THE SPARK SHOWDOWN

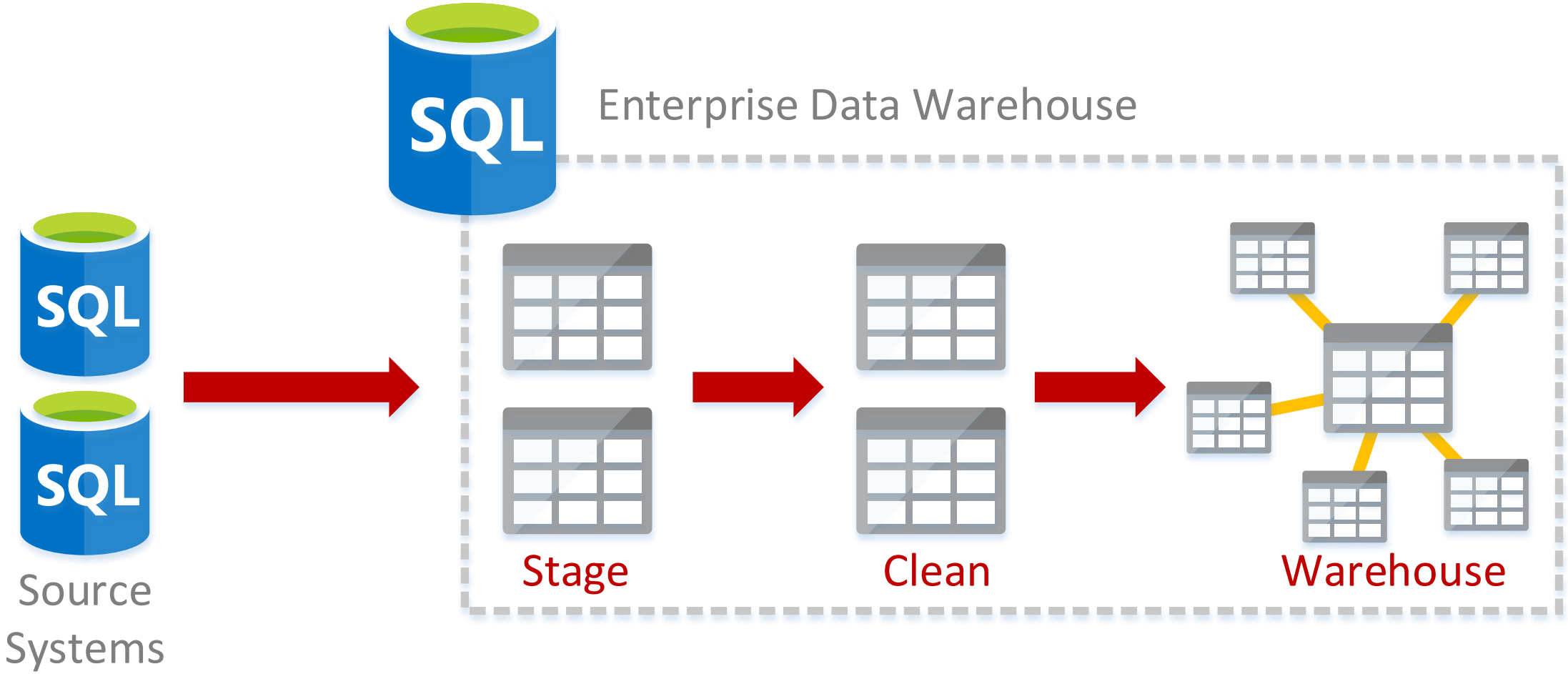
Simon Whiteley
@MrSiWhiteley

databricks

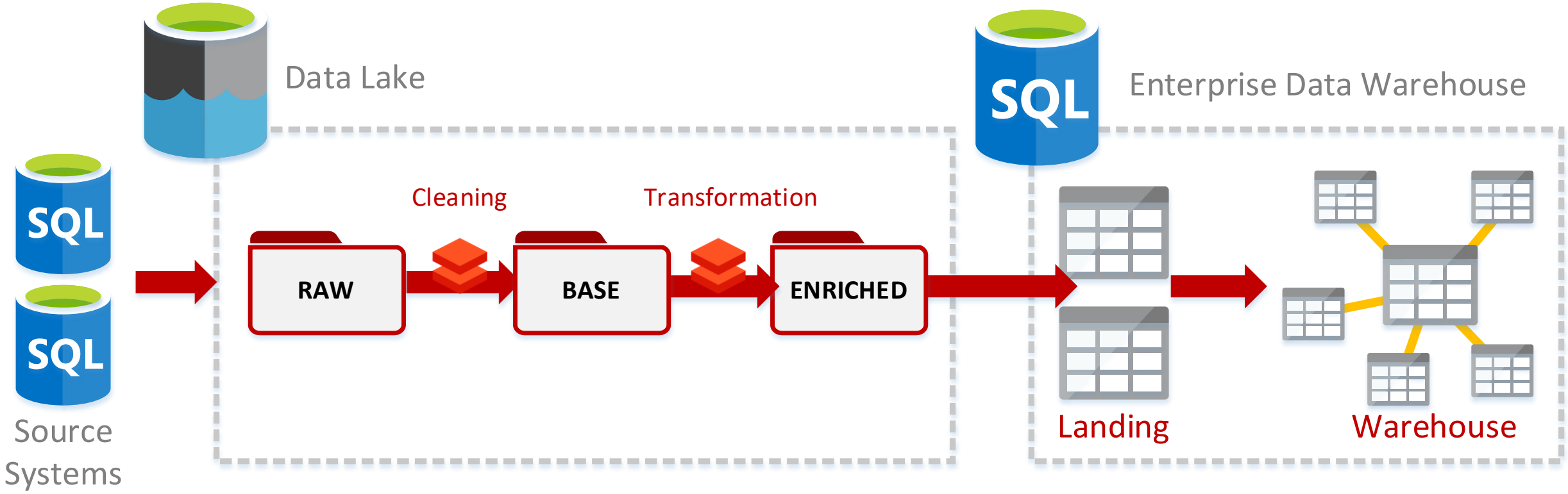
Synapse Analytics

HERE'S HOW IT WORKS

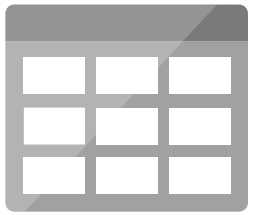
THE RETIRED CHAMPION



THE NEW GENERATION

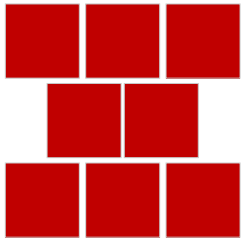


THE SPARK API ABSTRACTIONS



DataFrame API

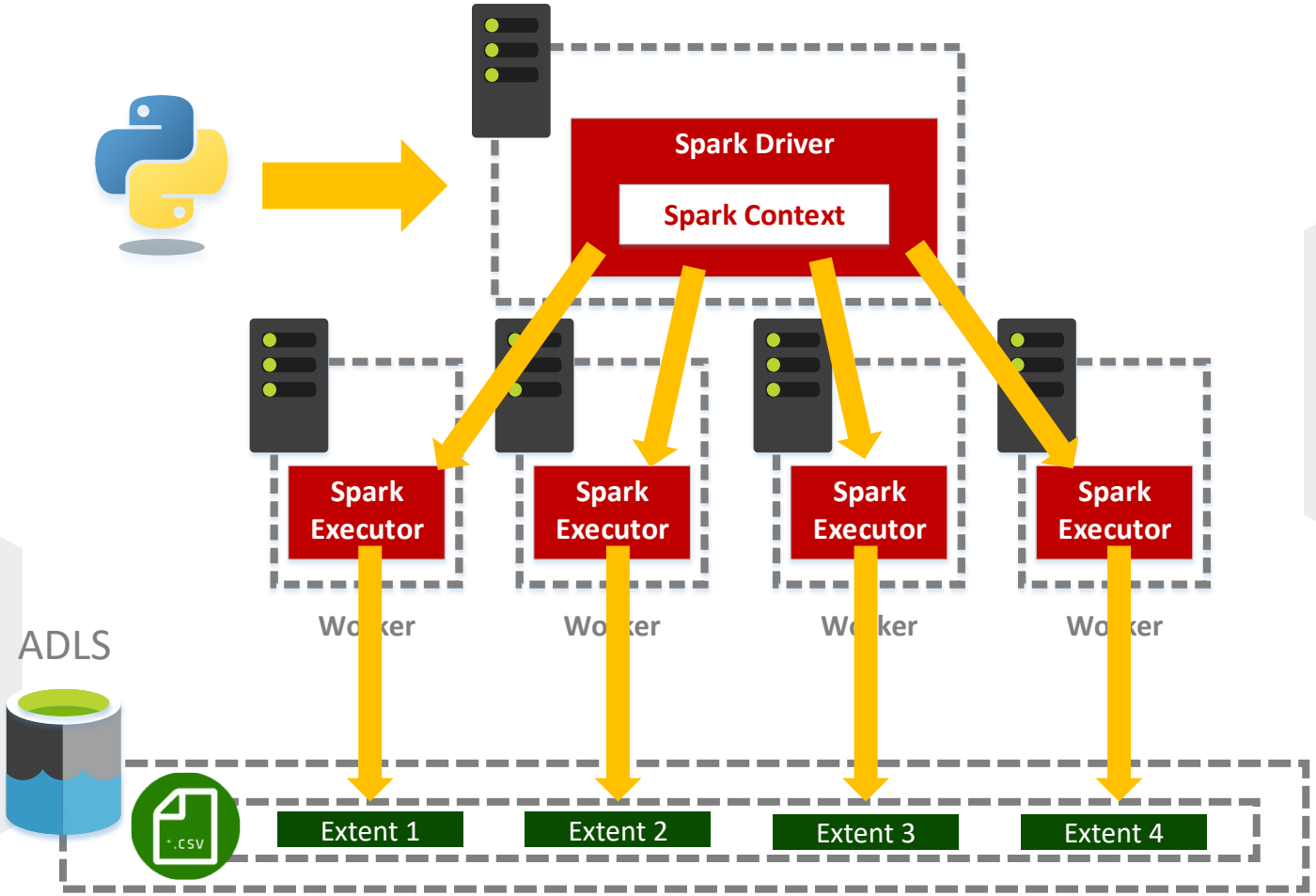
SQL API



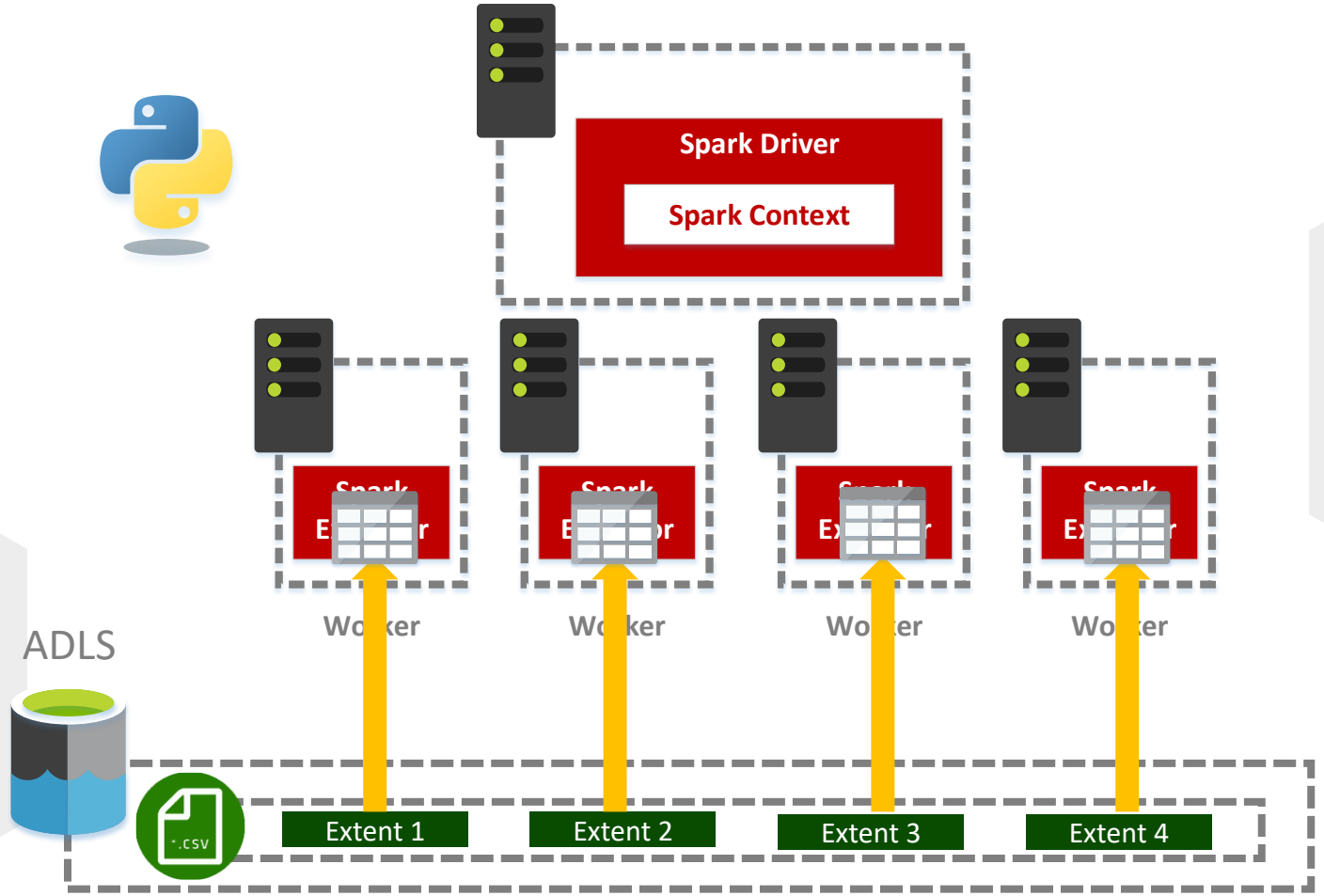
Resilient Distributed Datasets – In-Memory Data Blocks

Core Spark Engine – 80% Scala Code Libraries

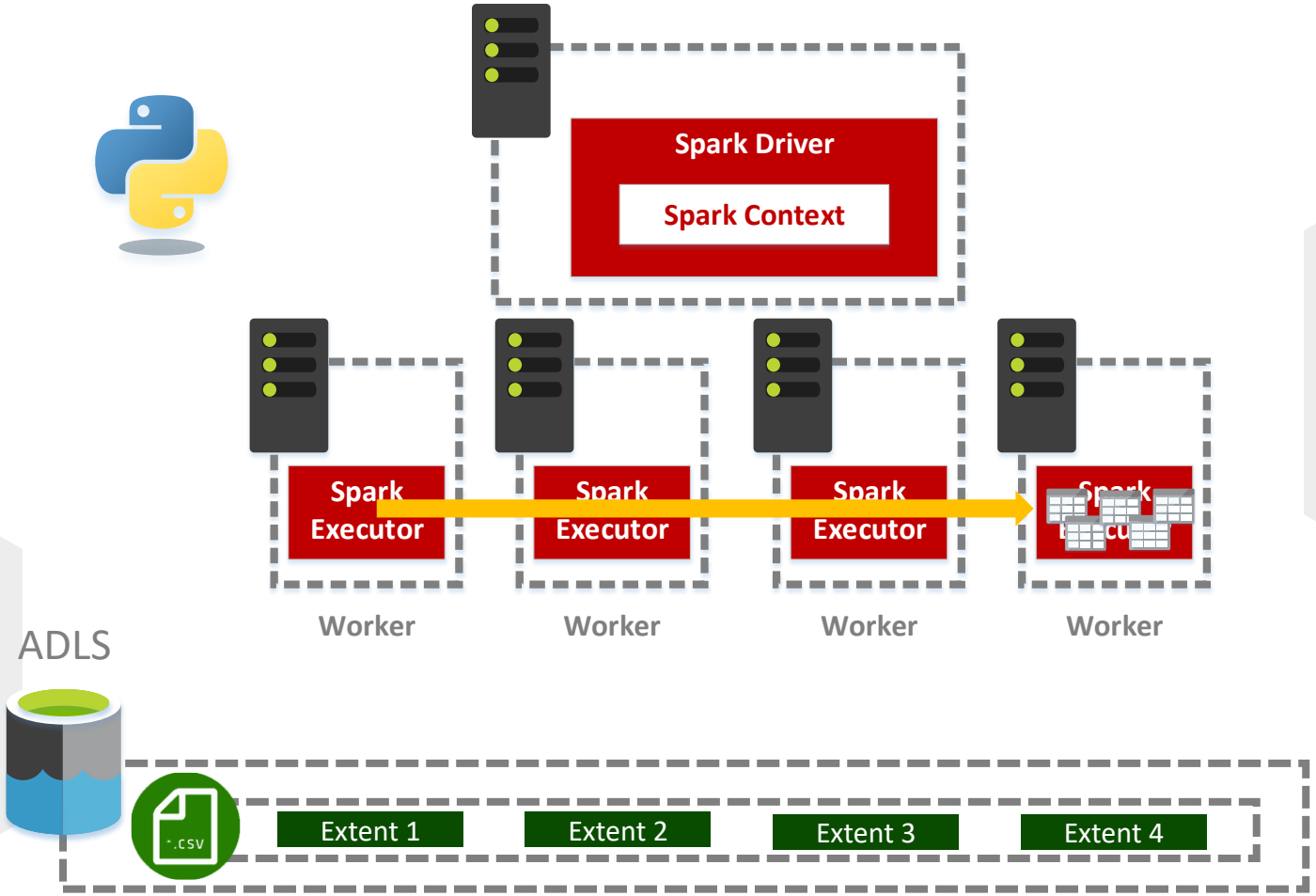
DISTRIBUTED COMPUTE



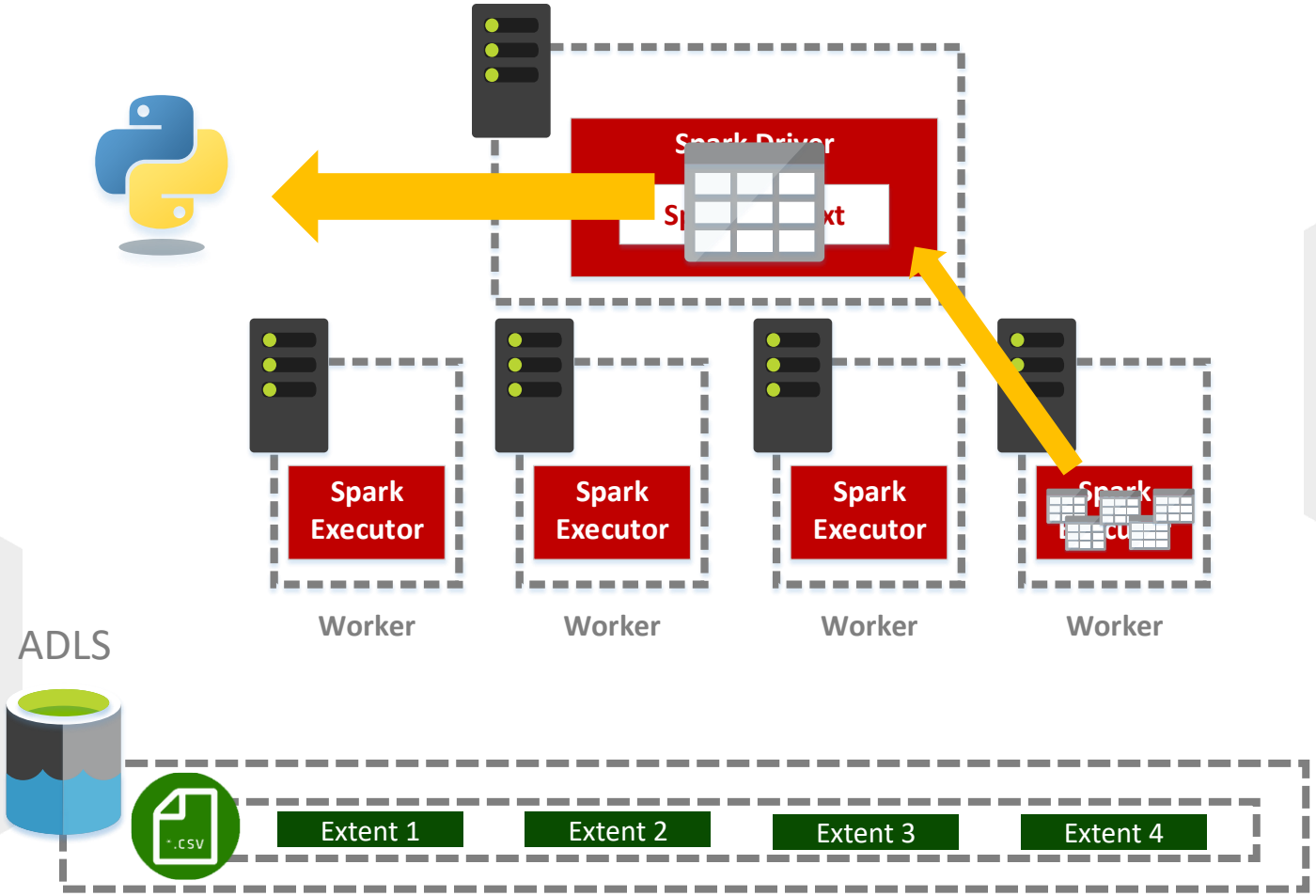
DISTRIBUTED COMPUTE



DISTRIBUTED COMPUTE



DISTRIBUTED COMPUTE



TODAY'S COMPETITORS

VS

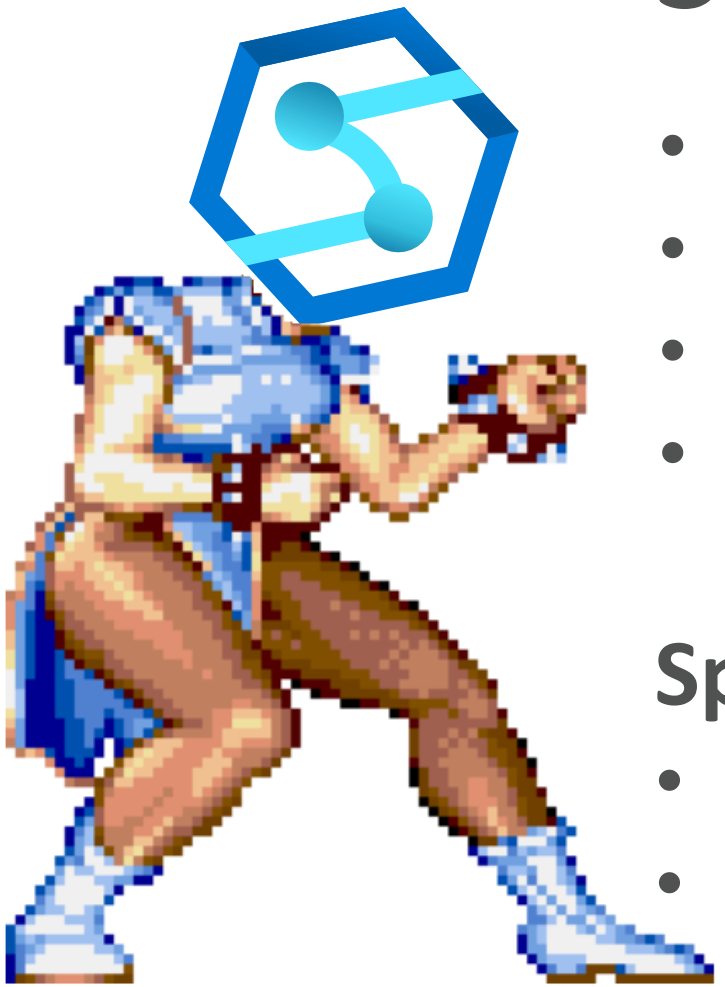
Databricks

- Released 2016 (AWS)
- AWS/Azure Cross Platform
- Databricks Proprietary Runtime
- Built by the inventors of Spark

Special Skills:

- Workspace Features
- Delta Engine

Synapse Analytics

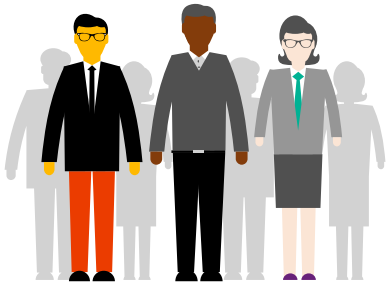


- Still in Preview
- Azure Only
- Vanilla Spark Runtime
- Fresh look at how Spark can work

Special Skills:

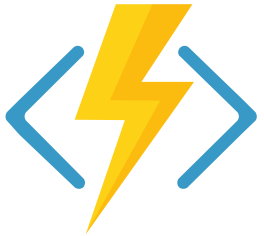
- Integrations
- Spark.NET

AZURE DATABRICKS WORKSPACE

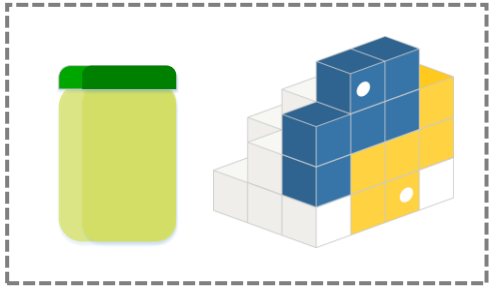


User Management

Notebooks



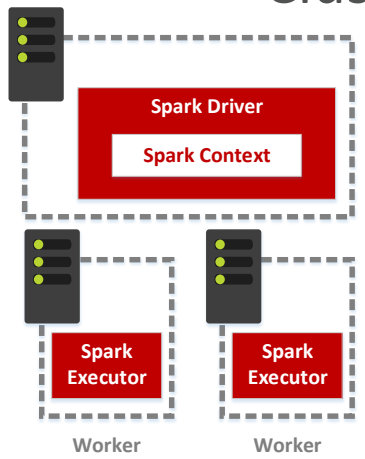
Jobs

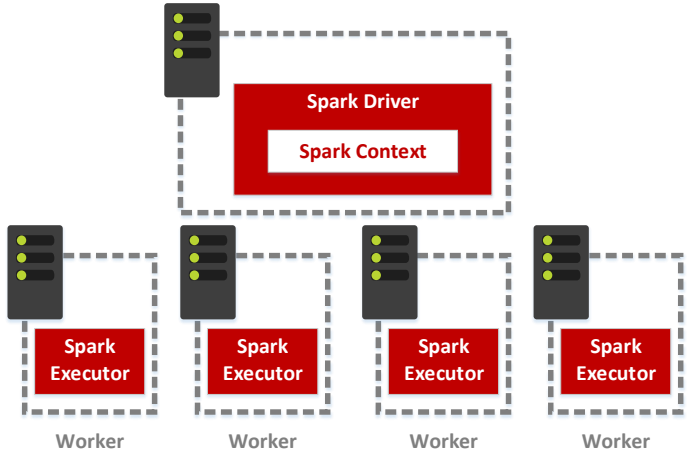


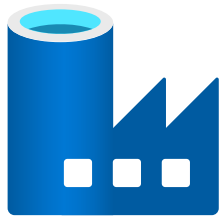
Library

DBFS

Cluster Management



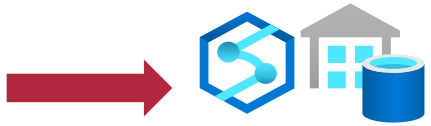




Azure Data Factory V2

Generally Available

SQL Data Warehouse (SQLDW)



Azure Synapse Analytics (SQLDW)

Azure Synapse Analytics (Workspaces)

Pipelines

NEW

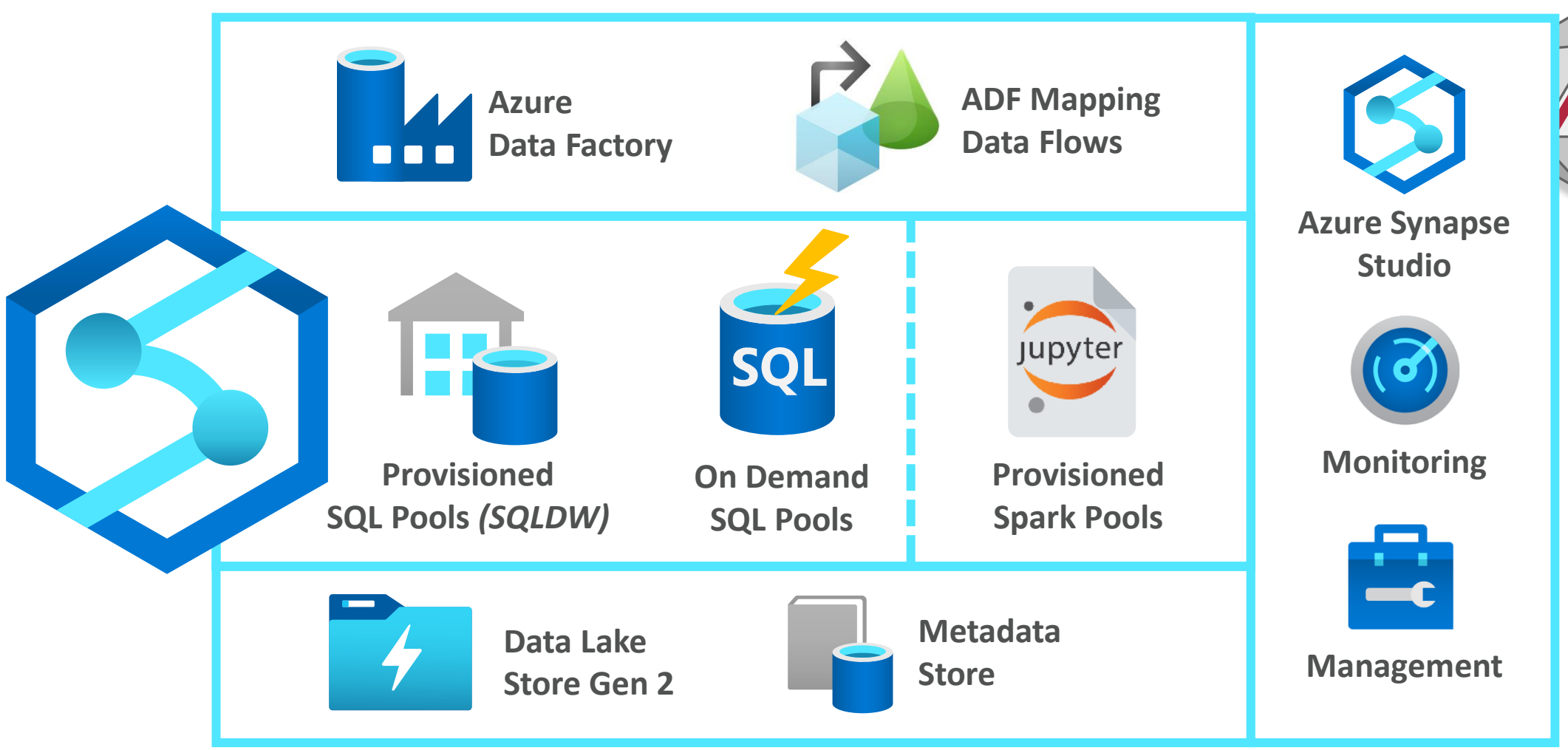
Provisioned Spark Pools

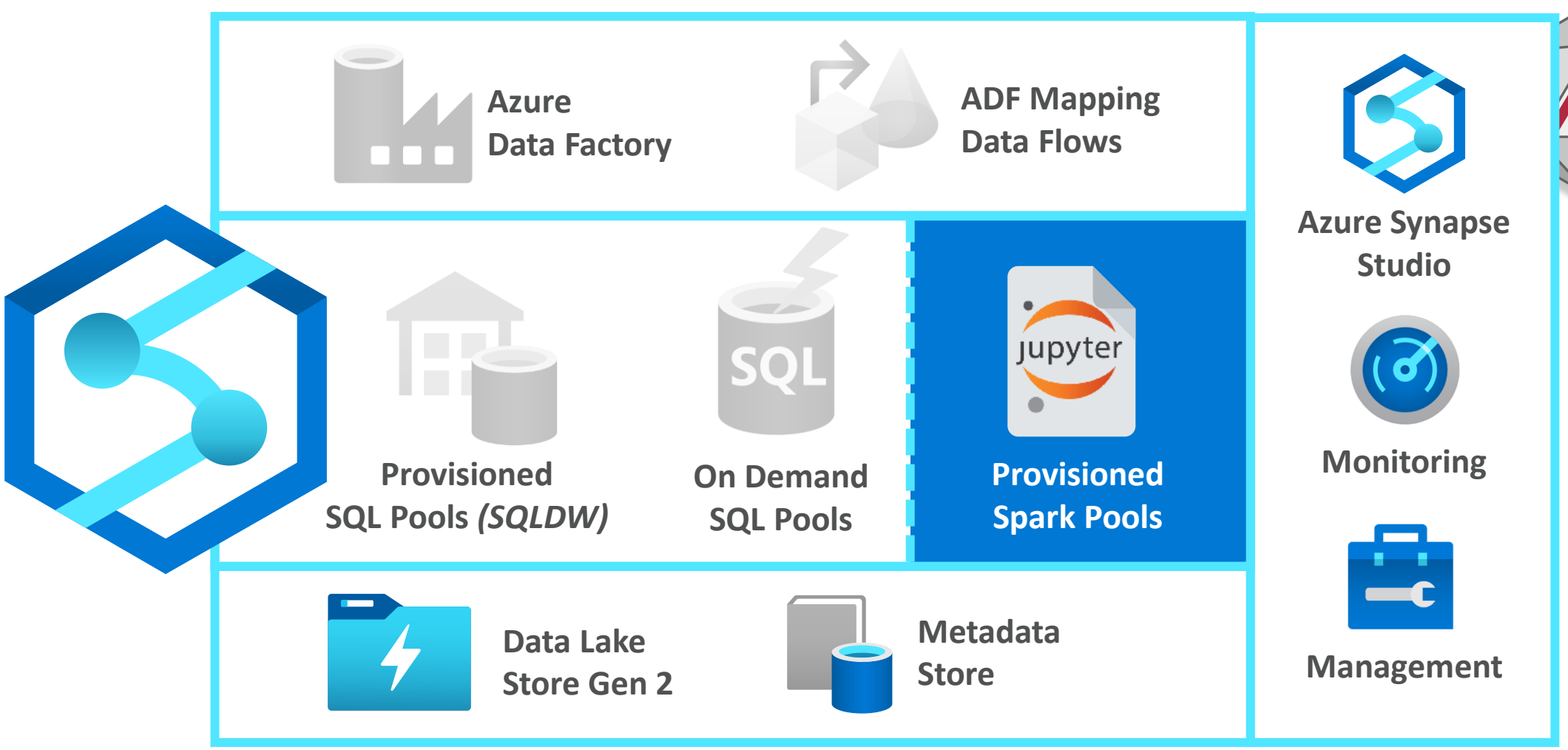
NEW

On Demand SQL Pools

Provisioned SQL Pools (aka SQLDW)

Public Preview


Azure
Data Factory
ADF Mapping
Data Flows
Azure Synapse
Studio
Provisioned
SQL Pools (SQLDW)
On Demand
SQL Pools
Provisioned
Spark Pools
Monitoring
Management
Data Lake
Store Gen 2
Metadata
Store


Azure
Data Factory
ADF Mapping
Data Flows
Azure Synapse
Studio
Provisioned
SQL Pools (SQLDW)
On Demand
SQL Pools
Provisioned
Spark Pools
Monitoring
Data Lake
Store Gen 2
Metadata
Store
Management

ROUND ONE - POWER

Why do we care?

- Functions
- Optimisations
- Spark 3.0

Delta Engine & Photon

3.0 Optimisation

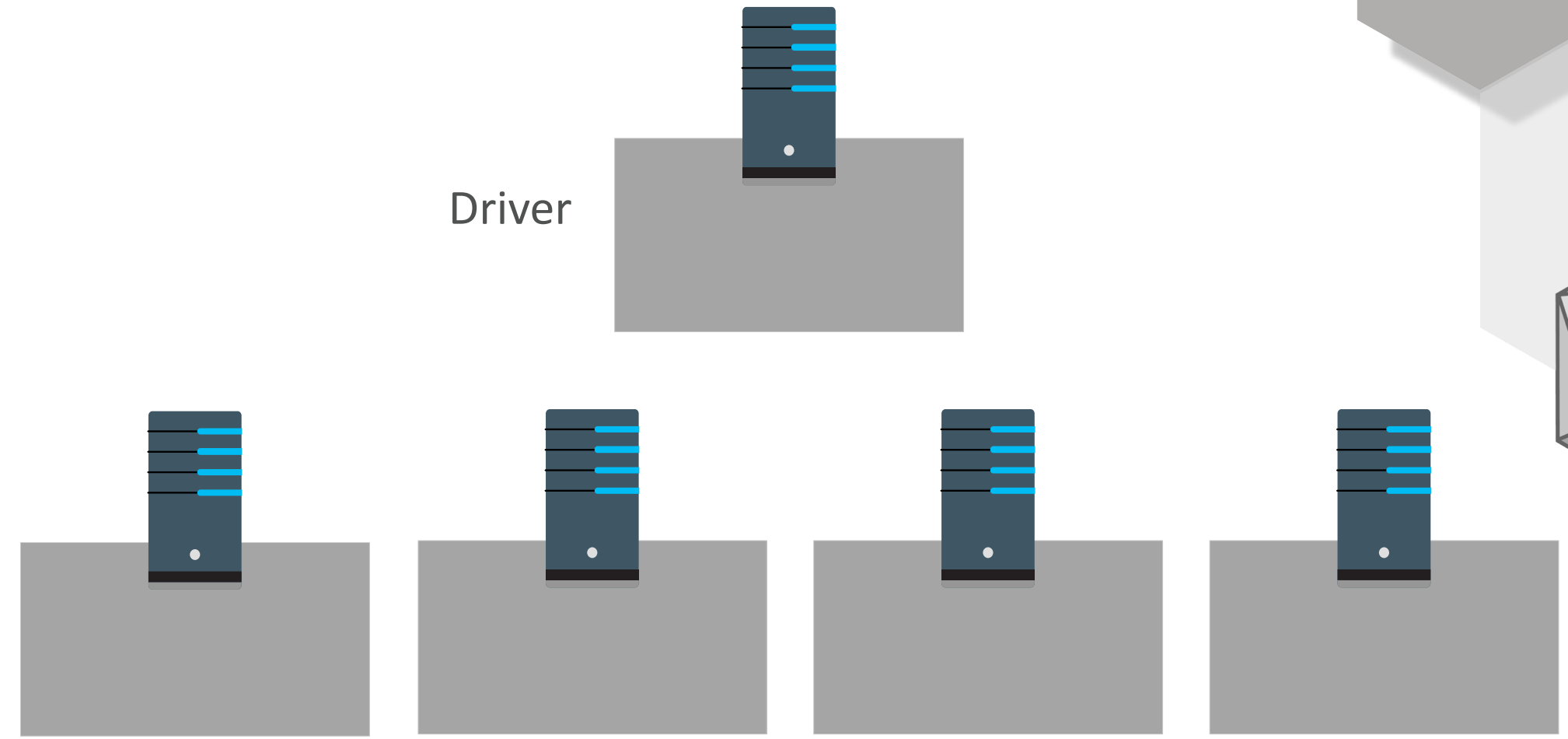
Spark 3.0

Databricks

Spark 3.0

Synapse

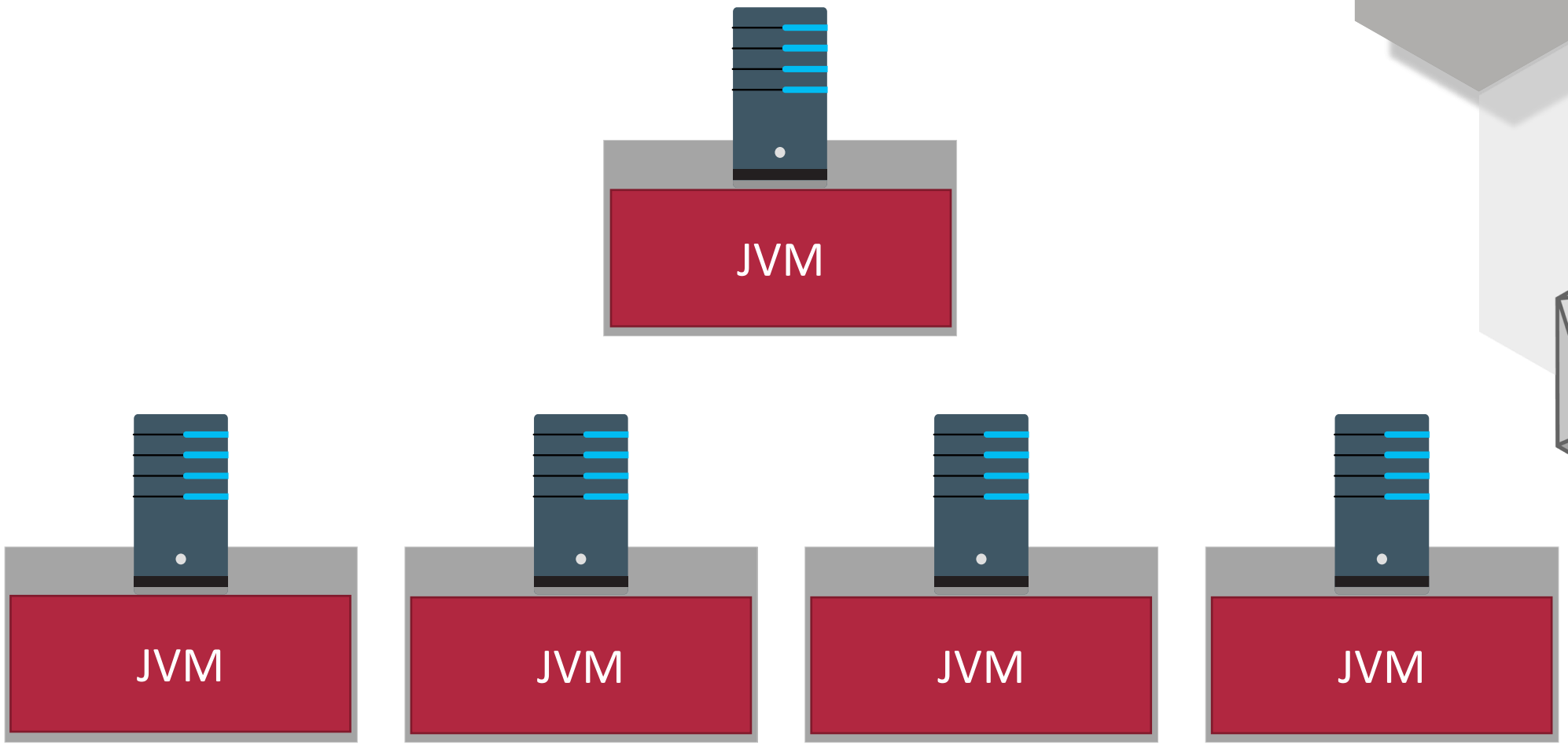
PHYSICAL ARCHITECTURE



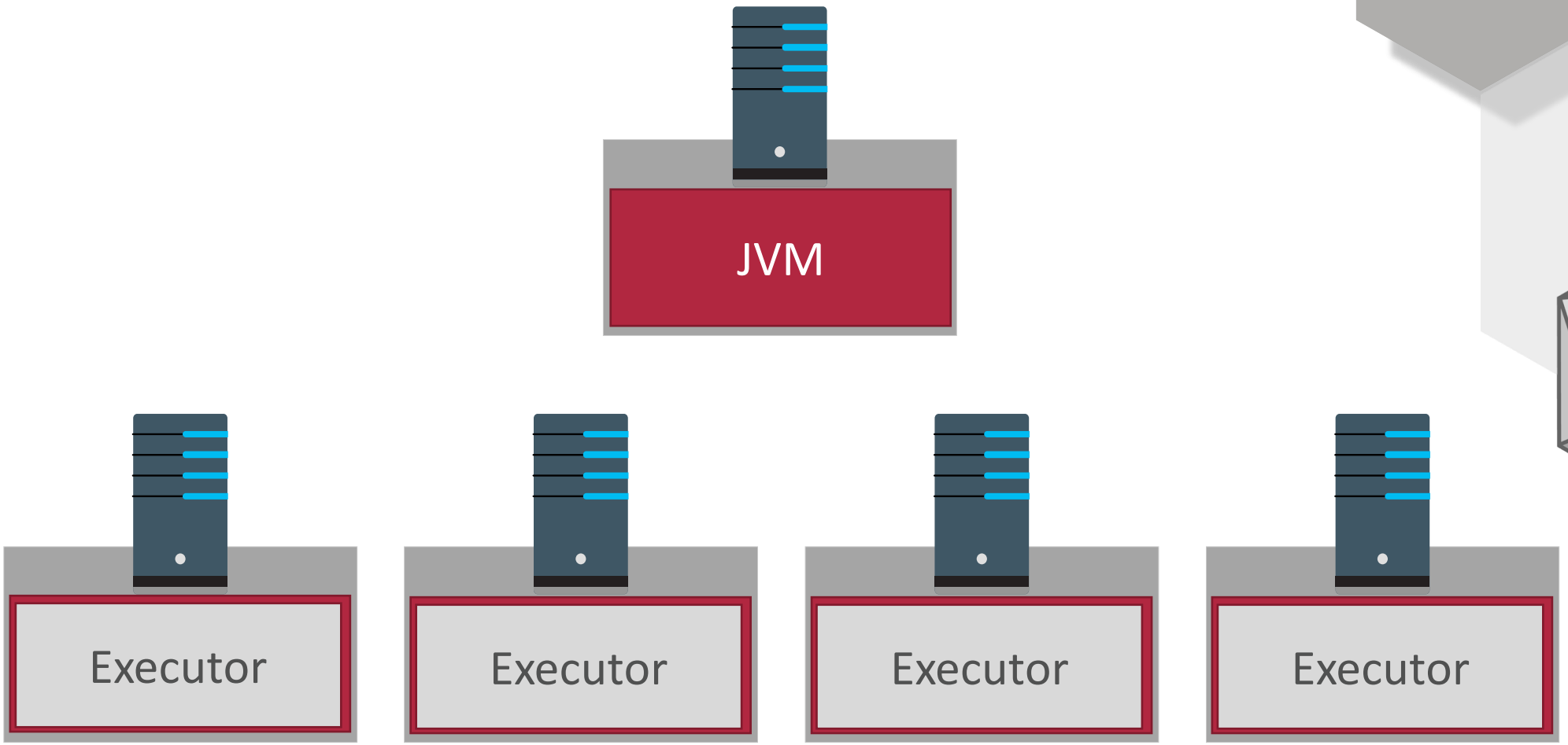
Workers

ADVANCING ANALYTICS

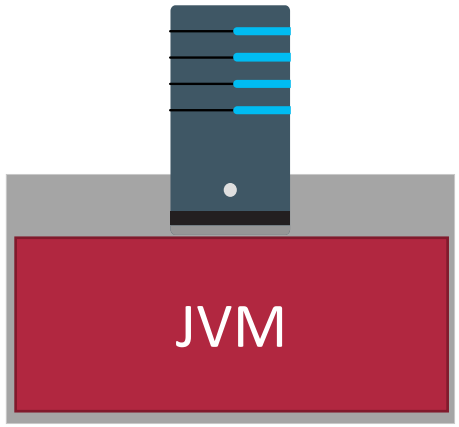
PHYSICAL ARCHITECTURE

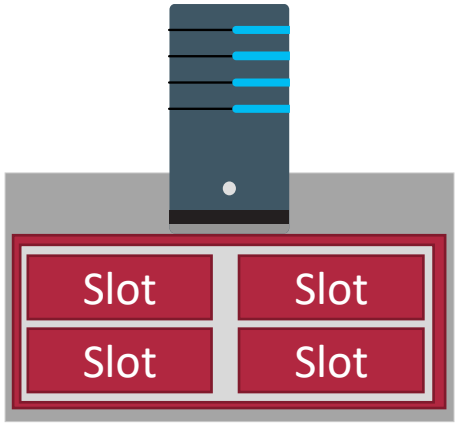


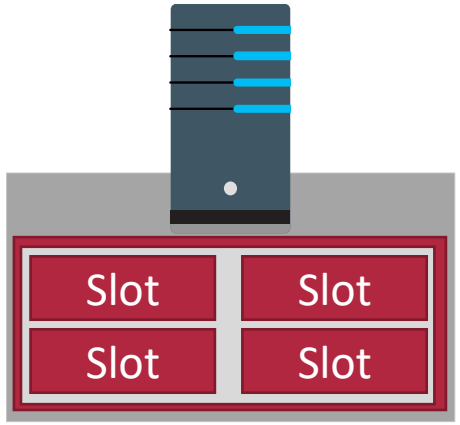
PHYSICAL ARCHITECTURE

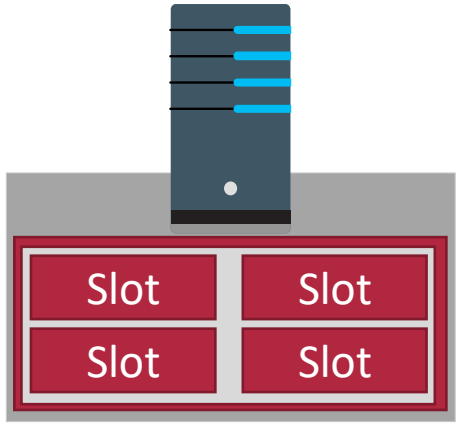


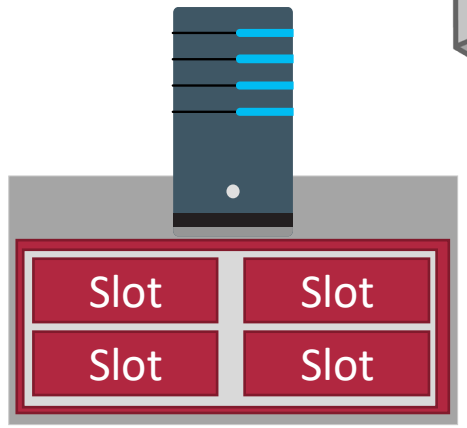
PHYSICAL ARCHITECTURE





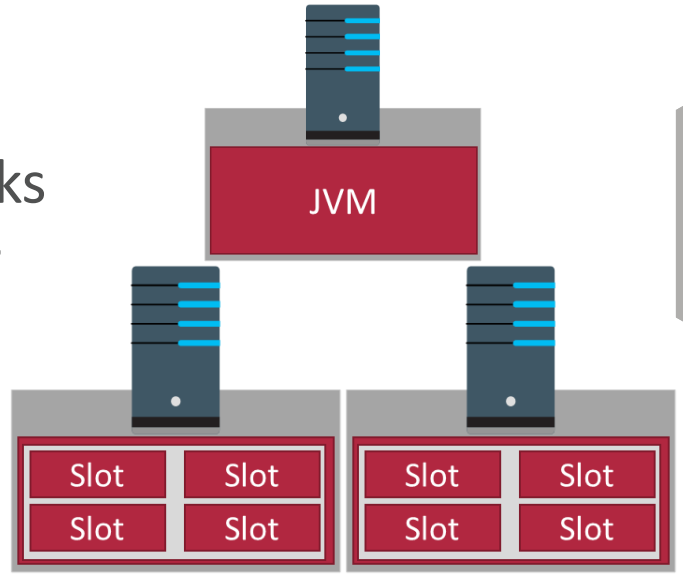


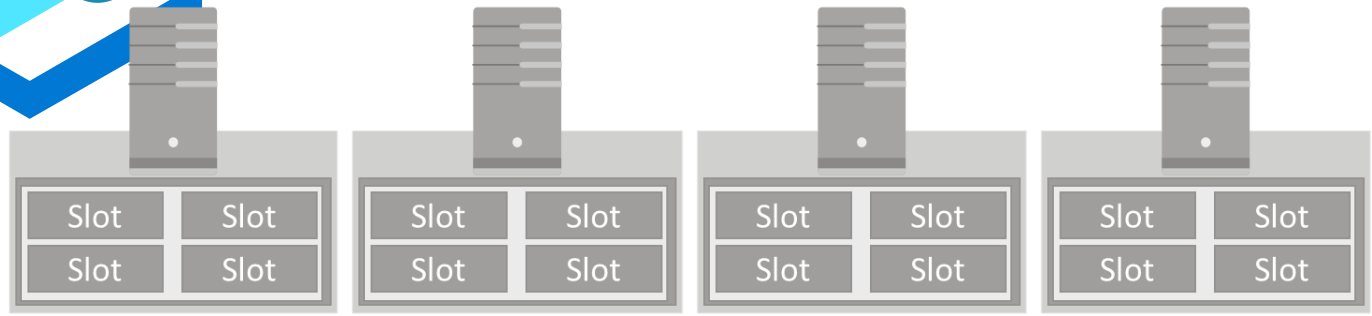




ADVANCING ANALYTICS

Databricks Cluster

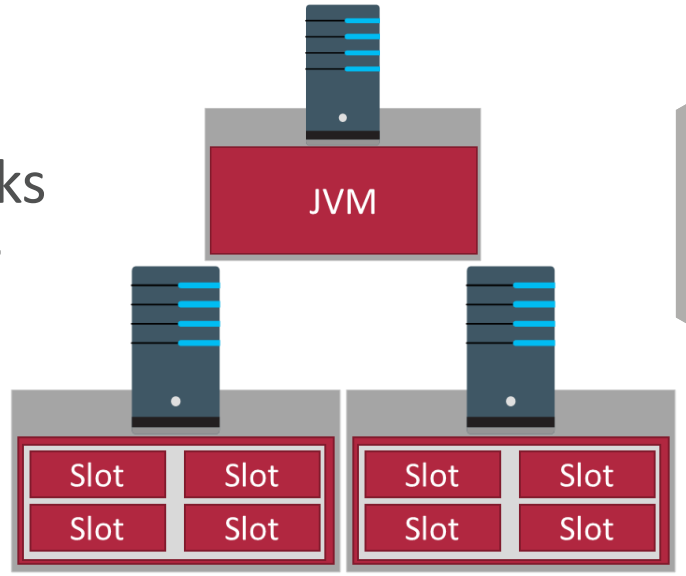




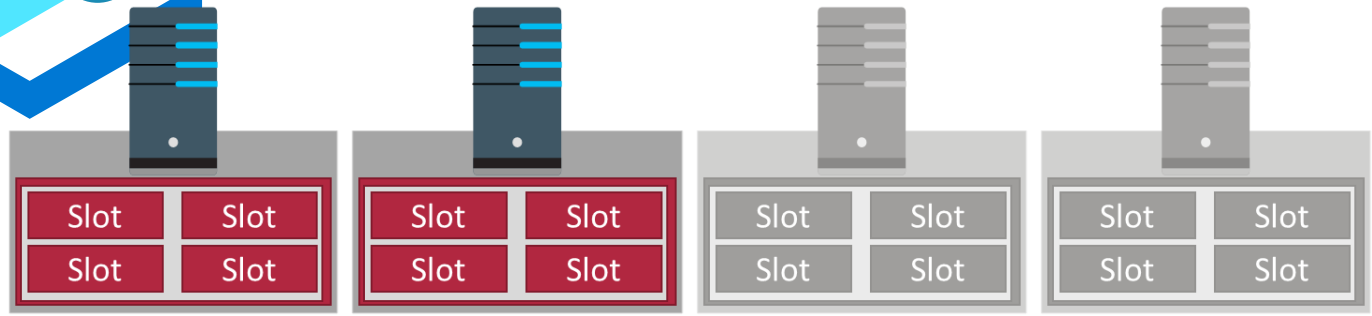
Spark Pool

ADVANCING ANALYTICS

Databricks Cluster



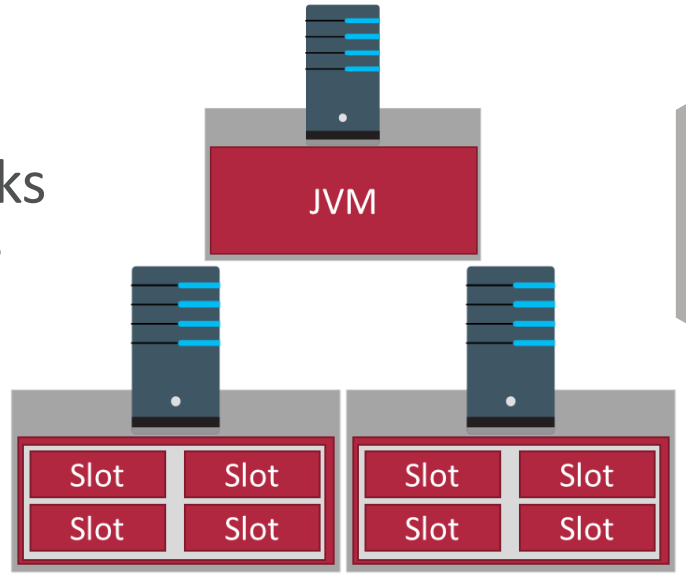
Session



Spark Pool

ADVANCING ANALYTICS

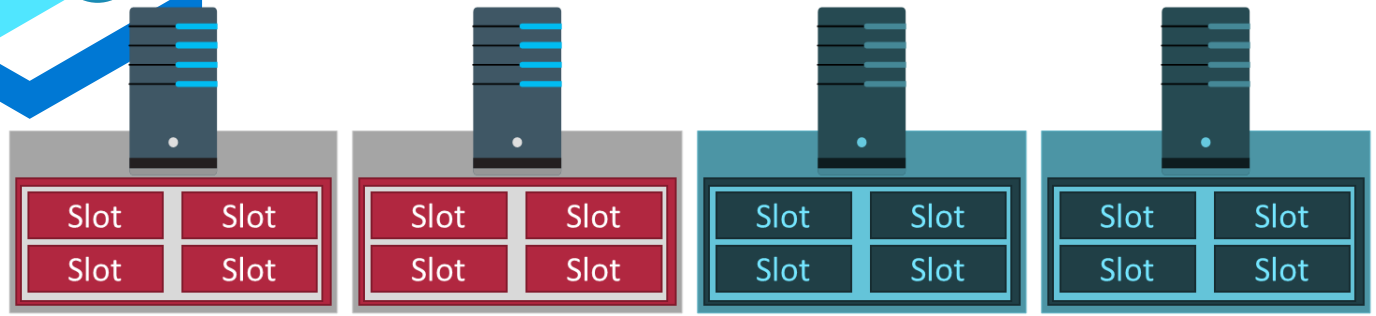
Databricks Cluster



Session 1

Session 2

Session 3



Spark Pool

ADVANCING ANALYTICS

ROUND TWO - SPECIAL MOVES

DATABRICKS NOTEBOOKS

DBUtils *(Databricks Utilities)*

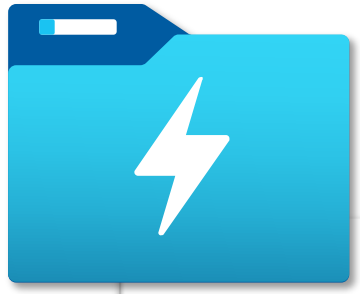
- Libraries
- File Management
- Secrets
- Widgets

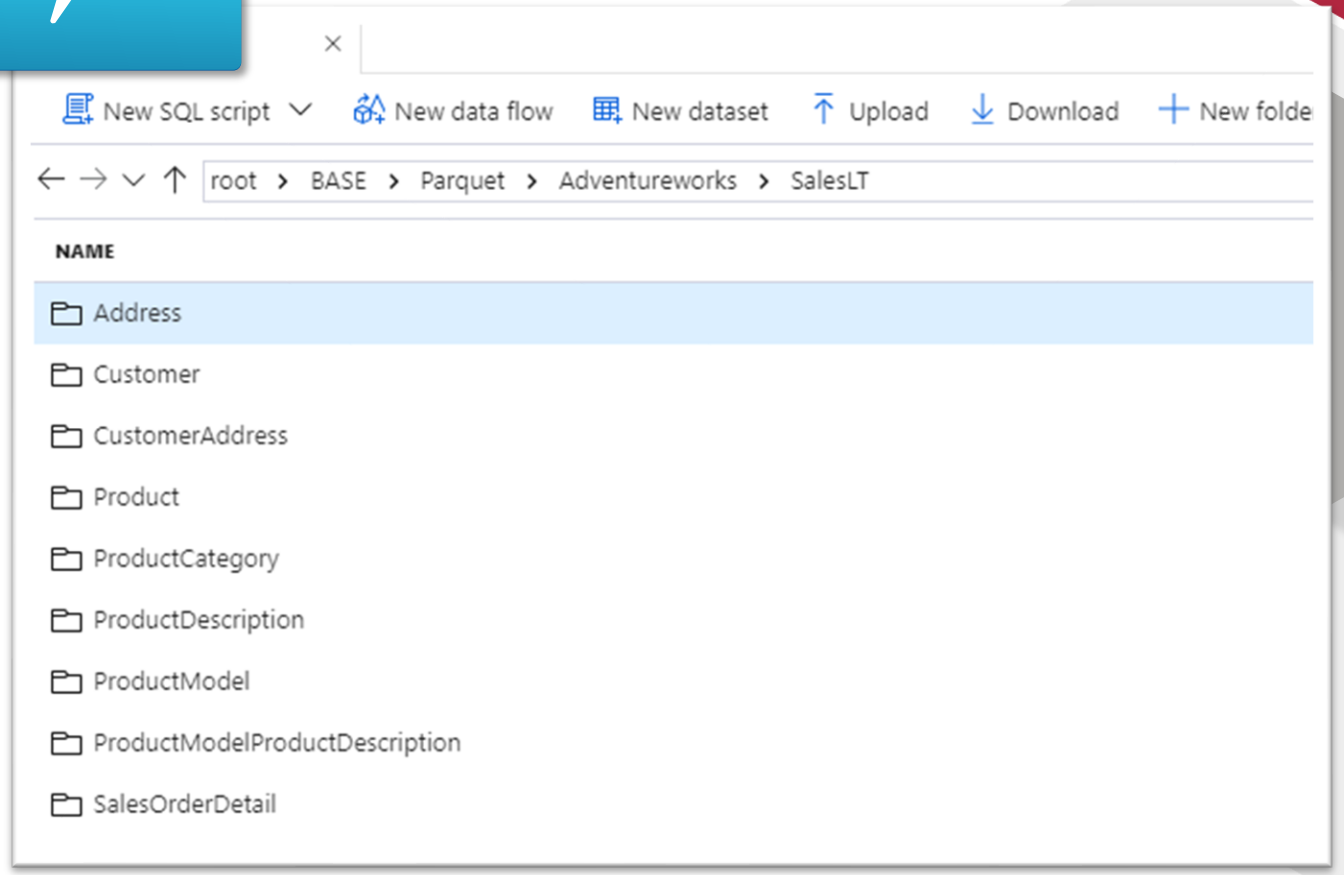
Display / Charting

Version Control

More Natural UI Integration

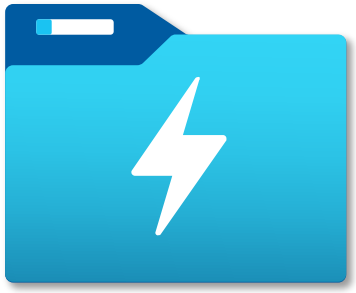
SYNAPSE INTEGRATIONS

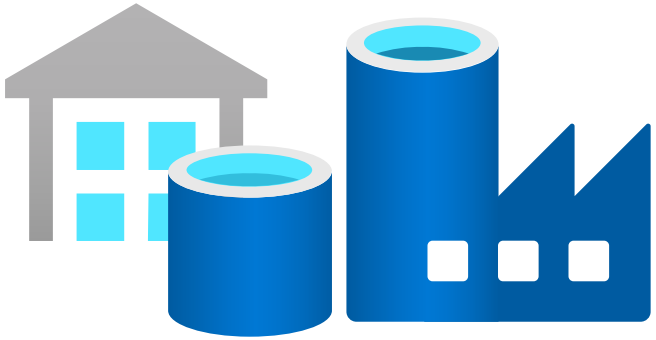




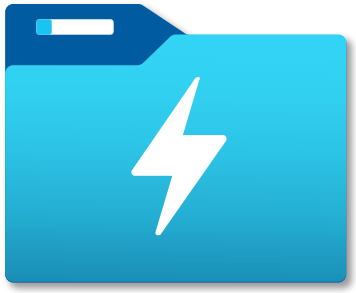
ADVANCING ANALYTICS

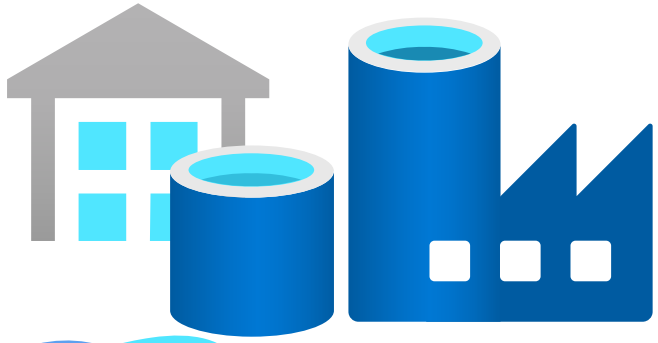
SYNAPSE INTEGRATIONS

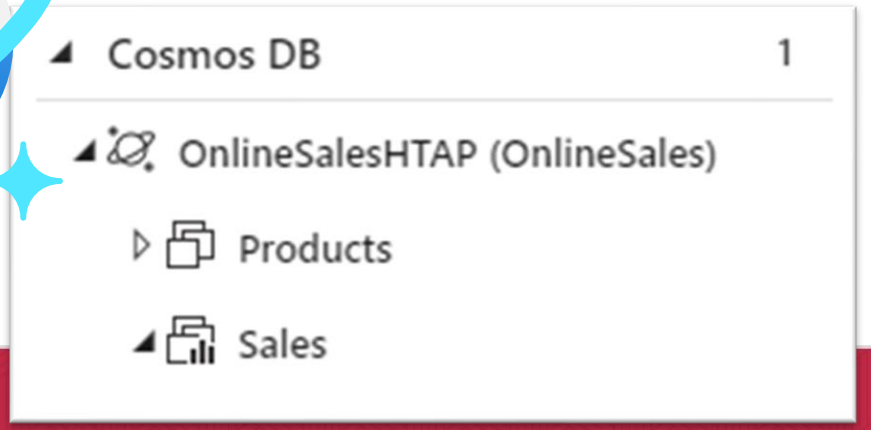




SYNAPSE INTEGRATIONS

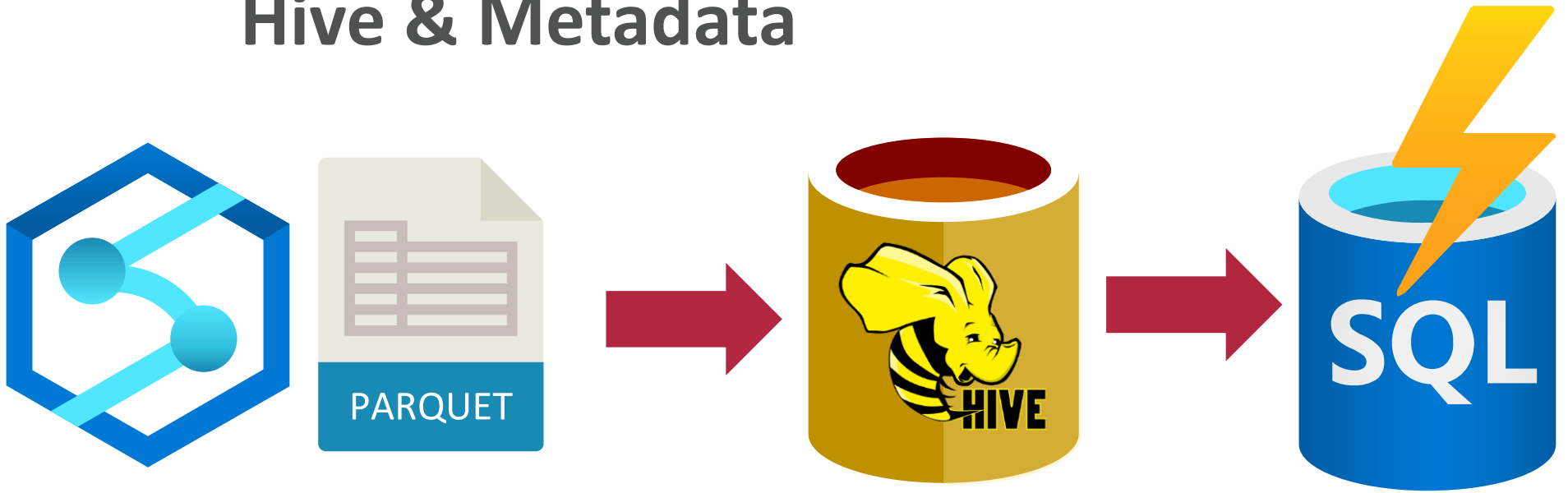






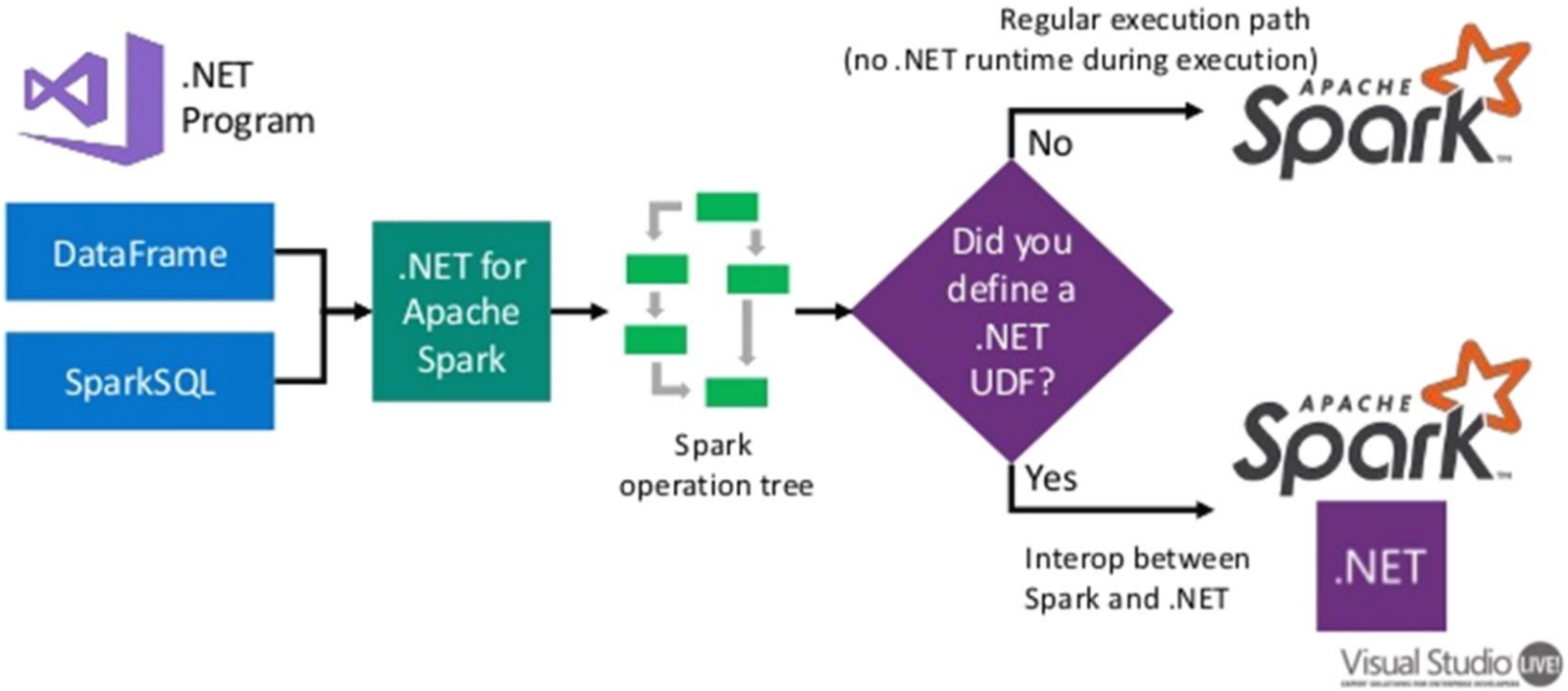
ADVANCING ANALYTICS

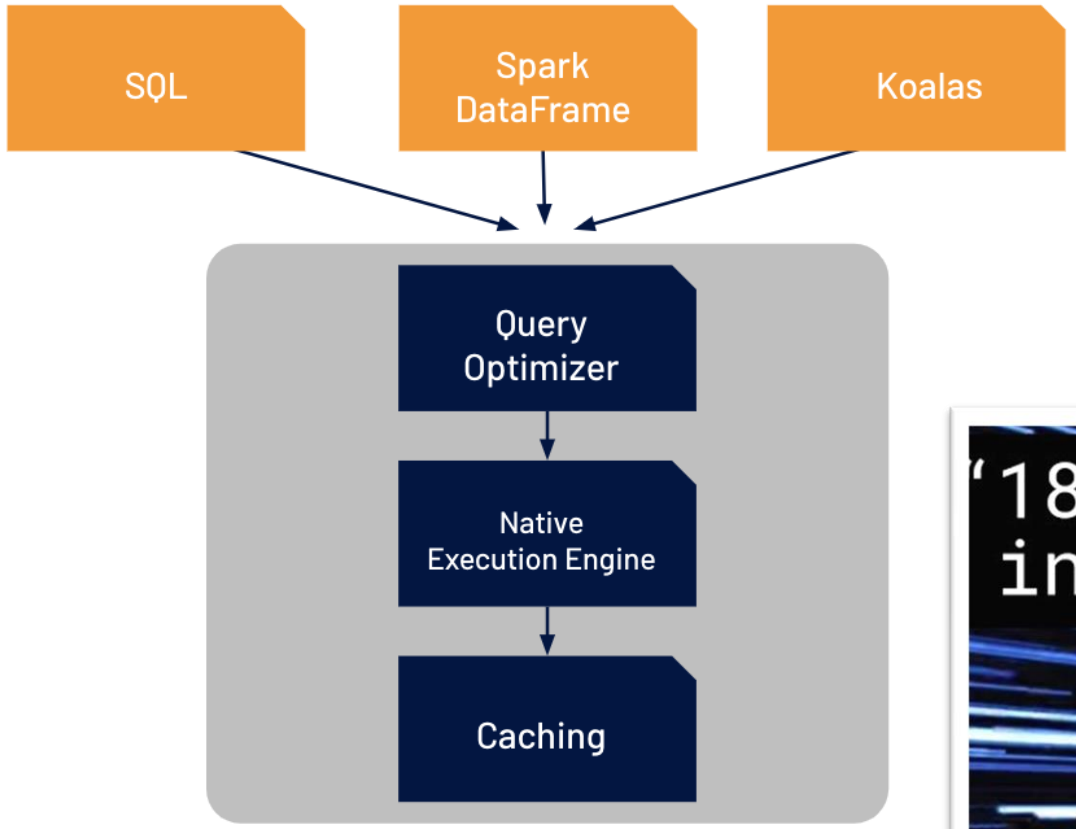
Hive & Metadata

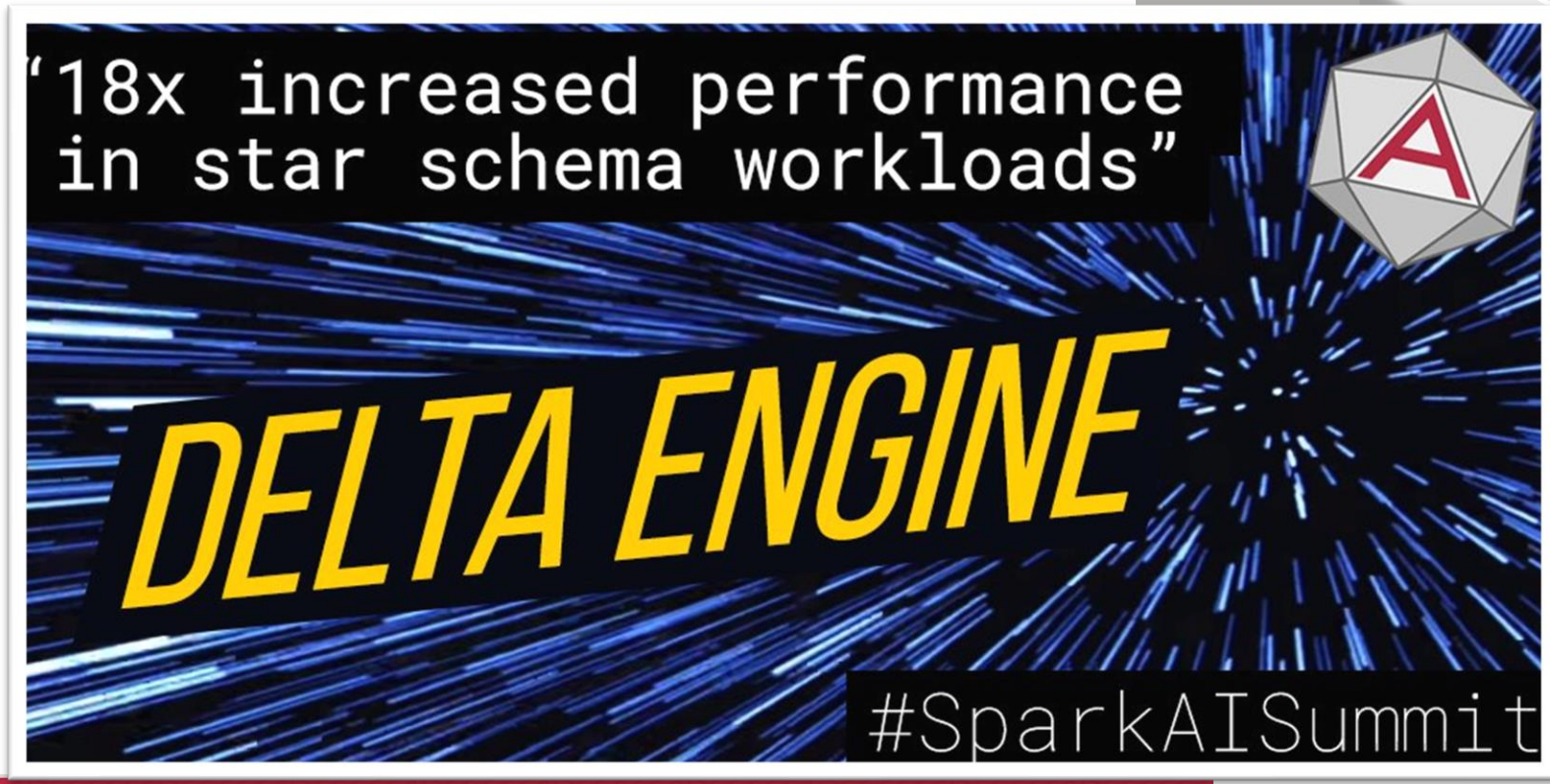


SQL On-Demand

What is happening when you write .NET Spark code?




SQL
Spark DataFrame
Koalas
Query Optimizer
Native Execution Engine
Caching


“18x increased performance in star schema workloads”
DELTA ENGINE
#SparkAISummit

HYPERSPACE

An extensible indexing subsystem for Apache Spark™

Scala Python .NET

Our first hyperspace: the covering index

Table A

a b c

Table B

a p q

Without Indexes

```
SELECT b, c
FROM Table A, B
JOIN ON A.a = B.a
```

With Covering Indexes

Step 1: Shuffle (data is not sorted)

Table A Table B

Step 2: Sort both sides

Table A Table B

Step 3: Merge

Result

Step 1: Optimizer picks index (pre-shuffled, pre-sorted)

Idx A Idx B

Step 2: Merge

Result

Shuffle eliminated

Since shuffle is the most expensive step, this query might run faster at scale

SPARK+AI SUMMIT


mlflow


DELTA LAKE


DELTA LAKE

ROUND THREE - PRICE

Apache Spark Pools in Azure Synapse

Memory Optimised

NODE SIZE:

Small (4 vCPU(s)/32 GB)

2	×	730 Hours	×	£0.13	=	£774.76
Instances				Per vCore-hour		

INSTANCE:

L4s: 4 Core(s), 32 GB RAM, 1 Databricks Unit(s), £0.270/hour

= £590.87
Average per month
(£0.00 charged upfront)

3	×	730 Hours
Virtual machines		

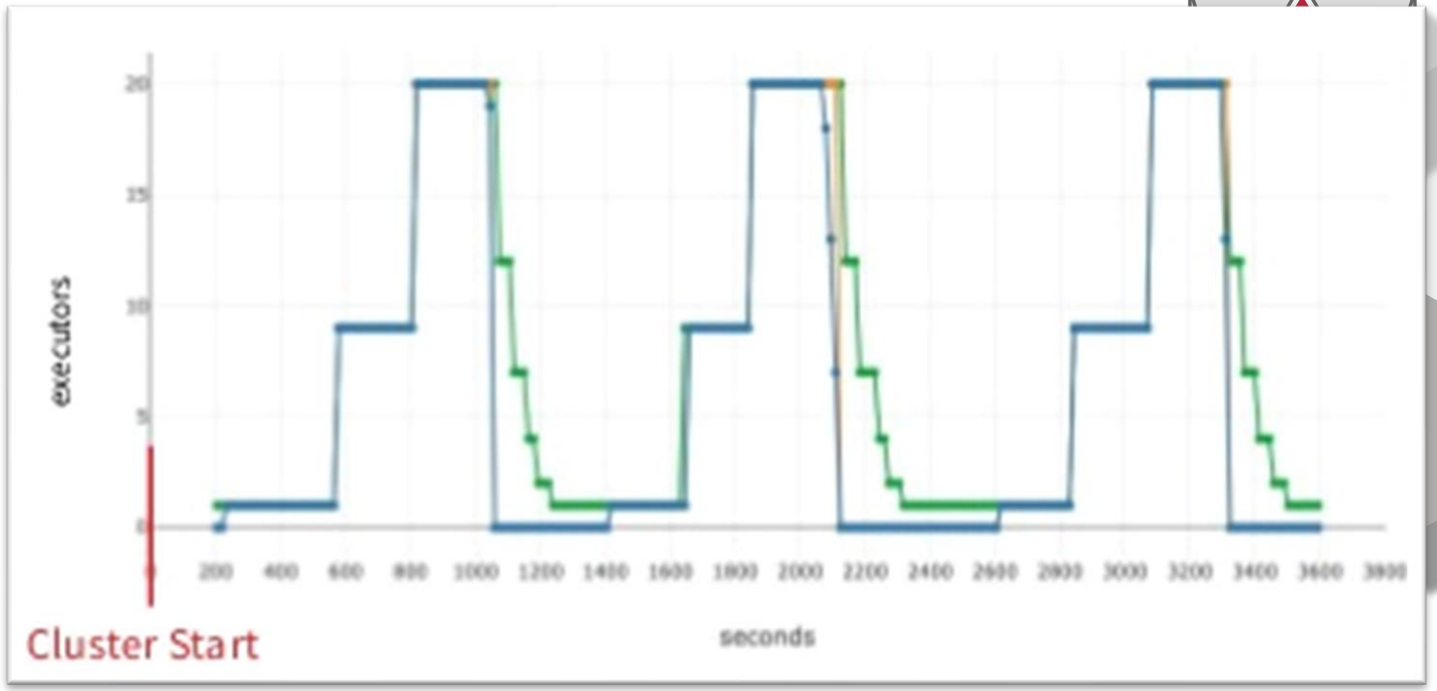
DBU (Databricks Unit)

3.00	×	£0.410	×	730 Hours	=	£897.72
DBU		Per DBU per hour				

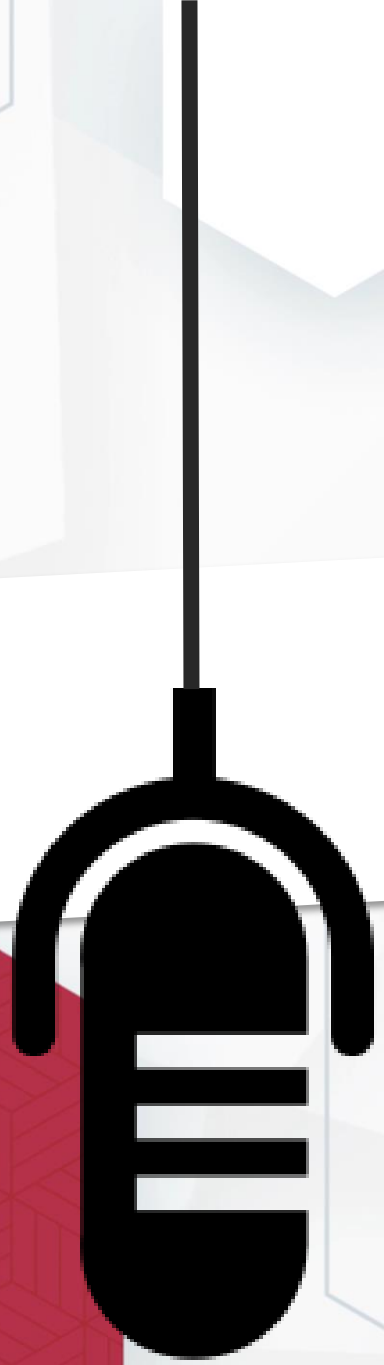
Upfront cost	£0.00
Monthly cost	£1,488.59

DBX costs more,
but is more efficient
for parallel use &
better optimised

Databricks Premium has advanced **Autoscale** feature



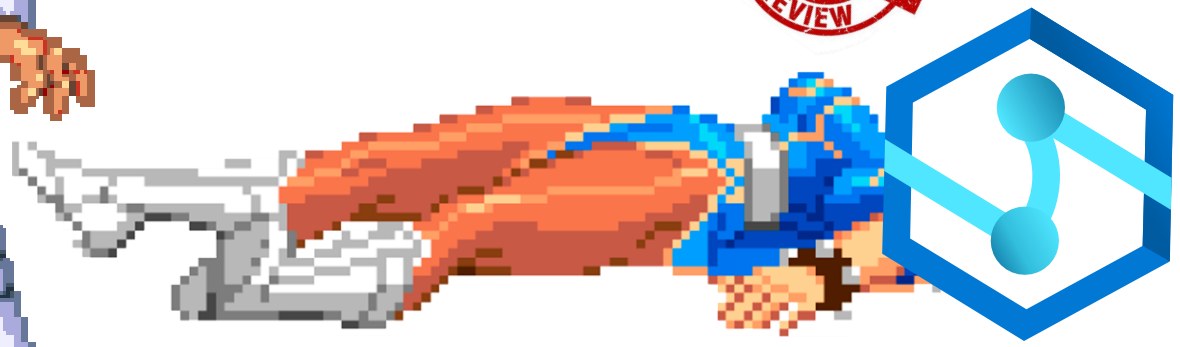
How Synapse will handle session-based autoscaling or high-concurrency workloads is still unknown



AND THE WINNER IS...

FOR PURE SPARK PERFORMANCE

For large volume, high-performance or high-concurrency spark usage, Databricks is a more mature, capable offering



But that's like comparing SQL Server **Enterprise** to SQL Server **Standard**.

Of course Enterprise does more, performs better, has additional features - it's the Premium offering

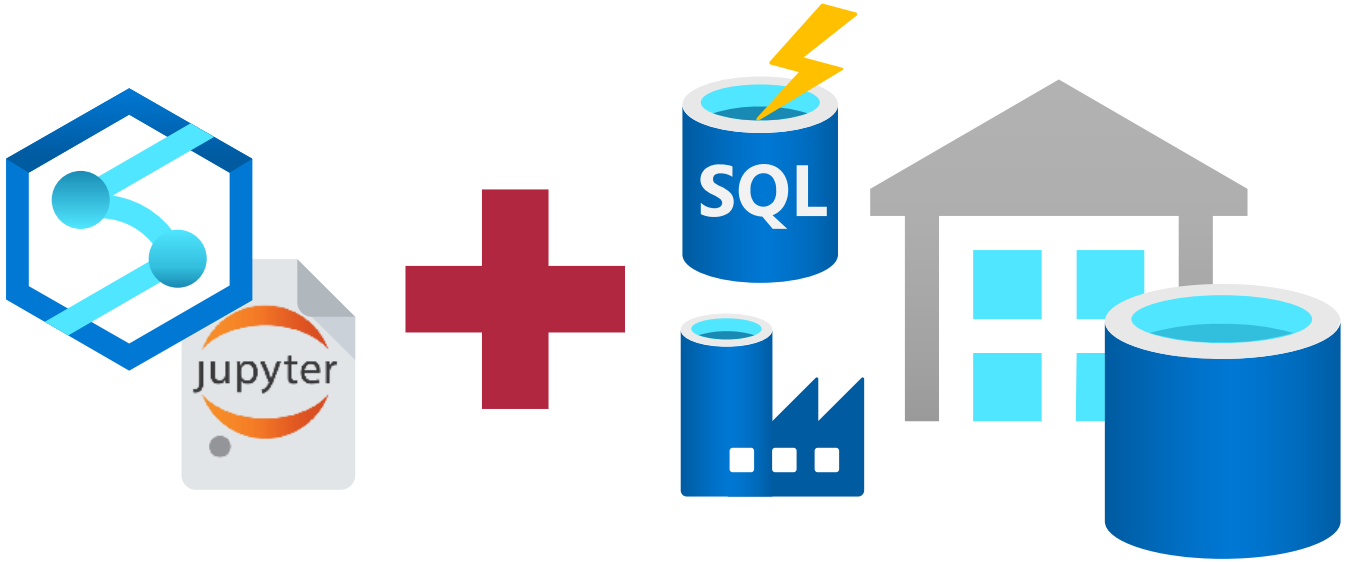
Premium Azure Spark

Standard Azure Spark

ADVANCING ANALYTICS

SCENARIOS:

You're building a platform that is largely SQL-based, but may have one or two Spark edge-cases



Synapse Spark Pools will be far easier and less complex to set up

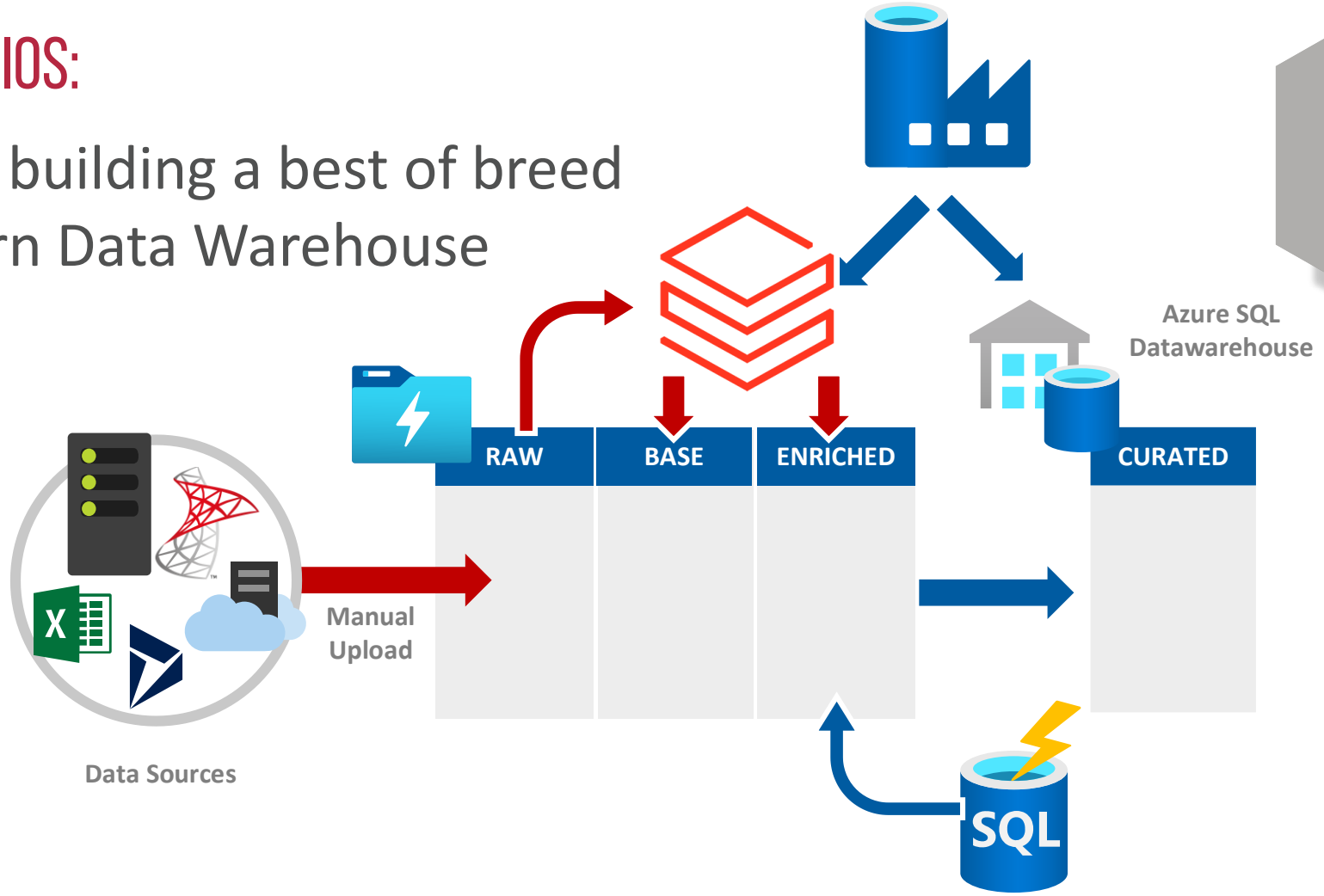
SCENARIOS:

You’re a Spark shop with a complex spark application and looking for a place to house it in the cloud

Databricks will have more features and provide better performance optimisations

SCENARIOS:

You're building a best of breed Modern Data Warehouse



Databricks and **Synapse Analytics** work **Better Together**

ADVANCING ANALYTICS