

PYTHON PIPELINE PRIMER

ADDING A LITTLE SPARK TO YOUR WAREHOUSE PROCESS



Simon Whiteley
@MrSiWhiteley



<https://github.com/SiWhiteley/DatabricksETL>

PYTHON PIPELINE PRIMER

Agenda

What is
Databricks?

Patterns &
Implementation

Orchestration

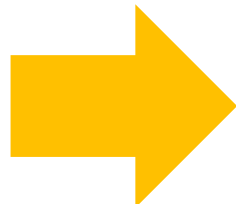
Data Factory
Dataflows

PYTHON PIPELINE PRIMER



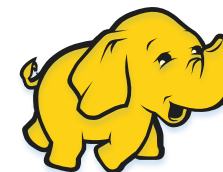
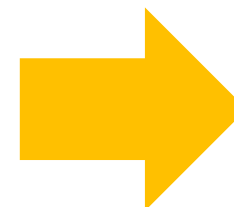
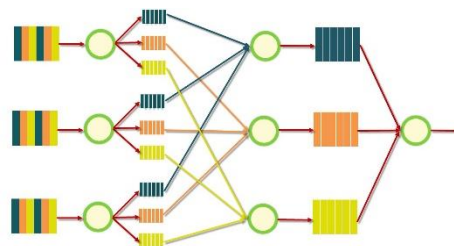
Google File System Papers
Released

2003



Google MapReduce Papers

2004



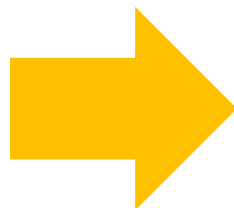
2006

Apache Hadoop
project created



Matei Zaharia starts Spark
project

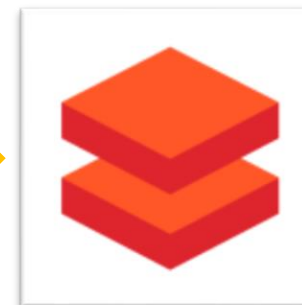
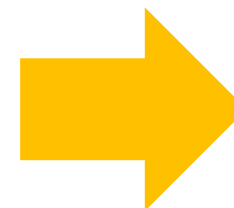
2012



THE
APACHE
SOFTWARE FOUNDATION

Project donated to Apache
Foundation

2013



Databricks founded by
Matei

2013

PYTHON PIPELINE PRIMER



2016

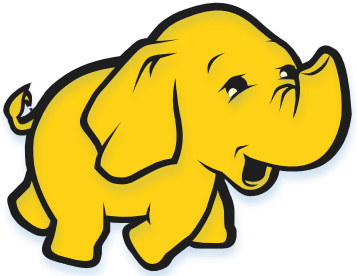
It's new to
Azure, not to
everyone else!



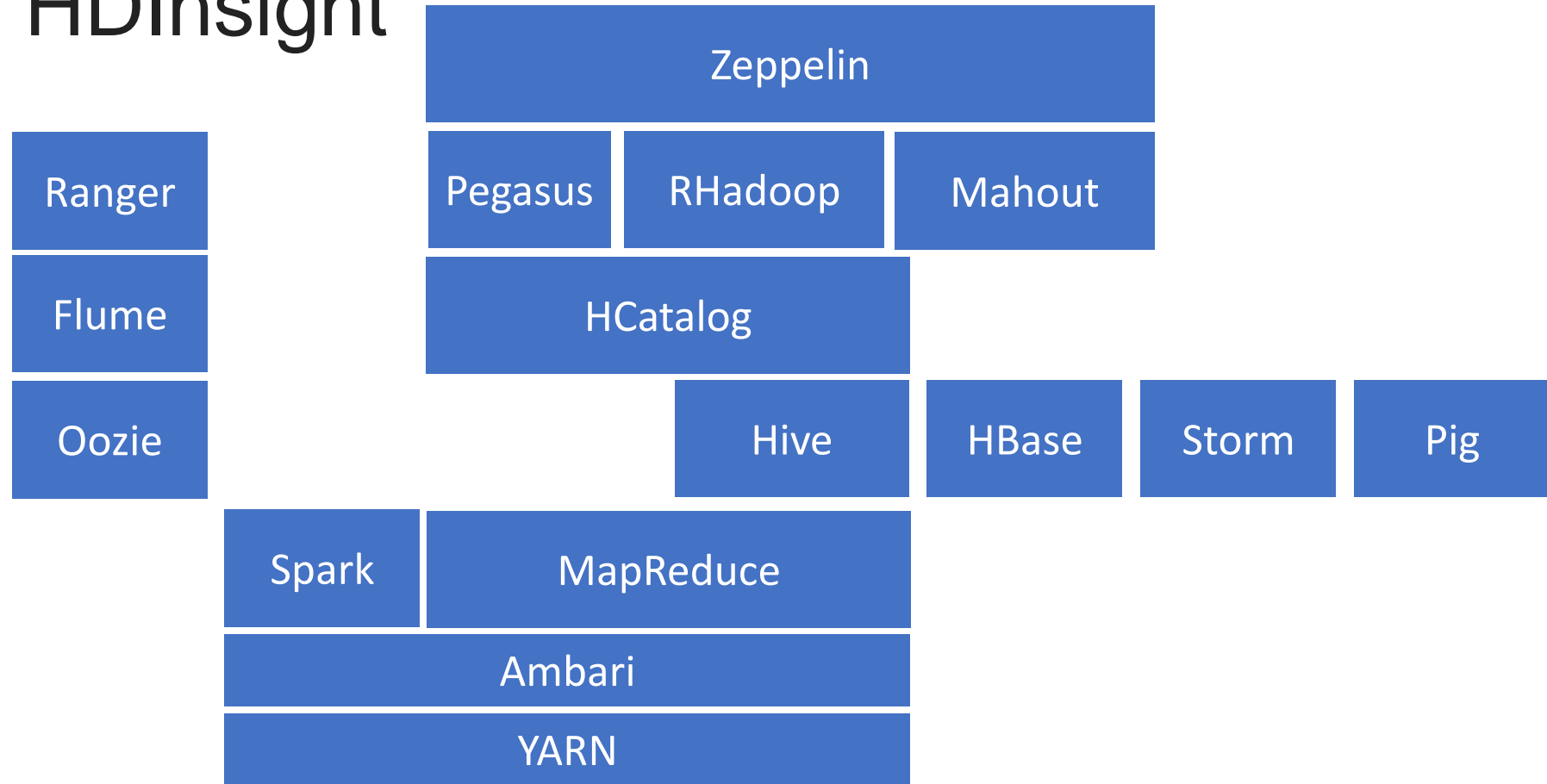
Microsoft
Azure

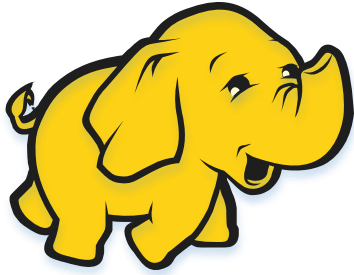
2018

PYTHON PIPELINE PRIMER



HDInsight





Open Source

20 min provisioning

Integrates Well

Secure

Hadoop, Spark, Kafka, Hbase,
HIVE, Storm...

Slow Release Cycle



Open Source

5 min provisioning

Integrates Well

Secure

Spark (Python/Scala/R)

Fast Release Cycle



Proprietary

1 min provisioning

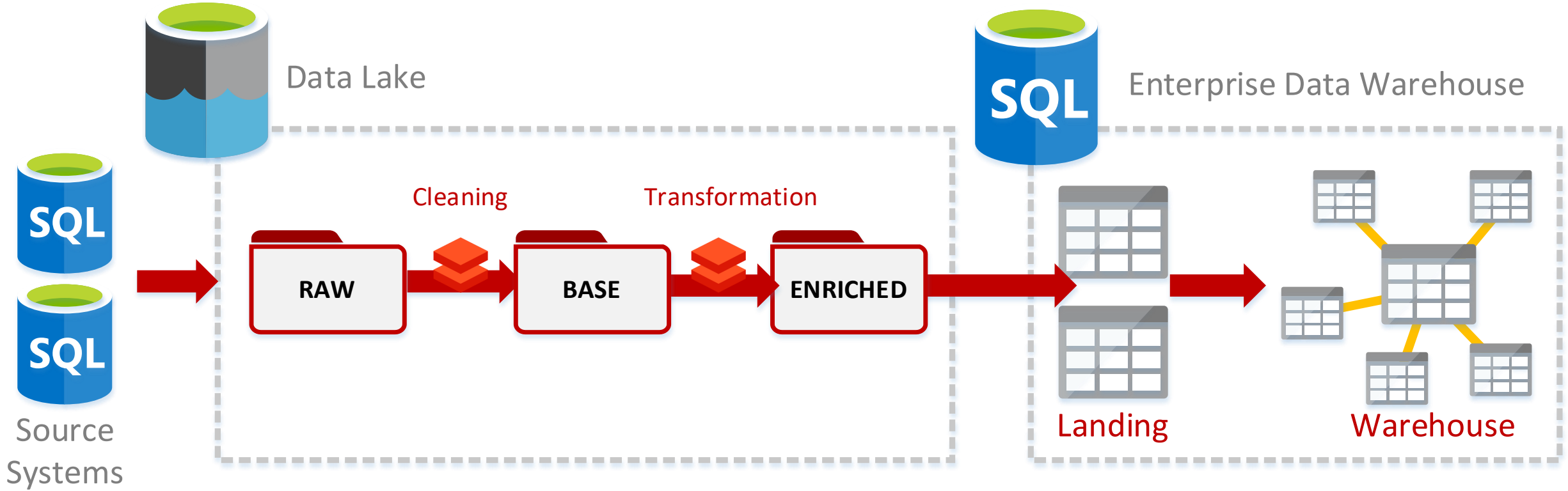
Integrates Poorly

Secure

U-SQL

Slow Release Cycle

PYTHON PIPELINE PRO...

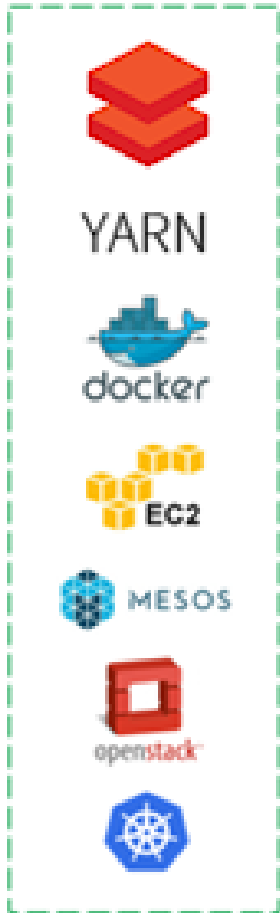


The background of the image is a dramatic, high-contrast photograph of dark, heavy clouds. A bright, glowing light source, possibly the sun or moon, is partially visible through the clouds in the upper center, creating a strong backlighting effect. A solid blue diagonal banner cuts across the middle of the image, providing a contrasting background for the white text.

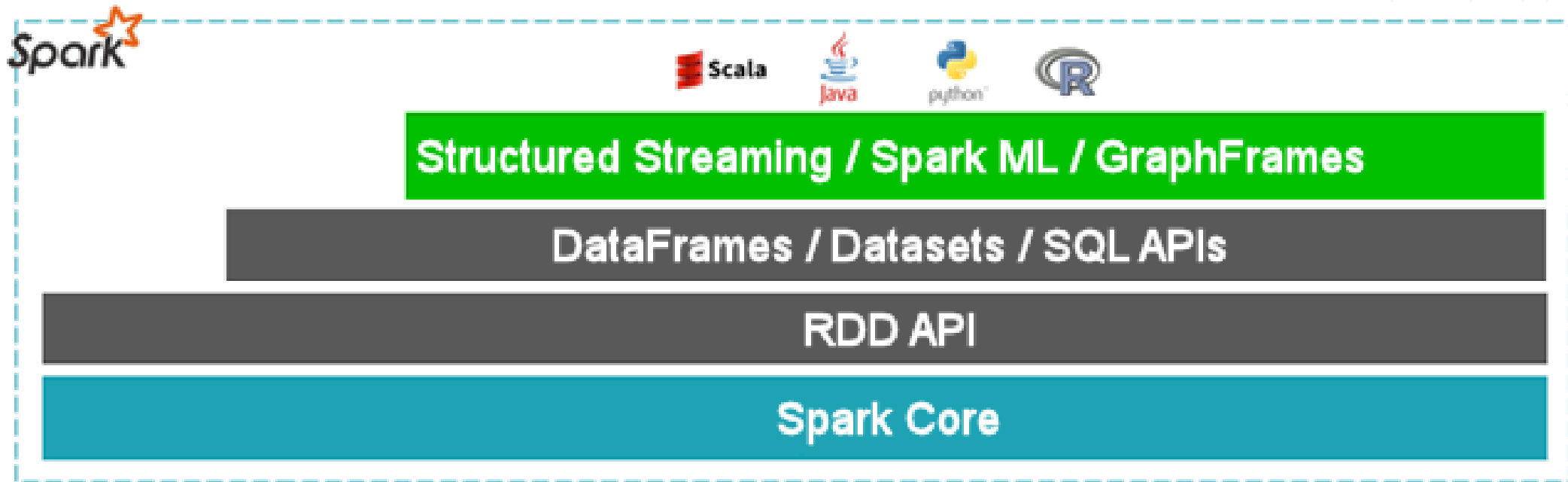
DATABRICKS BASICS

UNDER THE HOOD

Environments



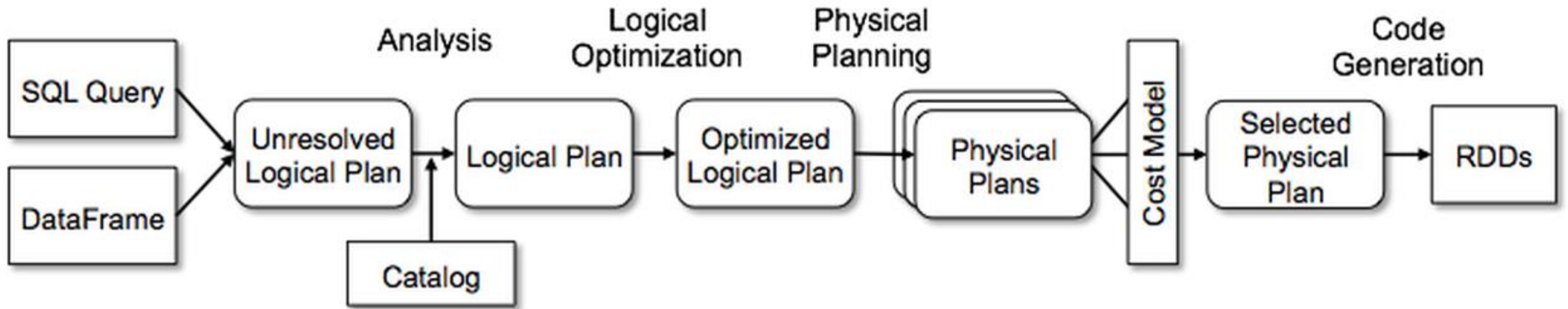
Workloads



Data Sources



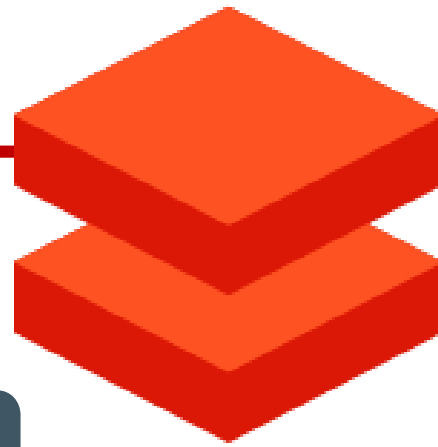
PYTHON PIPELINE PRIMER



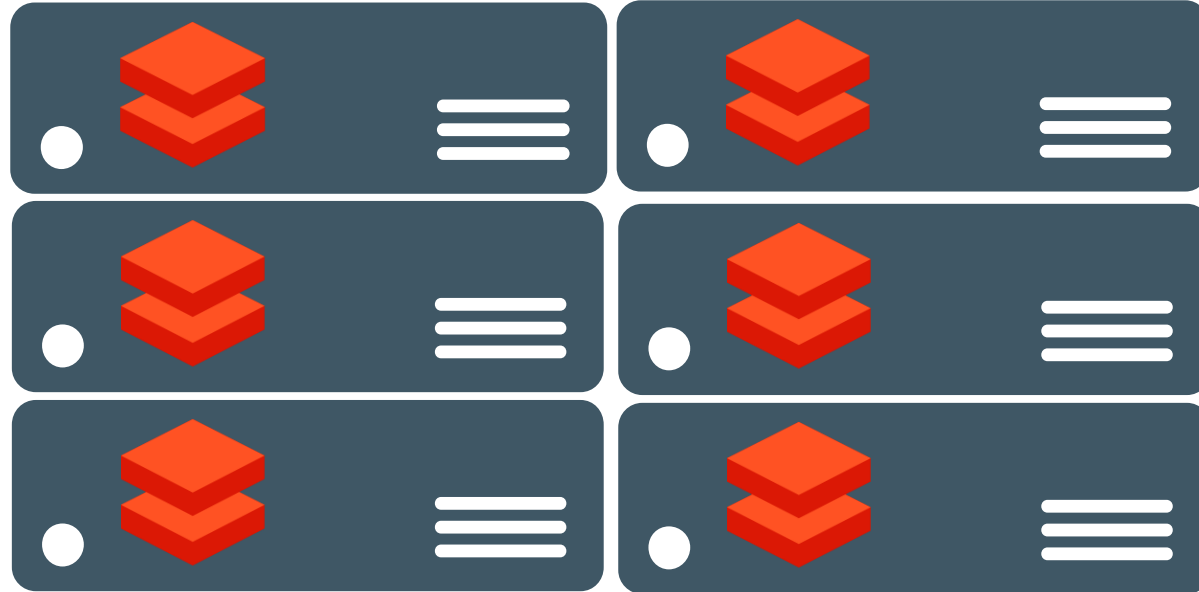
The background of the image is a dramatic, high-contrast photograph of dark, swirling clouds. A bright blue diagonal banner cuts across the middle of the image, providing a stark contrast to the dark, moody sky. The text 'PATTERNS & IMPLEMENTATION' is written in white, bold, sans-serif capital letters on this banner.

PATTERNS & IMPLEMENTATION

Workload Isolation



Processing Cluster



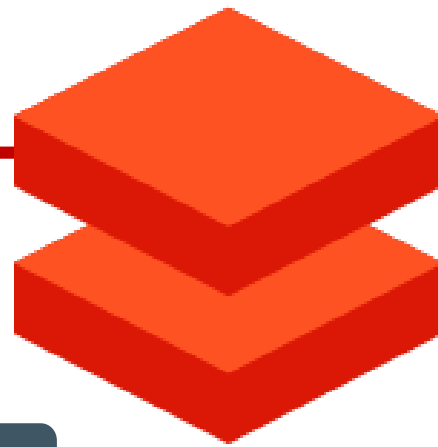
Streaming Cluster



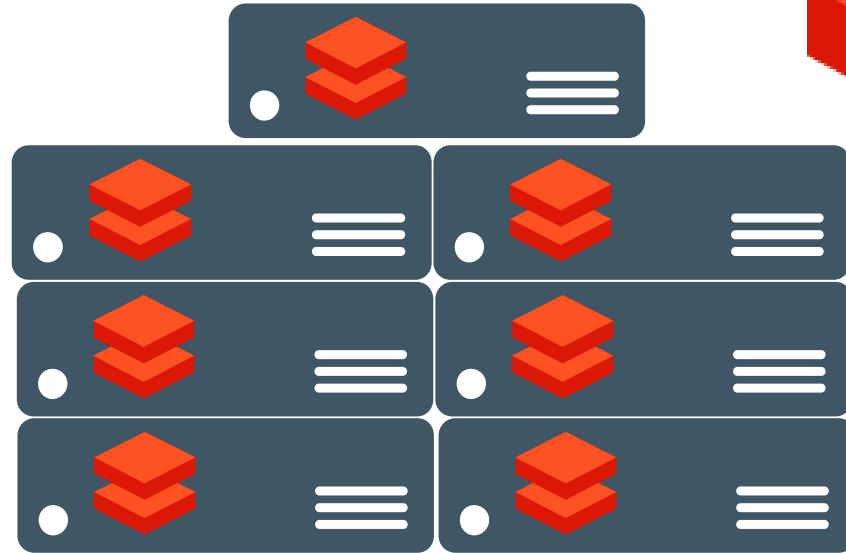
Interactive Cluster



Workload Isolation



Fact Cluster



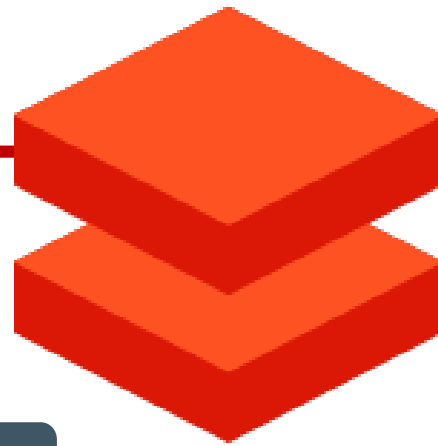
Tensorflow Cluster



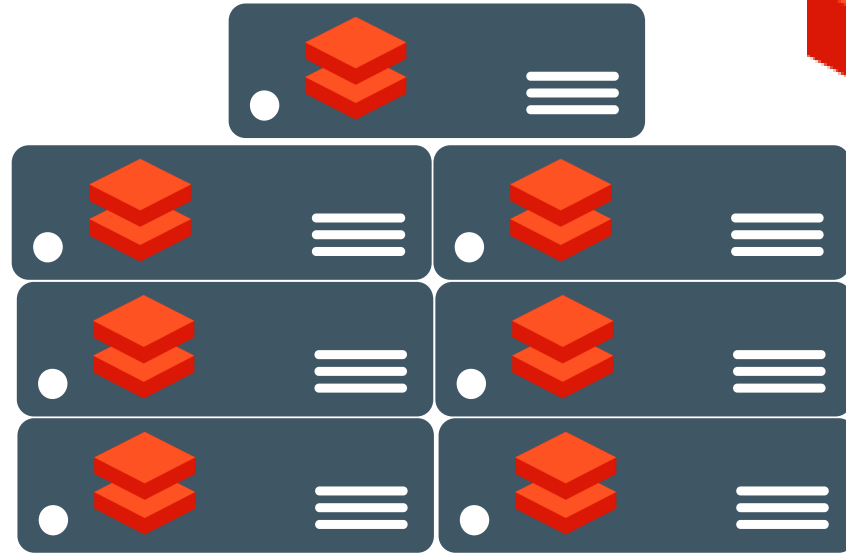
Dim Cluster



Workload Isolation



Fact Cluster



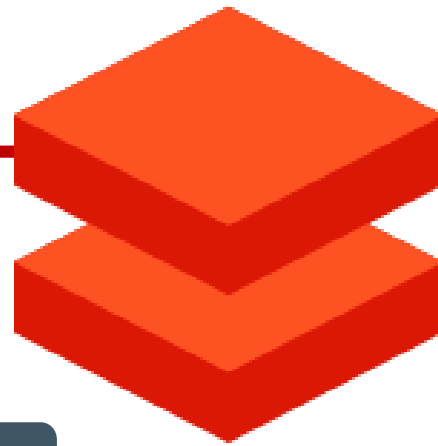
Tensorflow Cluster



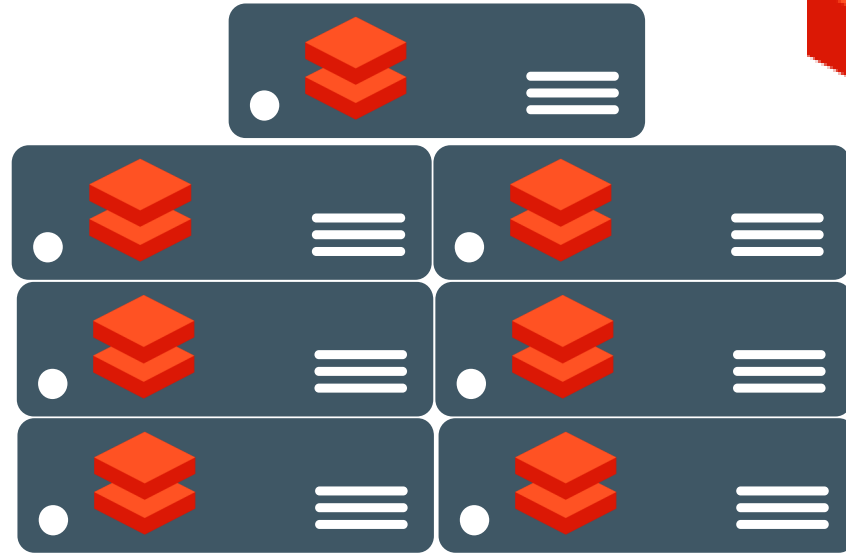
Dim Cluster



Workload Isolation



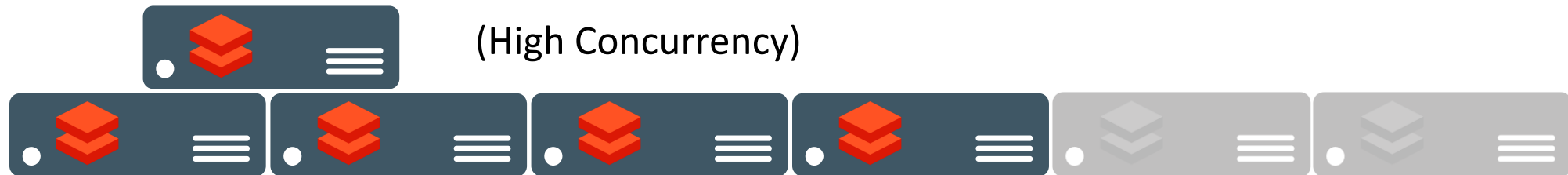
Fact Cluster



Tensorflow Cluster



Dim Cluster



SO.... WHAT SIZE?

Size of Driver

What is the largest dataset that we will perform need to return to the user? What actions do we need to perform outside of the spark engine? How performance / memory intensive is it? How many concurrent workers does my driver need to handle?

Size of Worker

What is the largest data set/single partition that needs to fit on a single executor?
How much memory should be left over for performing calculations?
How fast should each executor finish their job?

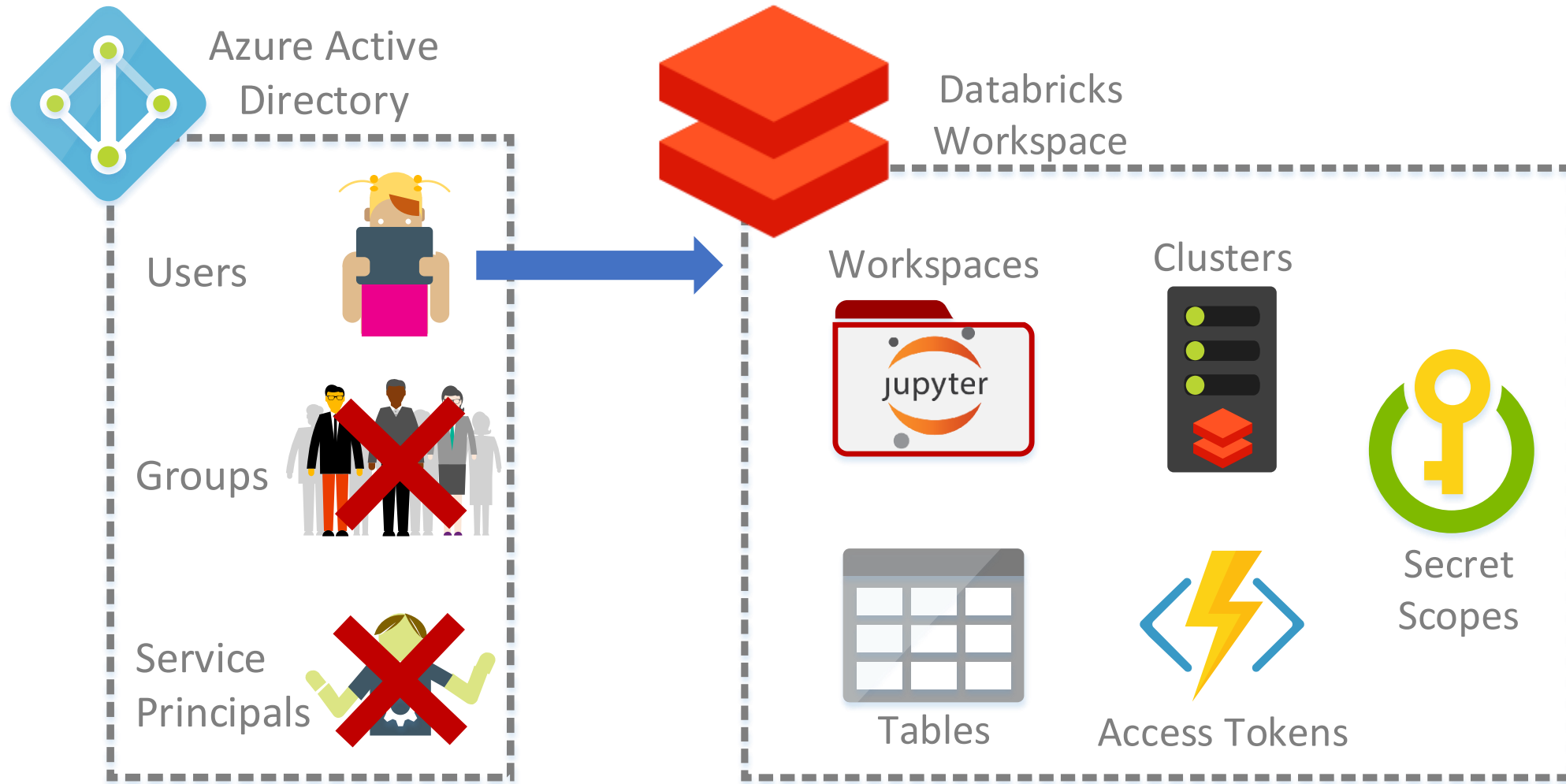
Number of Workers

What is the total amount of data that needs to be held in memory (both for in transformation queries, and cached tables)
How much concurrency do I need?

The background of the image consists of dark, heavy, and textured clouds, suggesting a storm or a night sky. A solid blue diagonal band cuts across the middle of the image, providing a contrasting background for the text.

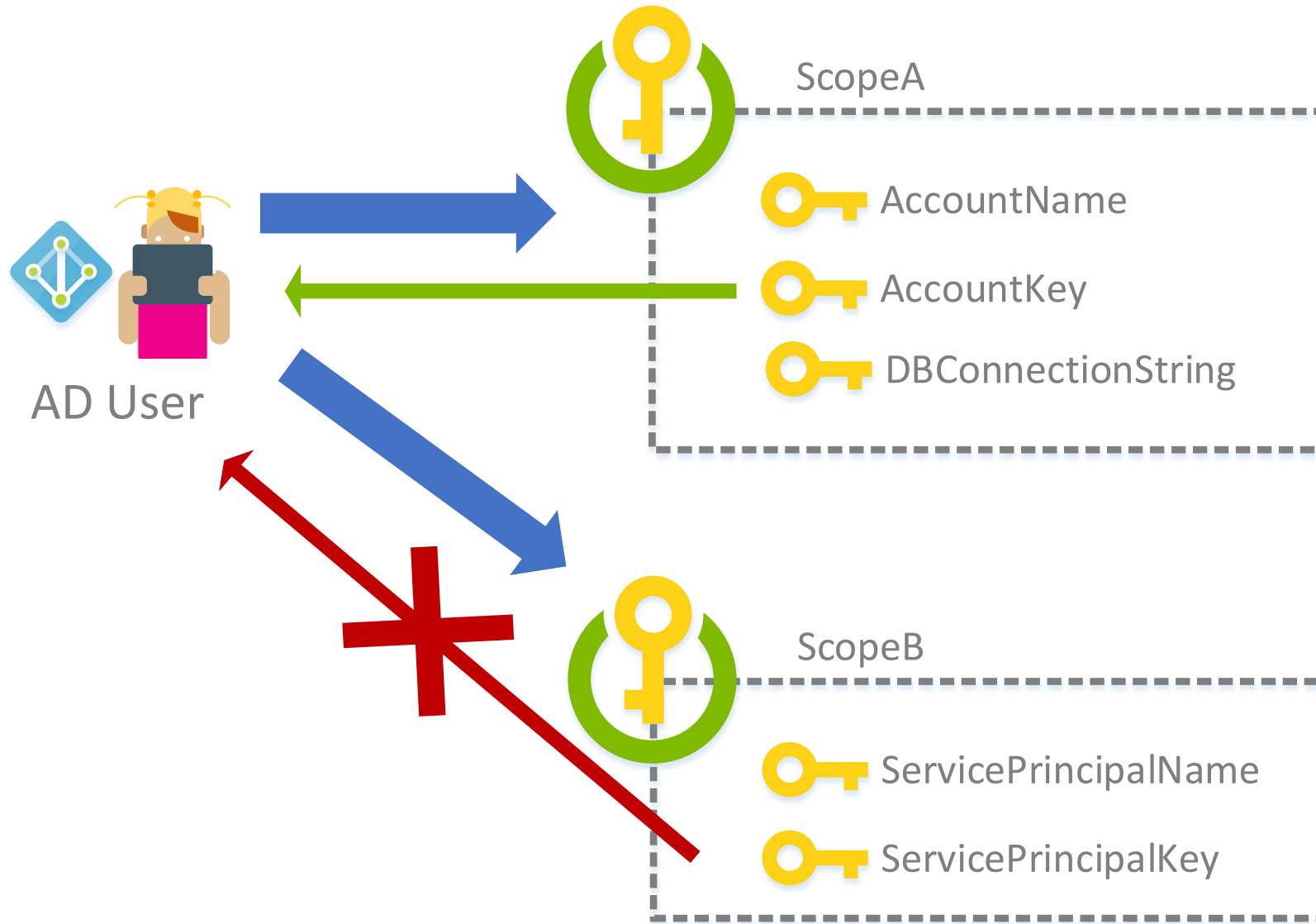
SECRETS

USER MANAGEMENT



PYTHON PIPELINE PRIMER

DATABRICKS SECRETS



Secrets are never displayed in databricks notebooks, even if you have access!

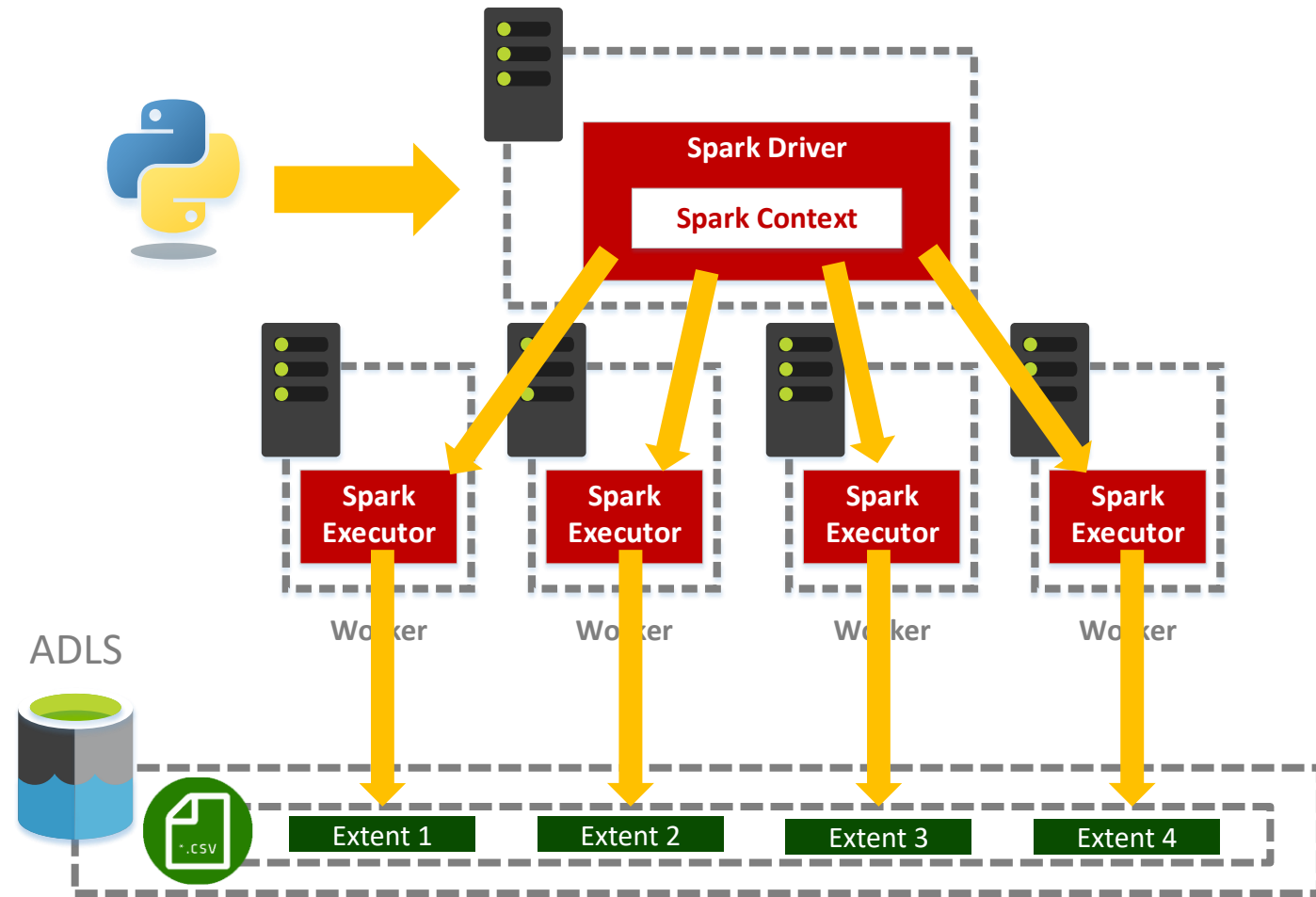
Any attempt to display the value will return **[REDACTED]**!

PYTHON PIPELINE

The background of the image consists of dark, heavy, and textured clouds, possibly storm clouds, with some lighter patches where light breaks through. A solid blue diagonal band cuts across the middle of the image, serving as a background for the text.

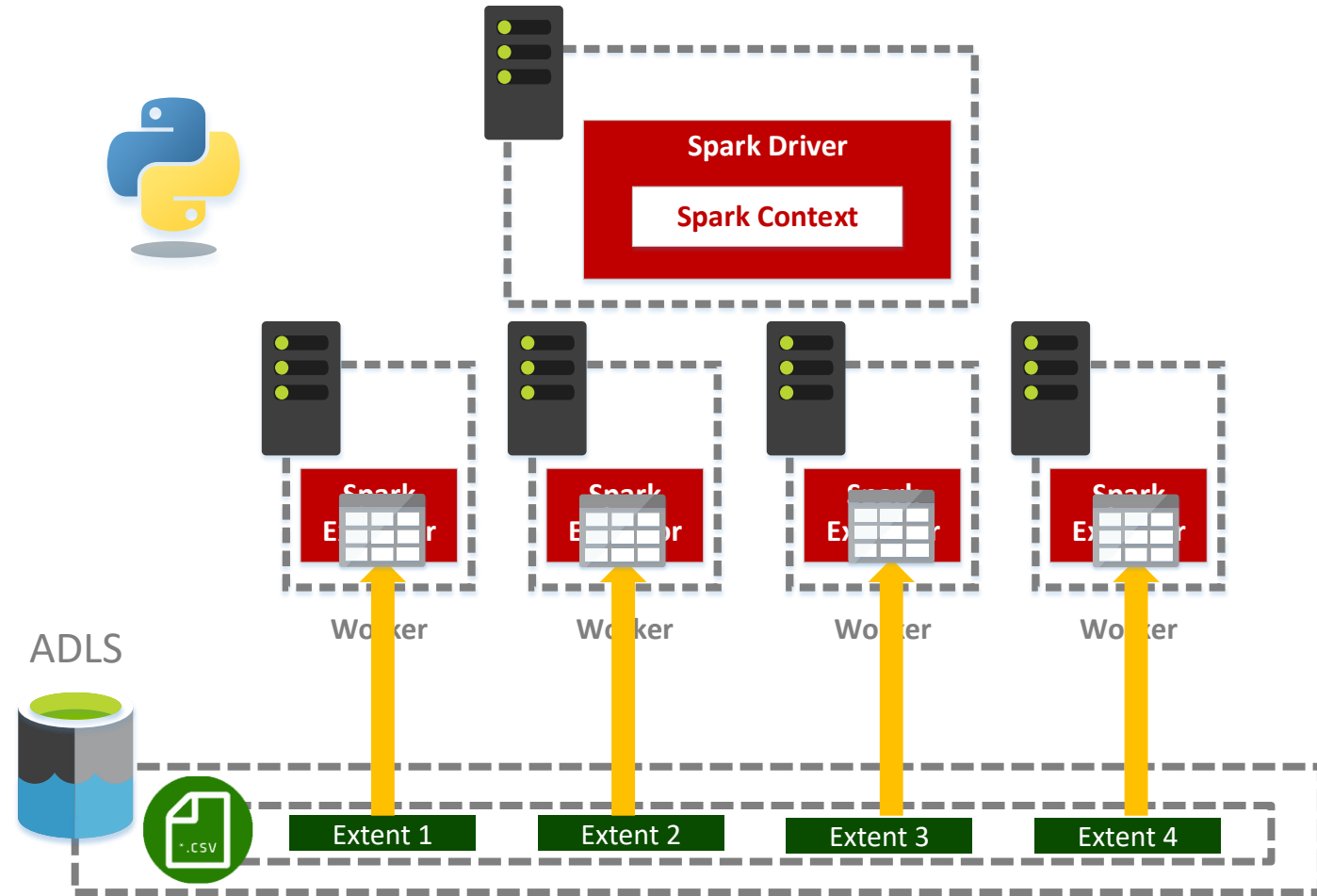
EXECUTIONS

DISTRIBUTED COMPUTE



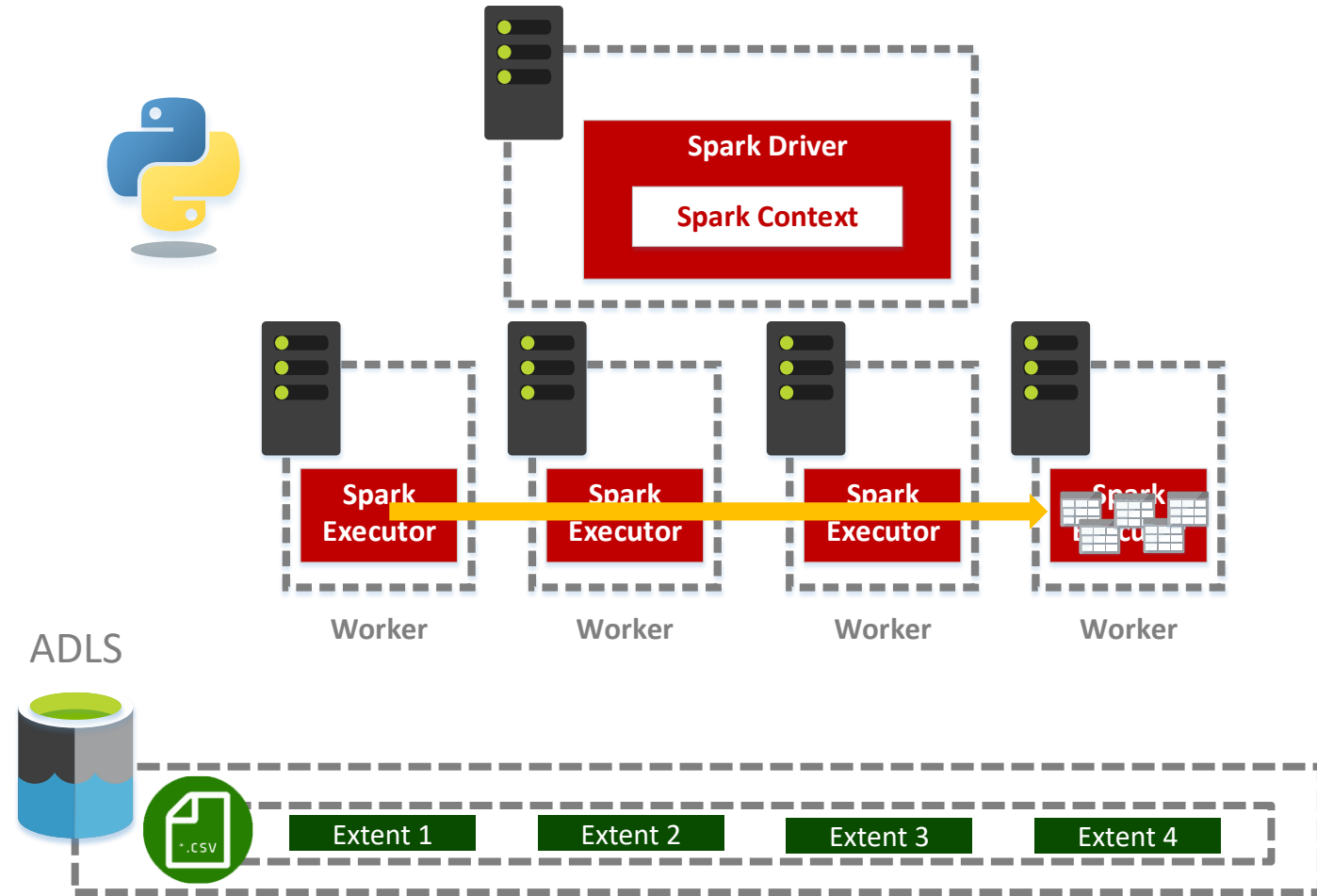
PYTHON PIPELINE PRIMER

DISTRIBUTED COMPUTE



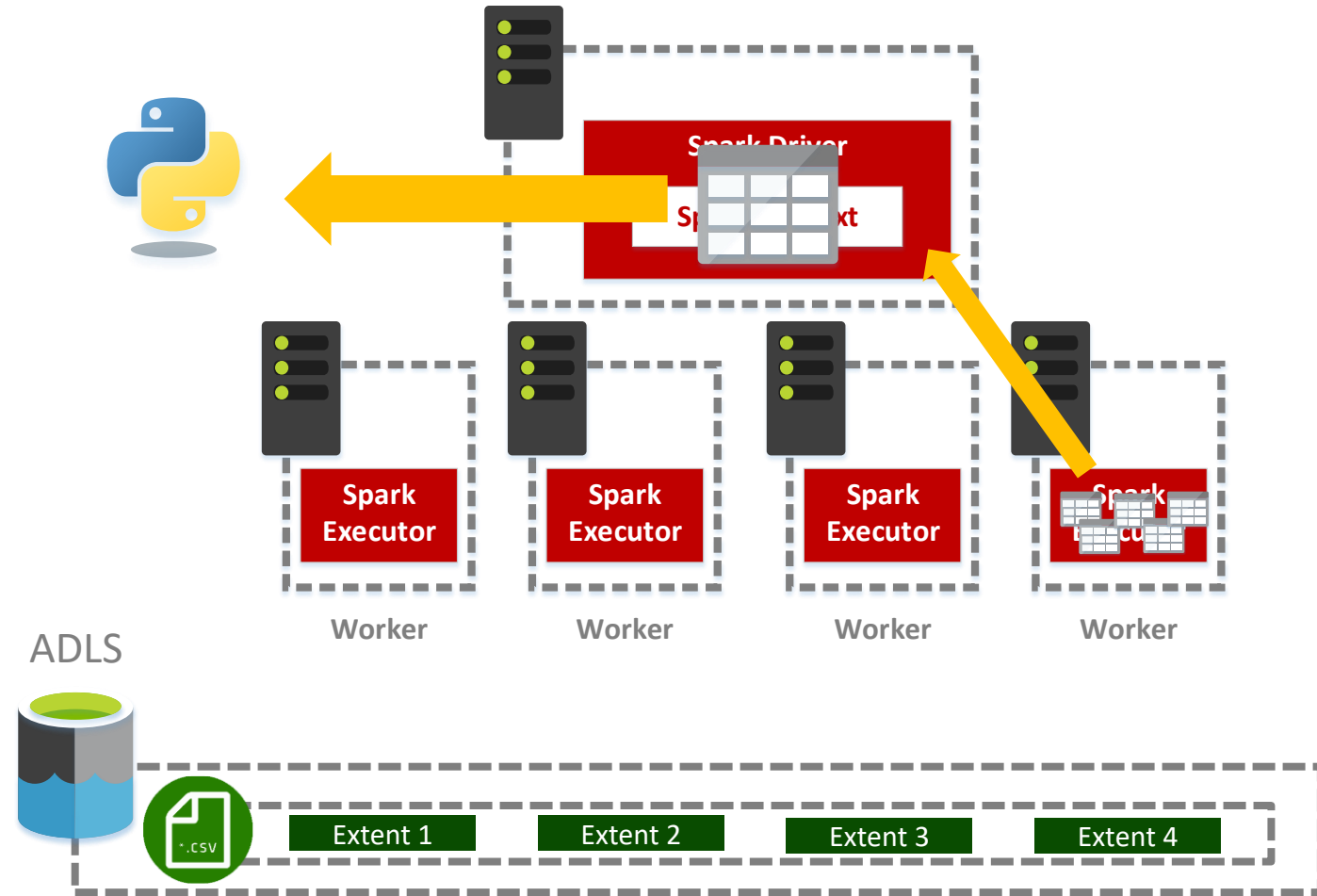
PYTHON PIPELINE PRIMER

DISTRIBUTED COMPUTE



PYTHON PIPELINE PRIMER

DISTRIBUTED COMPUTE



PYTHON PIPELINE PRIMER

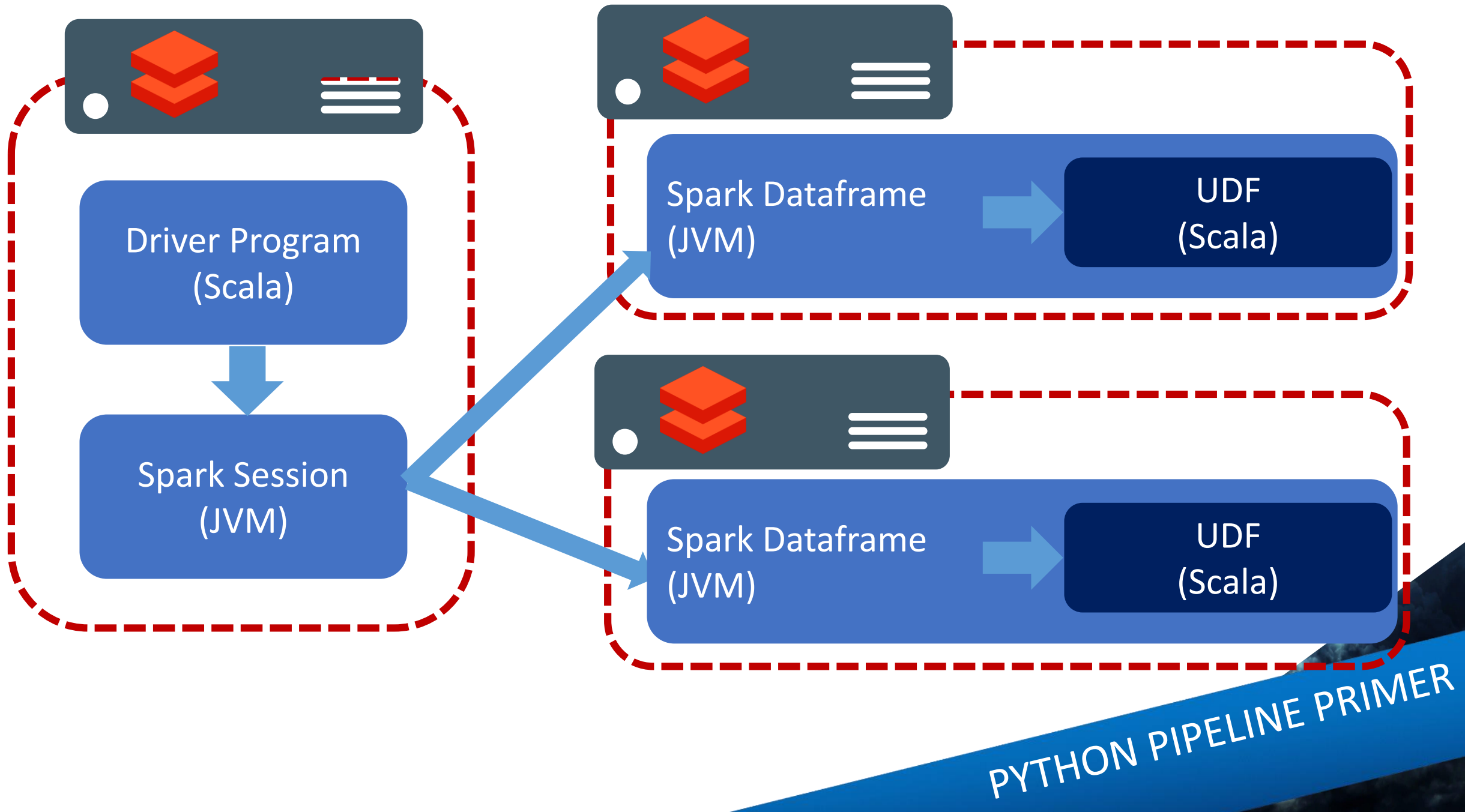
ALL LANGUAGES PERFORM THE SAME

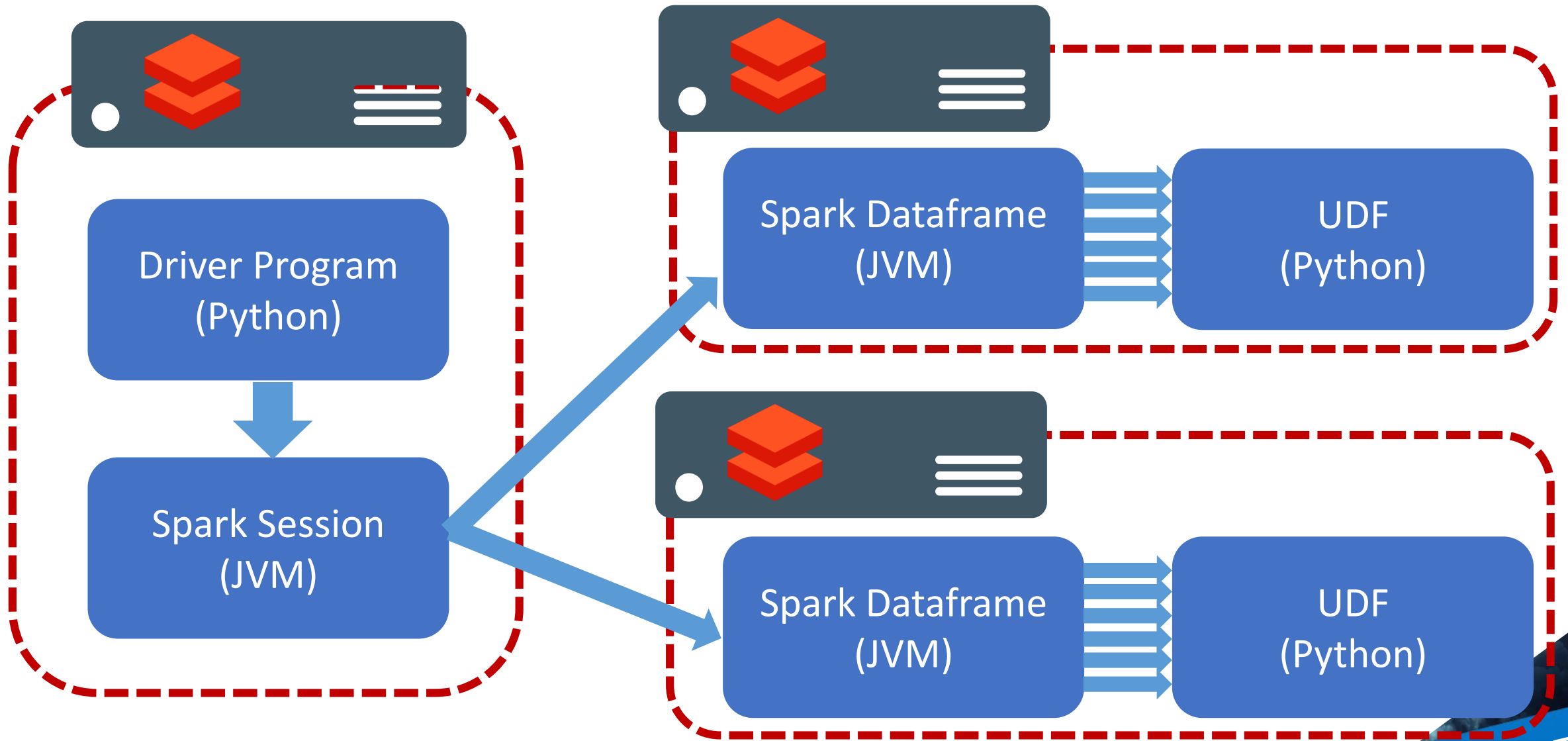
...EXCEPT...

PYTHON PIPELINE PRIMER

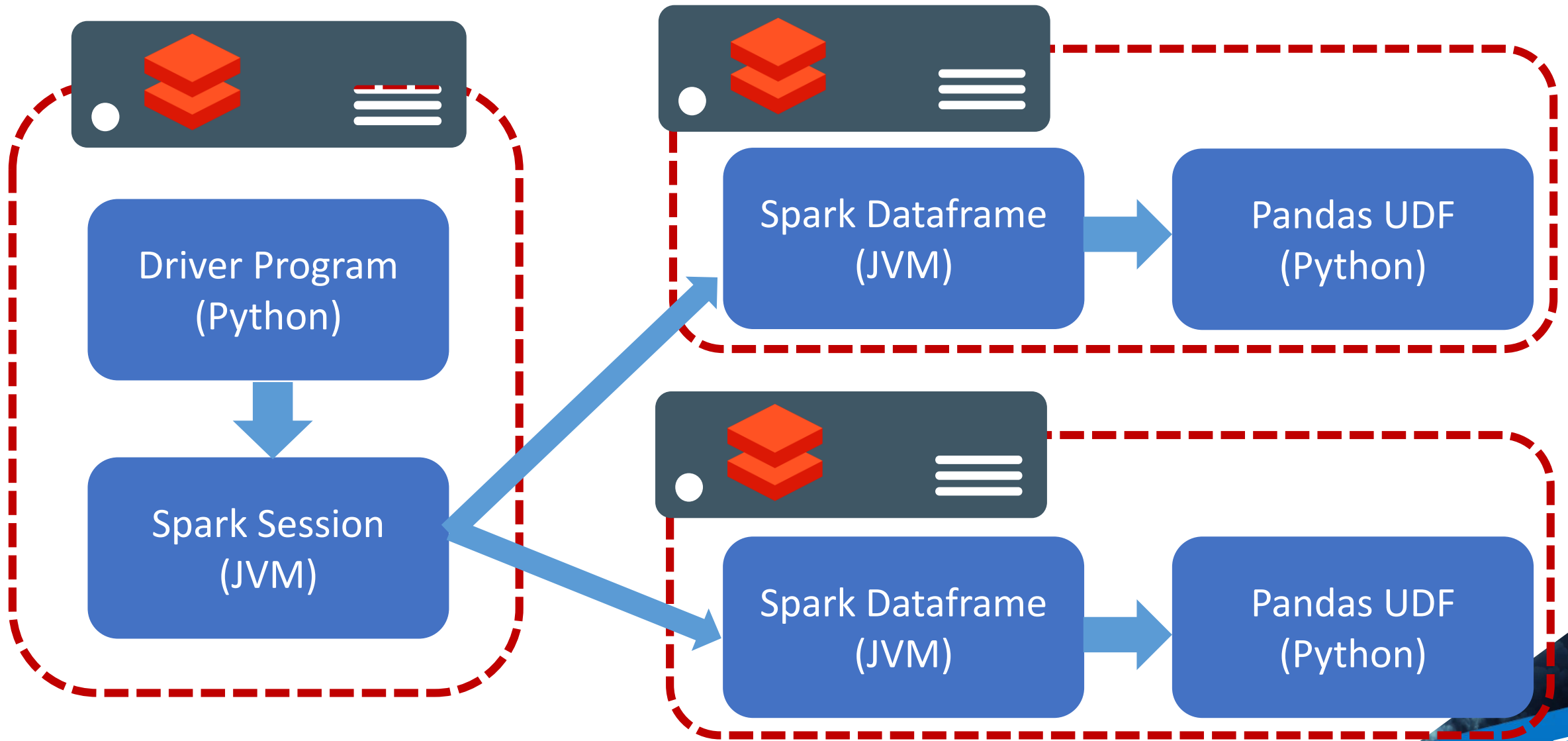
The background of the slide features a dramatic, high-contrast image of dark, swirling storm clouds. A solid blue diagonal band cuts across the middle of the image, providing a background for the text.

UDFS

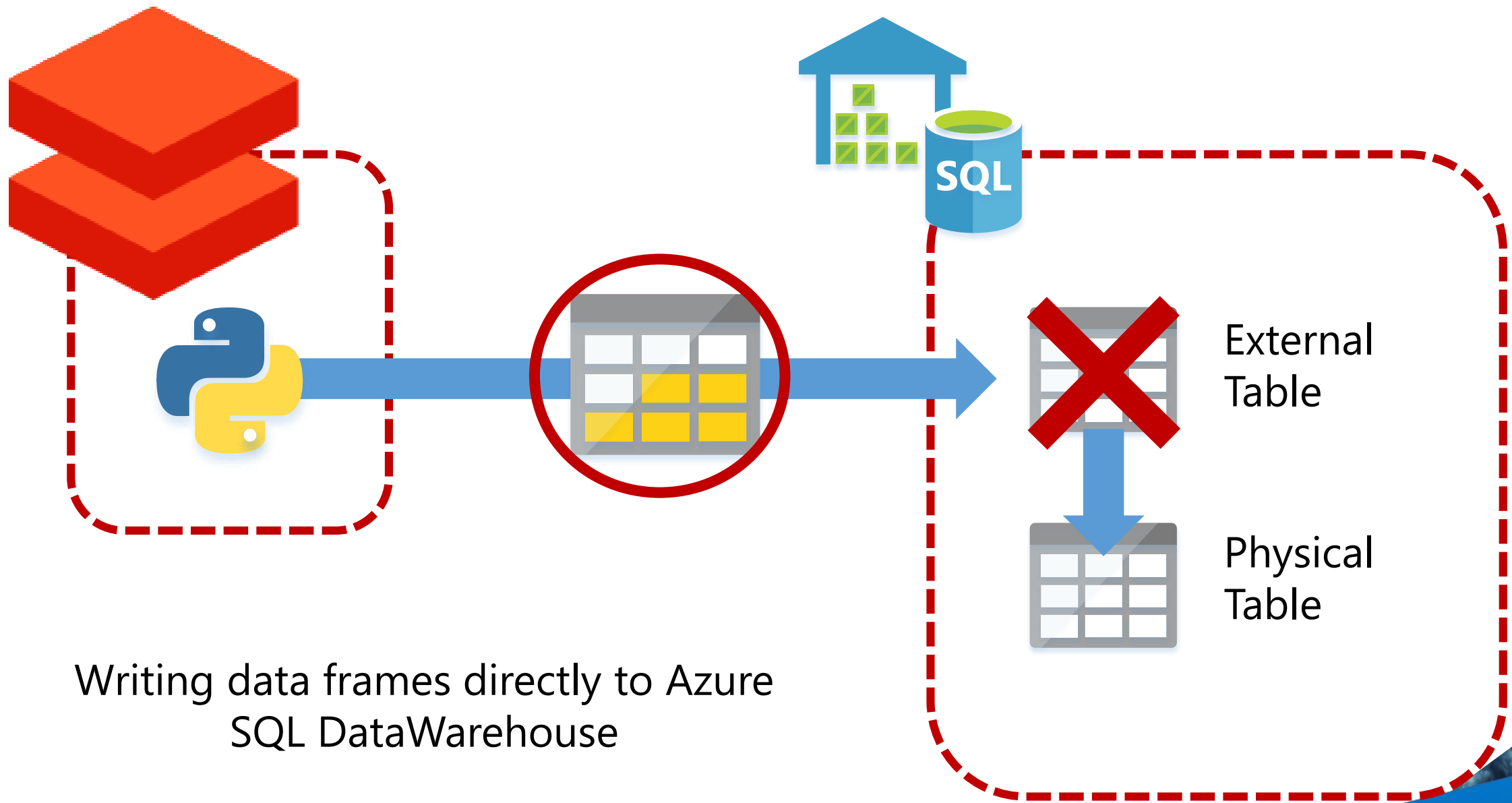




PYTHON PIPELINE PRIMER




PYTHON PIPELINE PRIMER



SCHEMA ON READ – INFER SCHEMA

Cmd 3

```
1 df = sqlContext.read.format("csv") \  
2   .option("header", "true") \  
3   .option("inferSchema", "true") \  
4   .load("abfss://root@dblake.dfs.core.windows.net/RAW/Public/Taxi/v1/SmallSlice.csv")
```

▼  df: pyspark.sql.dataframe.DataFrame

- Dispatching_base_num: string
- Pickup_DateTime: timestamp
- DropOff_datetime: string
- PUlocationID: integer
- DOlocationID: string



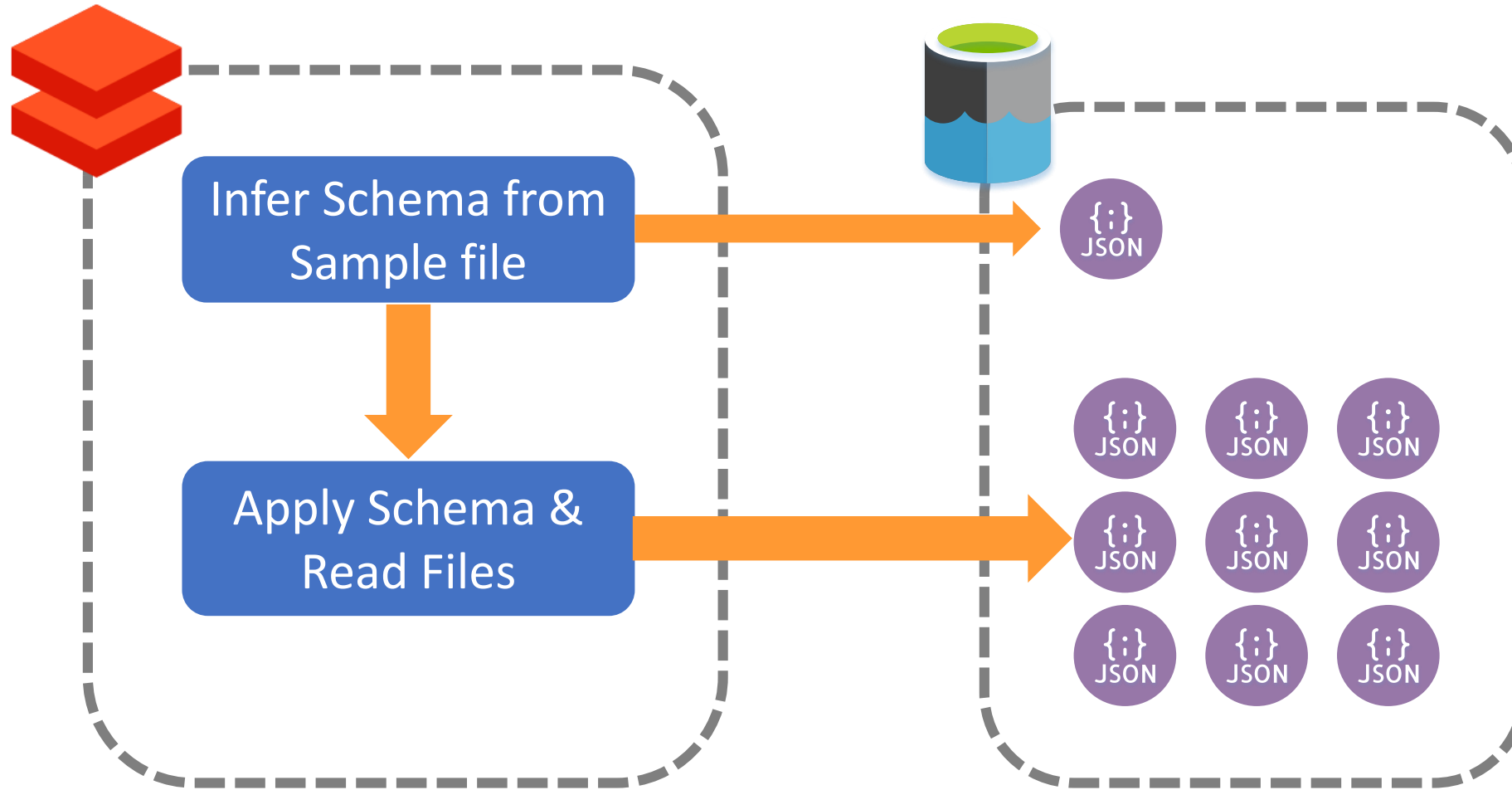
If the “inferSchema” option is used,
Spark reads a sample of each file
before creating a schema definition
for the dataset

SCHEMA ON READ – INFER SCHEMA



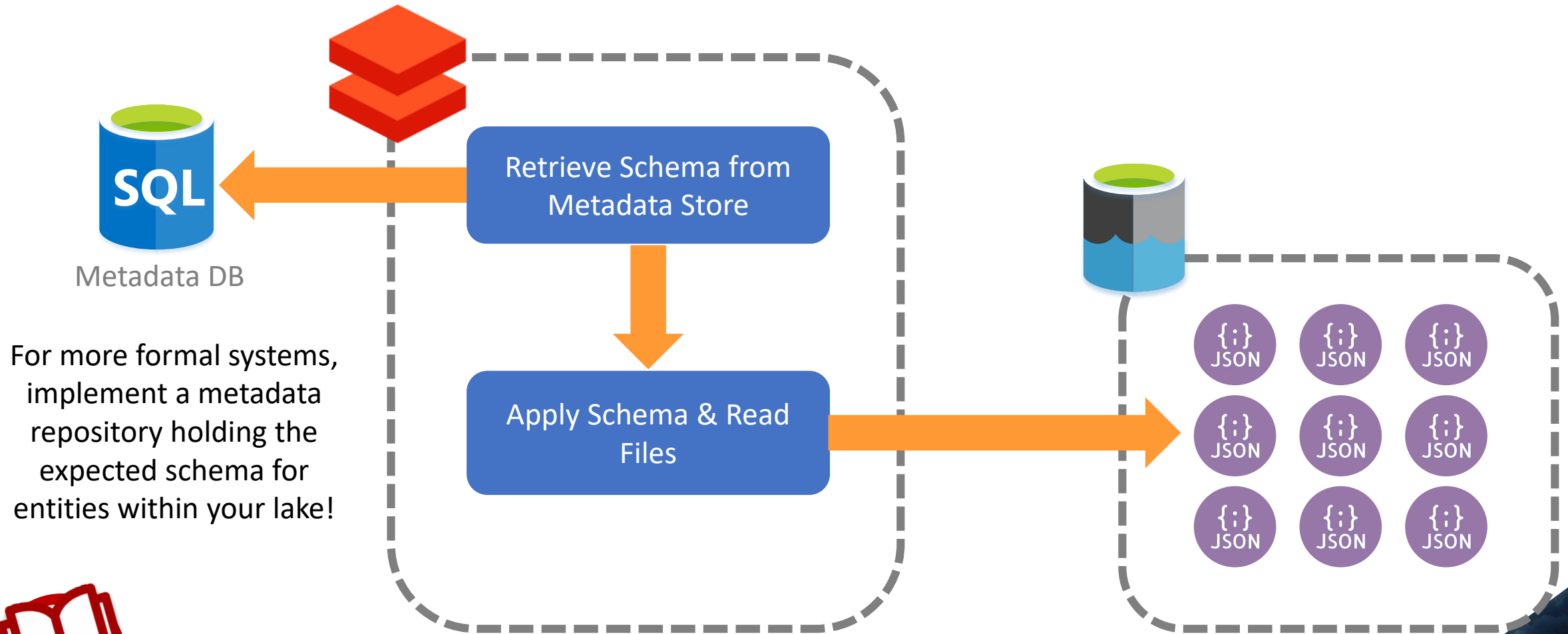
Inferring schema across very large, distributed datasets can perform very badly!

SCHEMA ON READ – SAMPLE FILES



Acquire schema metadata by inferring schema from a small file sample before reading large datasets

SCHEMA ON READ – METADATA STORE



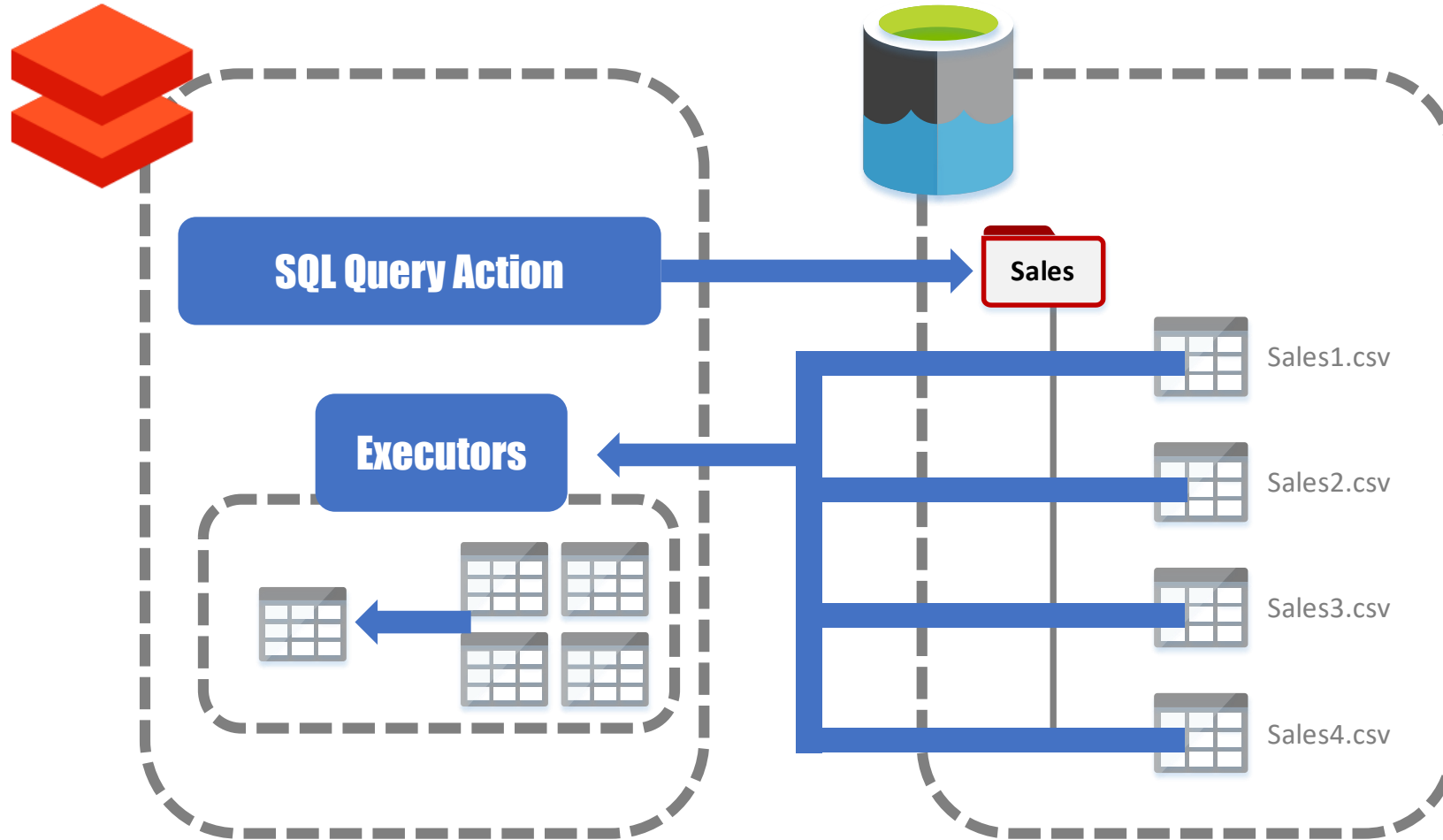
For more formal systems, implement a metadata repository holding the expected schema for entities within your lake!



Don't Forget – some file types have the schema metadata built into them! Files such as Parquet don't need a schema to be supplied!

READING FILES – NO PARTITIONS

*SELECT * FROM MyFiles WHERE Year = 2019 AND Month = 3*



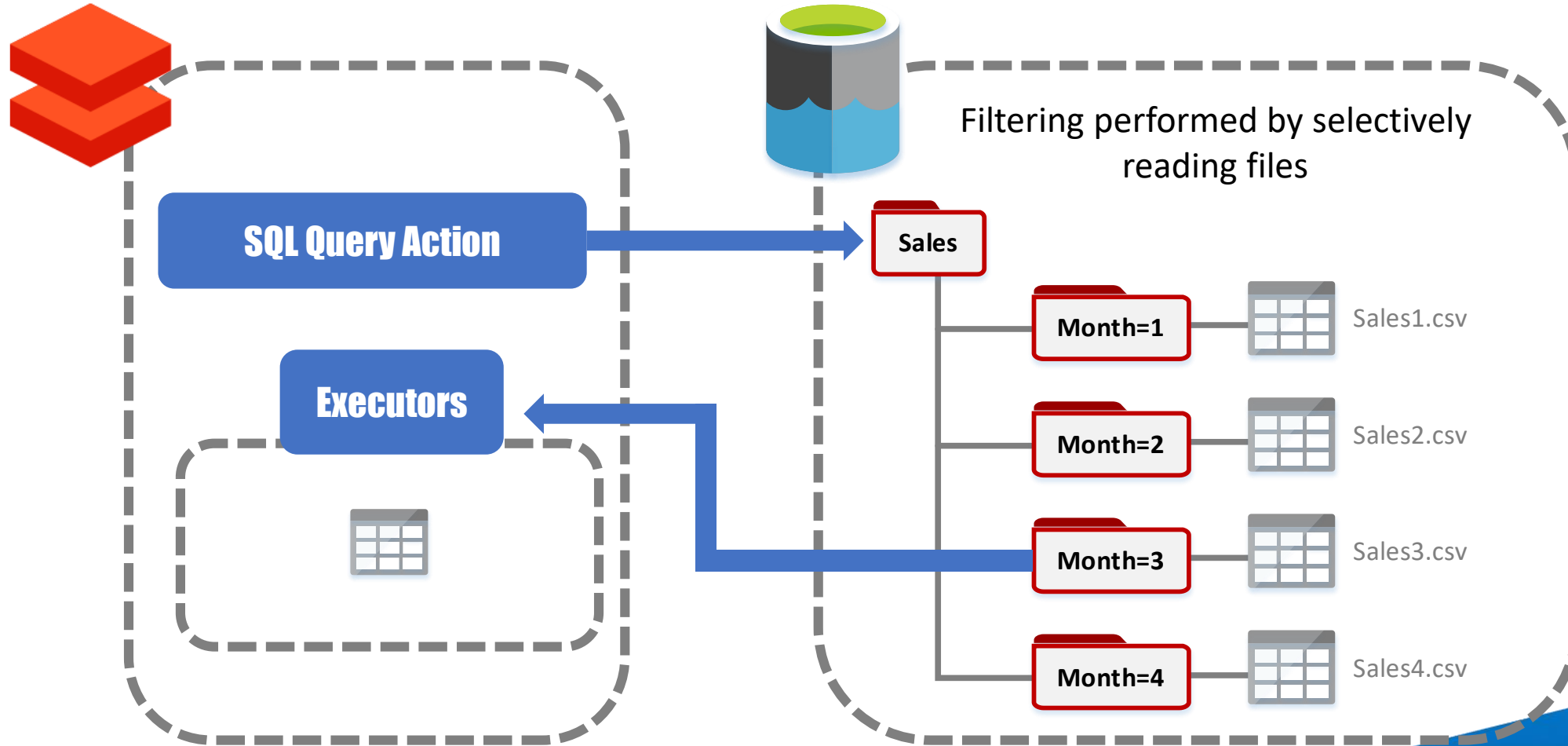
Filtering performed on executors
after reading all files

Databricks has no knowledge of what each file contains, so if you try to filter on a specific column, it will read all files then filter the DataFrame in memory

MODERN DATA WAREHOUSING

READING FILES – PARTITIONED

*SELECT * FROM MyFiles WHERE Year = 2019 AND Month = 3*

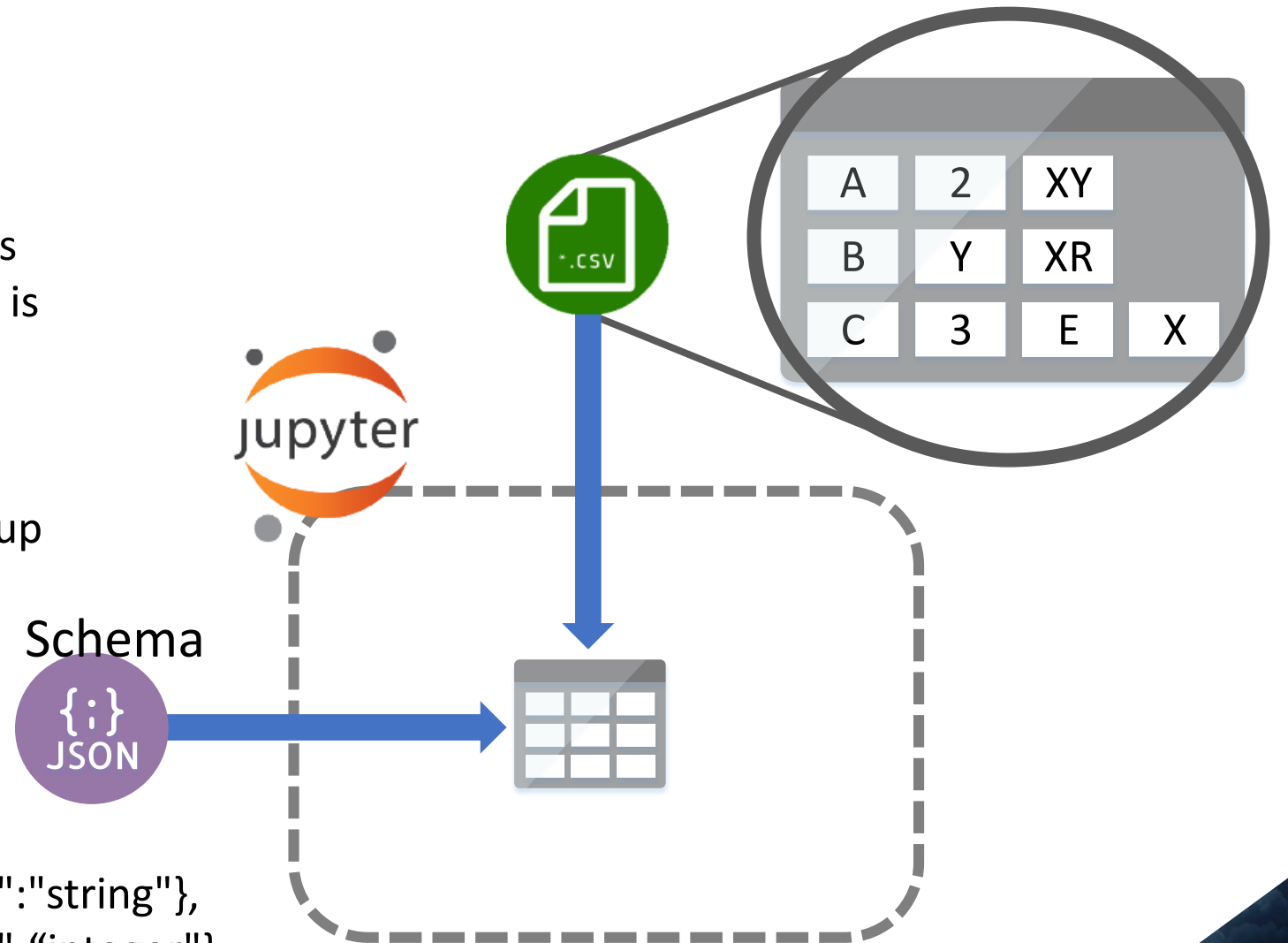


DATA VALIDATION

With Schema-On-Read file types such as CSV, we need to make sure that the data is in the format that we expect it to be.

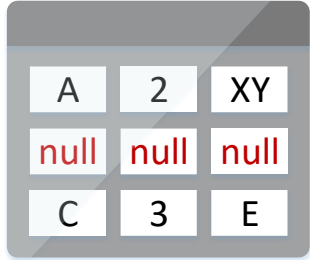
Let's assume that we're specifying the schema ourselves, from an external lookup

```
{"fields":  
  {"name":"Col1","nullable":true,"type":"string"},  
  {"name":"Col2","nullable":true,"type":"integer"},  
  {"name":"Col3","nullable":true,"type":"string"}  
}]
```



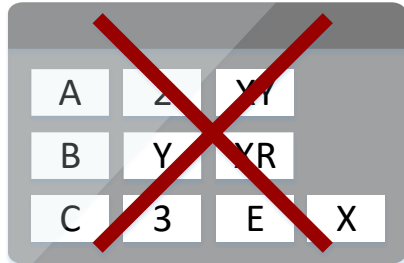
DATA VALIDATION

We have three different methods for handling failed parsing of a DataFrame when accessing text datasets such as csv:



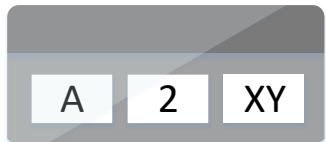
A	2	XY
null	null	null
C	3	E

- 1 **PERMISSIVE** – Extra columns are simply ignored, if any column fails to parse the entire row is nullified



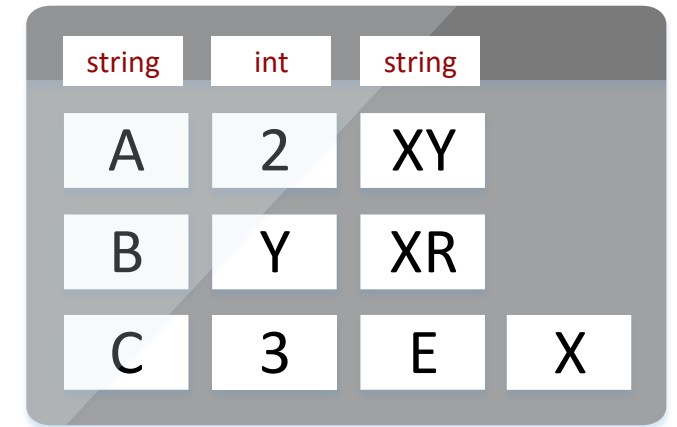
A	2	XY	
B	Y	XR	
C	3	E	X

- 2 **FAILFAST** – If any attributes are different to the specified schema, whether by failing to parse datatypes or attributes added/missing, the entire DataFrame will fail to load



A	2	XY
---	---	----

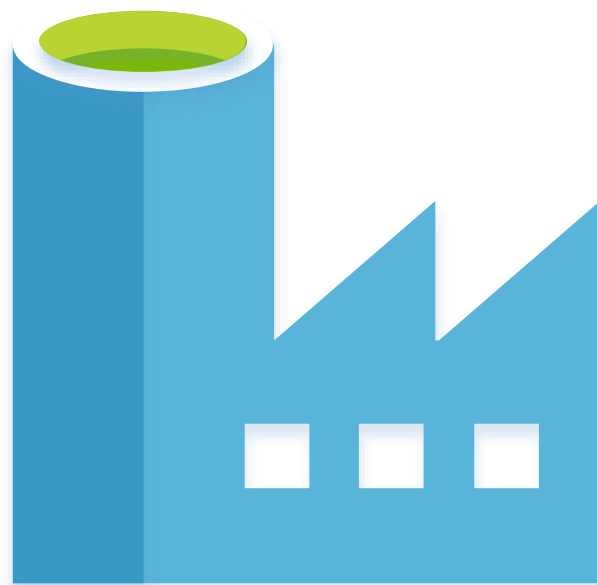
- 3 **DROPMALFORMED** – Any rows that differ from the schema will be silently dropped from the dataset, also known as the “Nothing to see here” approach to ETL...



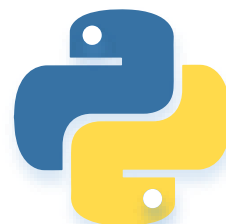
string	int	string	
A	2	XY	
B	Y	XR	
C	3	E	X

The background of the image is a dramatic, high-contrast photograph of dark, swirling clouds. A bright, glowing light source, possibly the sun or moon, is partially visible through the clouds in the upper center, creating a strong backlighting effect. A solid blue diagonal banner cuts across the middle of the image, providing a contrasting background for the text.

ORCHESTRATION



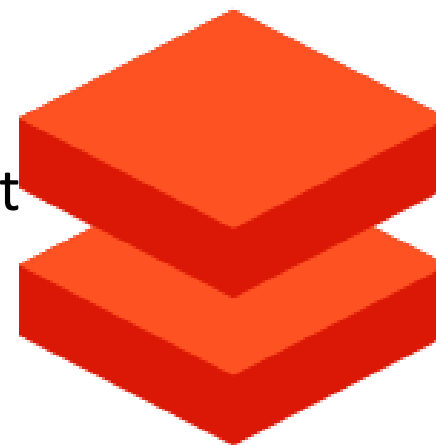
Jupyter Notebook



Python Script

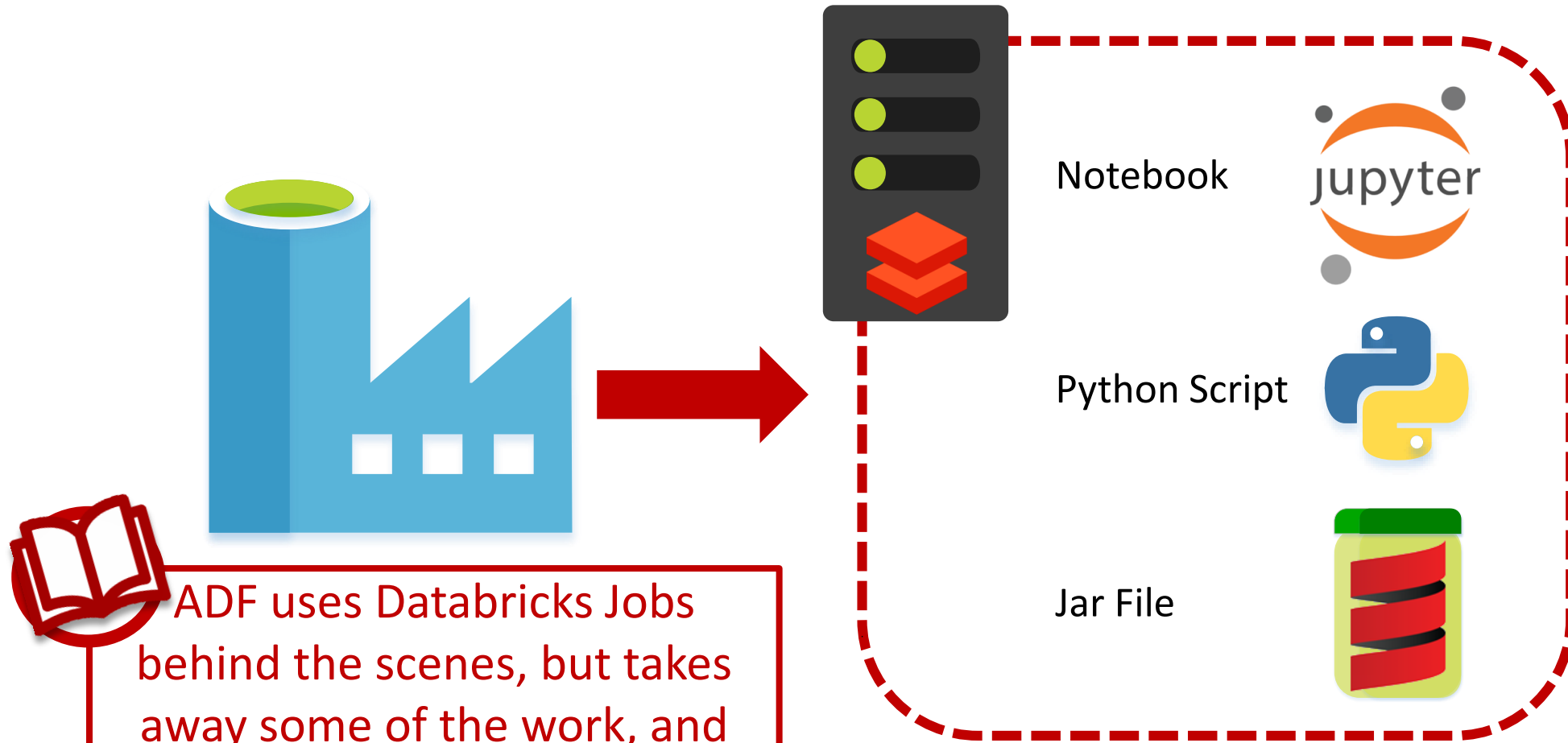


Jar File



PYTHON PIPELINE PRIMER

AZURE DATA FACTORY



ADF uses Databricks Jobs behind the scenes, but takes away some of the work, and means you can orchestrate your notebooks with other tasks!

PYTHON PIPELINE PRIMER

ANY QUESTIONS?



Simon Whiteley
@MrSiWhiteley