Azure Databricks Overview, Delta Lakes and MLFlow

VISION

Accelerate innovation by unifying data science, engineering and business

---

SOLUTION

Unified Analytics Platform powered by Apache Spark

WHO WE ARE

- Founded by the original creators of Apache Spark
- Contributes **75%** of the open source code, **10x** more than any other company
- Trained **100k+** Spark users on the Databricks platform

Azure Databricks

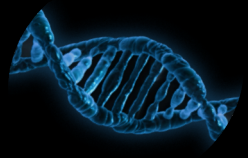
Spark has become the 1st unified engine (Defacto standard) for processing both Big Data & ML

AI has huge promise…..$1.2 Trillion Market

Huge disruptive innovations are affecting most enterprises on the planet

Keystone Research: *"Organizations that harness Data, Cloud and AI vastly outperform other companies in quartile"*

Transportation



Healthcare and Genomics

Internet of Things

Fraud Prevention

Prediction Personalization

However, only **1%** of companies are very successful with AI

Google

UBER

a

facebook

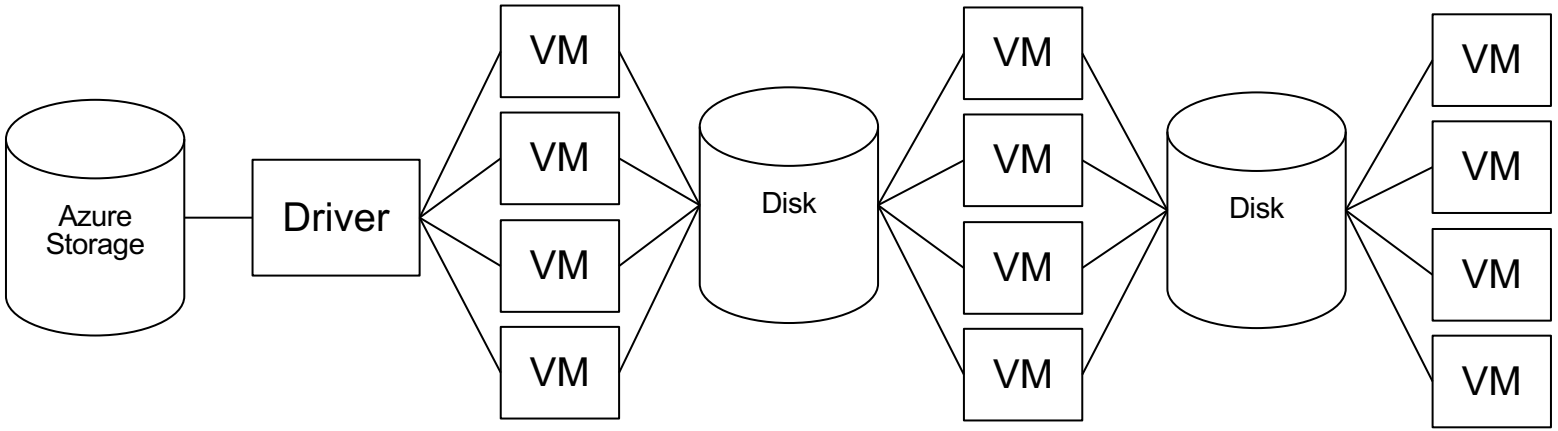
NETFLIX

- Access to large free data set with labels
- Thousands of engineers and data scientists
- Massive infrastructure for specific problems

Azure Databricks

Hadoop MapReduce

MapReduce in Hadoop

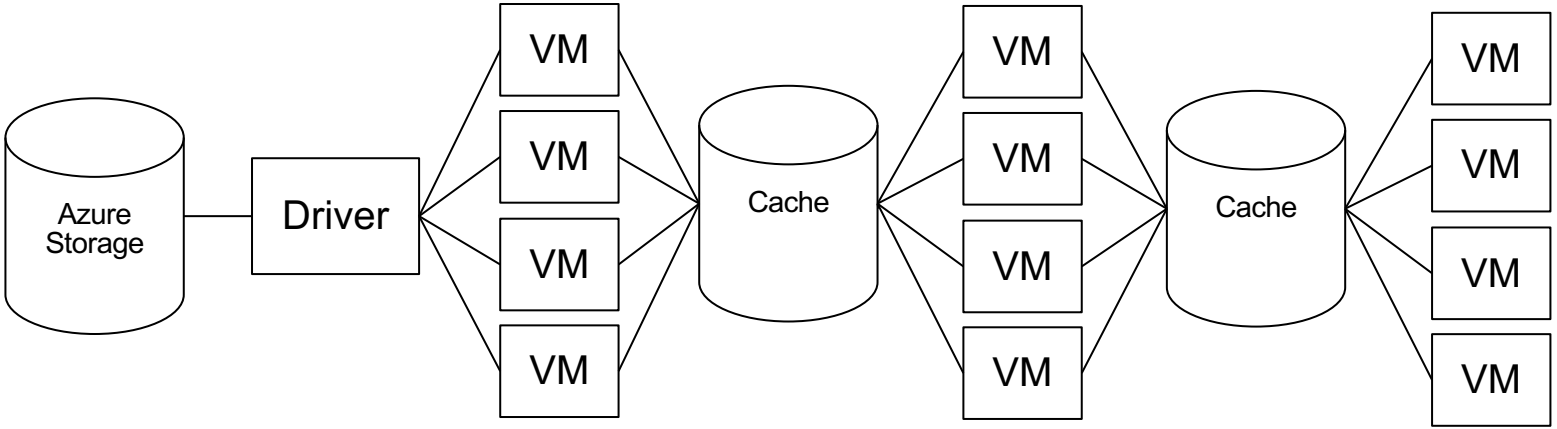


Azure Storage > Driver > VM/Parallelization > write to Disk > VM/Parallelization > write to disk > repeat…

Writing to disk takes time… every time you run this process in MapReduce

What is Azure Databricks?

Apache® Spark™ is FASTER and EASIER than MapReduce in Hadoop

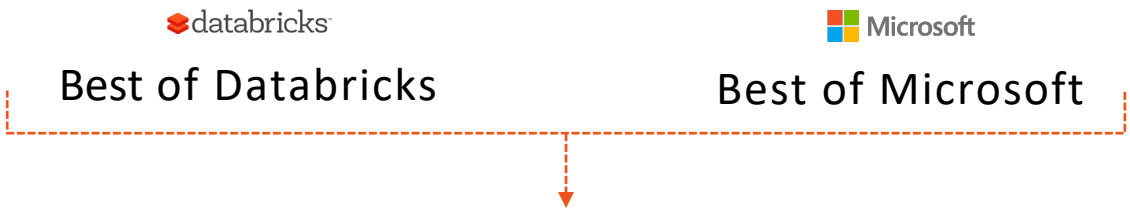


Faster – In Spark data stays in cache this give Spark the speed over MapReduce (writing to disk)

Easier – You can use the language you are most comfortable with in Spark (Python, Scala, R, SQL)

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)

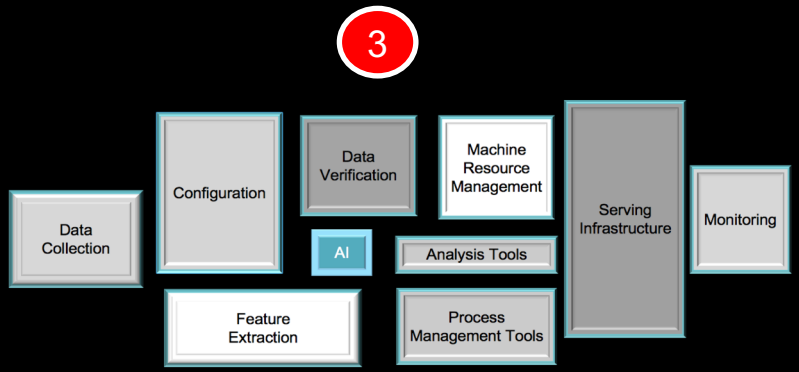
Enterprise-grade Azure security (Active Directory integration, compliance, enterprise-grade SLAs)

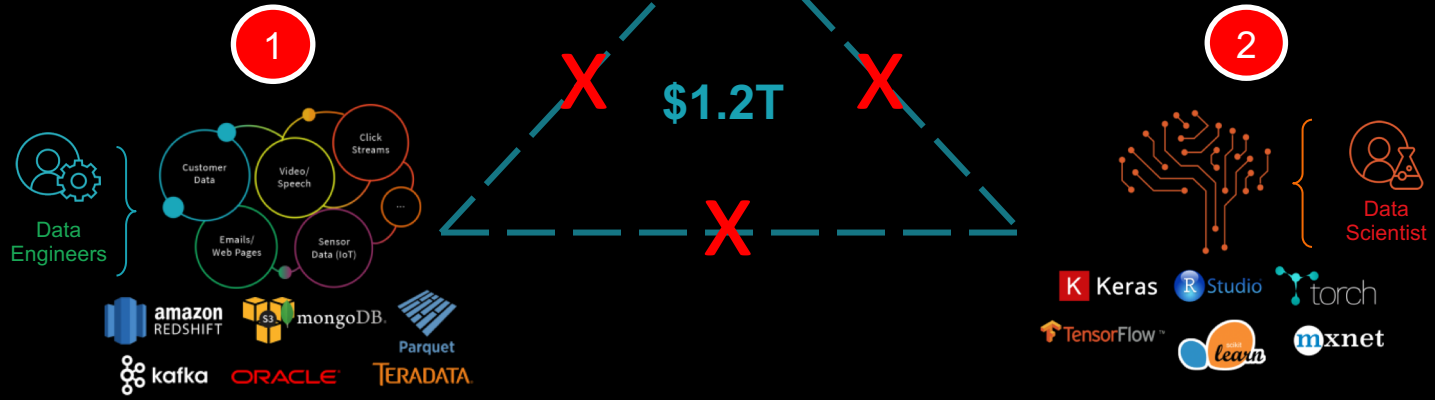
What is the Problem:

Gartner

"Through 2017, Most big-data projects will fail to go beyond piloting and experimentation and will be abandoned."

"Hidden Technical Debt in Machine Learning Systems," Google NIPS 2015"





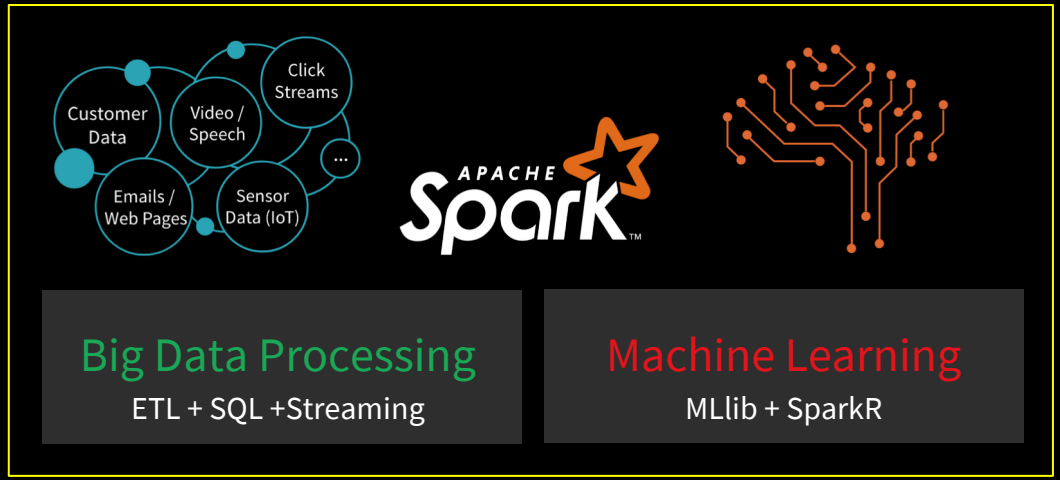
(Great for Data, but not AI)

(Great for AI, but not for data)

Apache Spark: The First Unified Analytics Engine

1 Engine

Uniquely combines Data & AI technologies

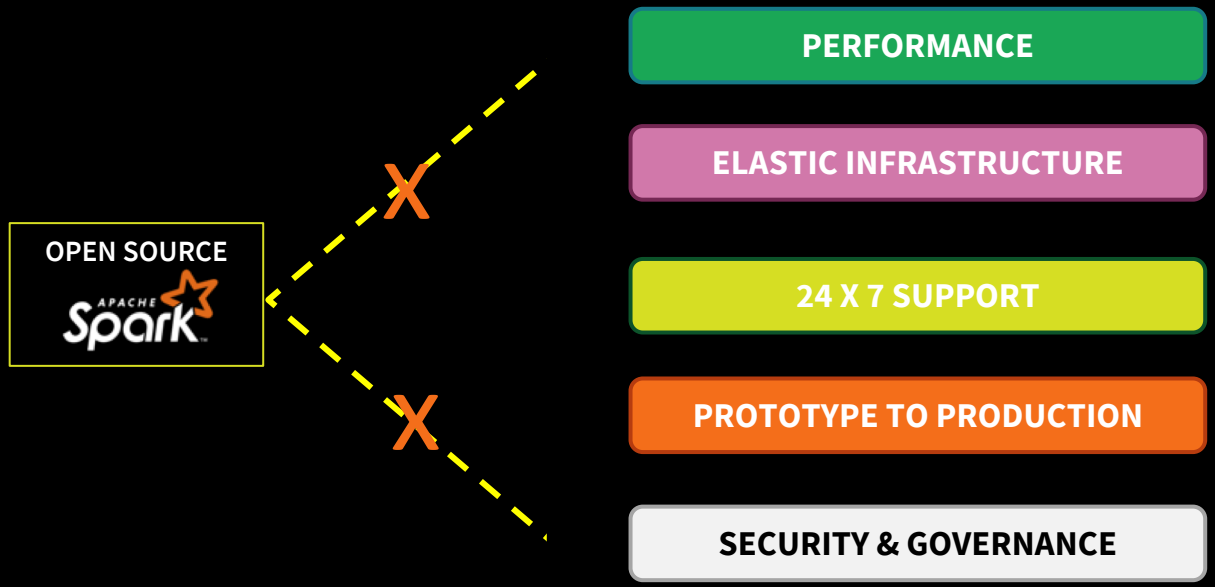


S3 Parquet hadoop kafka Azure

Azure Databricks

Apache Spark was only the 1st step to solve the problem:

...just an engine



Needs of the Enterprise

Azure Databricks

The Next Generation of Apache Spark:

…The Databricks Unified Analytics Platform

COLLABORATIVE PROTOTYPE TO PRODUCTION

EXPERT OPTIMIZATION 24x7 SUPPORT

ELASTIC INFRASTRUCTURE

PERFORMANCE

OPEN SOURCE

Databricks Runtime = ETL Acceleration

Databricks IO: Built-in connectors with optimized cloud access = 10x + speed increase

Ephemeral Instances

Instance Reuse

Spot Instance

Fall Back to On Demand

Elastic Storage

Expert Training by Spark Committers

Spark Support by Spark Committers

Professional Services to build and optimize

Zero-downtime upgrades

Next Spark Version

One-click from Notebook to Job

Single UI Workspace

REST APIs

Job Scheduler

Multi Spark Versions

Revision Control

Streaming Metrics Visualization

Snapshot Spark UI, Logs, & Job Output

Notebook Workflows

Enterprise Security Regulatory Compliance

Azure Databricks Platform Architecture

Azure Databricks

AZURE DATA SOURCES

Blob Storage

Data Lake Store

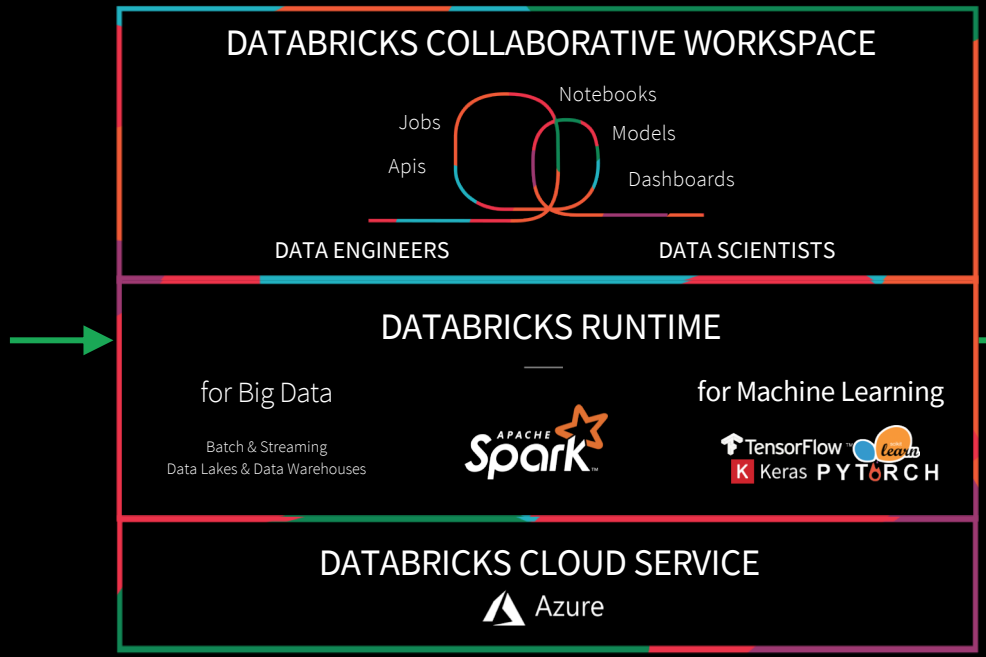
SQL Data Warehouse

Cosmos DB

Event Hub

IoT Hub

Azure Data Factory



Power BI

BI Reporting
Dashboards

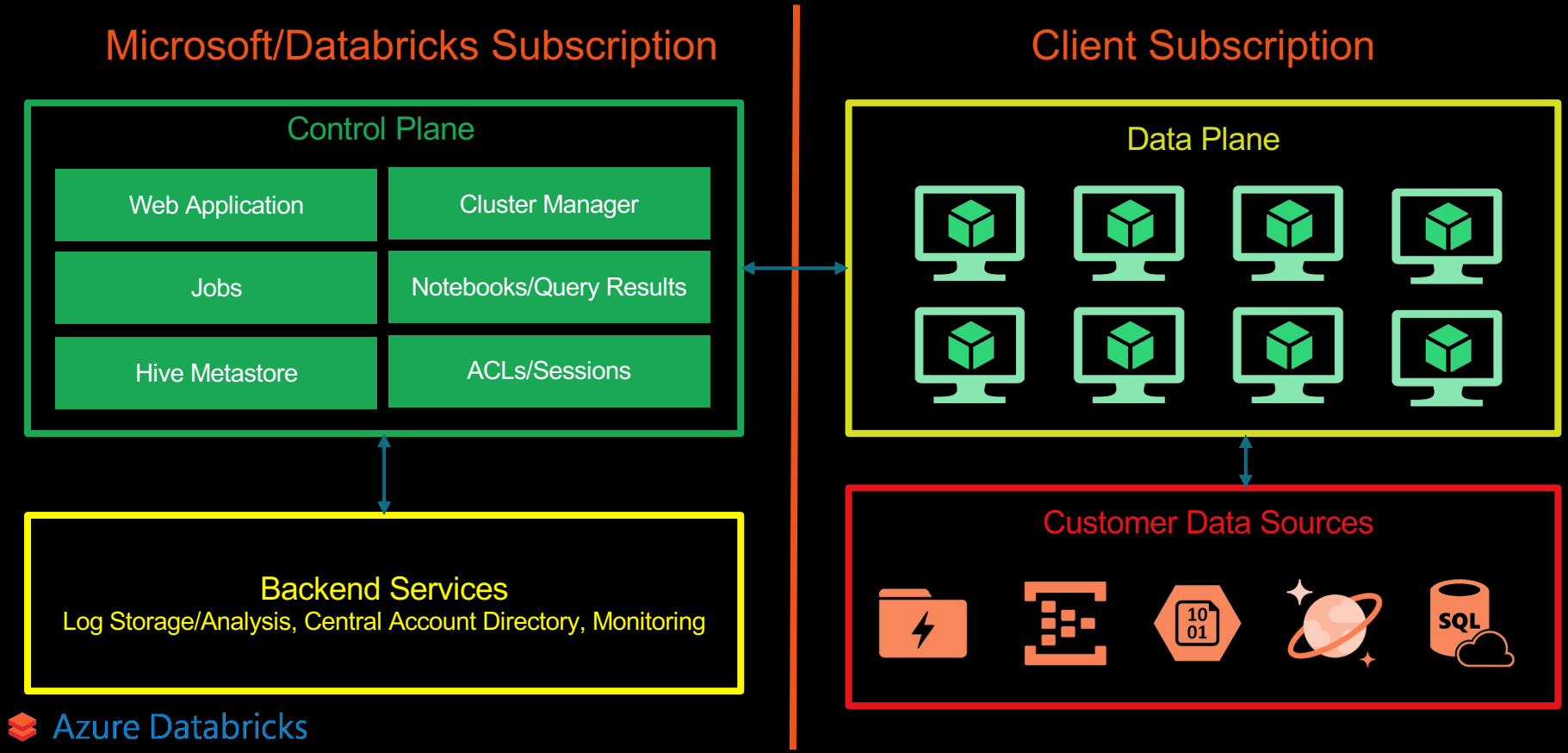
Microsoft Active Directory

Security Integration

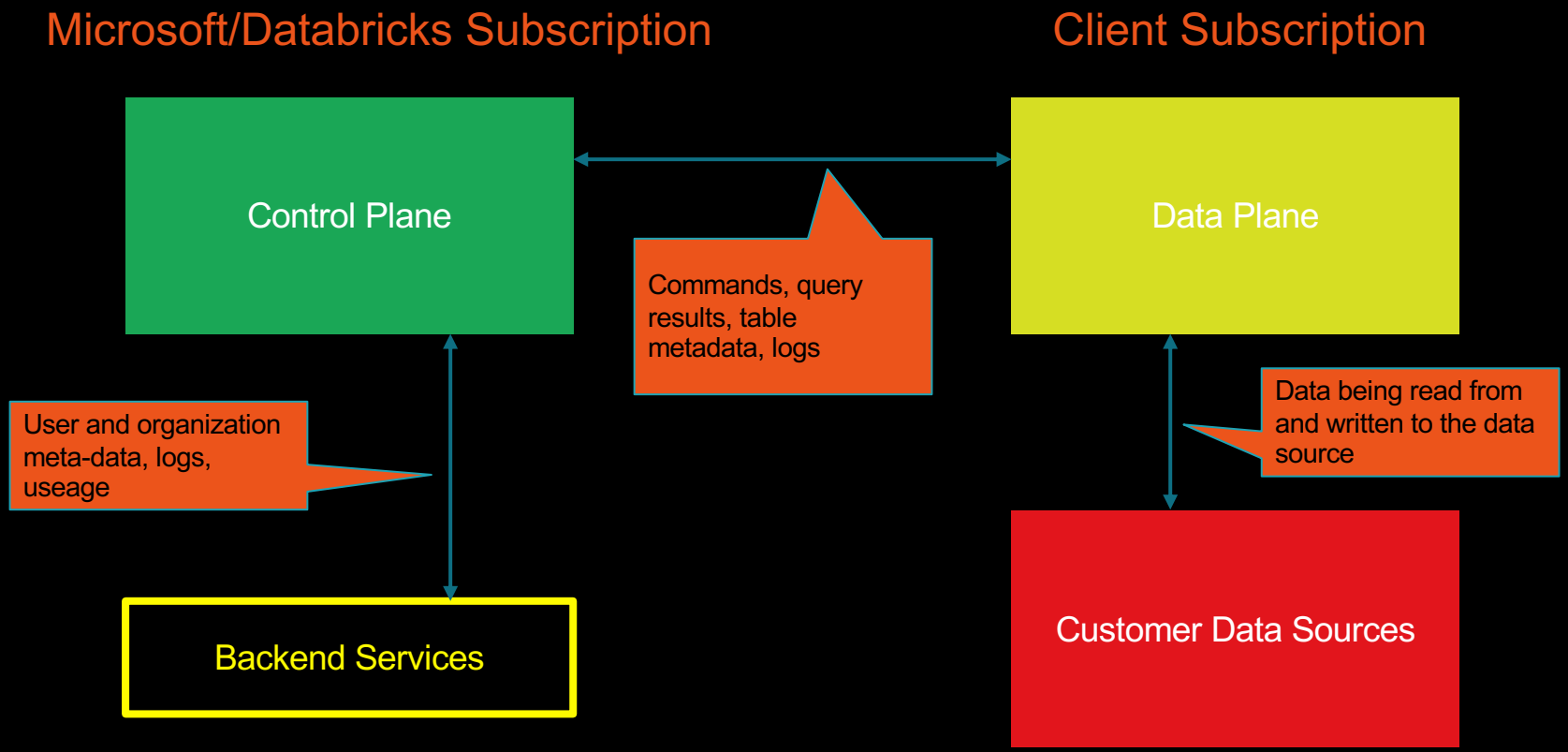
Azure Portal
One-Click setup
Unified Billing

Azure Databricks

Azure Databricks Platform Architecture

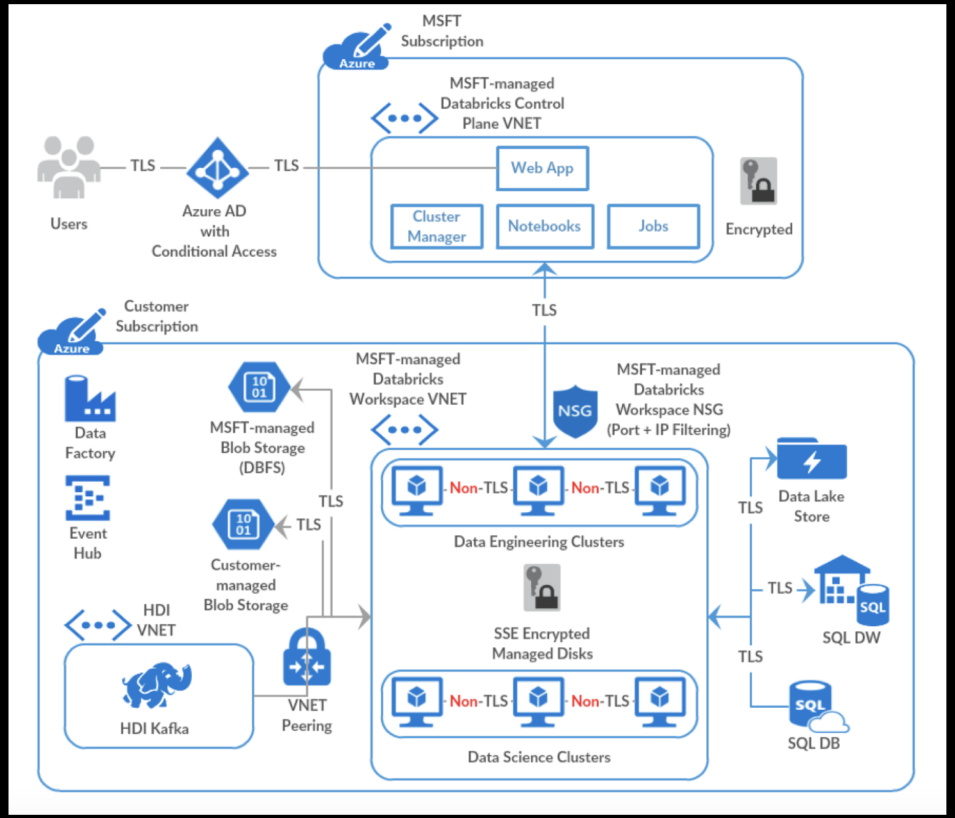


Azure Databricks Platform Architecture Cont'd



Azure Databricks Platform Architecture

Standard Deployment View with no inter-node TLS



Azure Databricks

Customer Case Study

SEVATEC SHIFT UP

Background

Sevatec supports a number of mission-critical applications across the federal government, to include Homeland Security, Department of Defense, Department of Transportation, Department of State, and other federal and civilian agencies.

CHALLENGE

Difficult to ingest and prepare data across 30+ disparate systems. Support a 2000+ users who are siloed and have different skill sets (BI users, statisticians, data engineers, data scientists, business)

Used a variety of disjointed tools to perform large data extractions which created substantial DevOps complexity
System was being heavily taxed by very IO intensive queries, impacting their ability to meet SLAs and the demands of the rest of their user community

DATABRICKS IMPACT

Democratize access to data across their various data sources through APIs and data source connectors. **Reduced access an ingest times from hours to minutes**.
Able to build build machine learning models at scale against the entire data set

Customer Case Study

GOAL	Analyze IoT data to predict switch failures and keep customers online
CHALLENGE	Inefficient detection of equipment failures resulted in a 60% detection rate of failures, leaving customers with more downtime
DATA	2 million switch records took 6 hours to process. Increased to 10 billion records with Databricks
DATABRICKS IMPACT	**10 billion records processed in 14 minutes** and a 94% detection rate meant 25,000 homes were kept online resulting in a better customer experience

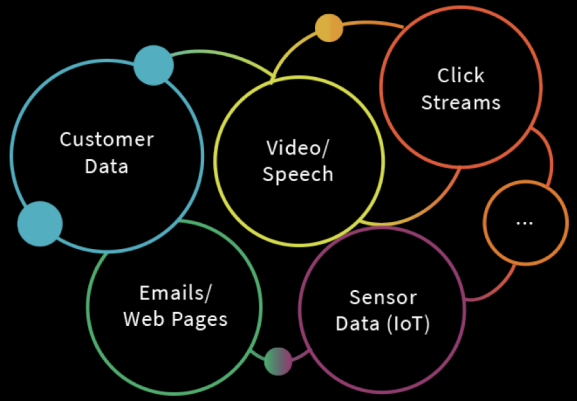
Broad Customer Adoption

- Now generally available (as of March 2018)
- Over 500 customers took part in the preview of Azure Databricks
- Widely adopted in many industries (e.g. Retail, Media & Entertainment, Healthcare)

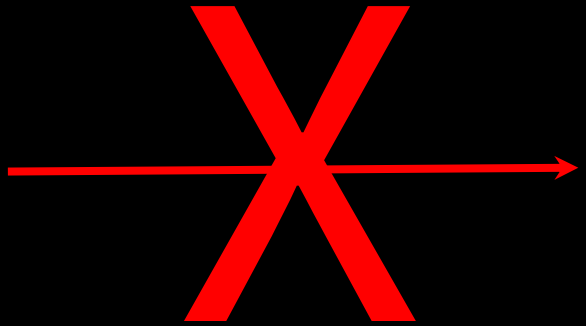
But the data is not ready for data science & ML

> 65% big data projects fail per Gartner

Unreliable Low Quality Data Slow Performance



Data Lake

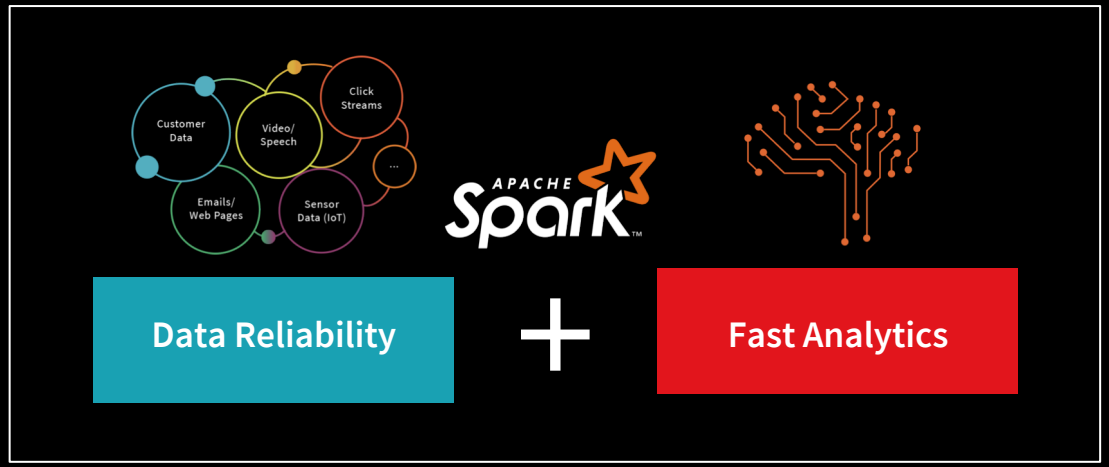


Data Science and ML

- Recommendation Engines
- Risk, Fraud, & Intrusion Detection
- Customer Analytics
- IoT & Predictive Maintenance
- Genomics & DNA Sequencing

Azure Databricks

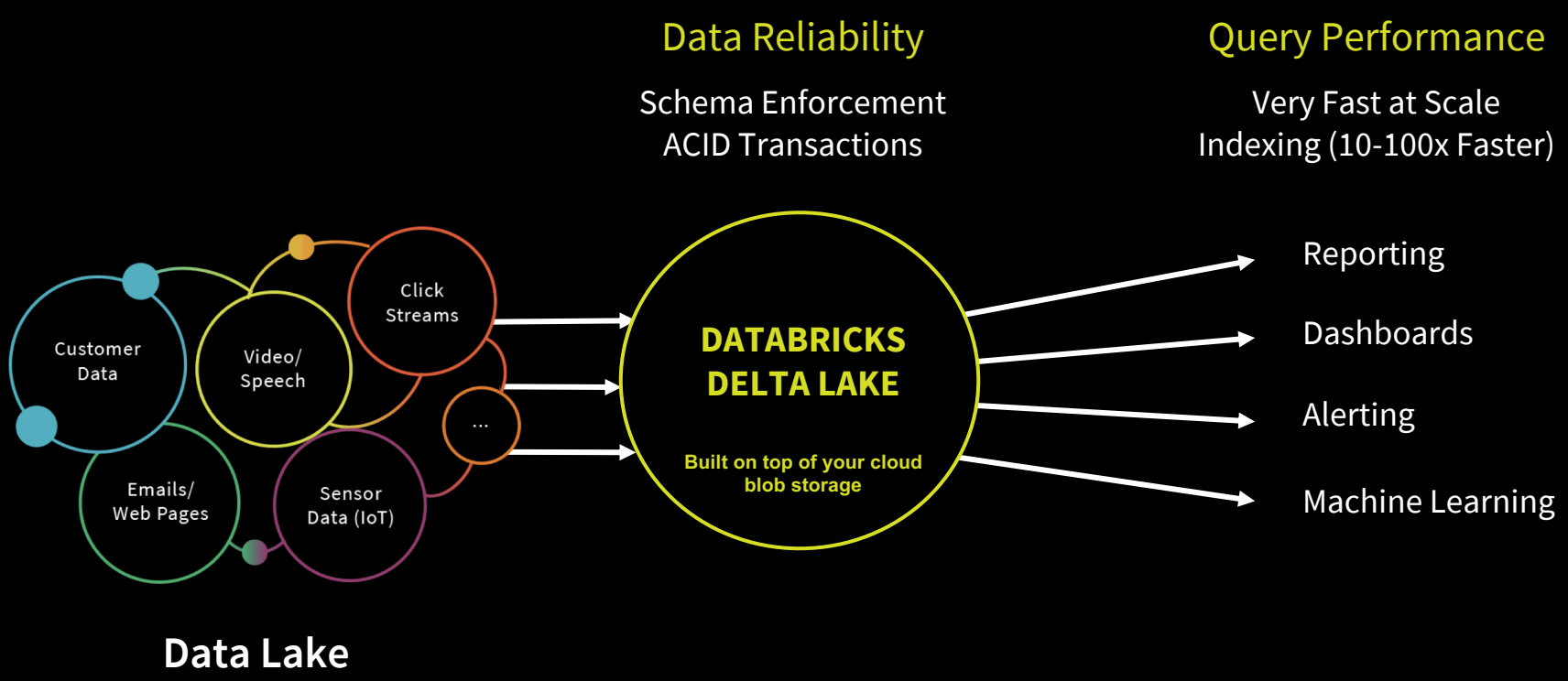
Databricks Delta Lake
Next Generation Unified Analytics Engine



Brings data reliability and performance to data lakes

Azure Databricks

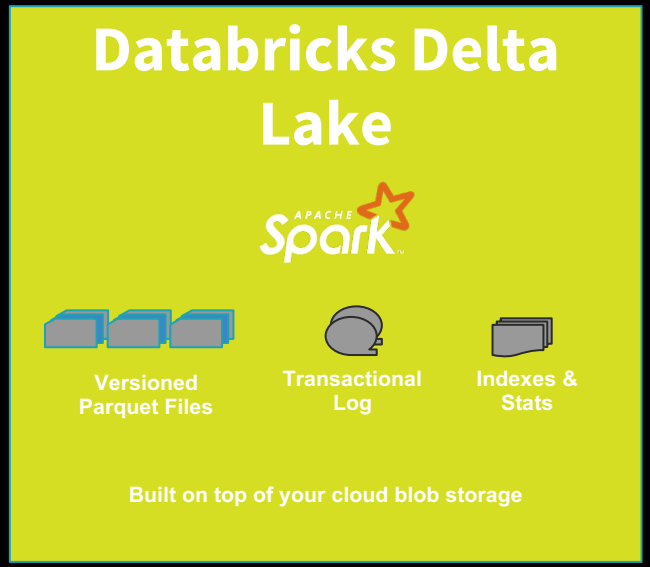
Databricks Delta Lake: makes data ready for Analytics



Azure Databricks

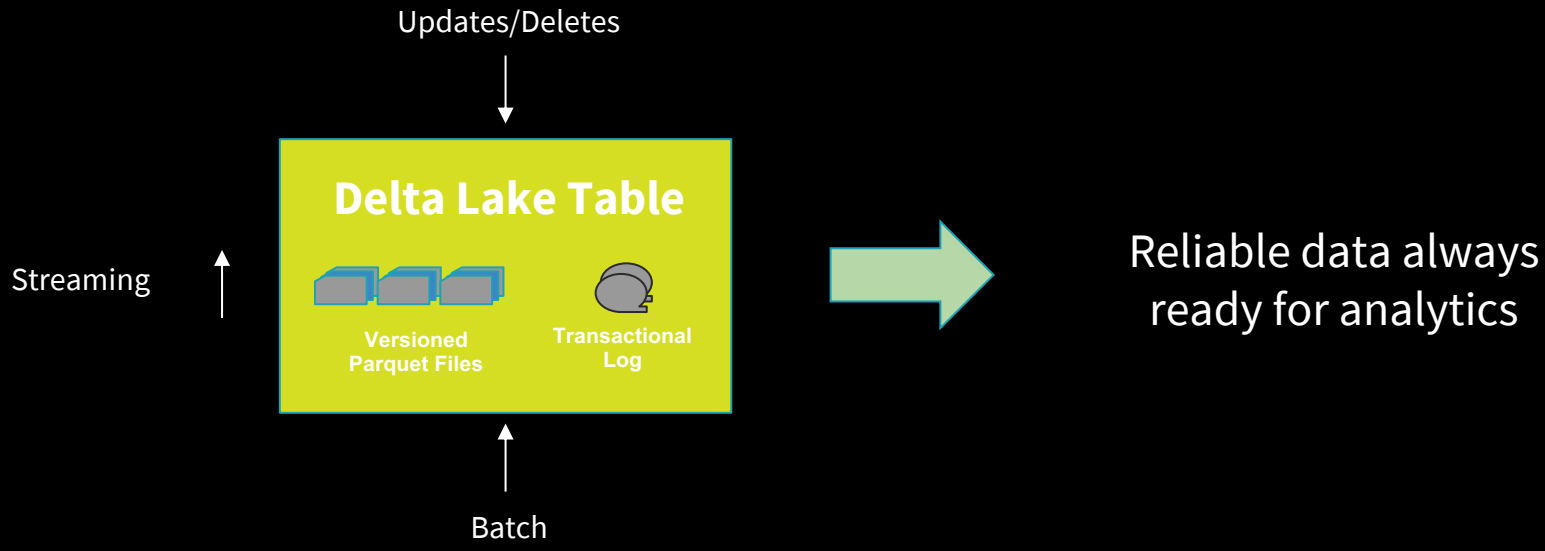
Databricks Delta Lake

Next-generation unified analytics engine



- Co-designed compute & storage
- Compatible with Spark API's
- Built on open standards (Parquet)

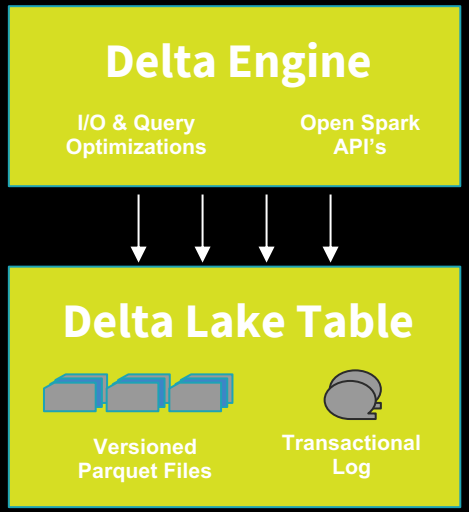
Delta Lake Makes Data Reliable



Key Features

- ACID Transactions
- Schema Enforcement
- Upserts
- Data Versioning

Delta Lake Makes Data More Performant



Fast, highly responsive queries at scale

Key Features

- Compaction
- Caching
- Data skipping
- Z-ordering

Get Started with Delta Lake using Spark APIs

Instead of **parquet**...

```
CREATE TABLE ...
USING parquet
...

dataframe
    .write
    .format("parquet")
    .save("/data")
```

... simply say **delta**

```
CREATE TABLE ...
USING delta
…

dataframe
    .write
    .format("delta")
    .save("/data")
```

Migrating your Spark jobs to Delta Lake

Step 1: Convert Parquet to Delta Tables

```
CONVERT TO DELTA parquet.`path/to/table` [NO STATISTICS]
[PARTITIONED BY (col_name1 col_type1, col_name2 col_type2, ...)]
```

Step 2: Optimize Layout for Fast Queries

```
OPTIMIZE events
WHERE date >= current_timestamp() - INTERVAL 1 day
ZORDER BY (eventType)
```

Upsert/Merge: Fine grained updates

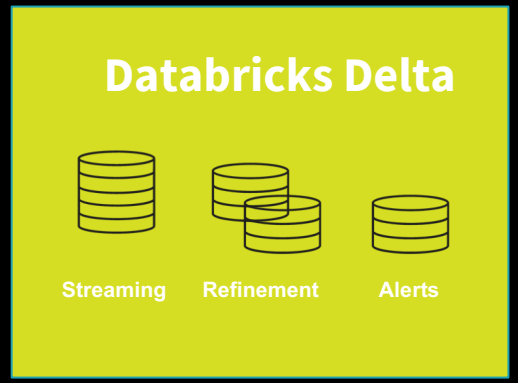
```
MERGE INTO customers     -- Delta table
USING updates
ON customers.customerId = source.customerId
WHEN MATCHED THEN
    UPDATE SET address = updates.address
WHEN NOT MATCHED
    THEN INSERT (customerId, address) VALUES (updates.customerId,
updates.address)
```

Apple: Threat Detection at Scale with Delta Lake

Detect signal across user, application and network logs; Quickly analyze the blast radius with ad hoc queries; Respond quickly in an automated fashion; Scaling across petabytes of data and 100's of security analysts

> 100TB new data/day

> 300B events/day



Data Science

Machine Learning

KEYNOTE TALK

BEFORE DELTA LAKE

- *Took 20 engineers; 24 weeks to build*
- *Only able to analyze 2 week window of data*

WITH DELTA LAKE

- *Took 2 engineers; 2 weeks to build*
- *Analyze 2 years of batch with streaming data*

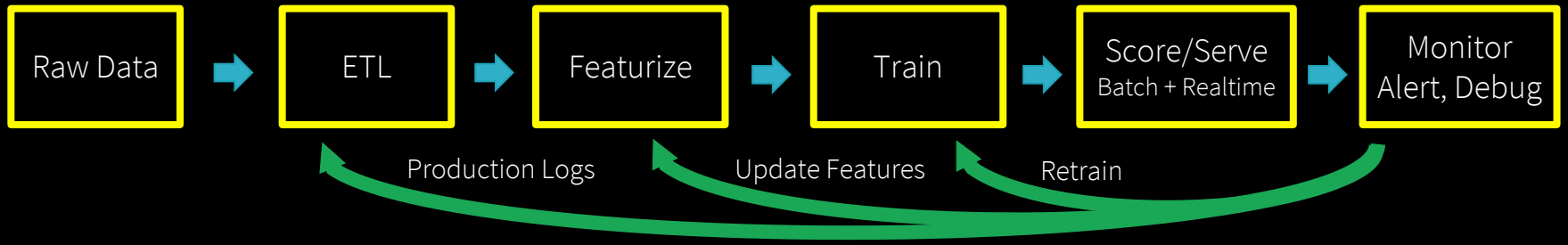
Introduction to mlflow

databricks

Machine Learning Development is Complex

ML Lifecycle and Challenges

mlflow
An open source platform for the machine learning lifecycle



Zoo of Ecosystem Frameworks

Tuning | Deploy | Model Mgmt

Collaboration | Scale | Governance

Feature Repository | Experiment Tracking | AutoML, Hyper-p. search | Remote Cloud Execution | Project Mgmt (scale teams) | Model Exchange | A/B Testing | CI/CD/Jenkins push to prod | Orchestration (Airflow, Jobs) | Lifecycle mgmt. | Data Drift | Model Drift

Azure Databricks

+mlflow

Vision: Make it painless for a single user to go from raw data to production scoring without leaving Databricks

What is mlflow

Open source project

Conventions, specs, tools

CLI, libraries, REST service

Community

Databricks is the best place to run MLflow

MLflow Components

mlflow Tracking

Record and query experiments: code, data, config, results

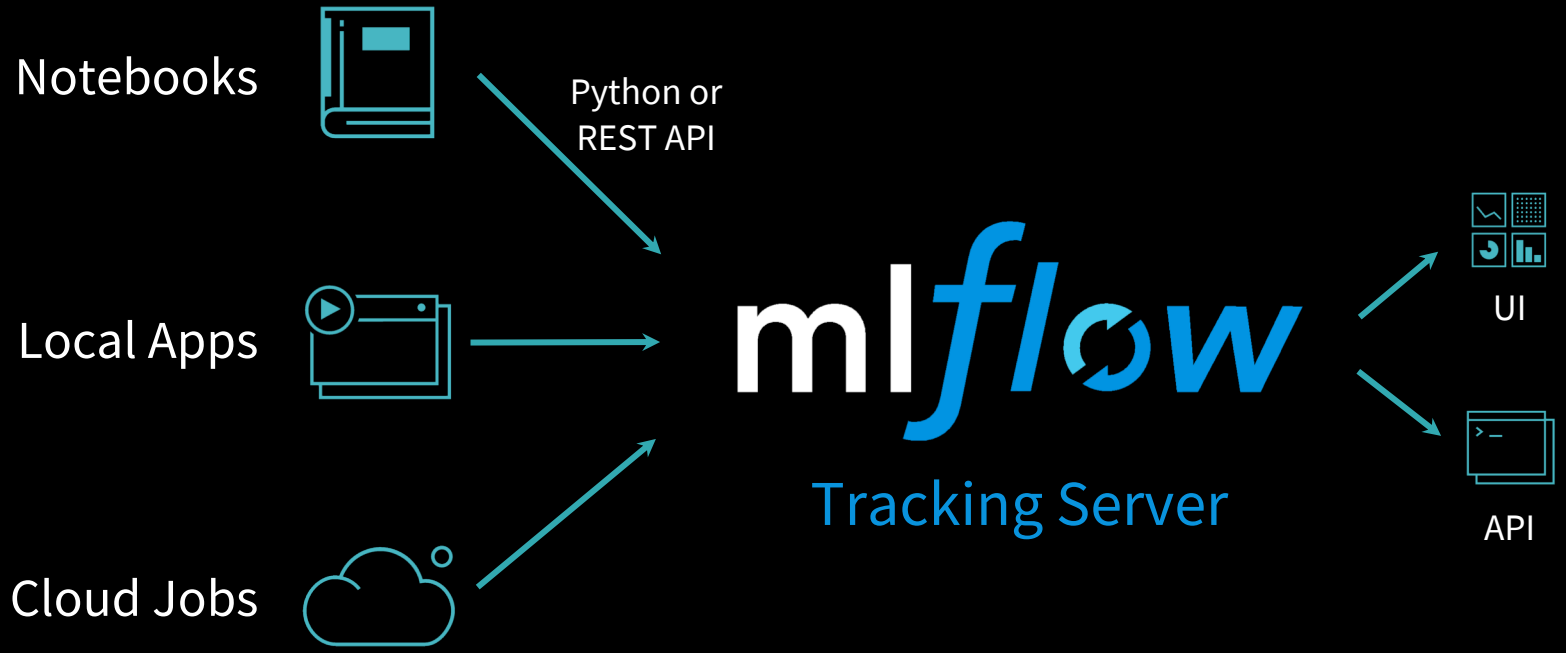
mlflow Projects

Packaging format for reproducible runs on any platform

mlflow Models

General model format that supports diverse deployment tools

MLflow Tracking



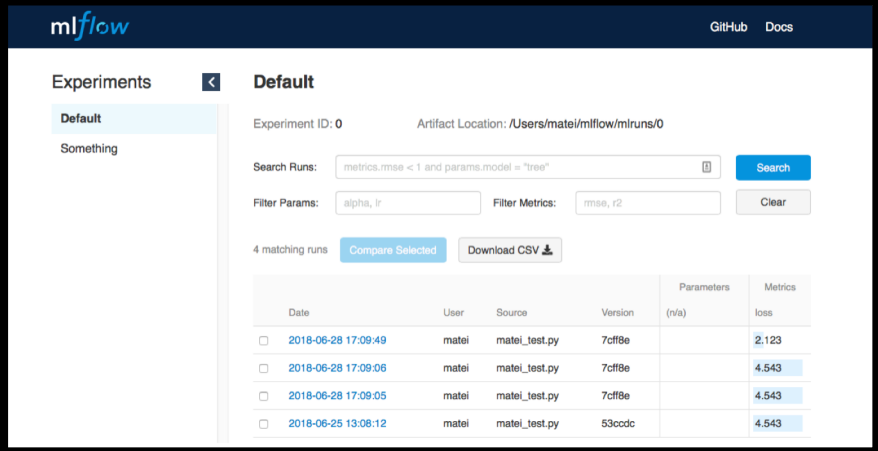
Key Concepts in Tracking

Parameters: key-value inputs to your code

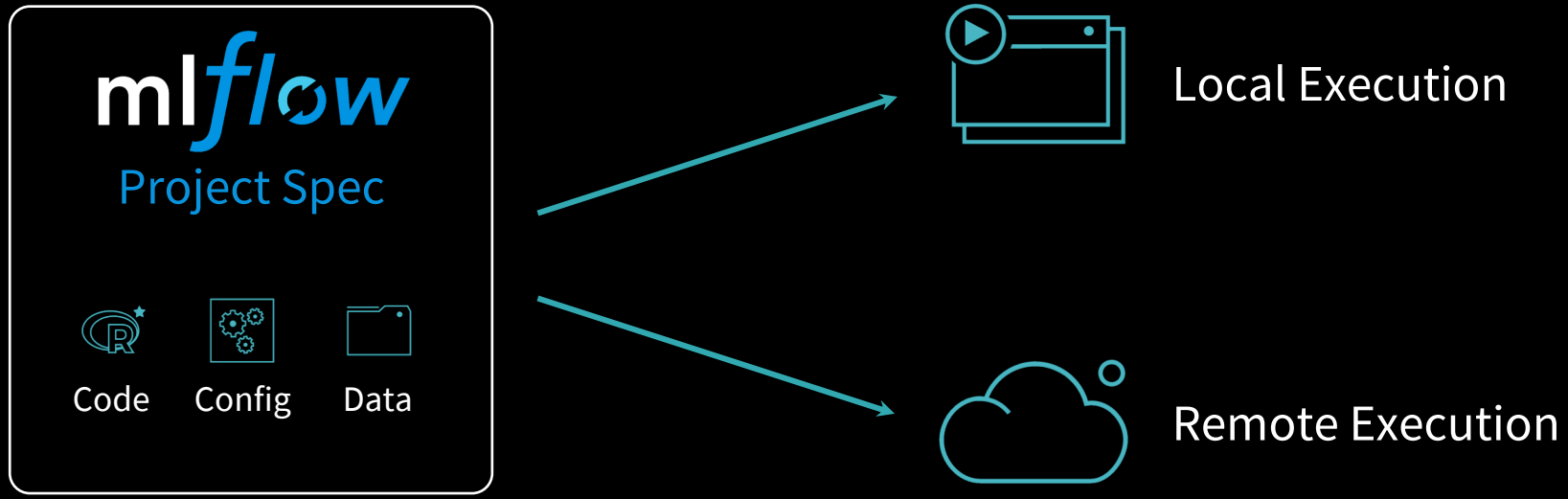
Metrics: numeric values (can update over time)

Artifacts: arbitrary files, including models

Source: what code ran?



MLflow Projects



Example MLflow Project

```
my_project/
├── MLproject
│
│
│
│
│
│
│
├── conda.yaml
├── main.py
└── model.py
    ...
```

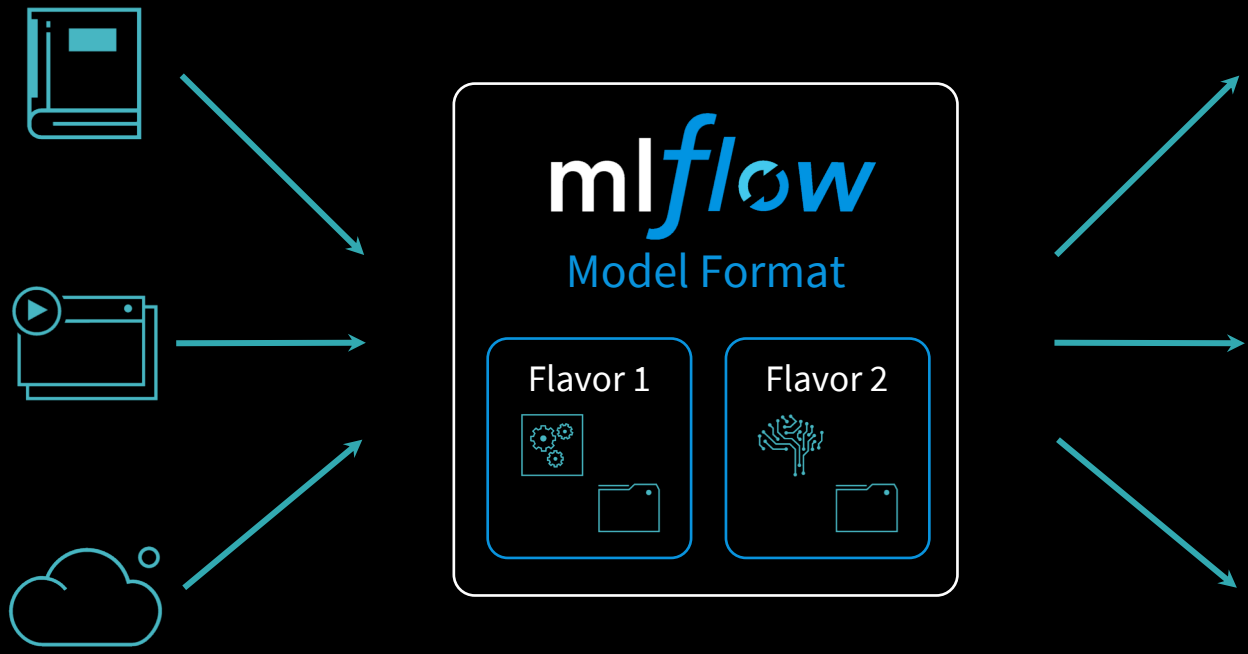
```
conda_env: conda.yaml

entry_points:
  main:
    parameters:
      training_data: path
      lambda: {type: float, default: 0.1}
    command: python main.py {training_data} {lambda}
```

```
$ mlflow run git://<my_project>
```

```
mlflow.run("git://<my_project>", ...)
```

MLflow Models



Azure Databricks

Example MLflow Model

```
my_model/
├── MLmodel
└── estimator/
    ├── saved_model.pb
    └── variables/
        ...
```

```
run_id: 769915006efd4c4bbd662461
time_created: 2018-06-28T12:34
flavors:
  tensorflow:
    saved_model_dir: estimator
    signature_def_key: predict
  python_function:
    loader_module: mlflow.tensorflow
```

Usable by tools that understand TensorFlow model format

Usable by any tool that can run Python (Docker, Spark, etc!)