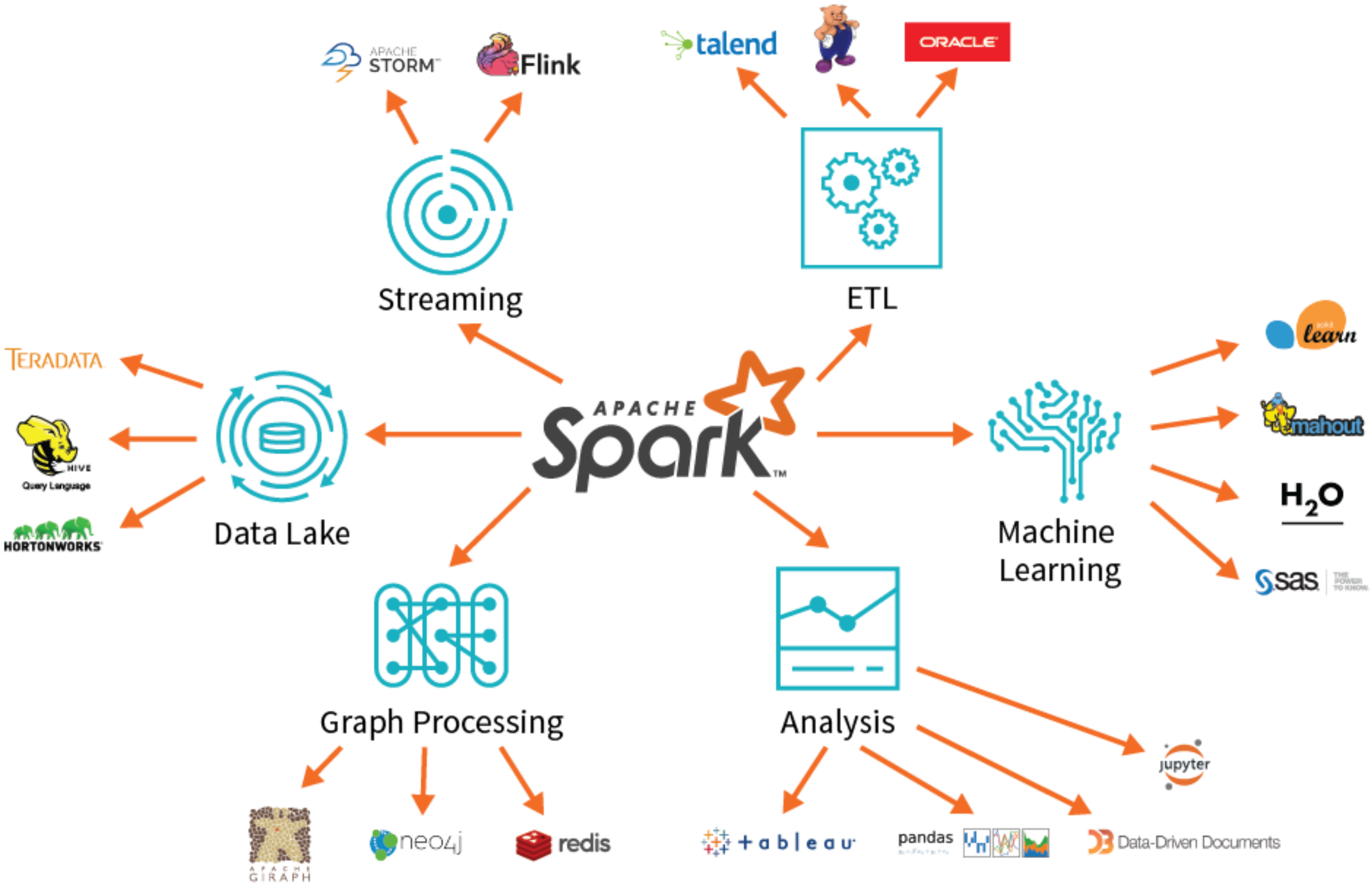
Apache Spark: An Intro

by Fatih Nayebi, Ph.D.

Master of Management Analytics, Desautels Faculty of Management, McGill


APACHE STORM
Flink
talend
ORACLE
Streaming
ETL
TERADATA
HIVE
Query Language
HORTONWORKS
Data Lake
APACHE Spark
Machine Learning
learn
mahout
H_2O
sas
Graph Processing
Analysis
jupyter
APACHE GIRAPH
neo4j
redis
tableau
pandas
Data-Driven Documents

Apache Spark

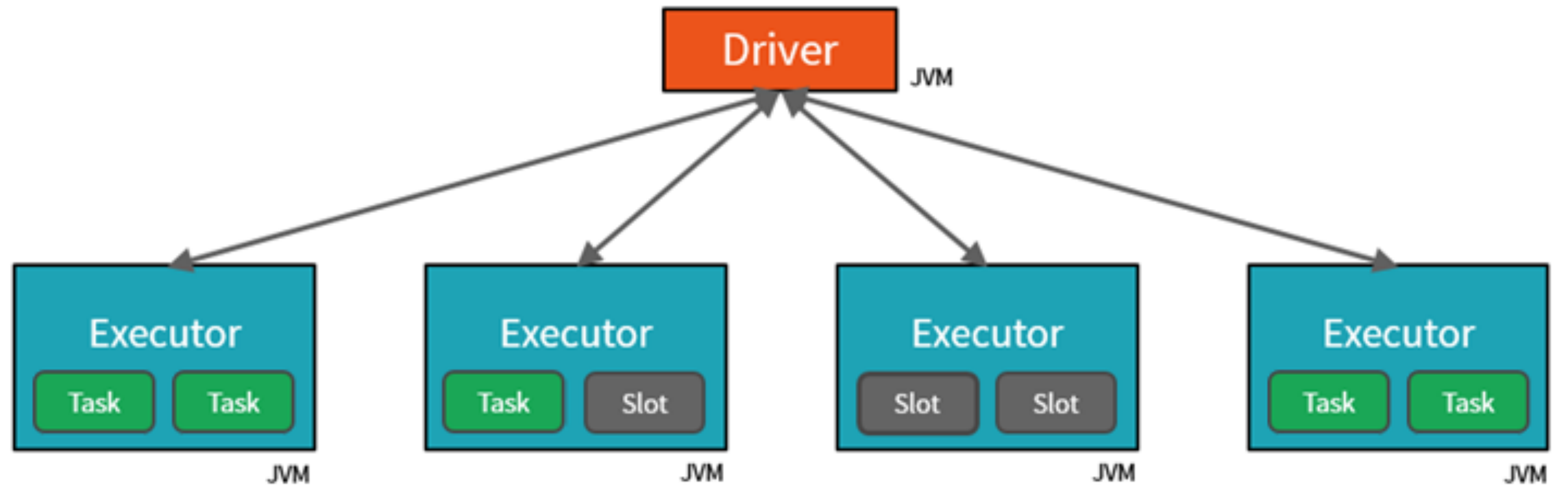
- Unified analytics engine for big data processing, with built-in modules for streaming, SQL, machine learning and graph processing.
- Research project at UC Berkley in 2009
- APIs: Scala, Python, Java, SQL, and R
- Micro batching

When to use Spark?

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

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 collections of elements in parallel

Transformation	Actions
Filter	Count
Sample	Take
Union	Collect

DataFrame

- Data with columns (built on RDDs)
- Improved performance via optimizations

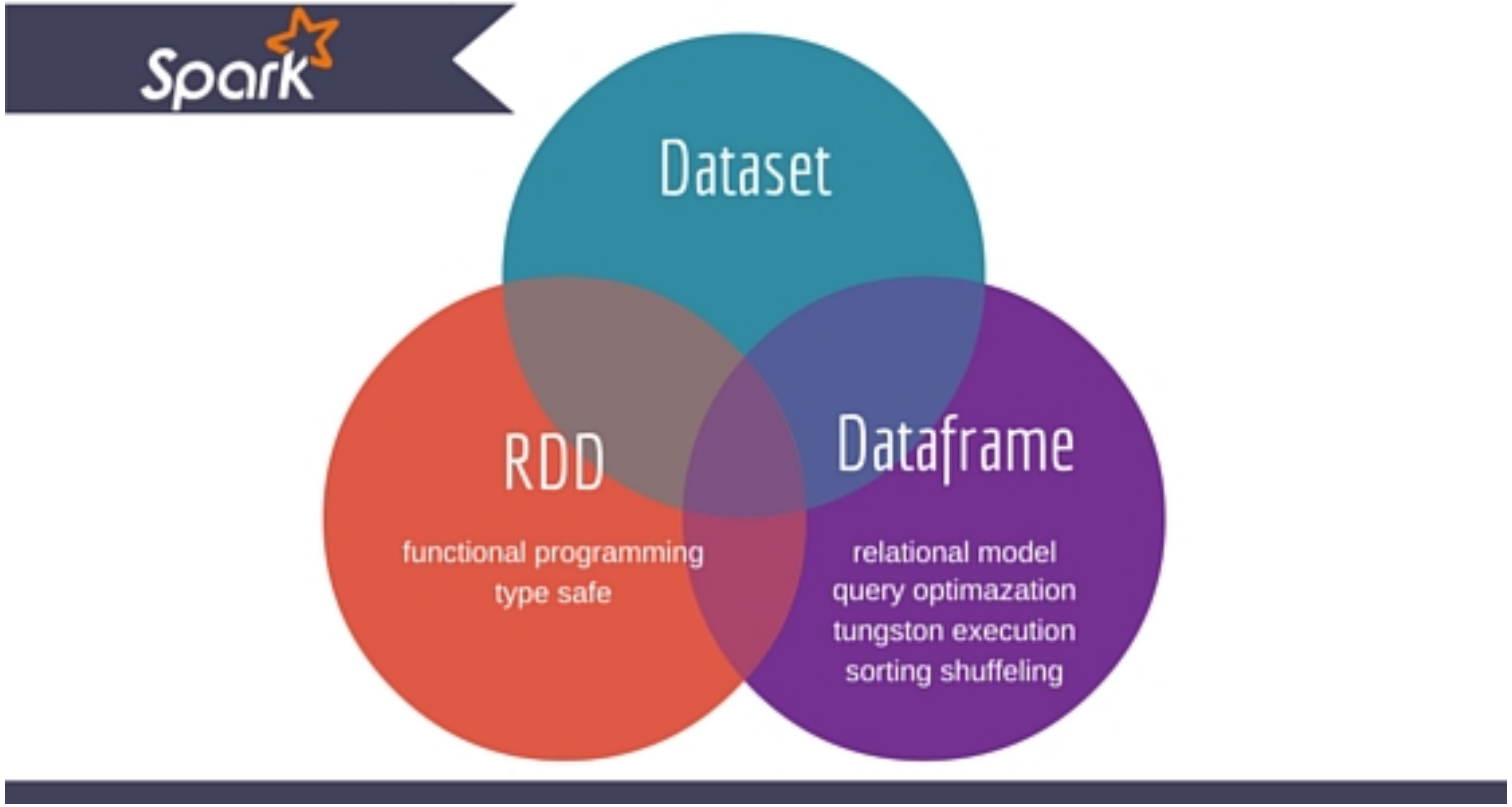
User-friendly API

```
dataRDD = sc.parallelize([("Jim", 20), ("Anne", 31), ("Jim", 30)])
(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)))
dataDF = dataRDD.toDF(["name", "age"])
dataDF.groupBy("name").agg(avg("age"))
```

Benefits

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

Dataset


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

Initializing SparkSession

- A SparkSession can be used to
  - create DataFrame
  - register DataFrame as tables
  - execute SQL over tables
  - cache tables
  - read parquet files.

Creating DataFrames from Spark Data Sources

- JSON
- Parquet
- TXT Files

JSON

```
df1 = spark.read.json("customer.json")
df1.show()
df2 = spark.read.load("people.json", format="json”)
```

Parquet Files

```
df3 = spark.read.load("users.parquet")
```

TXT Files

```
df4 = spark.read.text("people.txt")
```

Inspect Data

```
df.dtypes             # return df column names and data types
df.show()             # display the content of df
df.head()             # return first n rows
df.first()            # return first row
df.take(2)            # return the first 2 rows
df.schema             # return the schema of df
df.describe().show()  # compute summary statistics
df.columns            # return the columns of df
df.count()            # count the number of rows in df
df.distinct().count() # count the number distinct rows in df
df.printSchema()      # print the schema of df
df.explain()          # print the (logical & phsical) plans
```

Reference

- Apache Spark Key Terms, Explained
- Spark Overview by Brooke Wenig