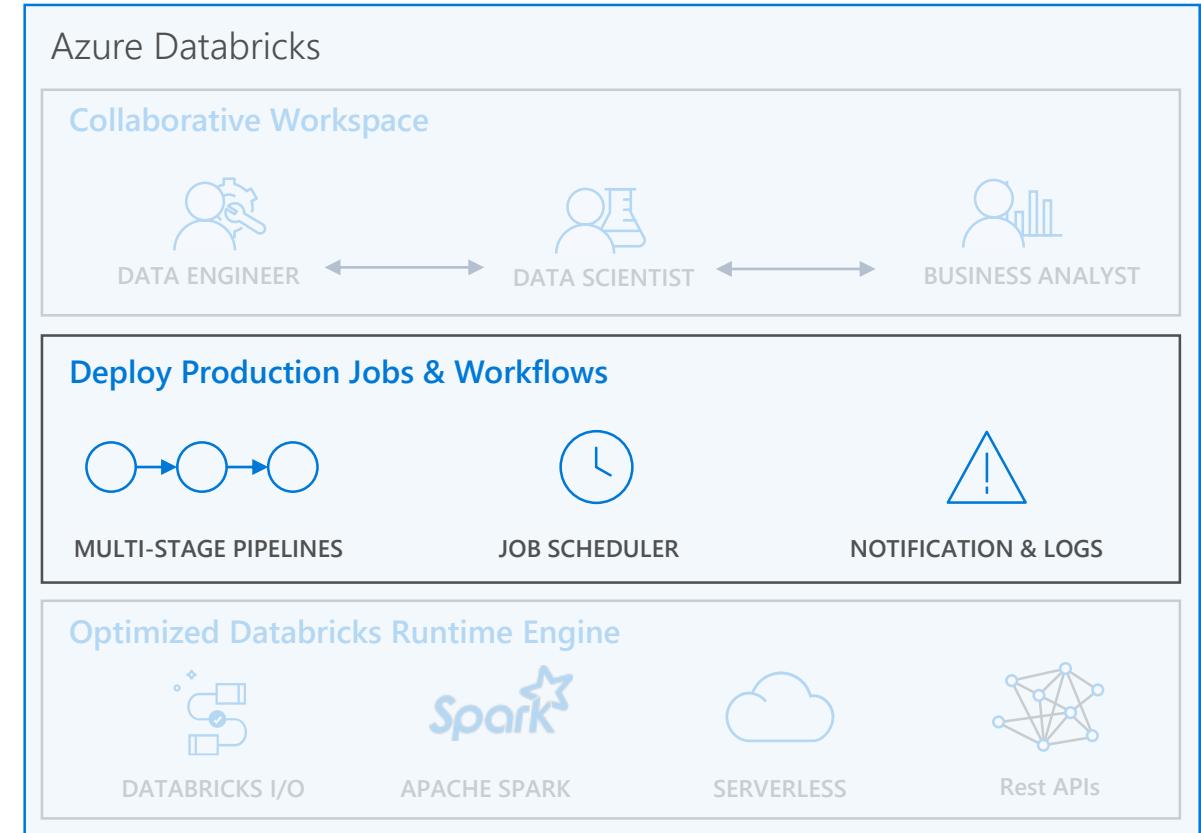
Turn notebooks or JARs into resilient Spark jobs with a click or an API call

## NOTIFICATIONS AND LOGS

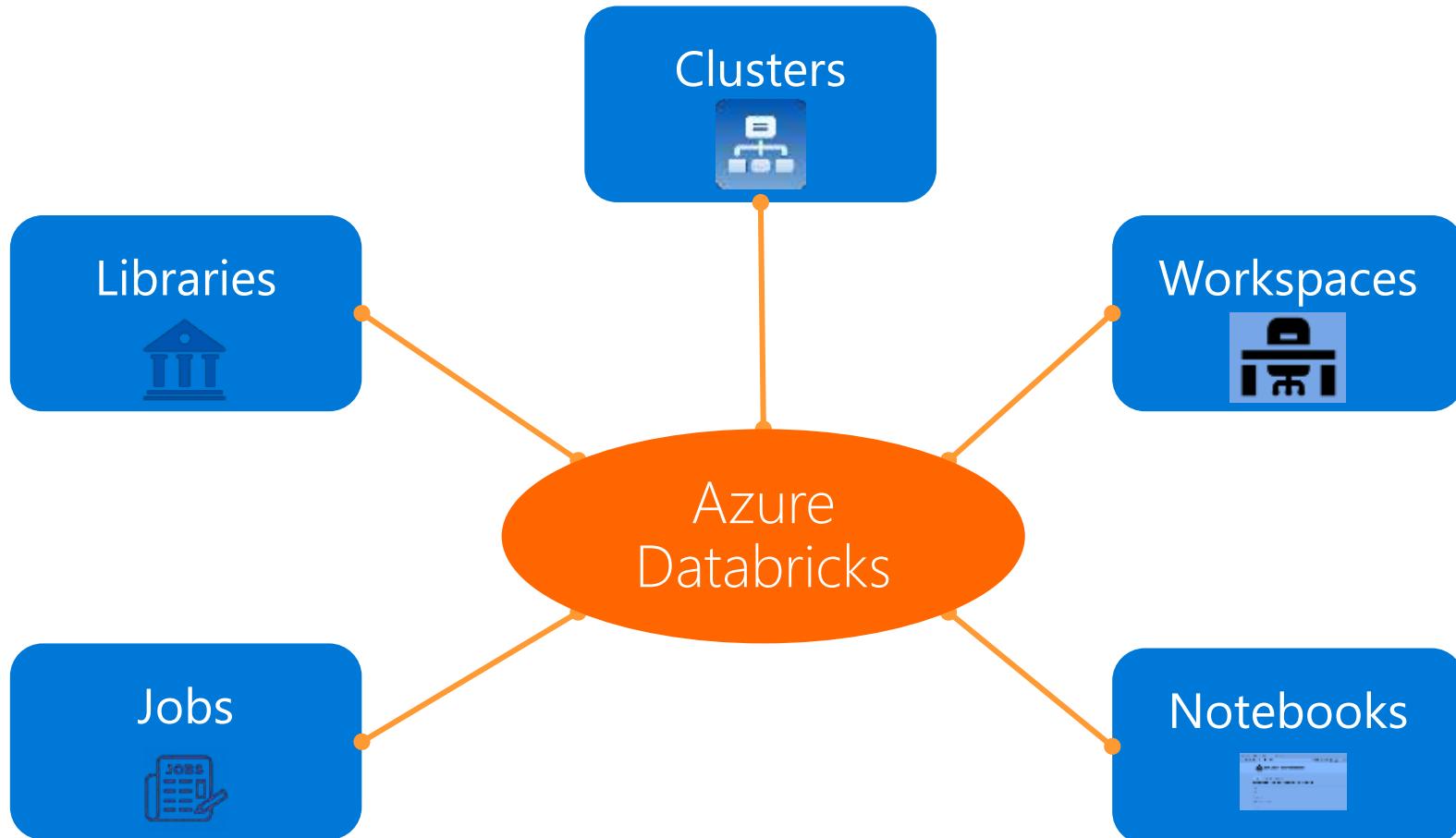
Set up alerts and quickly access audit logs for easy monitoring and troubleshooting

## INTEGRATE NATIVELY WITH AZURE SERVICES

Deep integration with Azure SQL Data Warehouse, Cosmos DB, Azure Data Lake Store, Azure Blob Storage, and Azure Event Hub



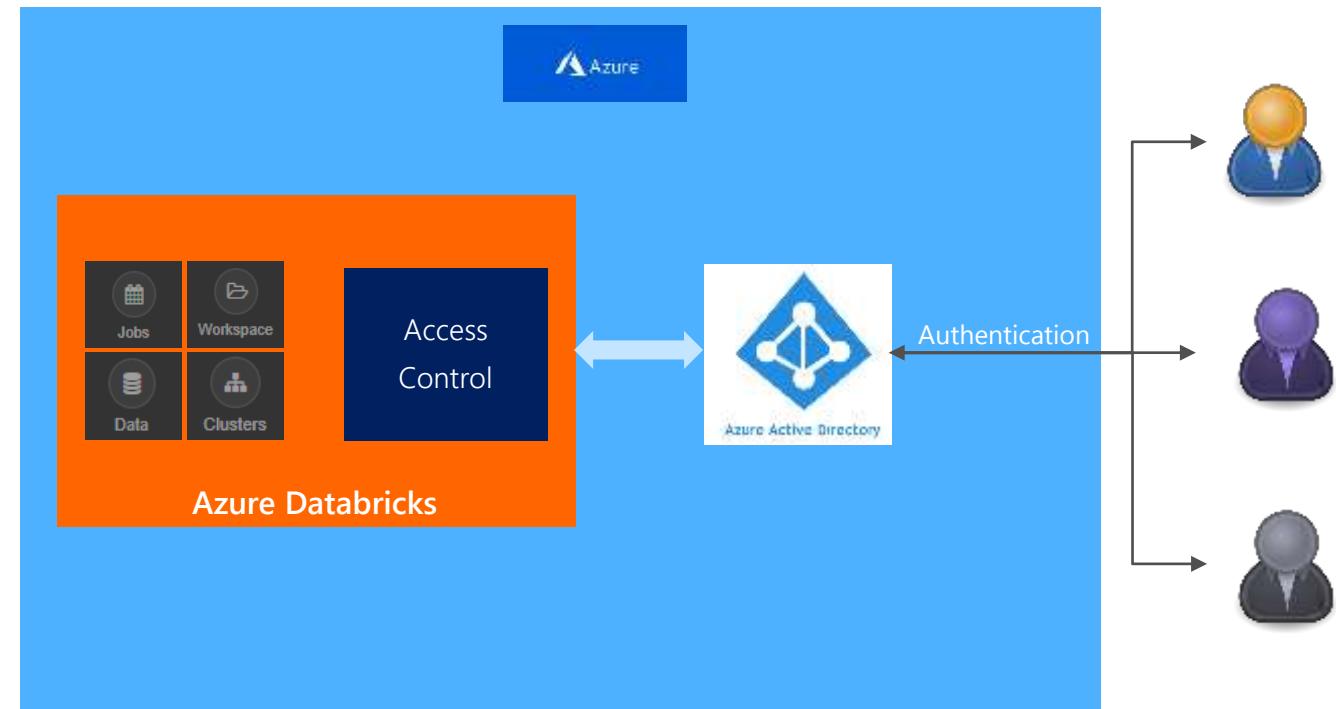
# AZURE DATA BRICKS CORE ARTIFACTS



# AZURE DATABRICKS INTEGRATION WITH AAD

Azure Databricks is integrated with AAD—so Azure Databricks users are just regular AAD users

- There is no need to define users—and their access control—separately in Databricks.
- AAD users can be used directly in Azure Databricks for all user-based access control (Clusters, Jobs, Notebooks etc.).
- Databricks has delegated user authentication to AAD enabling single-sign on (SSO) and unified authentication.
- *Notebooks, and their outputs, are stored in the Databricks account. However, AAD-based access-control ensures that only authorized users can access them.*



# CLUSTERS: AUTO SCALING AND AUTO TERMINATION

Simplifies cluster management and reduces costs by eliminating wastage

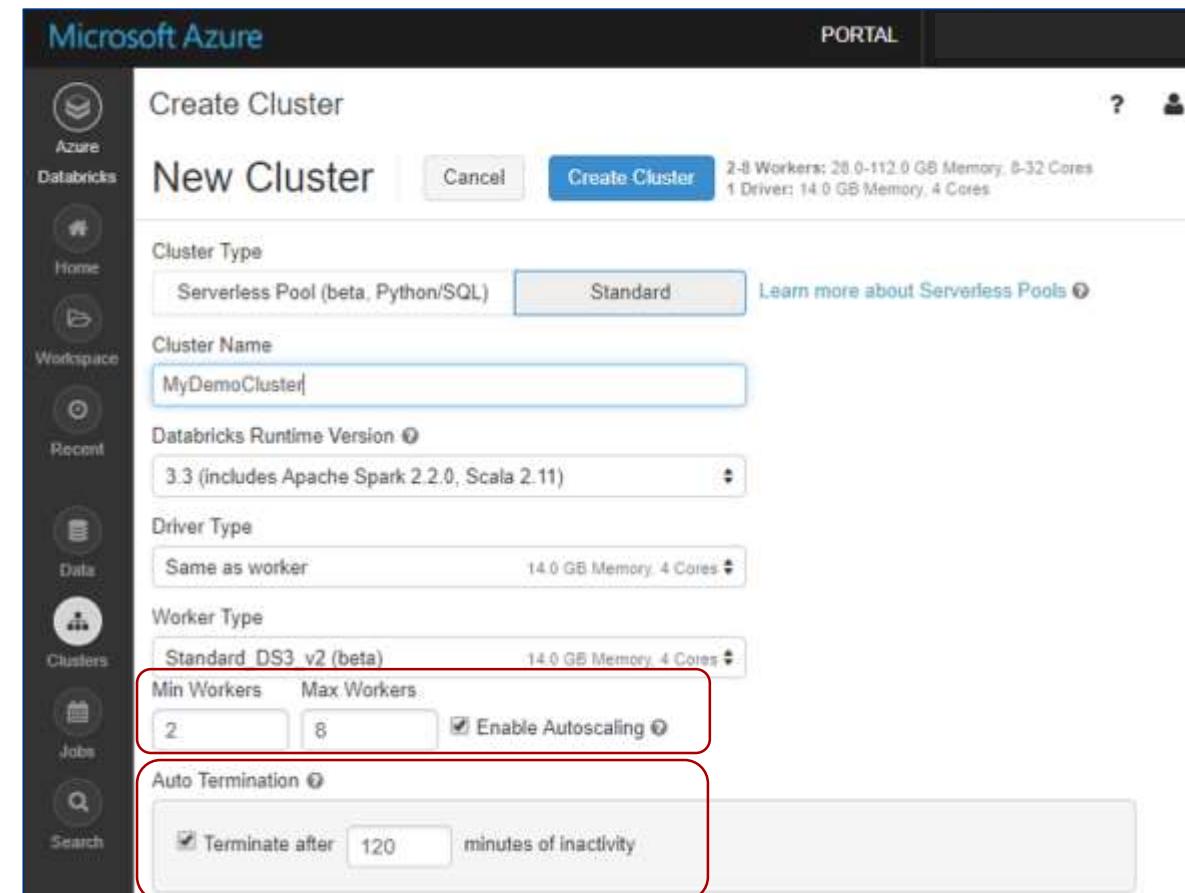
When creating Azure Databricks clusters you can choose Autoscaling and Auto Termination options.

Autoscaling: Just specify the min and max number of clusters. Azure Databricks automatically scales up or down based on load.

Auto Termination: After the specified minutes of inactivity the cluster is automatically terminated.

Benefits:

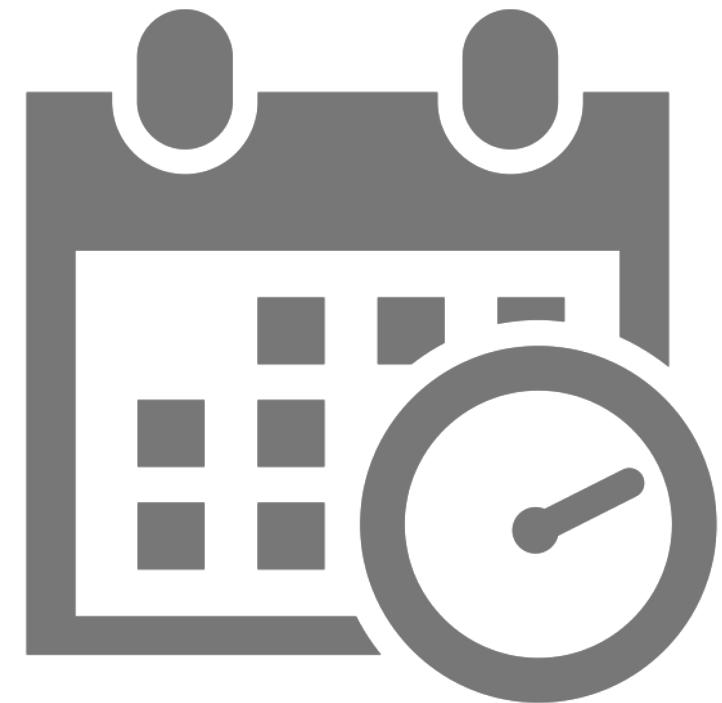
- You do not have to guess, or determine by trial and error, the correct number of nodes for the cluster
- As the workload changes you do not have to manually tweak the number of nodes
- You do not have to worry about wasting resources when the cluster is idle. You only pay for resource when they are actually being used
- You do not have to wait and watch for jobs to complete just so you can shutdown the clusters



## J O B S

Jobs are the mechanism to submit Spark application code for execution on the Databricks clusters

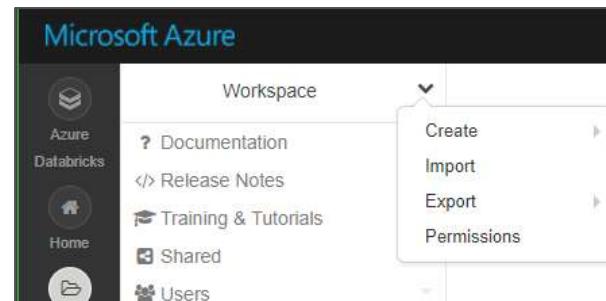
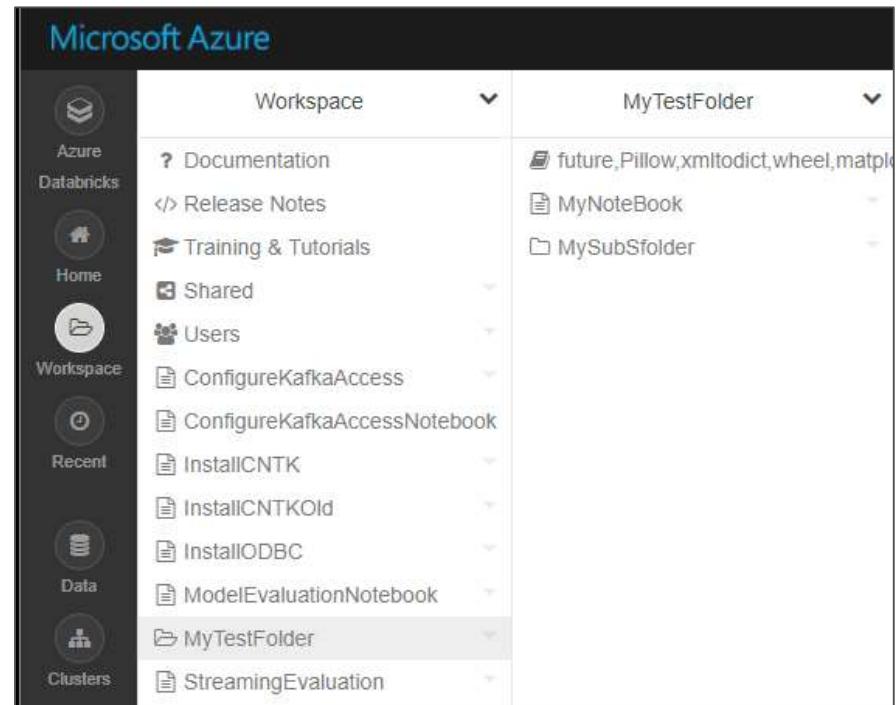
- Spark application code is submitted as a 'Job' for execution on Azure Databricks clusters
- Jobs execute either 'Notebooks' or 'Jars'
- Azure Databricks provide a comprehensive set of graphical tools to create, manage and monitor Jobs.



# WORKSPACES

Workspaces enables users to organize—and share—their Notebooks, Libraries and Dashboards

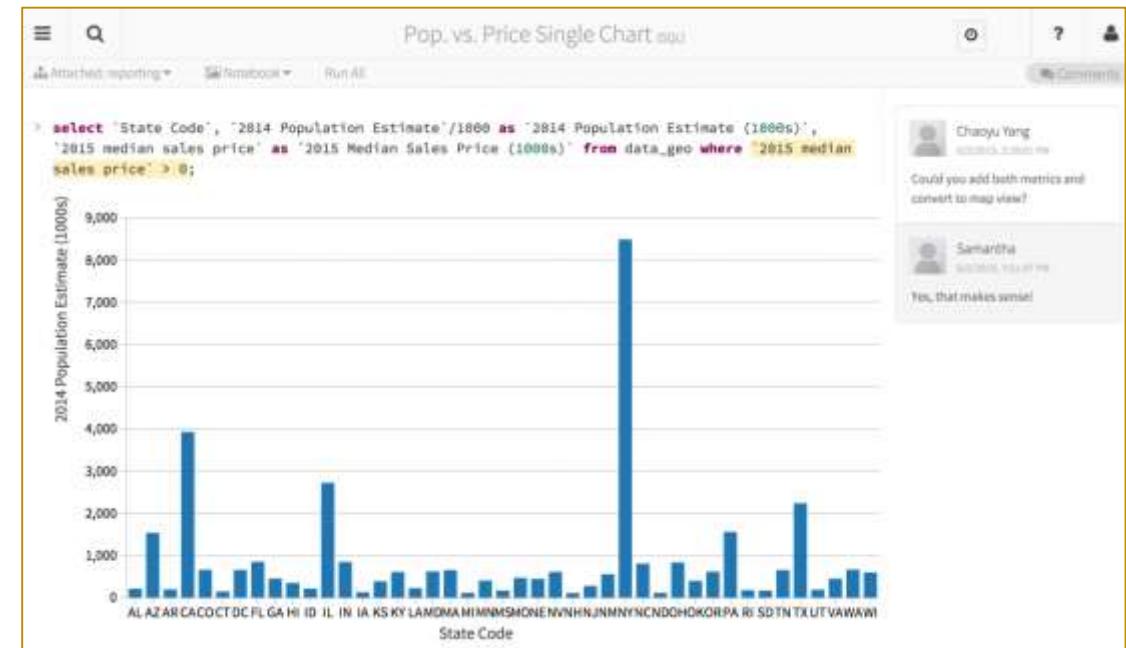
- Workspaces—sort of like Directories—are a convenient way to organize an user's Notebook, Libraries and Dashboards.
- Everything in a workspace is organized into hierarchical folders. Folders can hold Libraries, Notebooks, Dashboard or more (sub) folders.
  - Icons indicate the type of the object contained in a folder
- Every user has one directory that is private and unshared.
  - By default, the workspace and all its contents are available to users.
- Fine grained access control can be defined on workspaces (next slide) to enable *secure collaboration with colleagues*.



# AZURE DATA BRICKS NOTEBOOKS OVERVIEW

Notebooks are a popular way to develop, and run, Spark Applications

- Notebooks are not only for authoring Spark applications but can be *run/executed directly* on clusters
  - Shift+Enter
  - click the ▶ at the top right of the cell in a notebook
  - Submit via Job
- Notebooks support fine grained permissions—so they can be *securely shared* with colleagues for collaboration (see following slide for details on permissions and abilities)
- Notebooks are well-suited for prototyping, rapid development, exploration, discovery and iterative development



Notebooks typically consist of code, data, visualization, comments and notes

# LIBRARIES OVERVIEW

Enables external code to be imported and stored into a Workspace

- Libraries are containers to hold all your *Python, R, Java/Scala* libraries.
- Libraries resides within workspaces or folders.
- Libraries are created by importing the source code
- After importing libraries are immutable—can be deleted or overwritten only.
- You can customize installation of libraries via [Init Scripts](#) by writing custom UNIX scripts
- Libraries can also be managed via the [Library API](#)

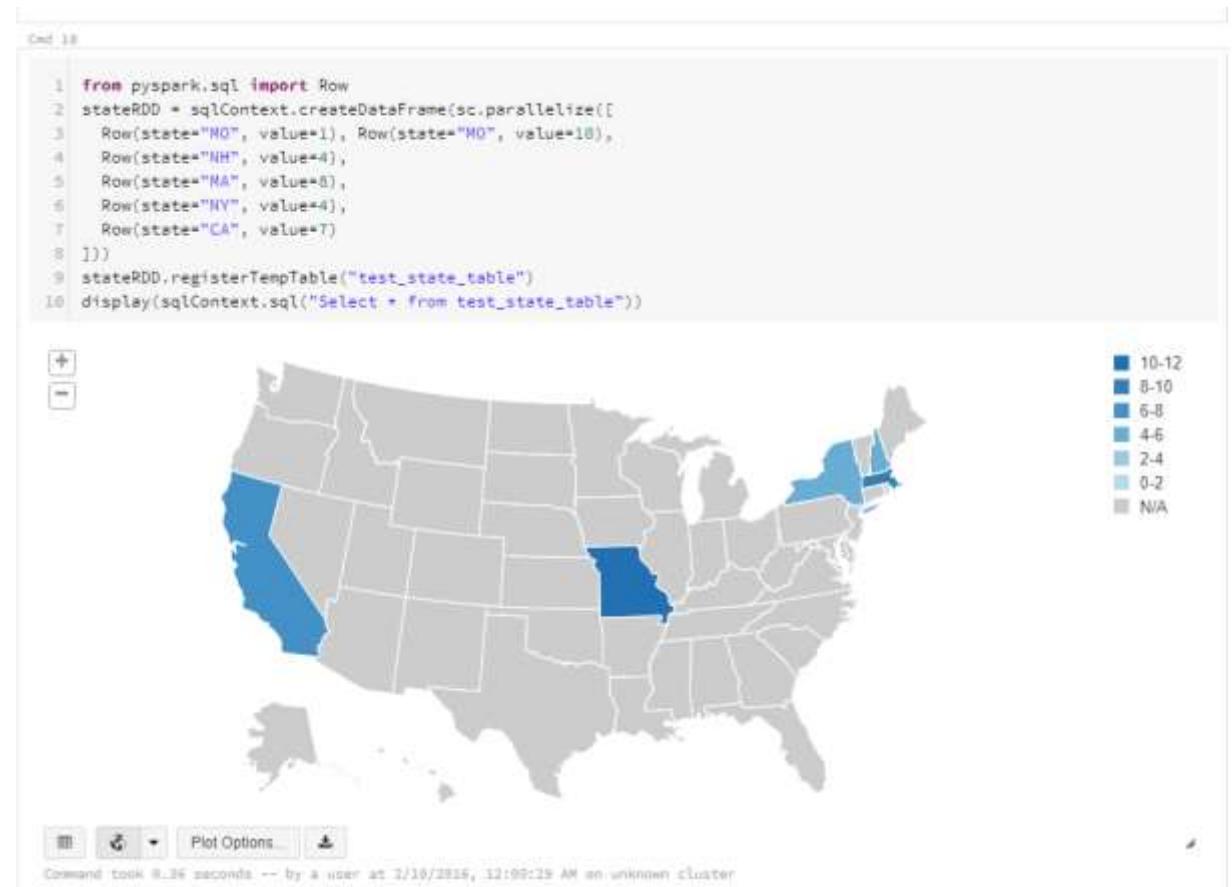
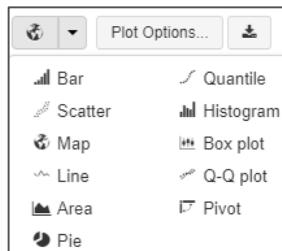
The image displays three separate screenshots of the Microsoft Azure portal's 'Create Library' interface, each showing a different way to import external code:

- Top Left (Python):** Shows the 'New Library' screen for Python. It includes fields for 'Language' (set to 'Upload Python Egg or PyPI'), 'PyPI Name' (containing 'PyPI Package (e.g. tensorflow or tensorflow<1.8.0)'), and an 'Install Library' button. Below this is an 'Upload Egg' section with a 'Library Name' field and a 'Drop library egg here to upload...' area.
- Top Right (R):** Shows the 'New Library' screen for R. It includes fields for 'Source' (set to 'R Library'), 'Install from' (set to 'CRAN-like Repository'), 'Repository' (containing 'https://cloud.r-project.org'), and a 'Package' field. A 'Create Library' button is at the bottom.
- Bottom (Java/Scala):** Shows the 'New Library' screen for Java/Scala. It includes fields for 'Source' (set to 'Upload Java/Scala JAR'), 'Library Name' (containing 'My Library'), and a 'JAR File' area with a 'Drop library JAR here to upload...' field. A 'Create Library' button is at the bottom.

# VISUALIZATION

Azure Databricks supports a number of visualization plots out of the box

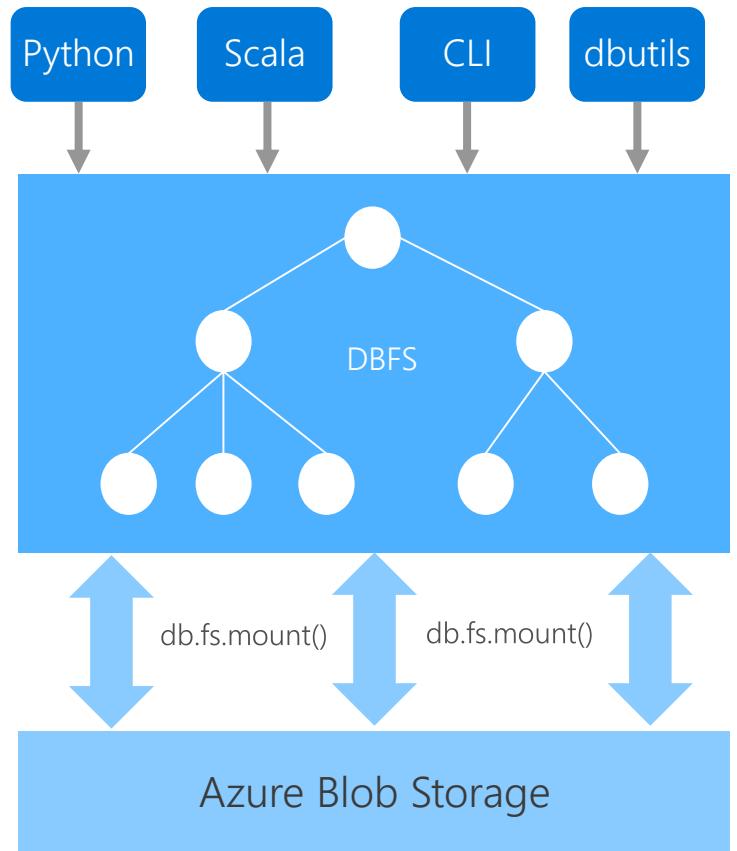
- All notebooks, *regardless of their language*, support Databricks visualizations.
- When you run the notebook the visualizations are rendered inside the notebook in-place
- The visualizations are written in HTML.
  - You can save the HTML of the entire notebook by exporting to HTML.
  - If you use Matplotlib, the plots are rendered as images so you can just right click and download the image
- You can change the plot type just by picking from the selection



# DATA BRICKS FILE SYSTEM (DBFS)

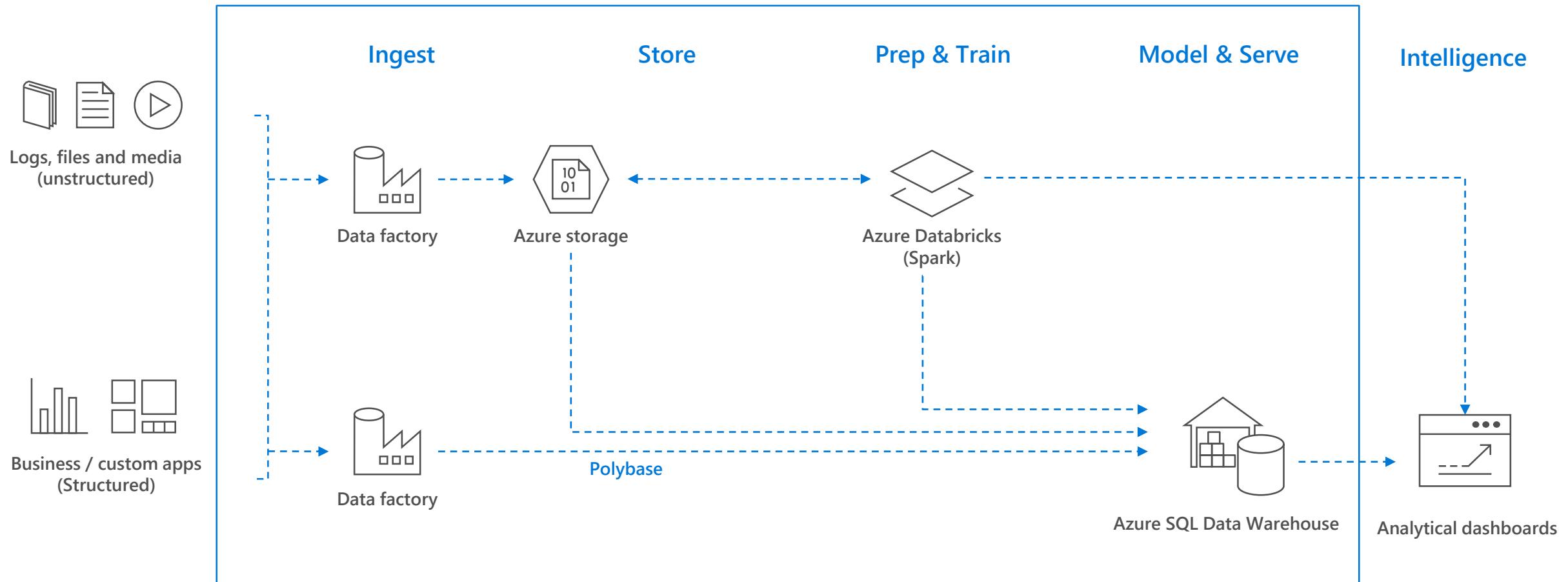
Is a distributed File System (DBFS) that is a layer over Azure Blob Storage

- Azure Storage buckets can be mounted in DBFS so that users can directly access them without specifying the storage keys
- DBFS mounts are created using `dbutils.fs.mount()`
- Azure Storage data can be cached locally on the SSD of the worker nodes
- Available in both Python and Scala and accessible via a DBFS CLI
- Data persist in Azure Blob Storage – is not lost even after cluster termination
- Comes pre-installed on Spark clusters in Databricks

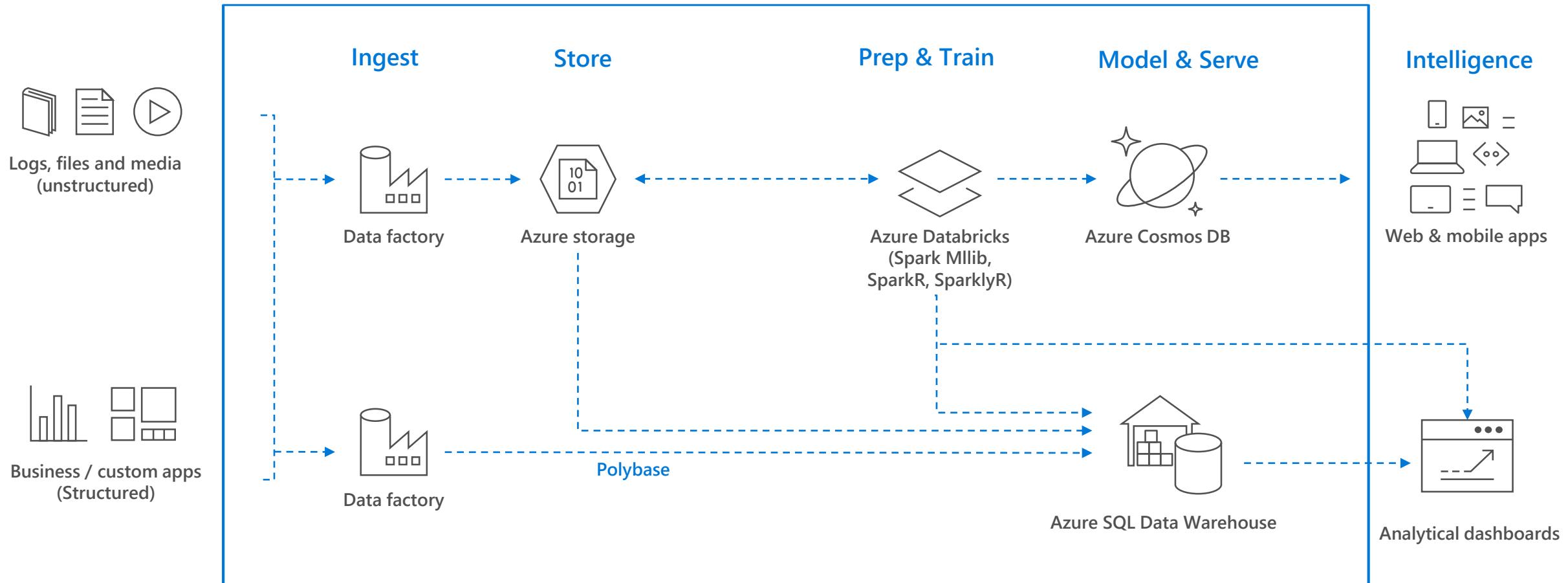


# Use Cases

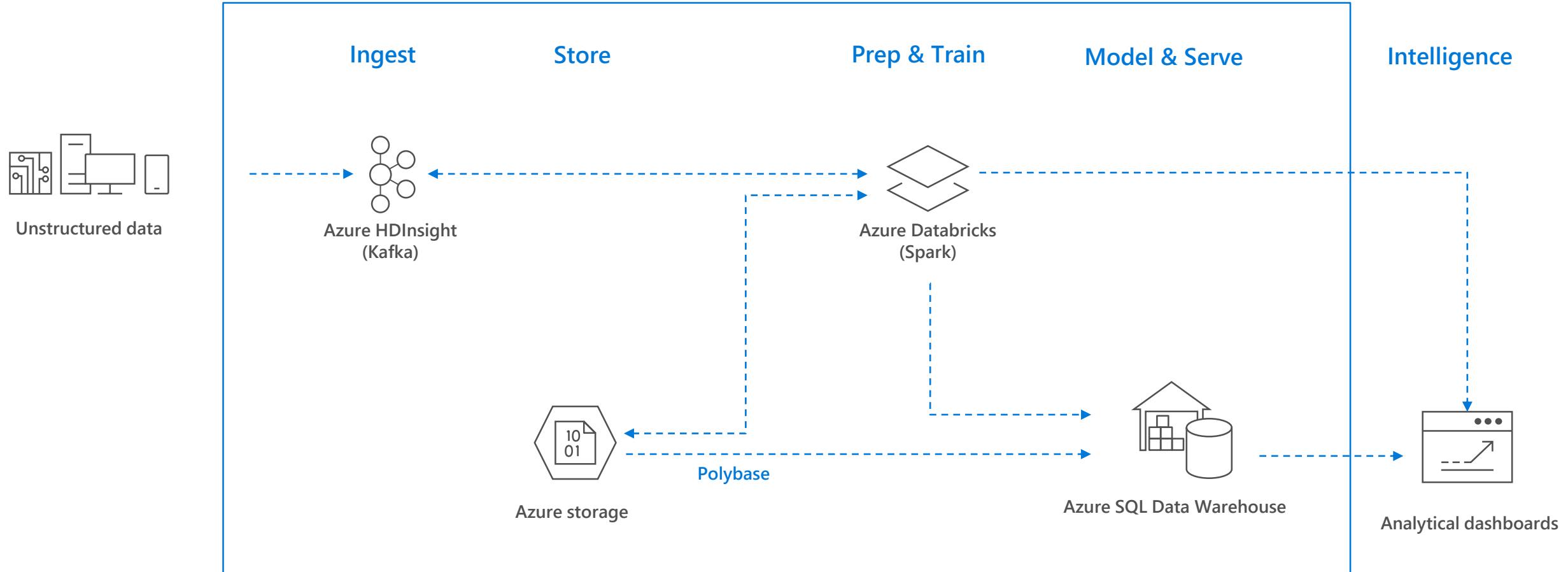
# Modern Big Data Warehouse



# Advanced Analytics on Big Data



# Real-time analytics on Big Data



# Big Data OSS - Comparison

## Azure HDInsight (1<sup>st</sup> party + Support)

### What it is

- **Hadoop** (Hortonworks' Distribution) as a managed service supporting a variety of open-source analytics engines such as Apache Spark, Hive LLAP, Storm, Kafka, HBase.
- Security via Ranger (Kerberos based)

### Pricing

- Priced to compete with AWS EMR. Standard offering.

### Use When

- Customer prefers a **PaaS** like experience to address big data use cases by working with different OSS analytics engines to address big data use cases. Cost sensitive.

## Azure Databricks (1<sup>st</sup> party + Support)

### What it is

- Databricks **Spark**, the most popular open-source analytics engine, as a managed service providing an easy and fast way to unlock big data use cases. Offers best-in-class notebooks experience for productivity and collaboration as well integration with Azure Data Warehouse, Power BI, etc
- Security via native Azure AD integration

### Pricing

- Priced to match Databricks on AWS. Premium offering.

### Use When

- Customer prefers **SaaS** like experience to address big data use cases and values Databricks' ease of use, productivity & collaboration features.

## 3<sup>rd</sup> Party Offerings

### What it is

Hadoop distributions from Cloudera, MapR & Hortonworks available on Azure Marketplace as IaaS VMs.

### Pricing

- N/A. Vendor prices their products.

### Use When

- Customer wants to move their on premises Hadoop distribution to Azure IaaS using their existing licenses.

# Analytics store

## Nuances and data engineering challenges

# Era Of Big Data

How it evolved and what is is for us in current market trends

# Big Data was the Missing Link for AI

## BIG DATA



Customer Data  
Emails/Web pages  
Click Streams  
Sensor data (IoT)  
Video/Speech  
...

## GREAT RESULTS



*Most companies are Struggling with Big Data*

BigData is not only Hadoop

# Hardest part of AI isn't AI

*“Hidden Technical Debt in Machine Learning Systems”, Google NIPS 2015*

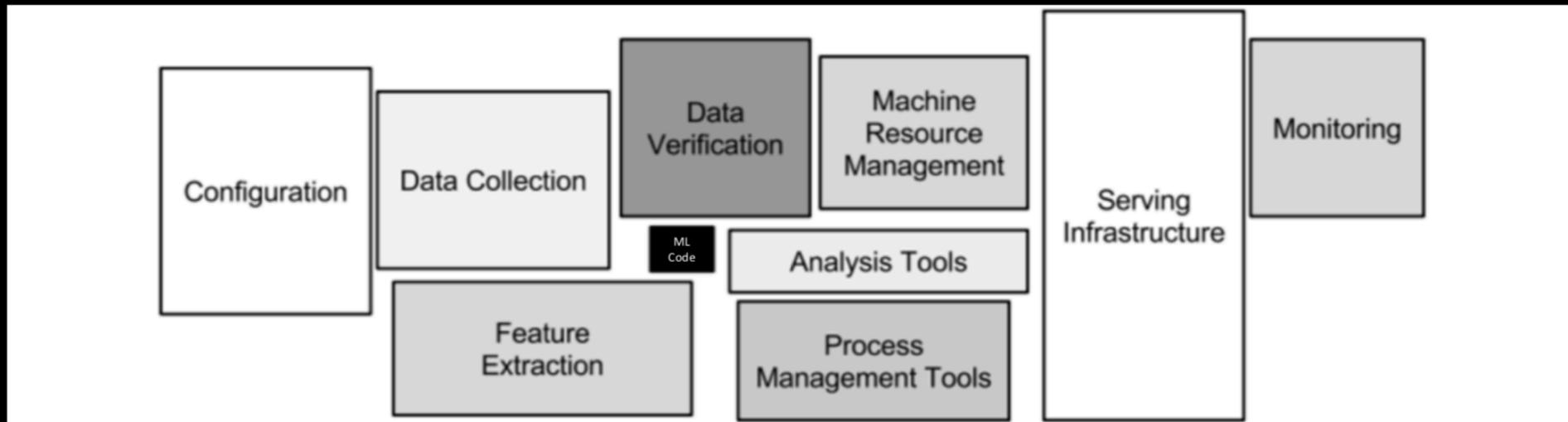


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

*The hardest part of AI is Big Data*

# The Evolution of Big Data

# Data Warehouse (DW)

**ETL important data to central DW and get Business Intelligence (BI)**

## THE GOOD

- Pristine Data
- Fast Queries
- Transactional

## THE BAD

- Expensive to Scale, not Elastic
- Requires ETL, Stale Data, No Real-Time
- No Predictions, No ML
- Closed formats (lock in)

*Not Future Proof – Missing Predictions, Real-time, Scale*

# The Era of the Data Lake

# Hadoop Data Lake

ETL all data to central scalable open lake for all use cases

## THE GOOD

- Massive scale
- Inexpensive Storage
- Open Formats (Parquet, ORC)
- Promise of ML & Real Time Streaming

## THE BAD

- Inconsistent Data
- Unreliable for Analytics
- Lack of Schema
- Poor Performance

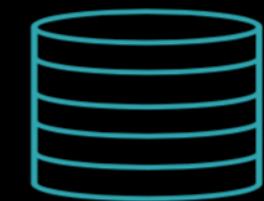
*Become a cheap messy data store with poor performance*

# The Current State of Data Platforms

# Evolution of a Cutting-Edge Data Pipeline



# Evolution of a Cutting-Edge Data Pipeline

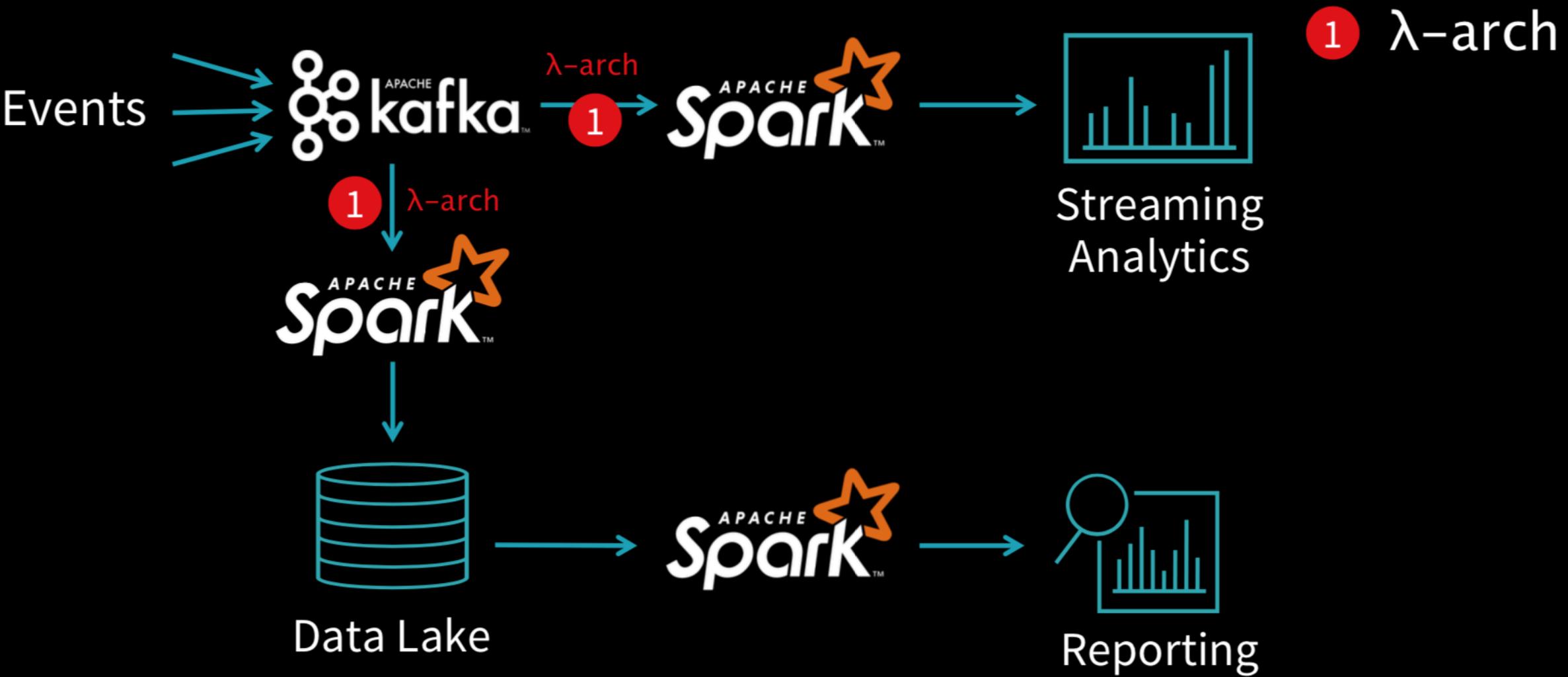


Data Lake

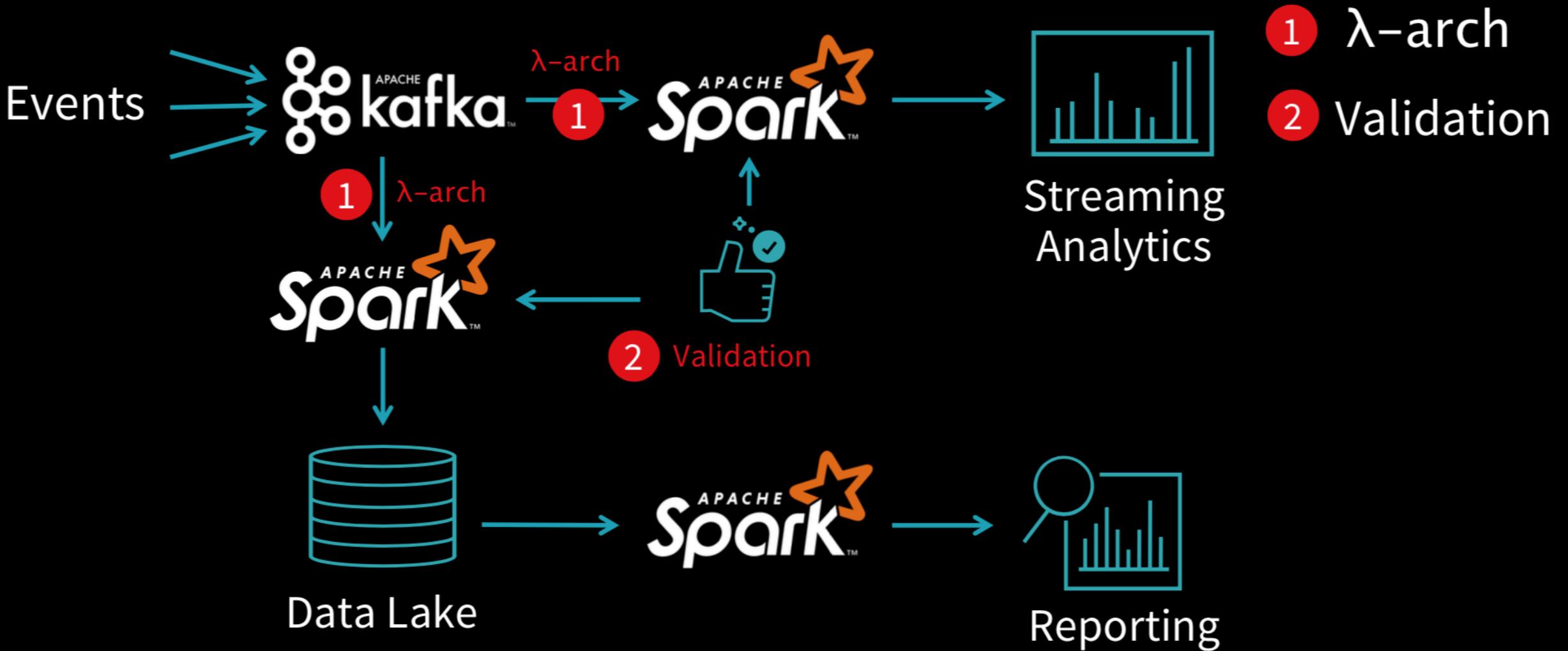


Reporting

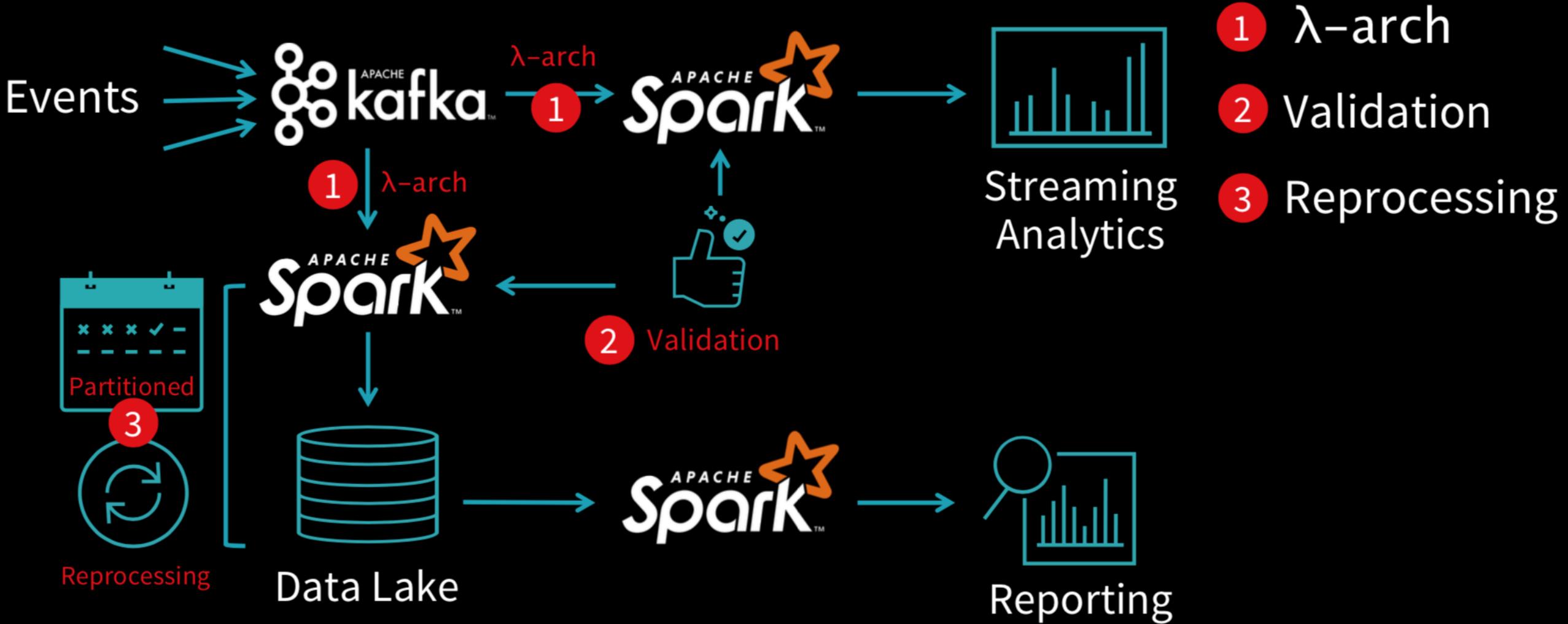
# Challenge #1: Historical Queries?



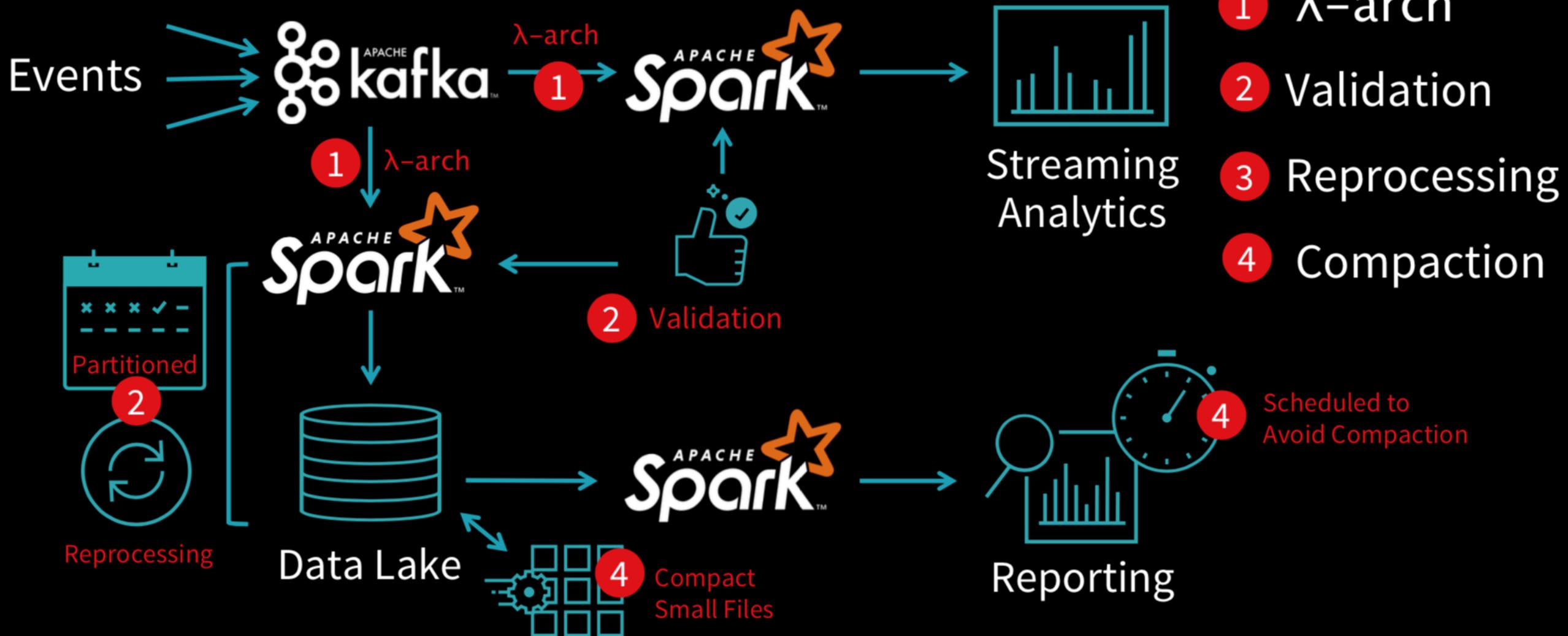
# Challenge #2: Messy Data?



# Challenge #3: Mistakes and Failures?

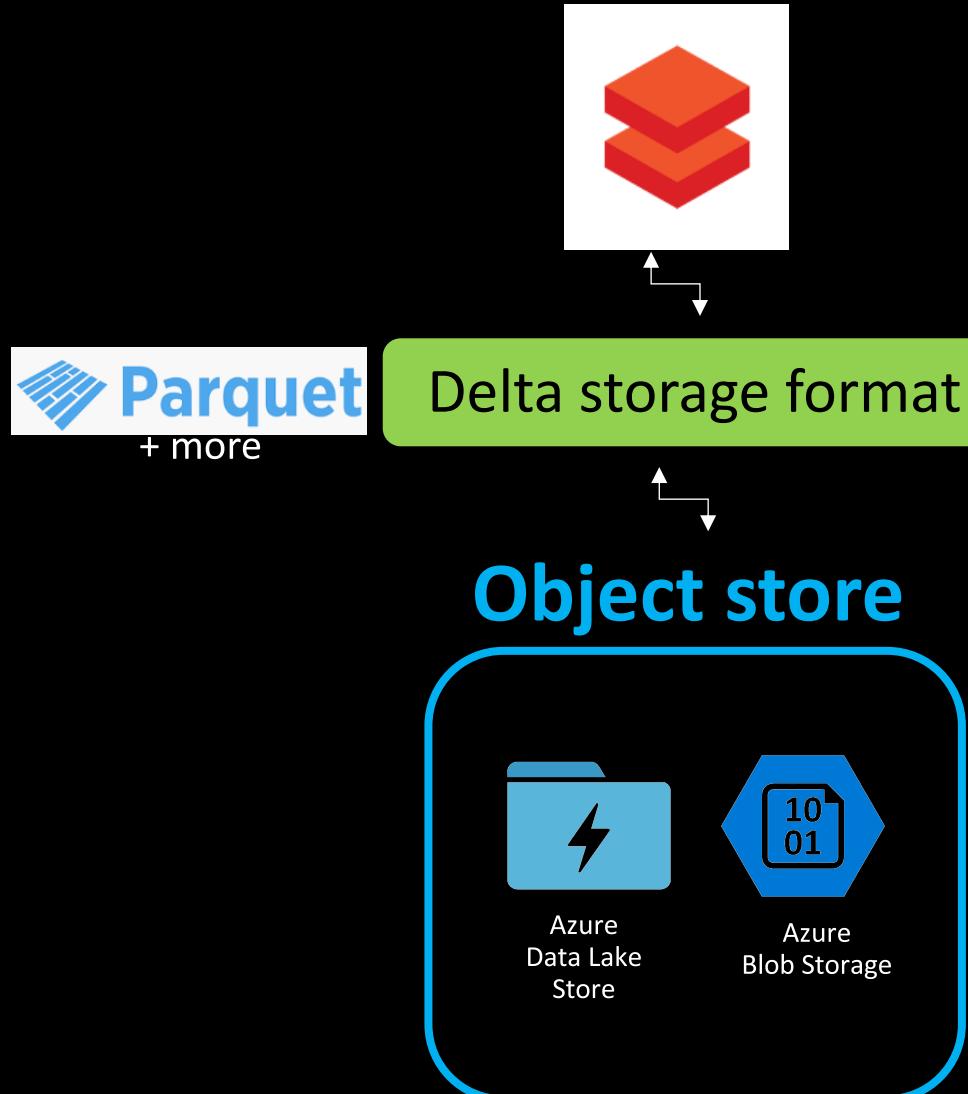


# Challenge #4: Query Performance?

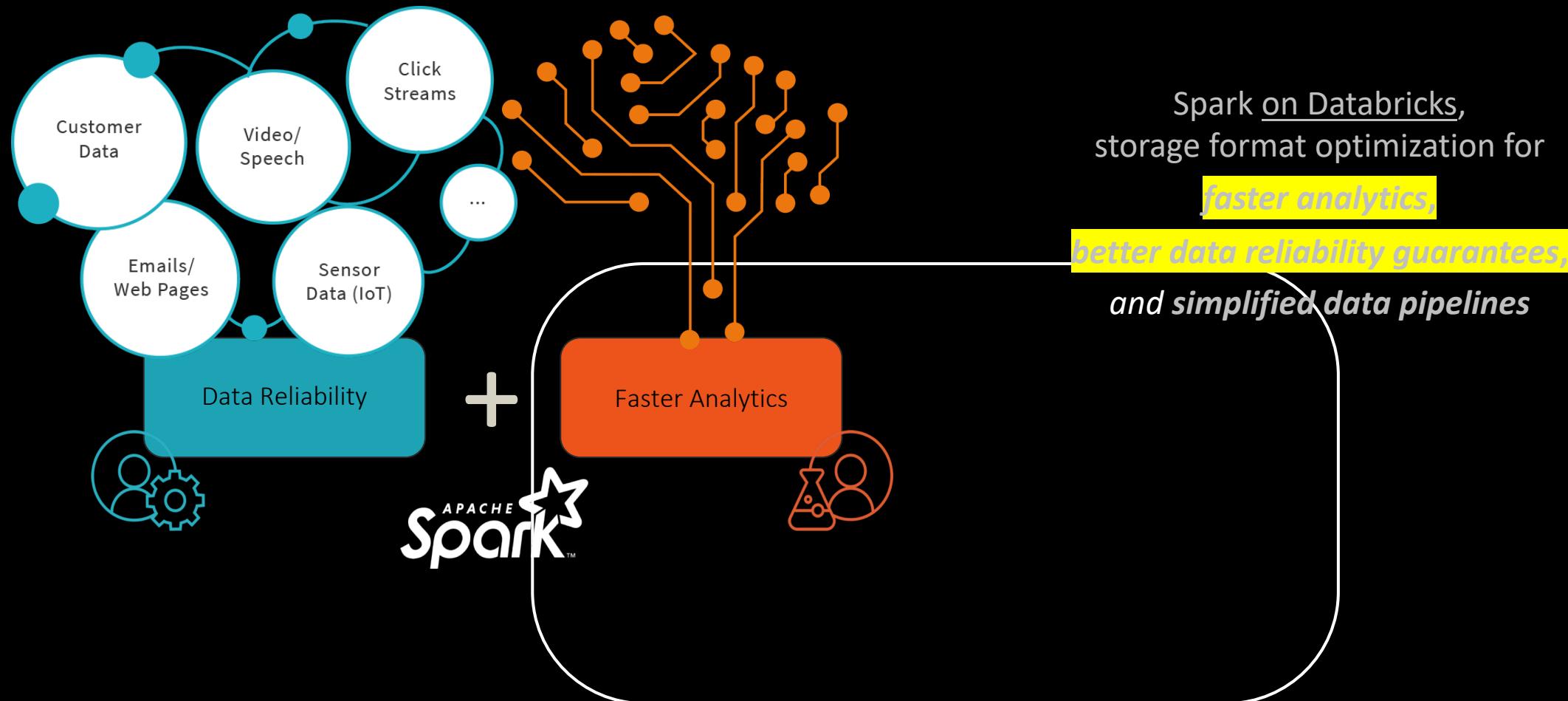


# What is Databricks Delta?

# Databricks Delta is fundamentally.....



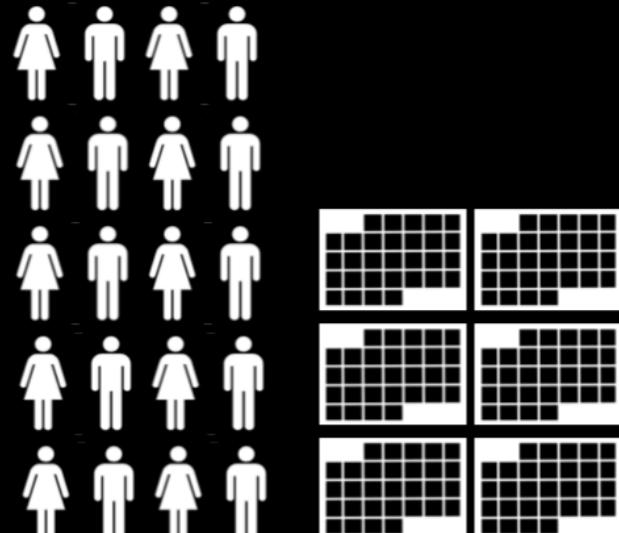
# Promise of Databricks Delta



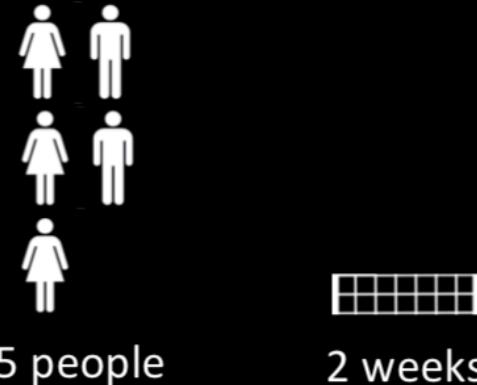
# Info Sec Project

**Project description:** enable streaming analytics and standard reporting on 1 trillion events per day with ACID transactions, serverless architecture, compressed indexed storage, decoupled from compute, transactional reliability, and scalable on demand.

2016



2017



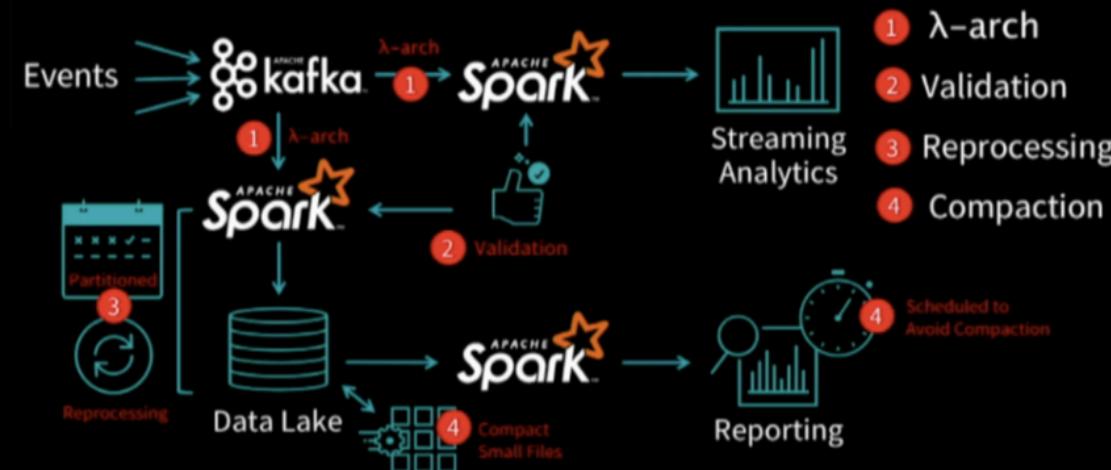
42 X

Reduction  
in  
Effort

How is a **42x**  
improvement possible?

# Databricks Delta

## Architecture Before Databricks Delta



## The Delta Architecture



<https://spark-summit.org/eu-2017/events/announcing-databricks-delta/>

# Throughput & Latency: Structured Streaming

## THROUGHPUT

- Improved throughput in 2.2 due to tight integration with Tungsten and Catalyst

## LATENCY

- Single event mode available in Spark 2.3 with sub-millisecond latency comparable to Flink

## INTEGRATION

- Nearly identical syntax with batch, just add stream to action directive, e.g. “writeStream”

## DEBUGGING

- Live monitoring and debugging in Databricks UI

