Getting Started with Apache Spark

databricks

Welcome and Housekeeping

- You should have received instructions on how to participate in the training session
- If you have questions, you can use the Q&A window in Go To Webinar
- The slides will also be made available to you as well as a recording of the session after the event

About Your Instructor

Doug Bateman is Director of Training and Education at Databricks. Prior to this role he was Director of Training at NewCircle.

Apache Spark - Genesis and Open Source

Spark was originally created at the AMP Lab at Berkeley. The original creators went on to found Databricks.

Spark was created to address bringing data and machine learning together

Spark was donated to the Apache Foundation to create the Apache Spark open source project

VISION

Accelerate innovation by unifying data science, engineering and business

---

SOLUTION

Unified Analytics Platform

WHO WE ARE

- Original creators of 

- 2000+ global companies use our platform across big data & machine learning lifecycle

Introducing Delta Lake

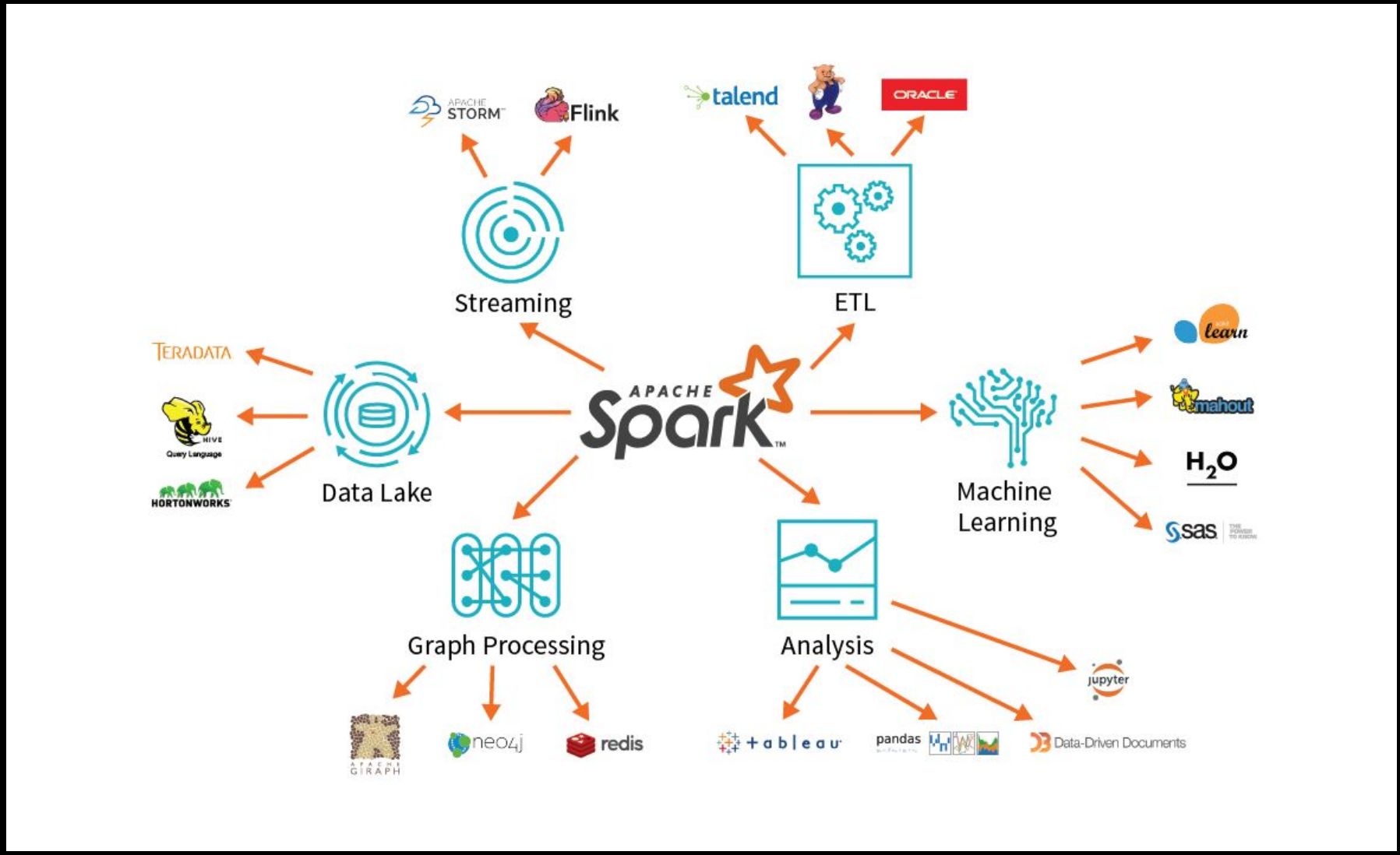
A New Standard for Building Data Lakes

Open Format Based on Parquet

With Transactions

Apache Spark API's

Apache Spark - A Unified Analytics Engine



Apache Spark

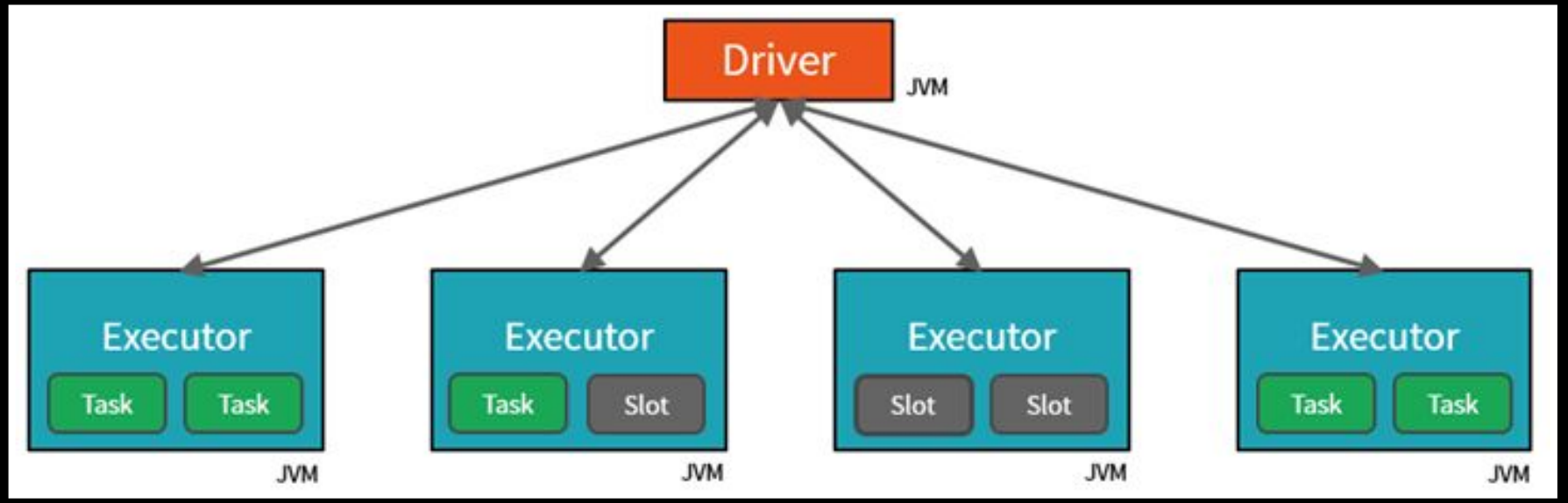
"Unified analytics engine for big data processing, with built-in modules for streaming, SQL, machine learning and graph processing"

- Research project at UC Berkeley in 2009
- APIs: Scala, Java, Python, R, and SQL
- Built by more than 1,200 developers from more than 200 companies

HOW TO PROCESS LOTS OF DATA?

M&Ms

Spark Cluster



One Driver and many Executor JVMs

Spark APIs

- RDD
- DataFrame
- Dataset

RDD

Resilient: Fault-tolerant

Distributed: Computed across multiple nodes

Dataset: Collection of partitioned data

- Immutable once constructed
- Track lineage information
- Operations on collection of elements in parallel

Transformations and Actions

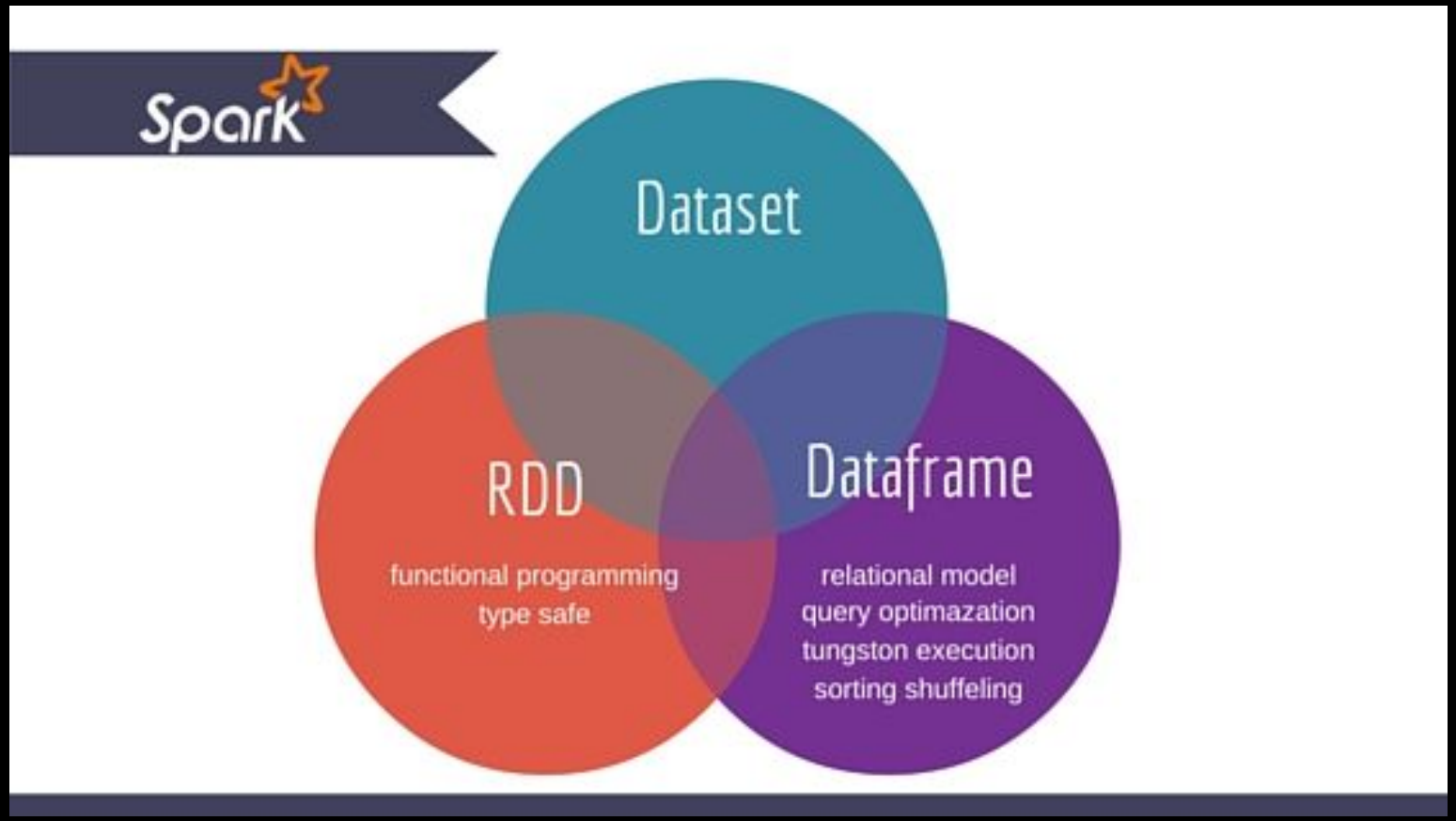
Transformations	Actions
Filter	Count
Sample	Take
Union	Collect

Dataframe

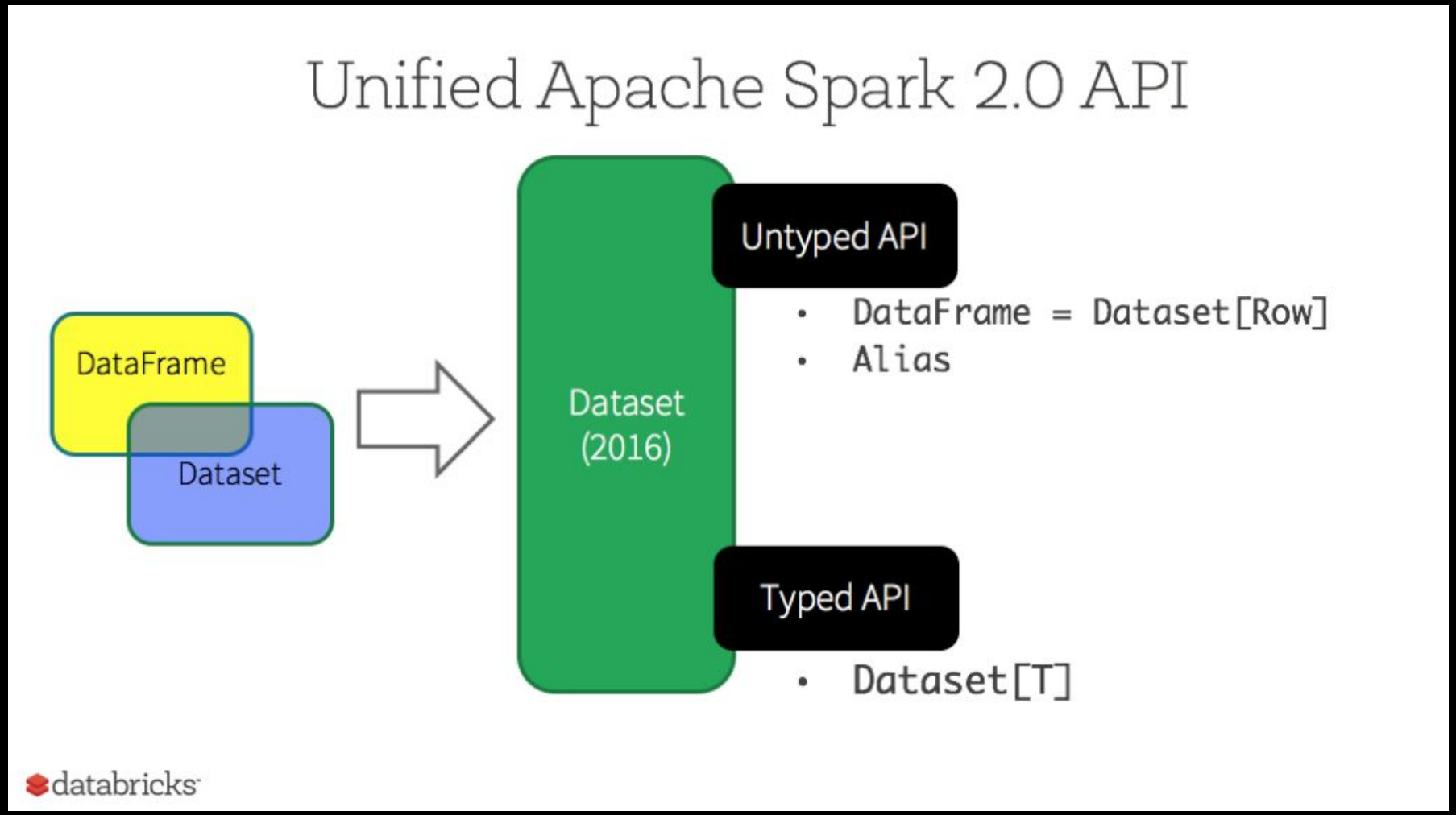
Data with columns (built on RDDs)

Improved performance via optimizations

Datasets


Spark
Dataset
RDD
functional programming
type safe
Dataframe
relational model
query optimazation
tungston execution
sorting shuffeling

Dataframe vs. Dataset



DATAFRAMES

Why Switch to Dataframes?

- User-friendly API

```
dataRDD = sc.parallelize([("Jim", 20), ("Anne", 31), ("Jim", 30)])

# RDD
(dataRDD.map(lambda (x,y): (x, (y,1)))
        .reduceByKey(lambda x,y: (x[0] +y[0], x[1] +y[1]))

        .map(lambda (x, (y, z)): (x, y / z)))

# DataFrame
dataDF = dataRDD.toDF(["name", "age"])

dataDF.groupBy("name").agg(avg("age"))
```

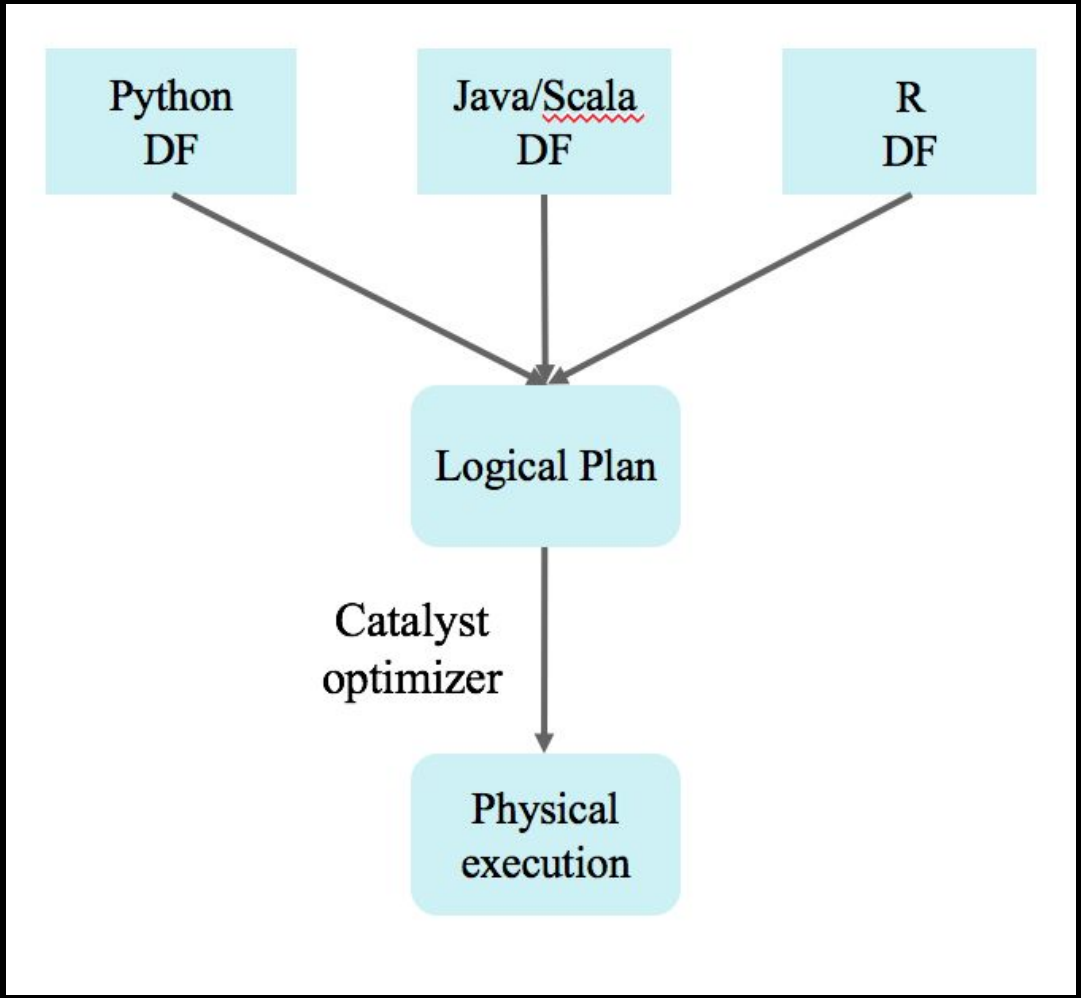
Why Switch to Dataframes?

- User-friendly API

Benefits:

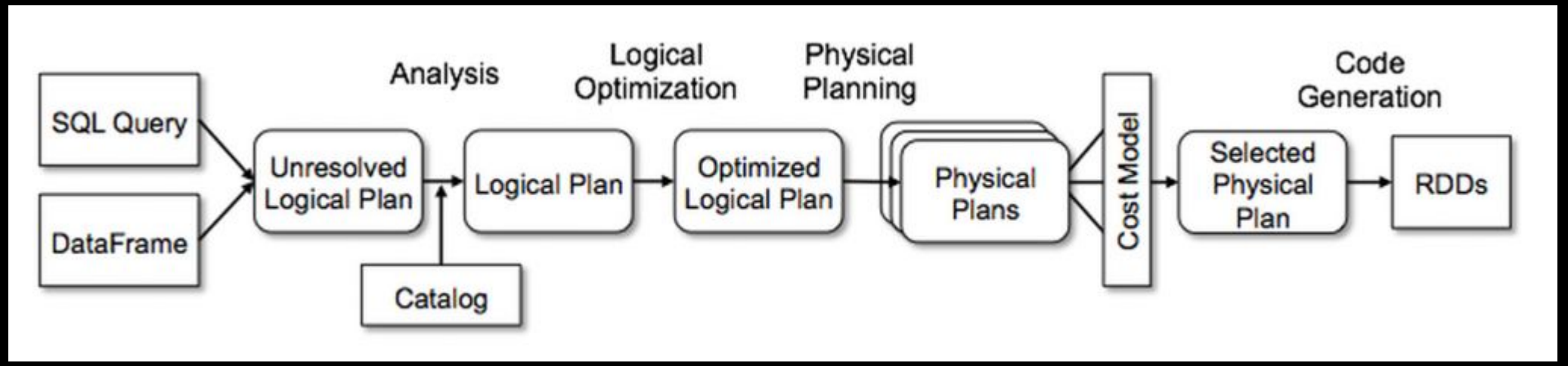
- SQL/DataFrame queries
- Tungsten and Catalyst optimizations
- Uniform APIs across languages

Why Switch to Dataframes?

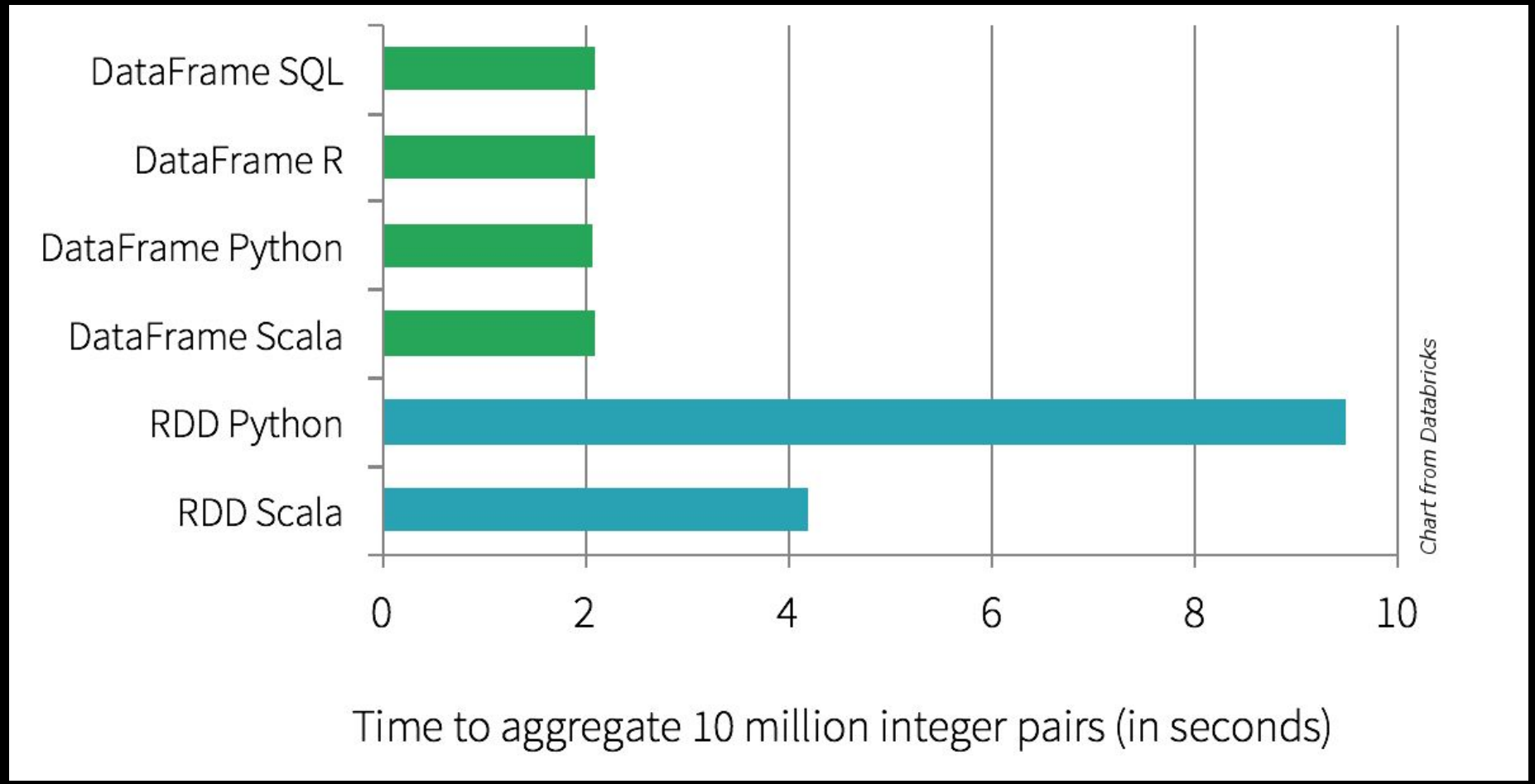


Wrapper to create logical plan

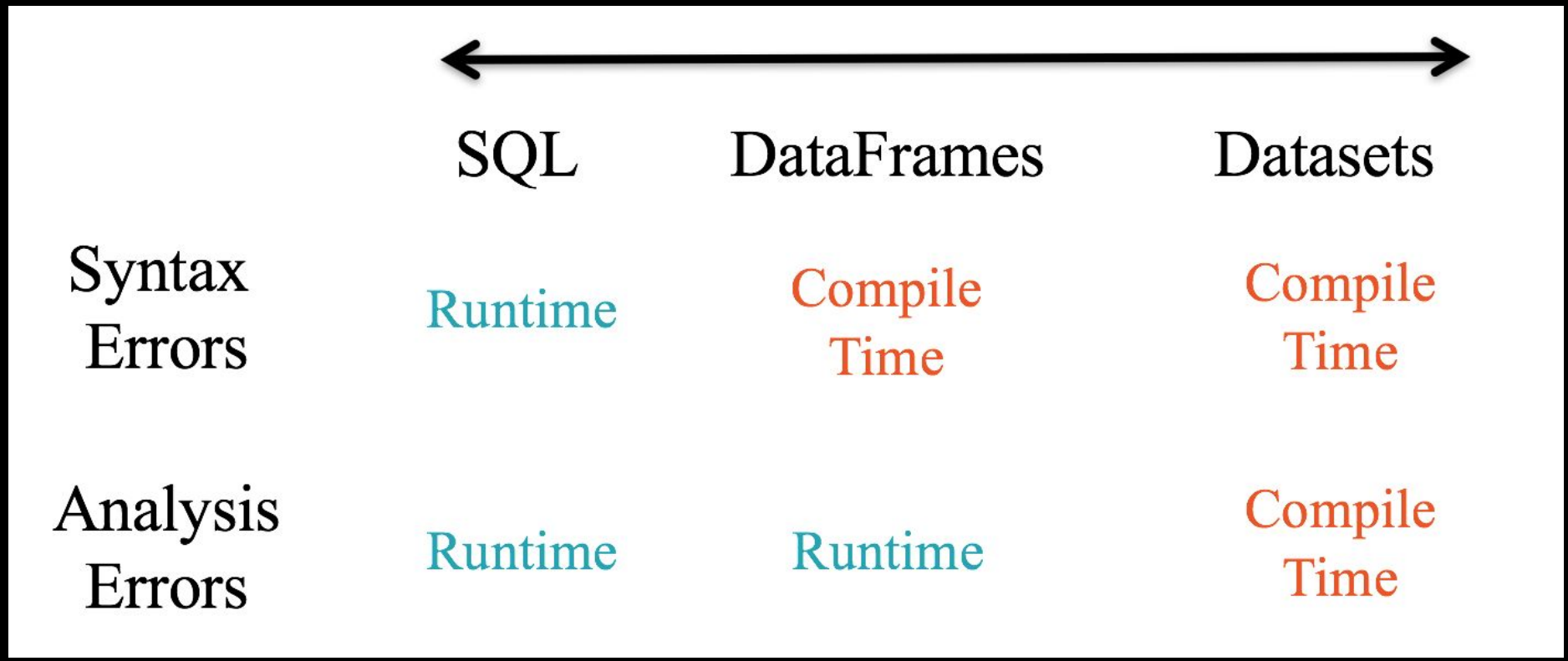
Catalyst: Under the Hood



Still Not Convinced?

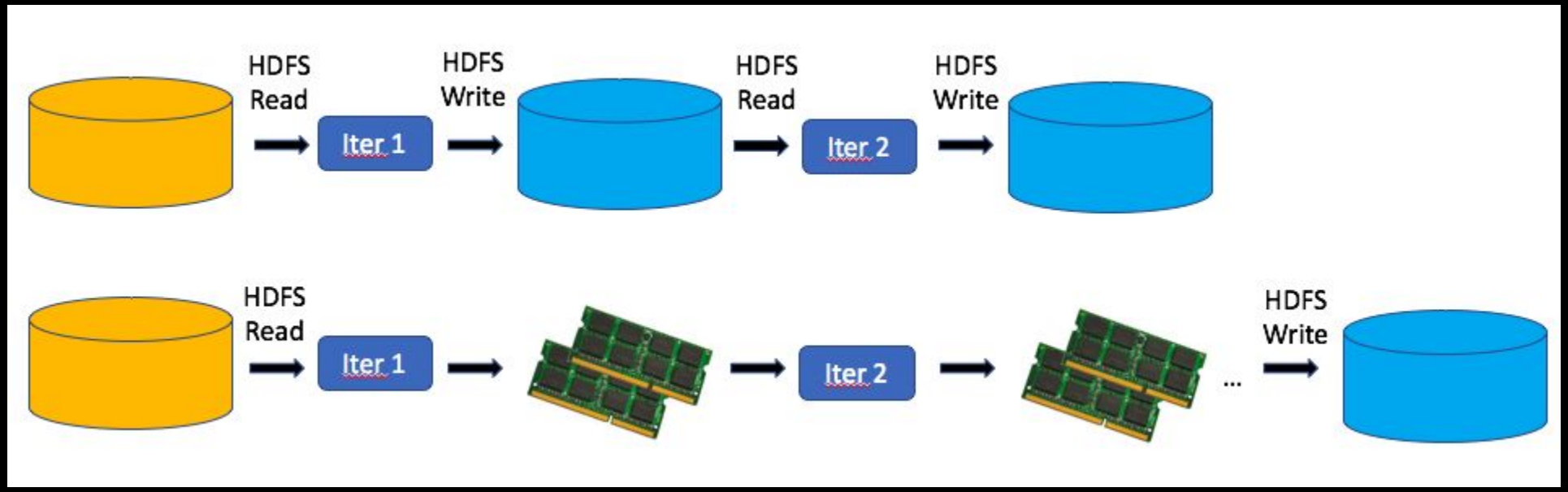


Structured APIs in Spark



WHY SWITCH FROM MAPREDUCE TO SPARK?

Spark vs. MapReduce



When to Use Spark

- Scale out: Model or data too large to process on a single machine
- Speed up: Benefit from faster results

Spark References

- Databricks
- Apache Spark ML Programming Guide
- Scala API Docs
- Python API Docs
- Spark Key Terms

Questions?

Further Training Options: http://bit.ly/DBTrng

- Live Onsite Training
- Live Online
- Self Paced

Meet one of our Spark experts: http://bit.ly/ContactUsDB