



Get Started with Databricks





What is Spark?



What is Spark ?

Open-source Cluster computing framework

- Massive Parallel Processing with linear scale

Built for

- Speed/Scalability
- Ease-of-Use
- Extensibility

Support for multiple languages

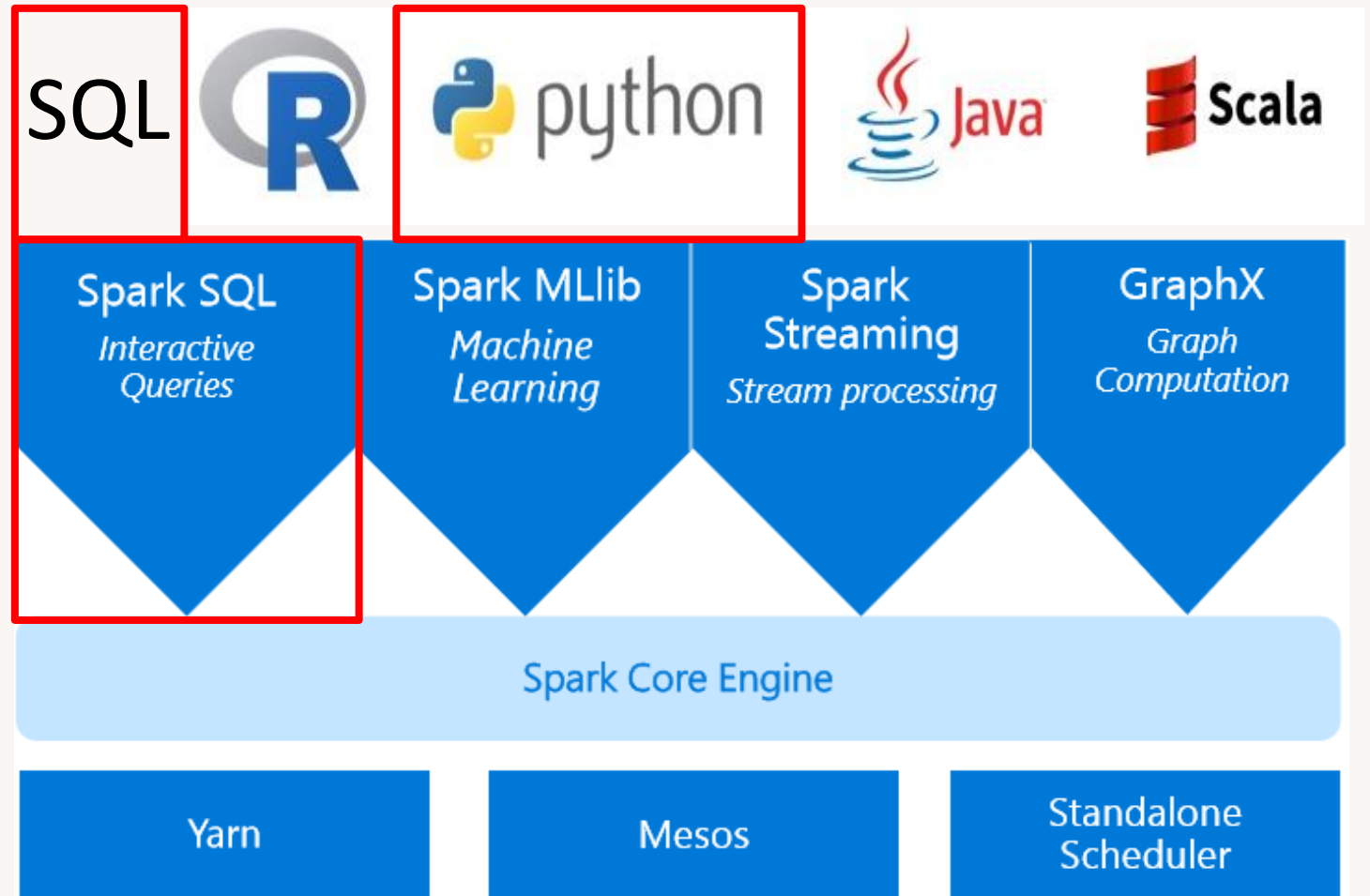
- Java
- Scala
- Python
- R
- SQL



What is Spark ?

Spark unifies

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



What is Spark ?

Used across all major Data & AI platforms and vendors:

- Databricks
- Azure Synapse
- Microsoft Fabric
- SAP HANA
- Amazon EMR
- Cloudera (Hortonworks)
- ...



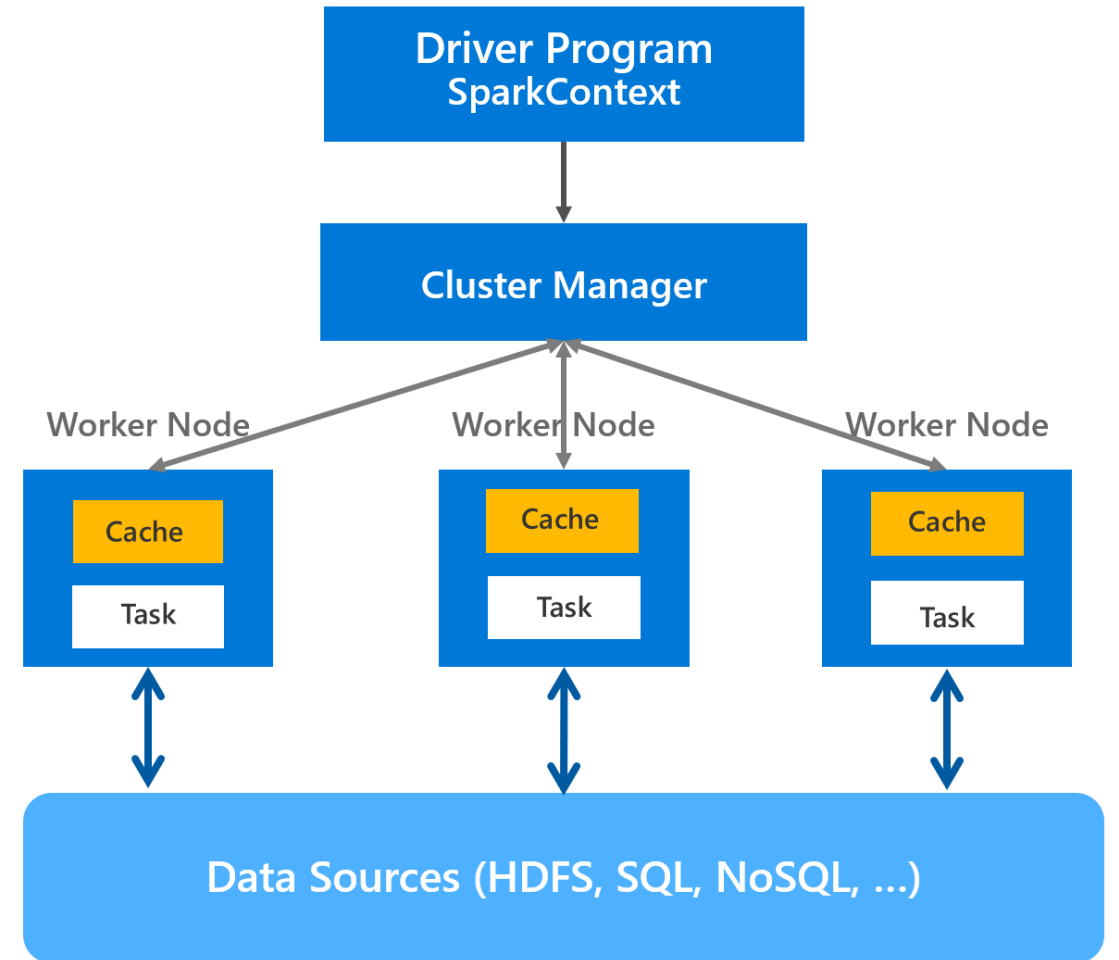
How does it work?

'Driver' runs the 'main' function and executes and coordinates the various parallel operations on the worker nodes.

The worker nodes read and write data from/to Data Sources including HDFS.

Worker node also caches transformed data in memory as RDDs (Resilient Distributed Data Sets)
The results of the operations are collected by the driver and returned to the client.

Worker nodes and the Driver Node execute as VMs in public clouds (AWS, Azure, GCP)





What is Databricks?



What is Databricks ?

Company that provides a Big Data processing service in the Cloud using Apache Spark

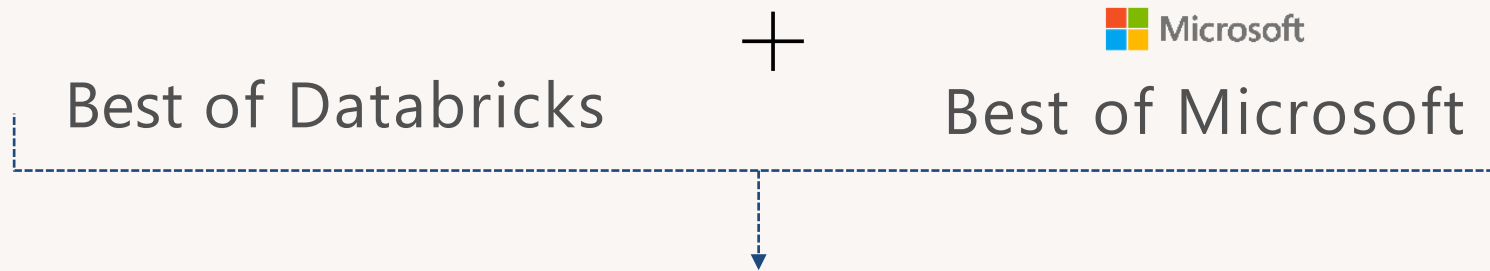
- Databricks on AWS
- Azure Databricks
- Databricks on Google Cloud
- **No on-prem solution!**


From the original Creators of Apache® Spark™



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.



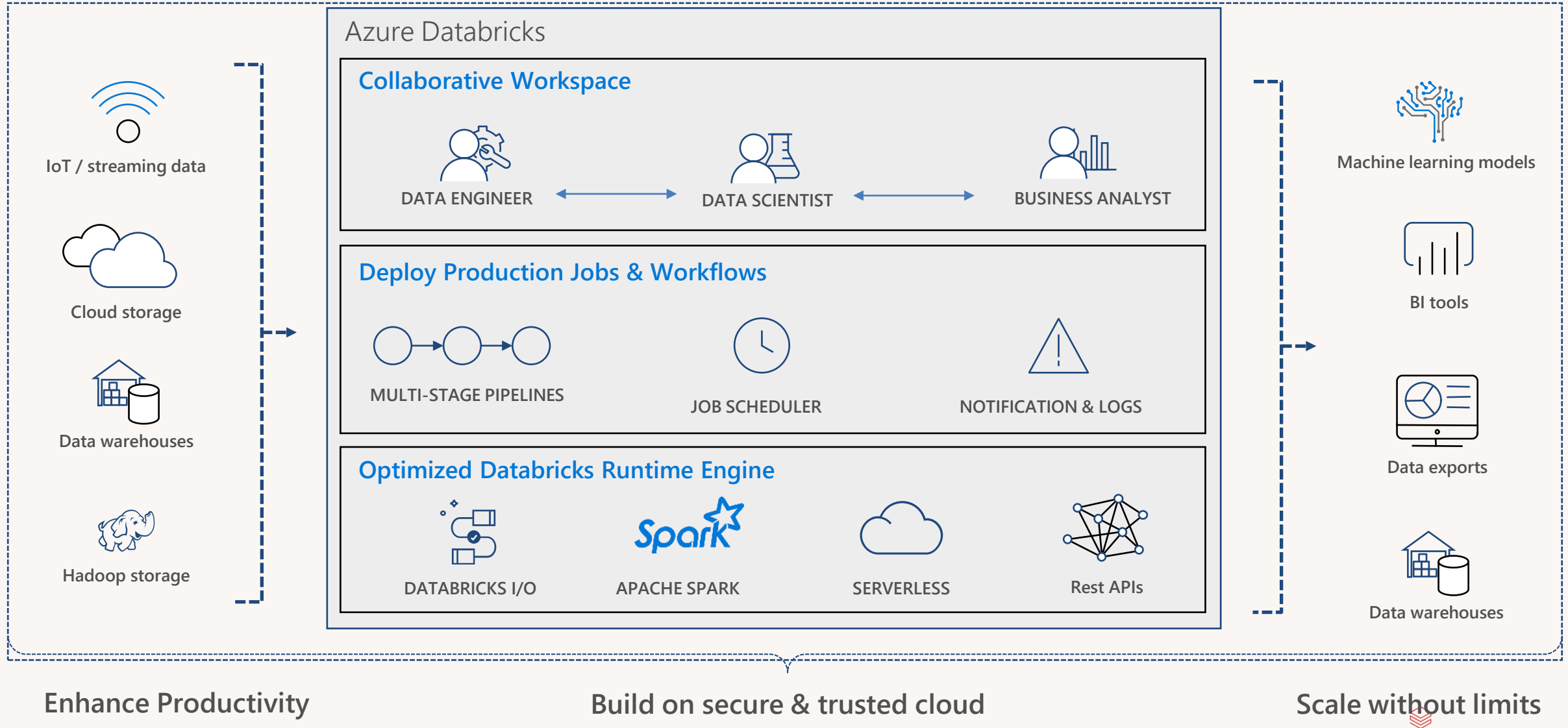
Native integration with Azure services (Power BI, SQL DW, Cosmos DB, Blob Storage)



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



Azure Databricks





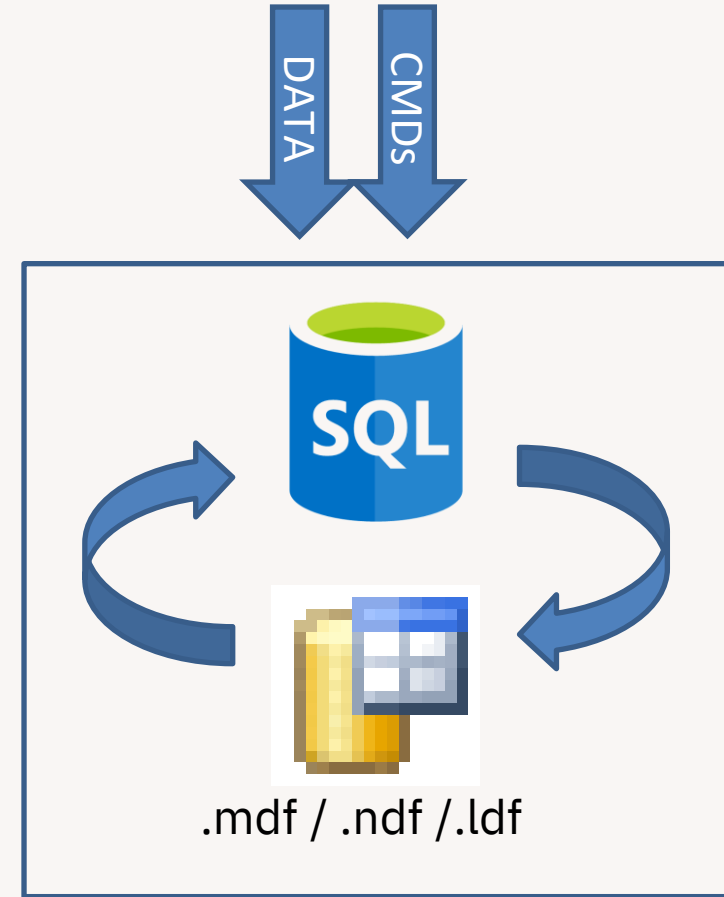
Spark vs. RDBMS



Classic RDBMS

Classic RDBMS

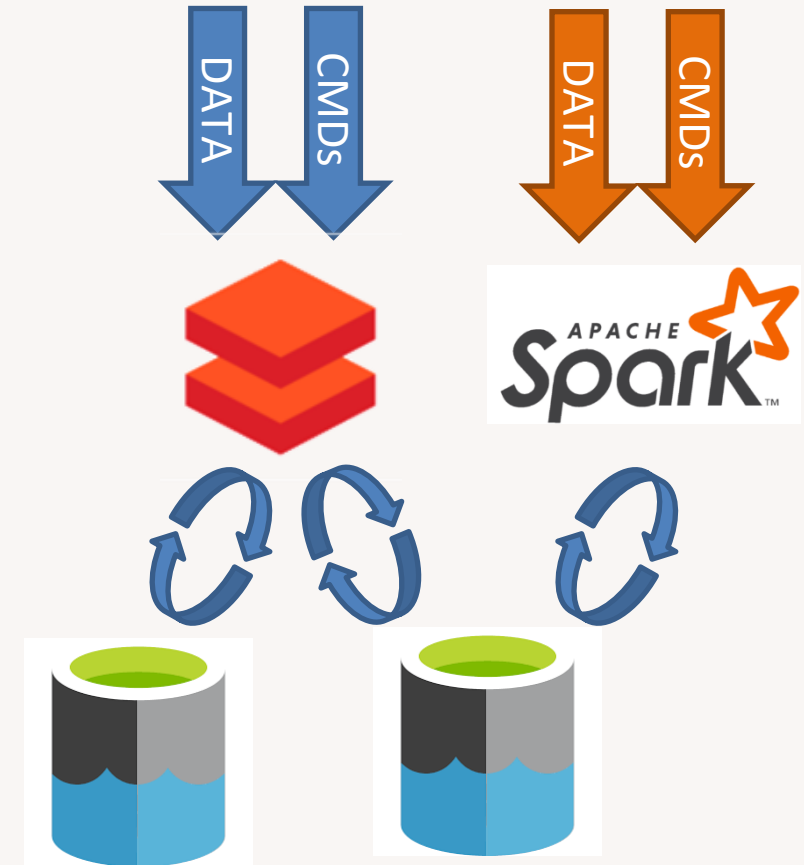
- Single point of access to your data
- Limited resources / no scale-out
- All process use the same resources
 - ETL vs. user queries
- Storage is managed internally



Databricks / Spark + Data Lake

Big Data processing with Spark

- Separation of storage and compute (!)
- Only spin up compute when necessary
- Can use multiple compute engines
- Cheap storage
- Can attach any/multiple storage(s)



Spark Overview

Spark SQL is a module for structured data processing
with multiple interfaces

SQL

DataFrame API

Python, Scala, Java, R



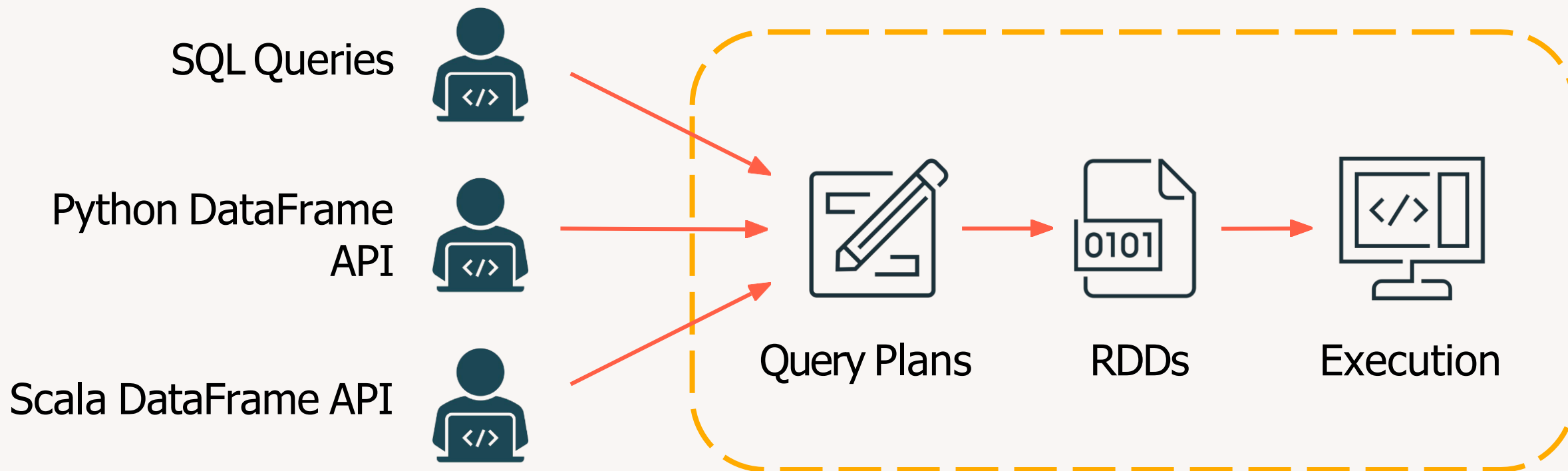
The same Spark SQL query can be expressed with **SQL** and the **DataFrame API**

```
SELECT id, result  
FROM exams  
WHERE result > 70  
ORDER BY result
```

```
spark.table("exams")  
  .select("id", "result")  
  .where("result > 70")  
  .orderBy("result")
```



Spark SQL executes all queries on the same engine



What is a RDD?

Resilient Distributed Dataset

RDD was the primary user-facing API in Spark since its inception. At the core, an RDD is an immutable distributed collection of elements of your data, partitioned across nodes in your cluster that can be operated in parallel with a low-level API that offers *transformations* and *actions*.



What is a DataFrame?

Like an RDD, a [DataFrame](#) is an immutable distributed collection of data. Unlike an RDD, data is organized into named columns with defined types, like a table in a relational database. Designed to make large data sets processing even easier, DataFrame allows developers to impose a structure onto a distributed collection of data, allowing higher-level abstraction; it provides a domain specific language API to manipulate your distributed data; and makes Spark accessible to a wider audience, beyond specialized data engineers.

