



Microsoft Big Data Lab

Advanced Analytics with Databricks

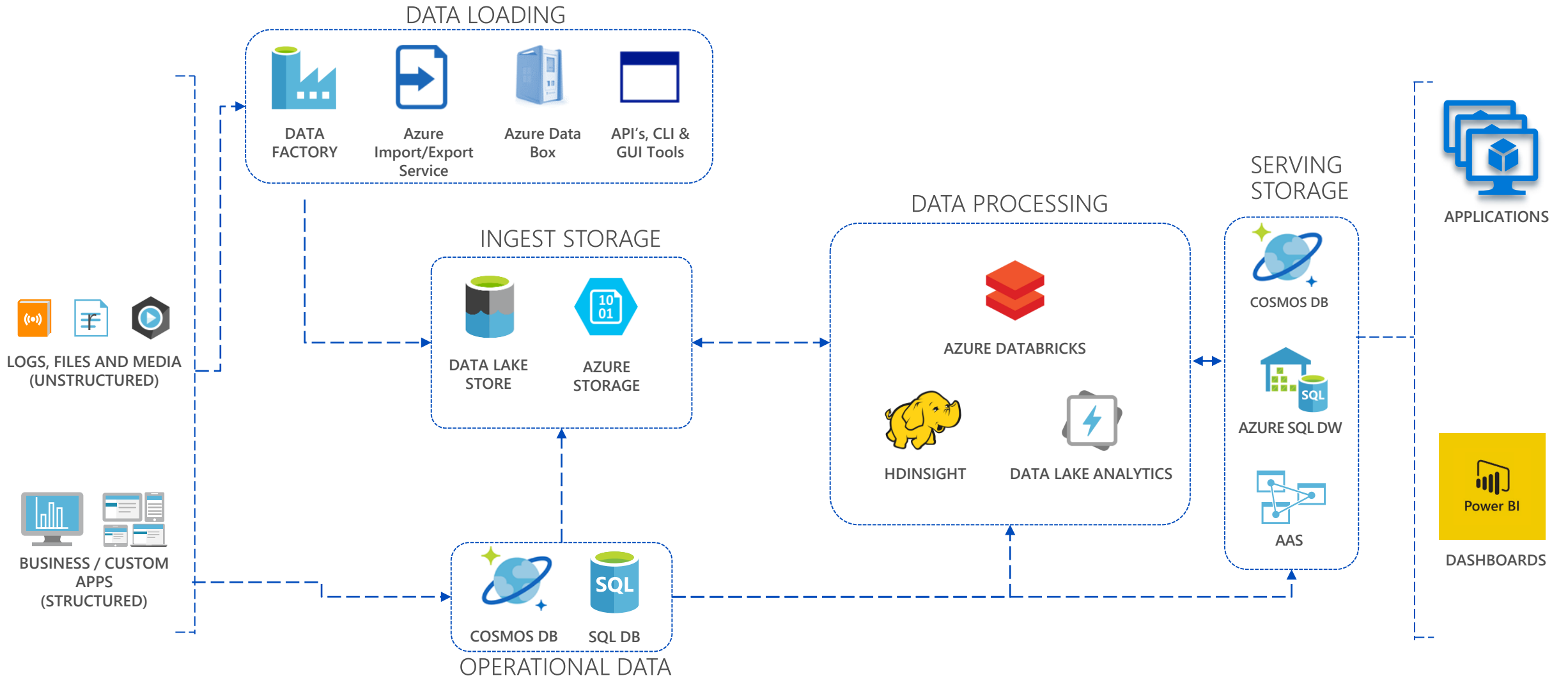
Anaig Marechal
Data Scientist

Geneva – 09.04.2019

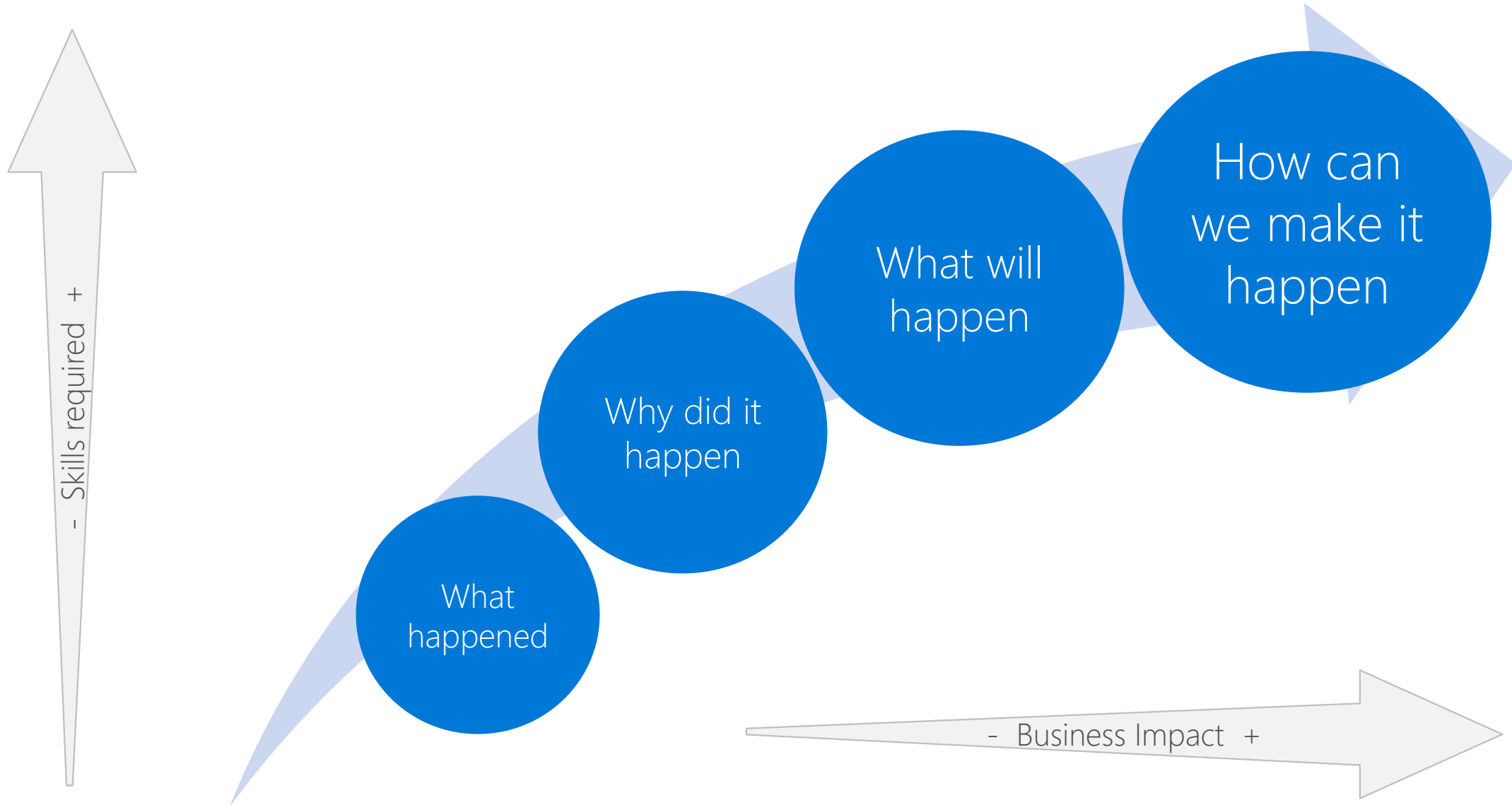


DATA WAREHOUSING PATTERN IN AZURE

Loading and preparing data for analysis with a data warehouse

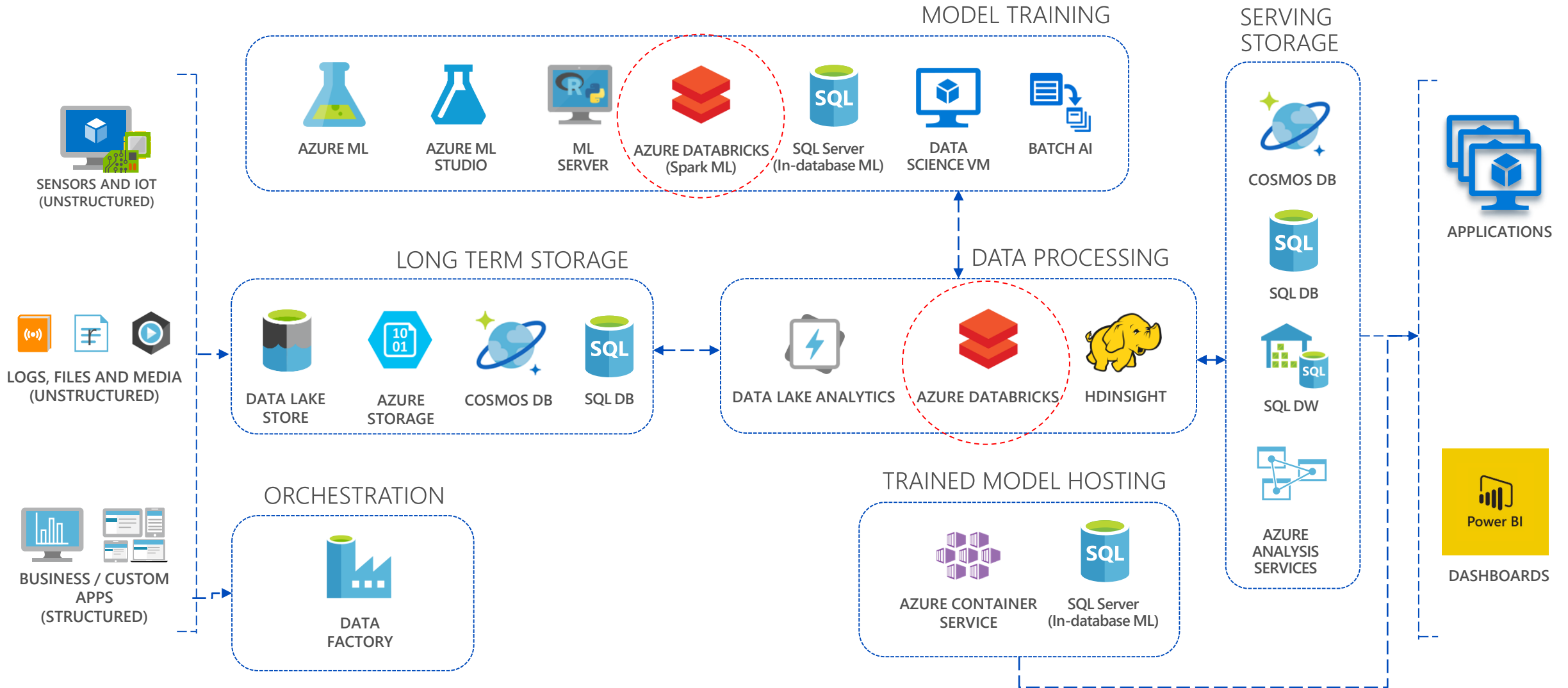


From looking back to looking forward

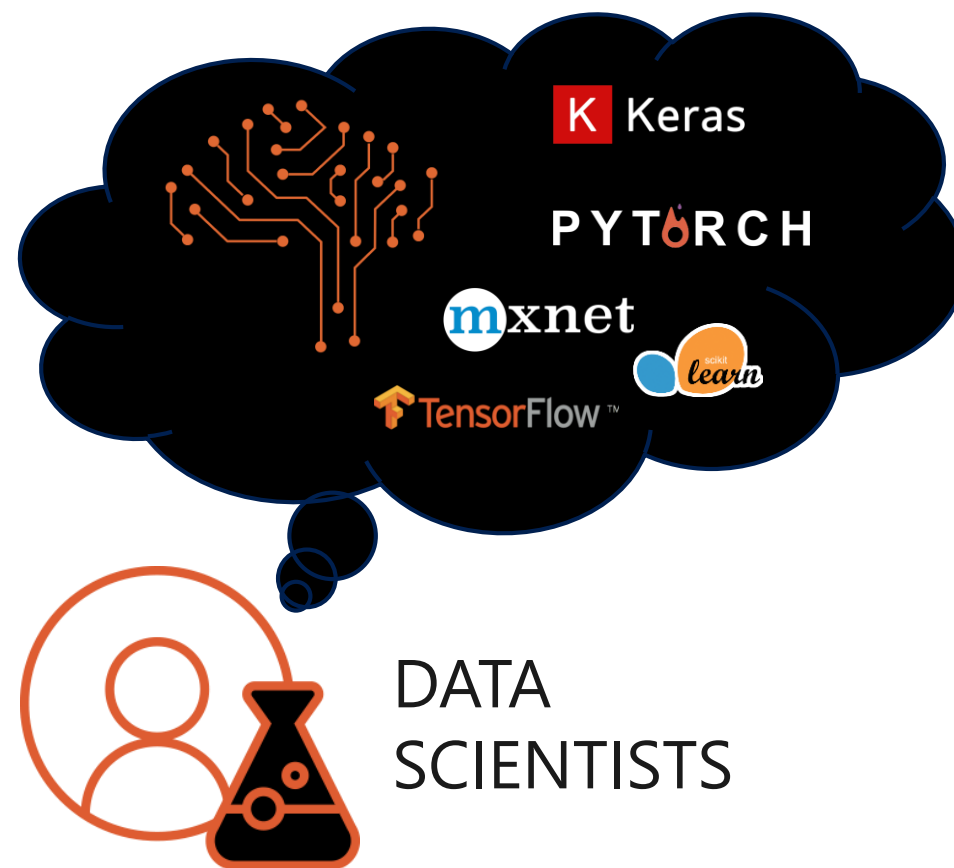
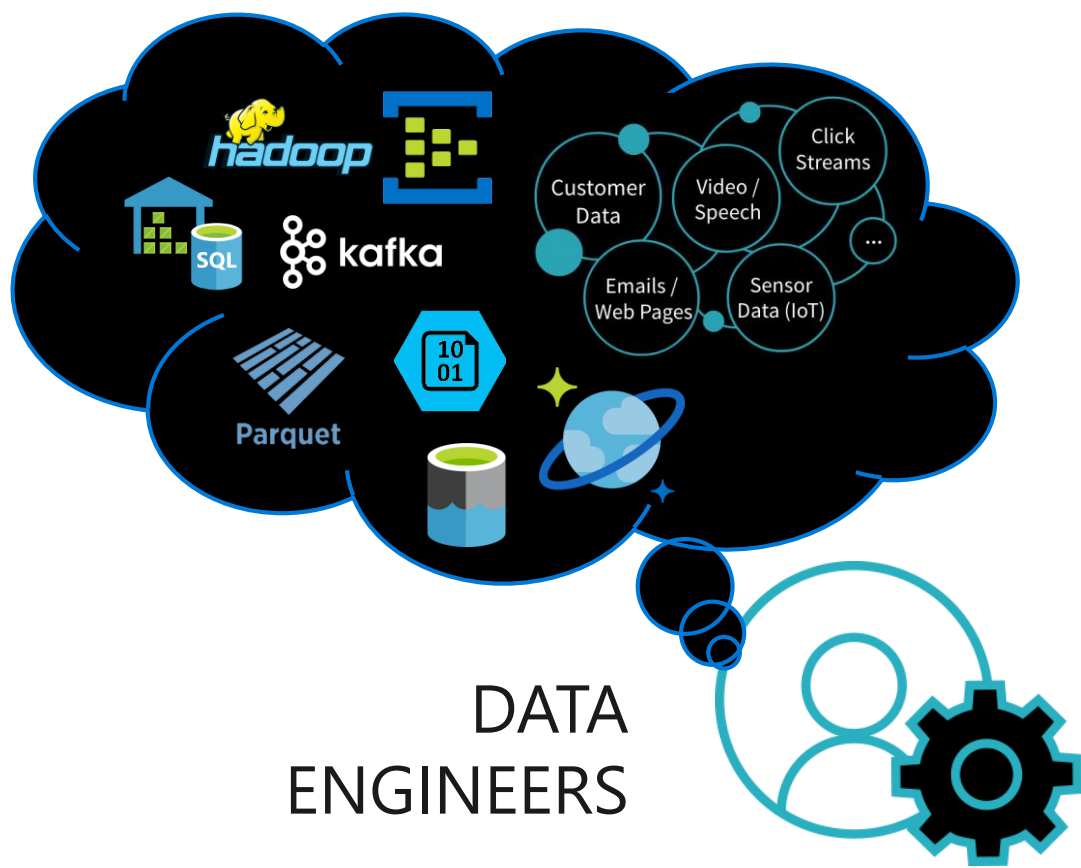


ADVANCED ANALYTICS PATTERN IN AZURE

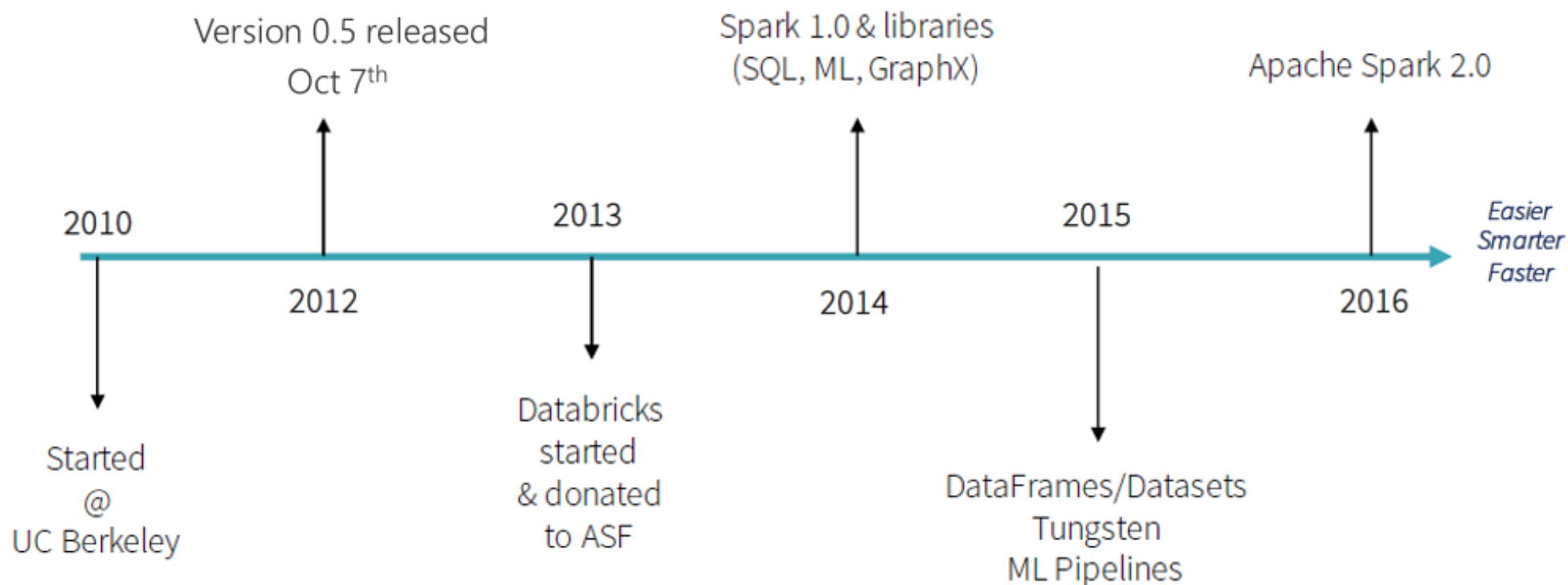
Performing data collection/understanding, modeling and deployment



Data & AI People are in Silos



SPARK: A BRIEF HISTORY

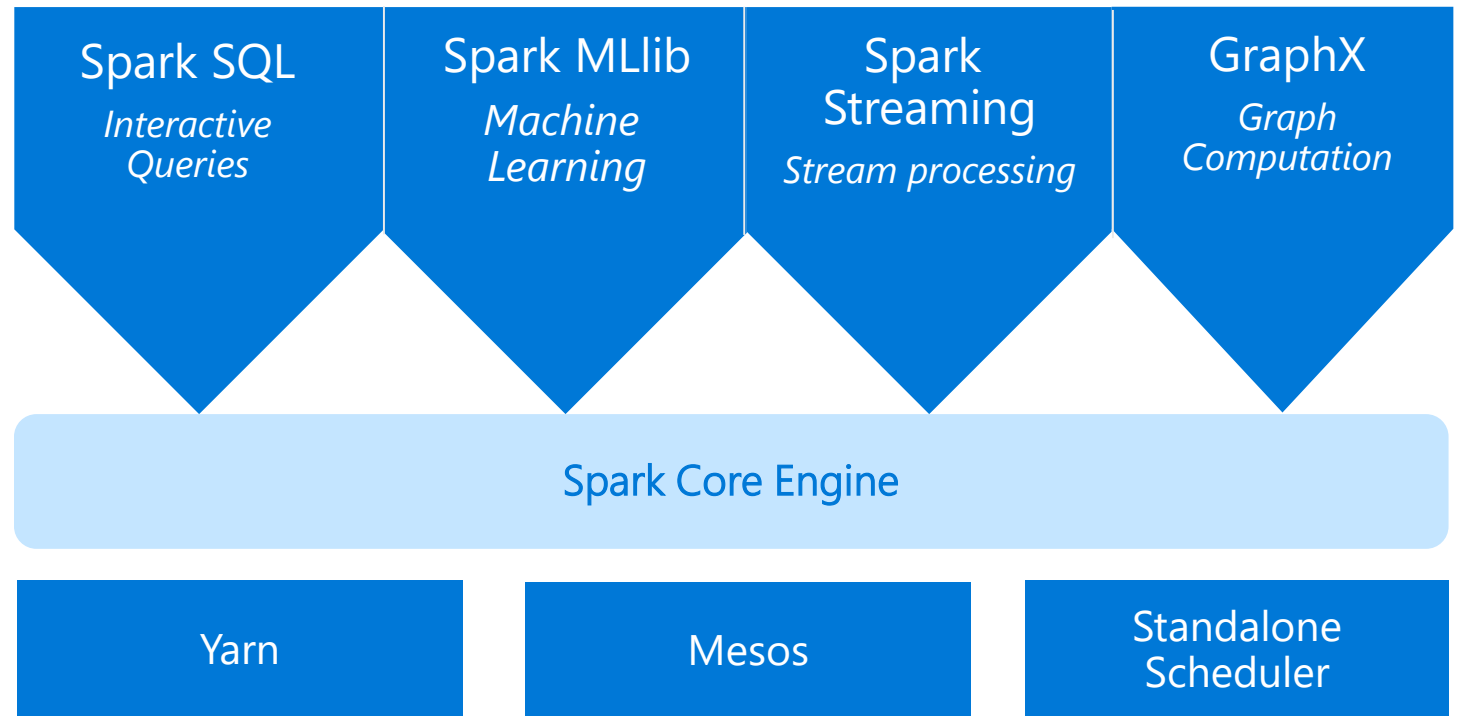


A P A C H E S P A R K

An unified, open source, parallel, data processing framework for Big Data Analytics

Spark Unifies:

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



Azure Databricks

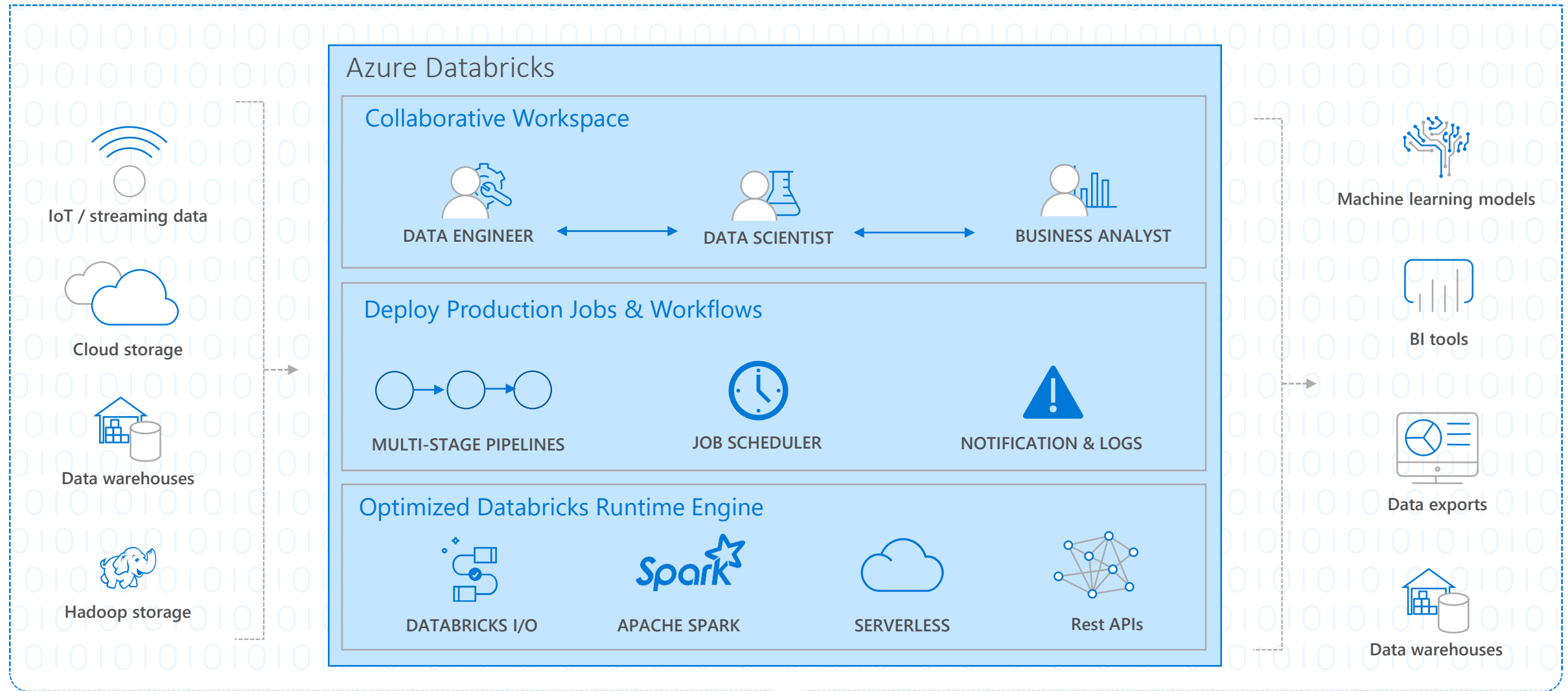
Databricks Spark as a managed service on Azure

A Z U R E D A T A B R I C K S

- Azure Databricks is a **first party** service on Azure.
 - Unlike with other clouds, it is not an Azure Marketplace or a 3rd party hosted service.
- Azure Databricks is integrated seamlessly with Azure services:
 - [Azure Portal](#): Service can be launched directly from Azure Portal
 - [Azure Storage Services](#): Directly access data in Azure Blob Storage and Azure Data Lake Store
 - [Azure Active Directory](#): For user authentication, eliminating the need to maintain two separate sets of users in Databricks and Azure.
 - [Azure SQL DW and Azure Cosmos DB](#): Enables you to combine structured and unstructured data for analytics
 - [Apache Kafka for HDInsight](#): Enables you to use Kafka as a streaming data source or sink
 - [Azure Billing](#): You get a single bill from Azure
 - [Azure Power BI](#): For rich data visualization
- Eliminates need to create a separate account with Databricks.



A Z U R E D A T A B R I C K S

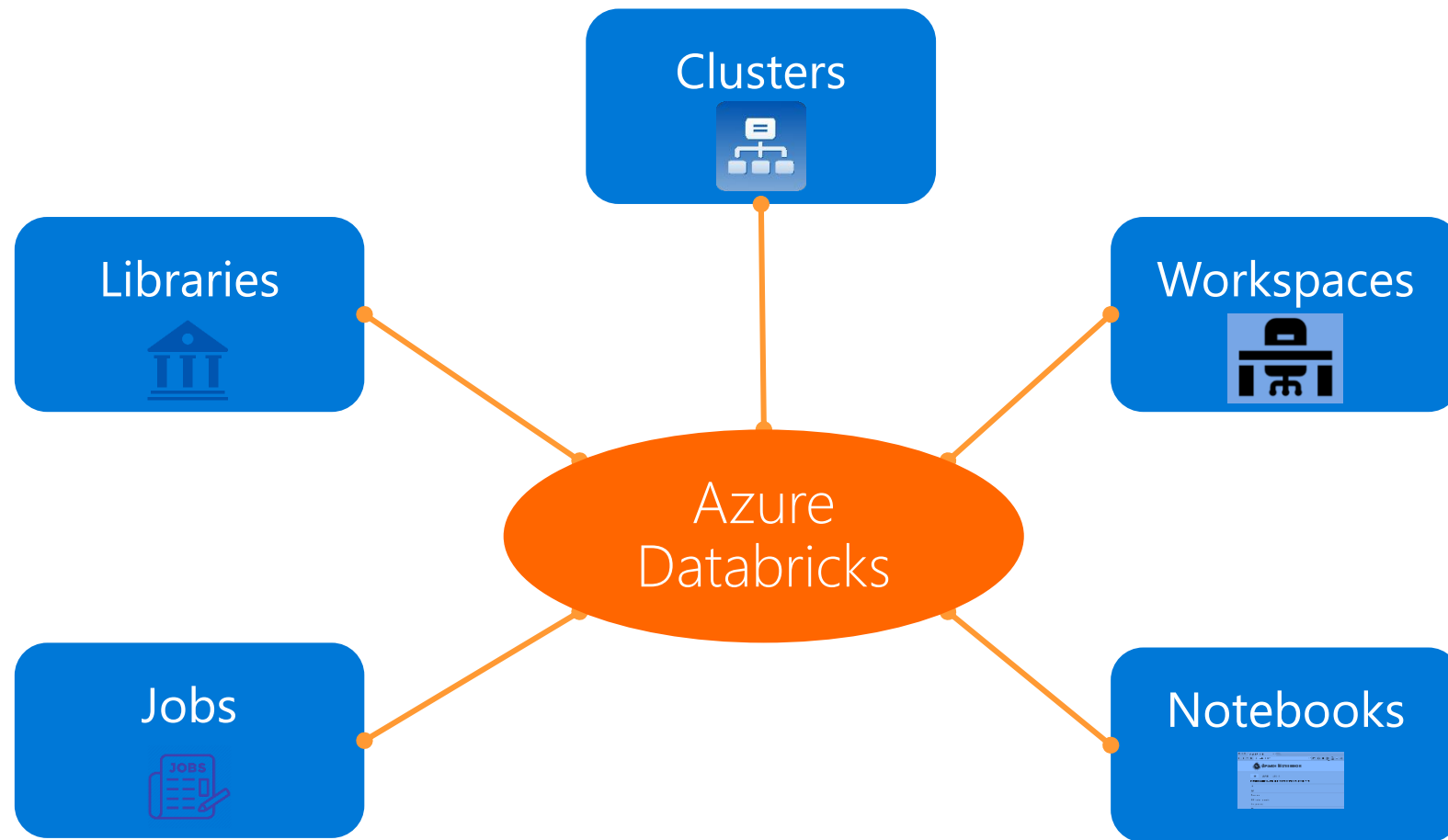


Enhance Productivity

Build on secure & trusted cloud

Scale without limits

AZURE DATABRICKS CORE ARTIFACTS

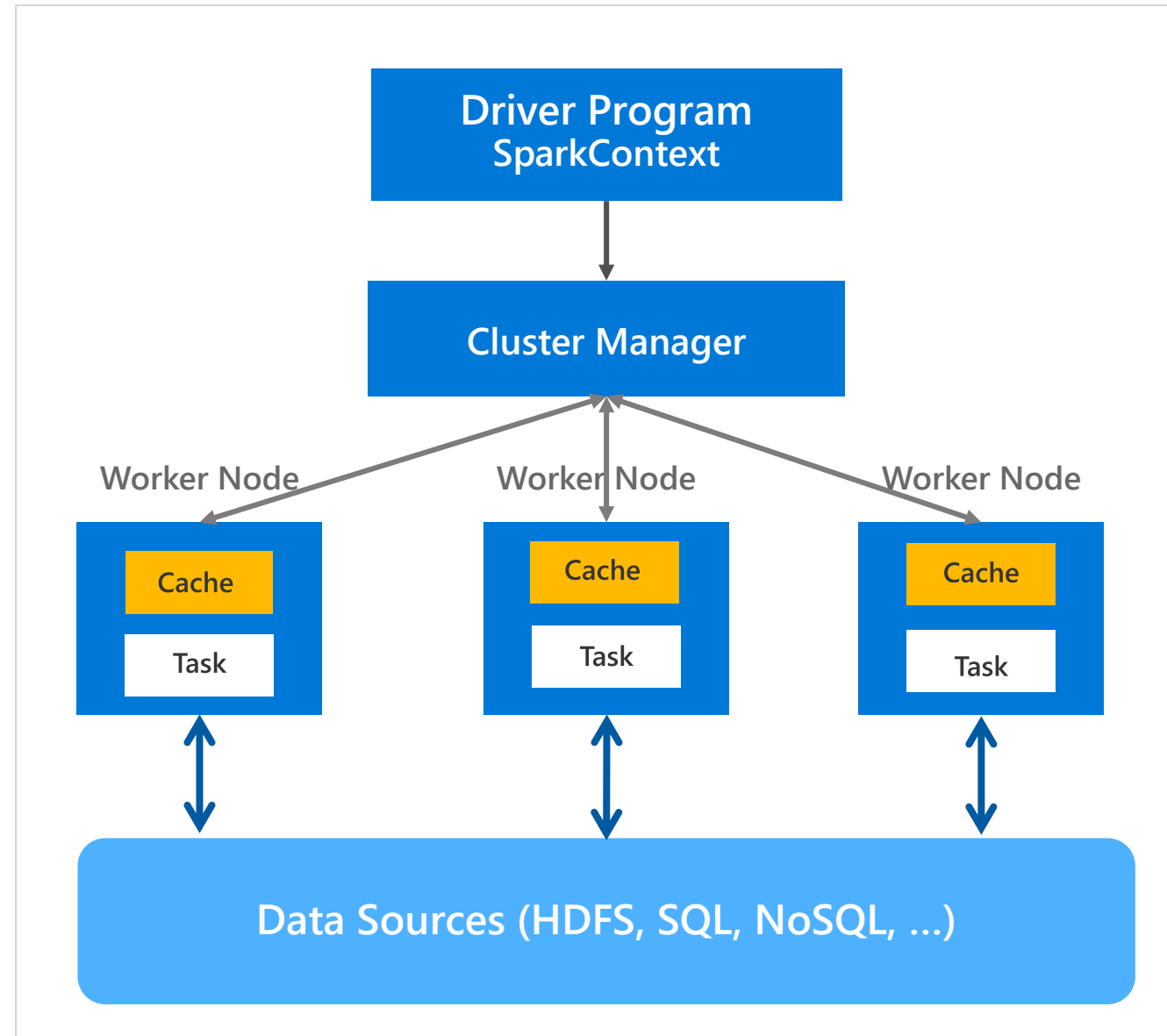


Azure Databricks

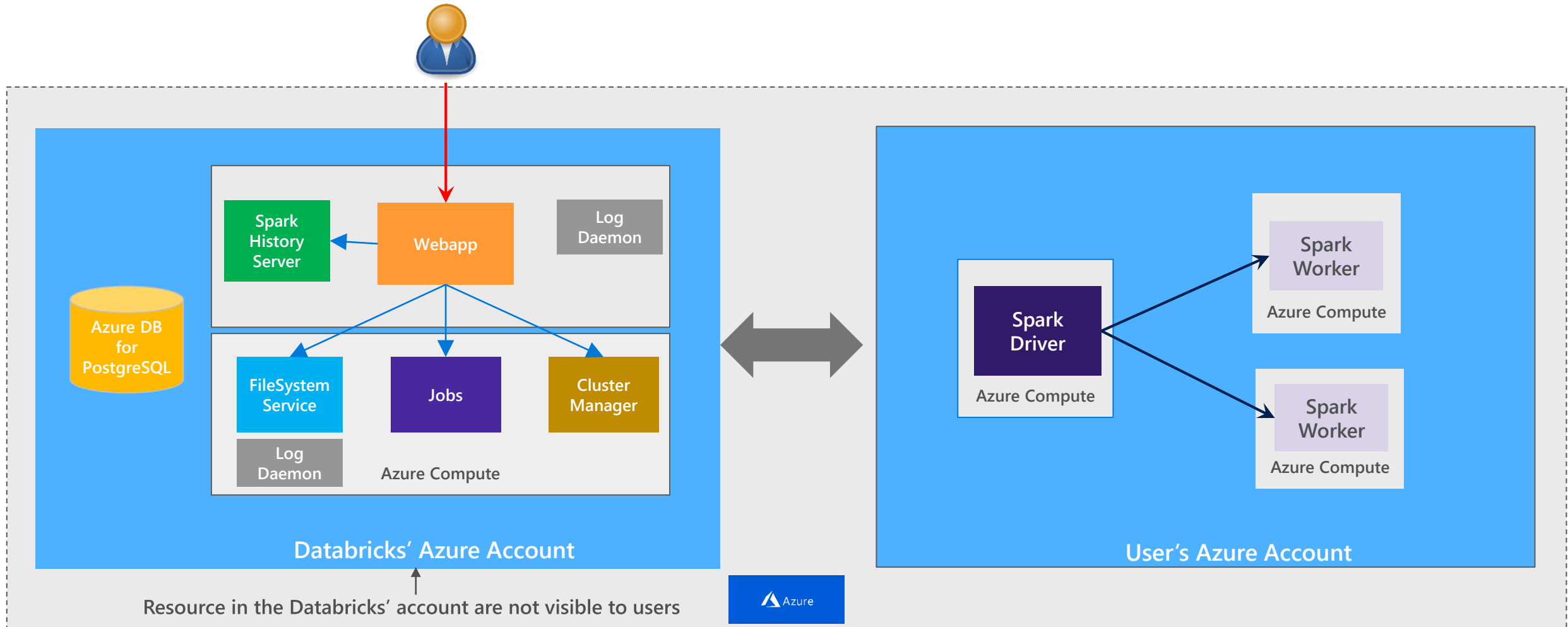
Core Concepts

GENERAL SPARK CLUSTER ARCHITECTURE

- 'Driver' runs the user's 'main' function and executes the various parallel operations on the worker nodes.
- The results of the operations are collected by the driver
- The worker nodes read and write data from/to Data Sources including HDFS.
- Worker node also cache transformed data in memory as RDDs (Resilient Data Sets).
- Worker nodes and the Driver Node execute as VMs in public clouds (AWS, Google and Azure).



AZURE DATABRICKS CLUSTER ARCHITECTURE



CLUSTER CREATION

- While creating a cluster you can specify:
 - Number of nodes
 - Autoscaling and Auto Termination policy
 - Auto Termination policy
 - Spark Configuration details
 - The Azure VM instance types for the Driver and Worker Nodes

General Purpose	
Standard_D3_v2 (beta)	14.0 GB Memory, 4 Cores
✓ Standard_DS3_v2 (beta)	14.0 GB Memory, 4 Cores
Standard_DS4_v2 (beta)	28.0 GB Memory, 8 Cores
Standard_DS5_v2 (beta)	56.0 GB Memory, 16 Cores
Standard_D4s_v3 (beta)	16.0 GB Memory, 4 Cores
Standard_D8s_v3 (beta)	32.0 GB Memory, 8 Cores
Standard_D16s_v3 (beta)	64.0 GB Memory, 16 Cores
Memory Optimized	
Standard_DS11_v2 (beta)	14.0 GB Memory, 2 Cores
Standard_DS12_v2 (beta)	28.0 GB Memory, 4 Cores
Standard_DS13_v2 (beta)	56.0 GB Memory, 8 Cores
Standard_DS14_v2 (beta)	112.0 GB Memory, 16 Cores
Standard_DS15_v2 (beta)	140.0 GB Memory, 20 Cores
Standard_E4s_v3 (beta)	32.0 GB Memory, 4 Cores
Standard_F8s_v3 (beta)	64.0 GB Memory, 8 Cores

Microsoft Azure PORTAL

Create Cluster

2-8 Workers: 28.0-112.0 GB Memory, 8-32 Cores
1 Driver: 14.0 GB Memory, 4 Cores

New Cluster

Cluster Type
Serverless Pool (beta, Python/SQL) **Standard** [Learn more about Serverless Pools](#)

Cluster Name

Databricks Runtime Version
3.3 (includes Apache Spark 2.2.0, Scala 2.11)

Driver Type
Same as worker 14.0 GB Memory, 4 Cores

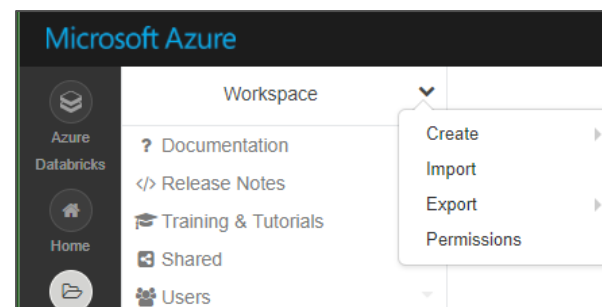
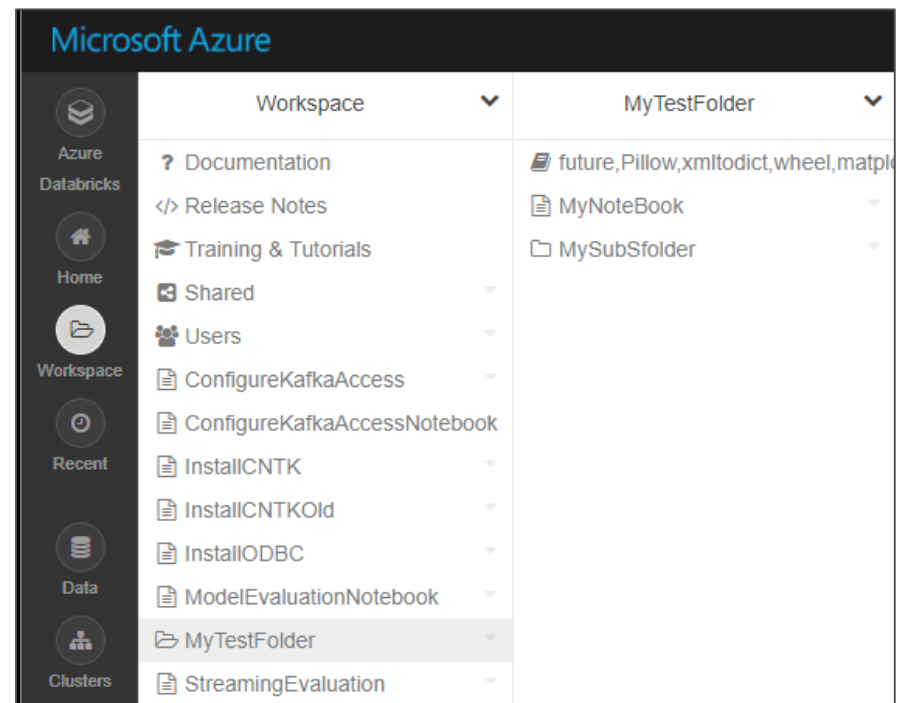
Worker Type
Standard_DS3_v2 (beta) 14.0 GB Memory, 4 Cores

Graphical wizard in the Azure Databricks portal to create a Standard Cluster

WORKSPACES


Workspaces enables users to organize—and share—their Notebooks, Libraries and Dashboards

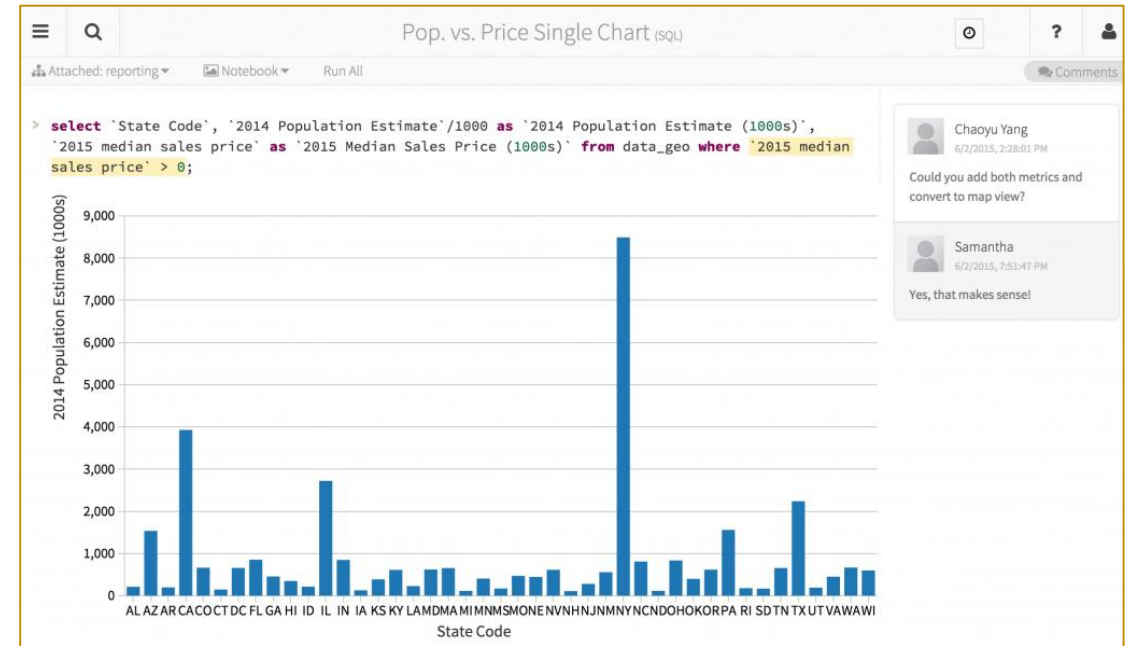
- Workspaces—sort of like Directories—are a convenient way to organize an user's Notebook, Libraries and Dashboards.
- Everything in a workspace is organized into hierarchical folders. Folders can hold Libraries, Notebooks, Dashboard or more (sub) folders.
 - Icons indicate the type of the object contained in a folder
- Every user has one directory that is private and unshared.
 - By default, the workspace and all its contents are available to users.
- Fine grained access control can be defined on workspaces (next slide) to enable *secure collaboration with colleagues*.



A Z U R E D A T A B R I C K S N O T E B O O K S O V E R V I E W

Notebooks are a popular way to develop, and run, Spark Applications

- Notebooks are not only for authoring Spark applications but can be *run/executed directly* on clusters
 - **Shift+Enter**
 - click the  at the top right of the cell in a notebook
 - Submit via Job
- Notebooks support fine grained permissions—so they can be *securely shared* with colleagues for collaboration (see following slide for details on permissions and abilities)
- Notebooks are well-suited for prototyping, rapid development, exploration, discovery and iterative development



Notebooks typically consist of code, data, visualization, comments and notes

MIXING LANGUAGES IN NOTEBOOKS

You can mix multiple languages in the same notebook

Normally a notebook is associated with a specific language. However, with Azure Databricks notebooks, you can mix multiple languages in the same notebook. This is done using the language magic command:

- `%python` Allows you to execute python code in a notebook (even if that notebook is not python)
- `%sql` Allows you to execute sql code in a notebook (even if that notebook is not sql).
- `%r` Allows you to execute r code in a notebook (even if that notebook is not r).
- `%scala` Allows you to execute scala code in a notebook (even if that notebook is not scala).
- `%sh` Allows you to execute shell code in your notebook.
- `%fs` Allows you to use Databricks Utilities - dbutils filesystem commands.
- `%md` To include rendered markdown

LIBRARIES OVERVIEW

Enables external code to be imported and stored into a Workspace

- Libraries are containers to hold all your *Python, R, Java/Scala* libraries.
- Libraries resides within workspaces or folders.
- Libraries are created by importing the source code
- After importing libraries are immutable—can be deleted or overwritten only.
- You can customize installation of libraries via [Init Scripts](#) by writing custom UNIX scripts
- Libraries can also be managed via the [Library API](#)

This screenshot shows the 'Create Library' page in the Microsoft Azure portal, specifically for creating a new Python library. The 'Language' dropdown is set to 'Upload Python Egg or PyPi'. Under the 'Install PyPi Package' section, there is a 'PyPi Name' field with the example 'simplejson==3.8.0' and an 'Install Library' button. The 'Upload Egg' section has a 'Library Name' field and a large 'Egg File' upload area with the instruction 'Drop library egg here to upload' and a 'Create Library' button at the bottom.

This screenshot shows the 'Create Library' page for creating a new R library. The 'Source' dropdown is set to 'R Library'. The 'Install from' dropdown is set to 'CRAN-like Repository', and the 'Repository' field contains 'https://cloud.r-project.org'. There is a 'Package' field and a 'Create Library' button at the bottom.

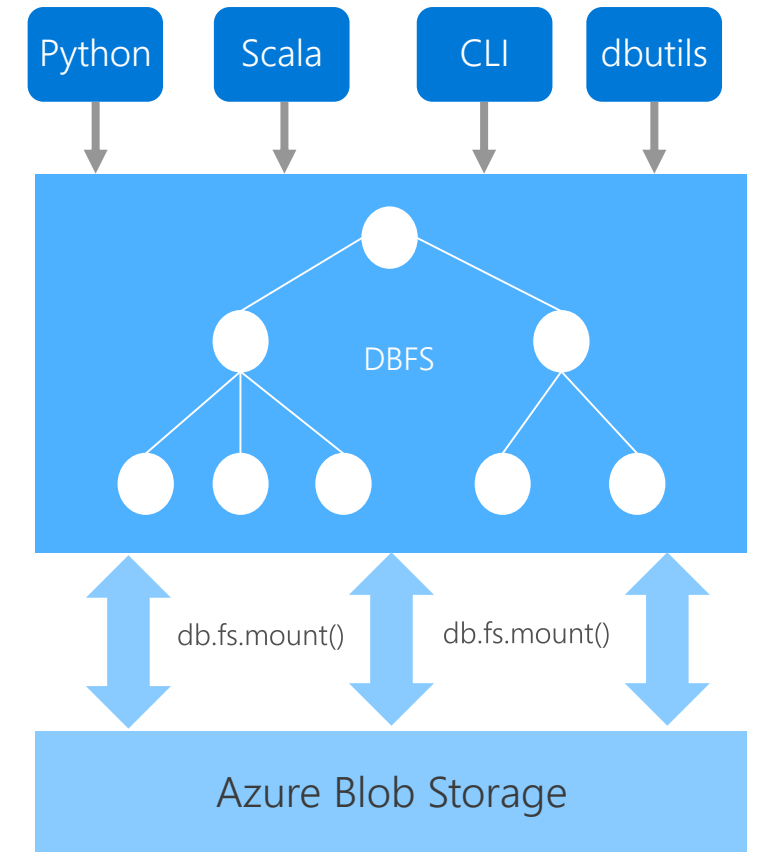
This screenshot shows the 'Create Library' page for creating a new Java/Scala library. The 'Source' dropdown is set to 'Upload Java/Scala JAR'. The 'Library Name' field contains 'My Library'. There is a 'JAR File' upload area with the instruction 'Drop library JAR here to upload' and a 'Create Library' button at the bottom.

This screenshot shows the 'Create Library' page for creating a new Maven library. The 'Source' dropdown is set to 'Maven Coordinate'. The 'Install Maven Artifacts' section has a 'Coordinate' field with the example 'com.databricks:spark-csv_2.10:1.0.0' and a 'Search Spark Packages and Maven Central' button. There is an 'Advanced Options' section and a 'Create Library' button at the bottom.

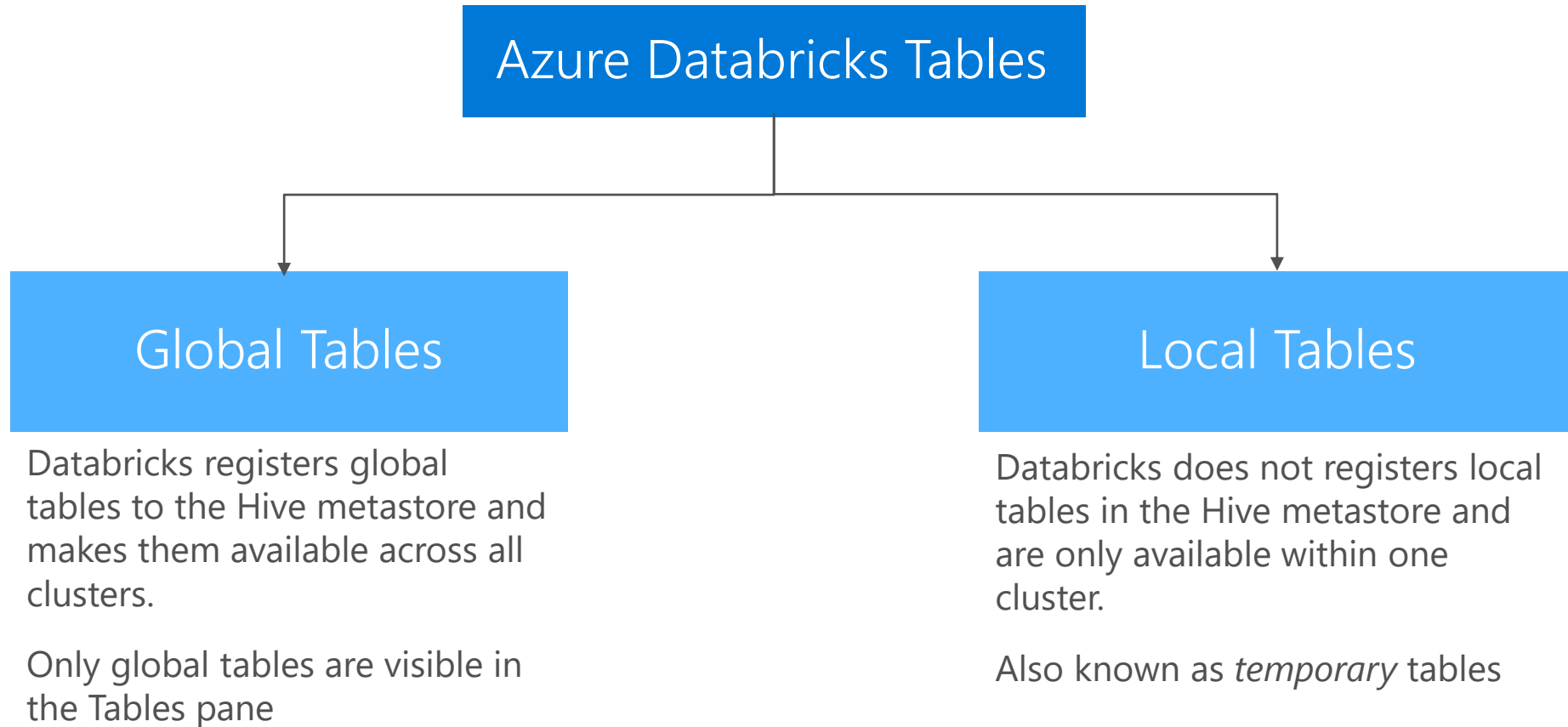
DATABRICKS FILE SYSTEM (DBFS)

Is a distributed File System (DBFS) that is a layer over Azure Blob Storage

- Azure Storage buckets can be mounted in DBFS so that users can directly access them without specifying the storage keys
- DBFS mounts are created using `dbutils.fs.mount()`
- Azure Storage data can be cached locally on the SSD of the worker nodes
- Available in both Python and Scala and accessible via a DBFS CLI
- Data persist in Azure Blob Storage – is not lost even after cluster termination
- Comes pre-installed on Spark clusters in Databricks



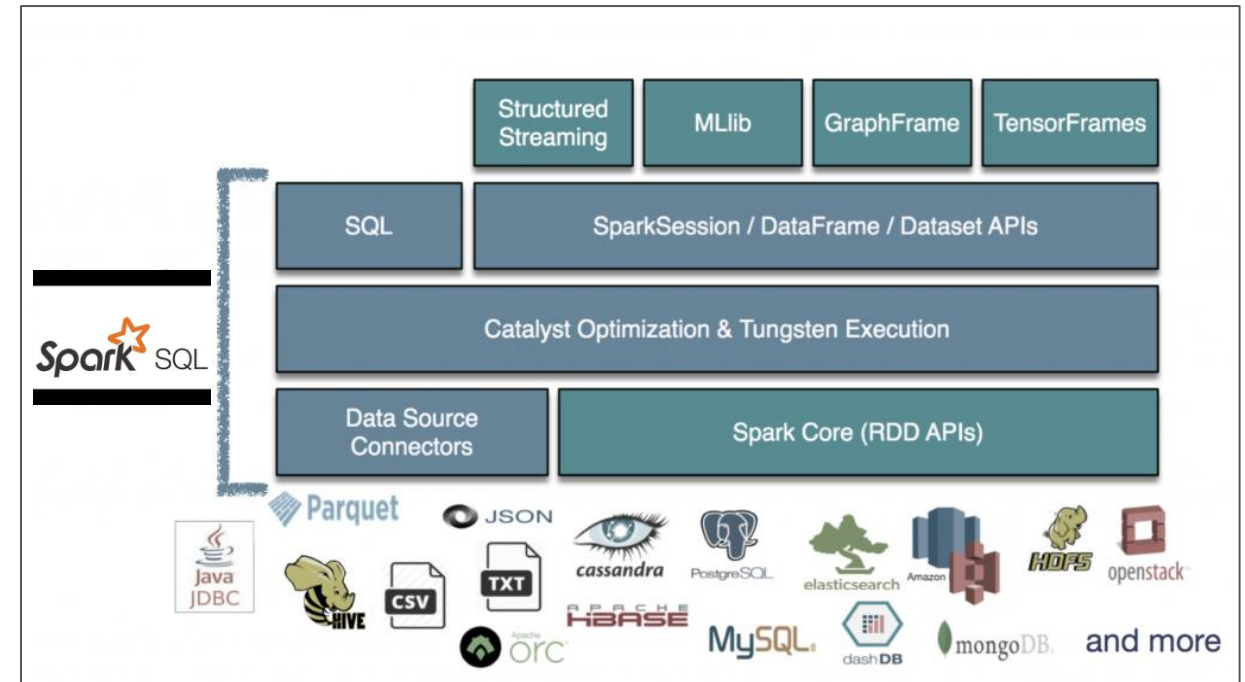
LOCAL AND GLOBAL TABLES



SPARK SQL OVERVIEW

Spark SQL is a distributed SQL query engine for processing structured data

- Can query data stored in wide variety of data sources—external databases, structured data files, Hive tables and more.
- Data can be queried using either SQL or HiveQL
- Has bindings in Python, Scala and Java
- Has built-in support for structured streaming.
- Built using the [Catalyst optimizer](#) and [Tungsten execution](#)



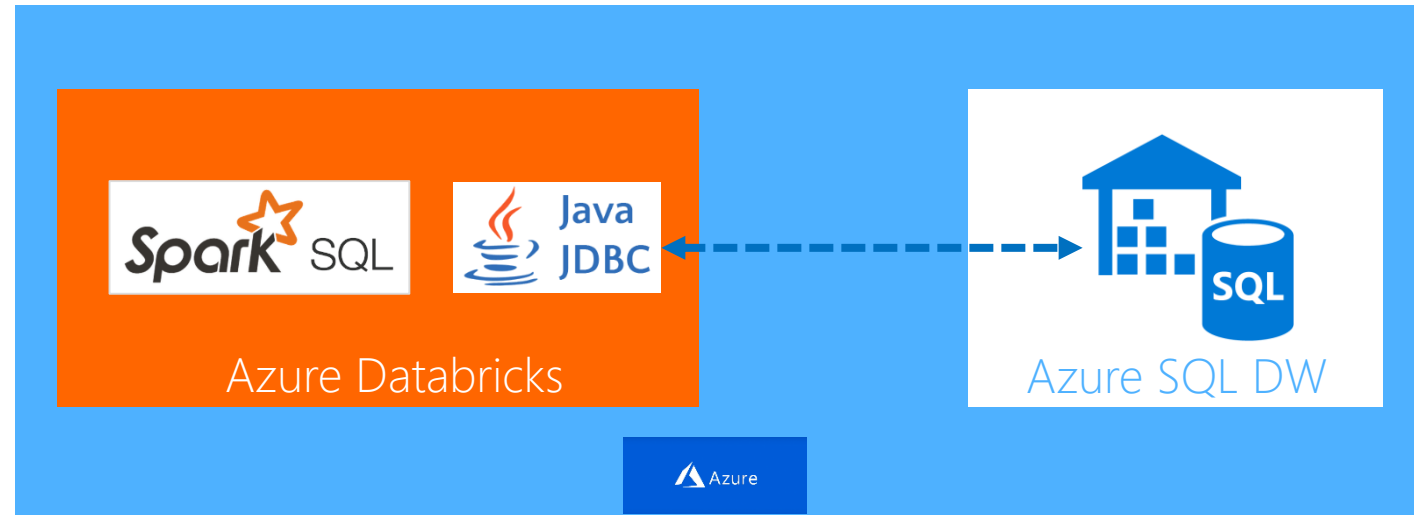
A Z U R E S Q L D W I N T E G R A T I O N

Integration enables structured data from SQL DW to be included in Spark Analytics



Azure SQL Data Warehouse is a SQL-based fully managed, petabyte-scale cloud solution for data warehousing

- You can bring in data from Azure SQL DW to perform advanced analytics that require both structured and unstructured data.
- Currently you can access data in Azure SQL DW via the [JDBC driver](#). From within your spark code you can access just like any other JDBC data source.
- If Azure SQL DW is authenticated via AAD then Azure Databricks user can seamlessly access Azure SQL DW.



POWER BI INTEGRATION

Enables powerful visualization of data in Spark with Power BI



Power BI is a business analytics tool that provides data Visualization, Report and Dashboard throughout an organization

Power BI Desktop can connect to Azure Databricks clusters to query data using JDBC/ODBC server that runs on the driver node.

- This server listens on port 10000 and it is not accessible outside the subnet where the cluster is running.
- Azure Databricks uses a public HTTPS gateway
- The JDBC/ODBC connection information can be obtained from the Cluster UI directly as shown in the figure.
- When establishing the connection, you can use a Personal Access Token to authenticate to the cluster gateway. Only users who have attach permissions can access the cluster via the JDBC/ ODBC endpoint.
- In Power BI desktop you can setup the connection by choosing the ODBC data source in the "Get Data" option.

The image shows two overlapping screenshots. The background screenshot is the 'Get Data' dialog in Power BI Desktop, with the 'Most Common' list expanded to show 'Excel', 'Power BI service', 'SQL Server', 'Analysis Services', 'Text/CSV', 'Web', 'OData feed', and 'Blank Query'. The foreground screenshot is the 'JDBC/ODBC' tab in the Azure Databricks cluster configuration page. It contains the following fields:

- Server Hostname:** westeurope.azuredatabricks.net
- Port:** 443
- Protocol:** HTTPS
- HTTP Path:** sql/protocolv1/o/3940194168315486/0925-153006-ugh295 (unique)
sql/protocolv1/o/3940194168315486/ntedemoapitest (alias, not guaranteed unique)
- JDBC URL:** jdbc:hive2://westeurope.azuredatabricks.net:443/default;transportMode=http;ssl=true;httpPath=sql/protocolv1/o/3940194168315486/0925-153006-ugh295
jdbc:hive2://westeurope.azuredatabricks.net:443/default;transportMode=http;ssl=true;httpPath=sql/protocolv1/o/3940194168315486/ntedemoapitest

C O S M O S D B I N T E G R A T I O N

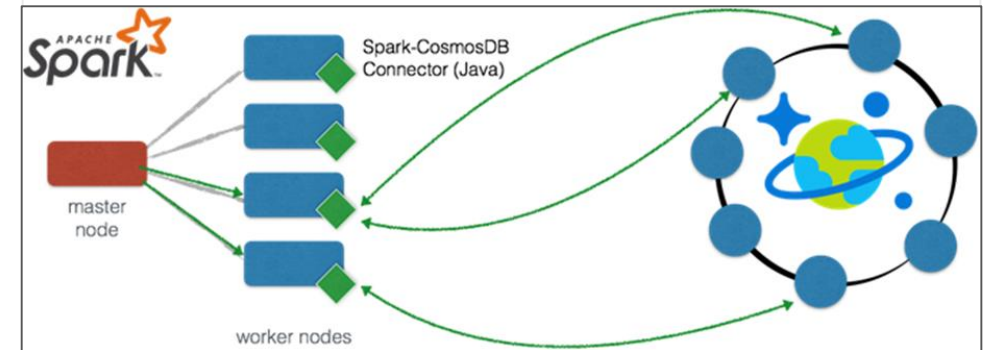
The Spark connector enables real-time analytics over globally distributed data in Azure Cosmos DB



[Azure Cosmos DB](#) is Microsoft's [globally distributed](#), multi-model database service for mission-critical applications

- With Spark connector for Azure Cosmos DB, Apache Spark can now interact with all Azure Cosmos DB data models: *Documents, Tables, and Graphs*.
 - efficiently exploits the native Azure Cosmos DB managed indexes and enables updateable columns when performing analytics.
 - utilizes push-down predicate filtering against fast-changing globally-distributed data
- Some use-cases for Azure Cosmos DB + Spark include:
 - Streaming Extract, Transformation, and Loading of data (ETL)
 - Data enrichment
 - Trigger event detection
 - Complex session analysis and personalization
 - Visual data exploration and interactive analysis
 - Notebook experience for data exploration, information sharing, and collaboration

The connector uses the [Azure DocumentDB Java SDK](#) and moves data directly between Spark worker nodes and Cosmos DB data nodes



A Z U R E B L O B S T O R A G E I N T E G R A T I O N

Data can be read from [Azure Blob Storage](#) using the Hadoop FileSystem interface. Data can be read from public storage accounts without any additional settings. To read data from a private storage account, you need to set an account key or a [Shared Access Signature \(SAS\)](#) in your notebook

Setting up an account key

```
spark.conf.set ( "fs.azure.account.key.{Your Storage Account Name}.blob.core.windows.net", "{Your Storage Account Access Key}")
```

Setting up a SAS for a given container:

```
spark.conf.set( "fs.azure.sas.{Your Container Name}.{Your Storage Account Name}.blob.core.windows.net", "{Your SAS For The Given Container}")
```

Once an account key or a SAS is setup, you can use standard Spark and Databricks APIs to read from the storage account:

```
val df = spark.read.parquet("wasbs://{Your Container Name}@m{Your Storage Account name}.blob.core.windows.net/{Your Directory Name}")  
dbutils.fs.ls("wasbs://{Your Container Name}@{Your Storage Account Name}.blob.core.windows.net/{Your Directory Name}")
```

A Z U R E D A T A L A K E I N T E G R A T I O N

To read from your Data Lake Store account, you can configure Spark to use service credentials with the following snippet in your notebook

```
spark.conf.set("dfs.adls.oauth2.access.token.provider.type", "ClientCredential")
spark.conf.set("dfs.adls.oauth2.client.id", "{YOUR SERVICE CLIENT ID}")
spark.conf.set("dfs.adls.oauth2.credential", "{YOUR SERVICE CREDENTIALS}")
spark.conf.set("dfs.adls.oauth2.refresh.url", "https://login.windows.net/{YOUR DIRECTORY ID}/oauth2/token")
```

After providing credentials, you can read from Data Lake Store using standard APIs:

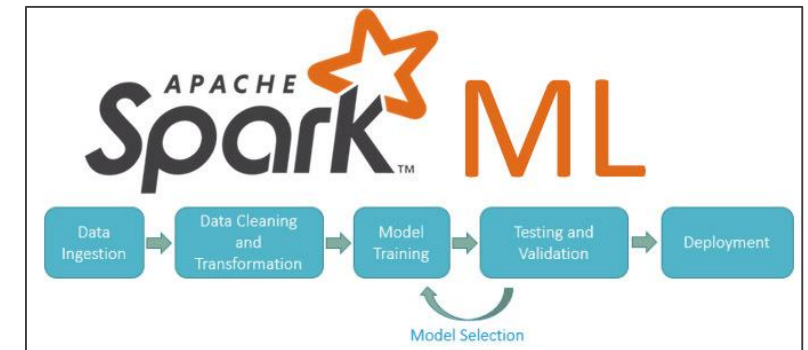
```
val df = spark.read.parquet("adl://{YOUR DATA LAKE STORE ACCOUNT NAME}.azuredatalakestore.net/{YOUR DIRECTORY NAME}")
dbutils.fs.list("adl://{YOUR DATA LAKE STORE ACCOUNT NAME}.azuredatalakestore.net/{YOUR DIRECTORY NAME}")
```

Machine Learning and Deep Learning

SPARK MACHINE LEARNING (ML) OVERVIEW

Enables Parallel, Distributed ML for large datasets on Spark Clusters

- Offers a set of parallelized machine learning algorithms (see next slide)
- Supports [Model Selection](#) (hyperparameter tuning) using [Cross Validation](#) and [Train-Validation Split](#).
- Supports Java, Scala or Python apps using [DataFrame](#)-based API (as of Spark 2.0). Benefits include:
 - An uniform API across ML algorithms and across multiple languages
 - Facilitates [ML pipelines](#) (enables combining multiple algorithms into a single pipeline).
 - Optimizations through Tungsten and Catalyst
- Spark MLlib comes pre-installed on Azure Databricks
- 3rd Party libraries supported include: [H2O Sparkling Water](#), [SciKit-learn](#) and [XGBoost](#)

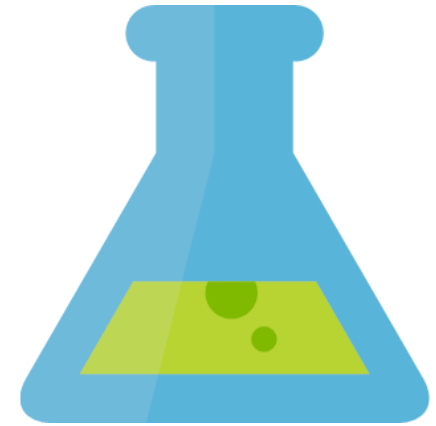


M M L S P A R K

[Microsoft Machine Learning Library](#) for Apache Spark (MMLSpark) lets you easily create scalable machine learning models for large datasets.

It includes integration of SparkML pipelines with the [Microsoft Cognitive Toolkit](#) and [OpenCV](#), enabling you to:

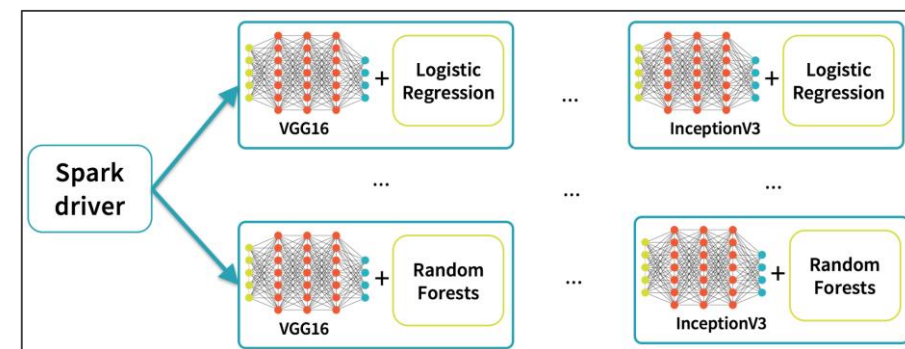
- Ingress and pre-process image data
- Featurize images and text using pre-trained deep learning models
- Train and score classification and regression models using implicit featurization



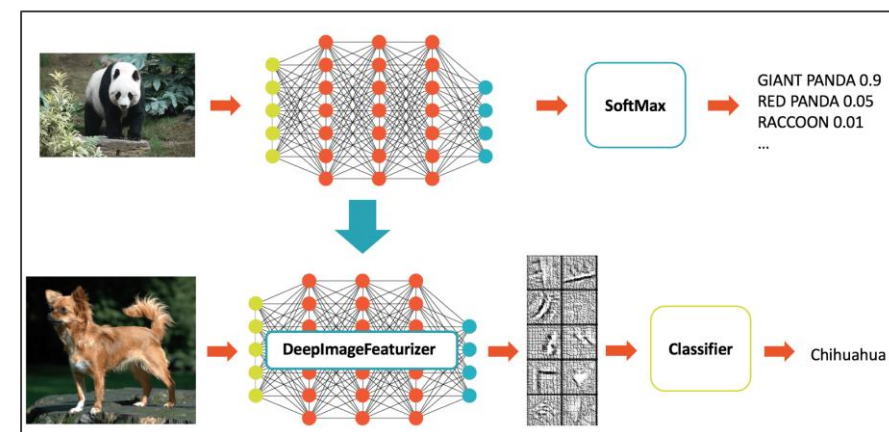
DEEP LEARNING

Azure Databricks supports and integrates with a number of Deep Learning libraries and frameworks to make it easy to build and deploy Deep Learning applications

- Supports Deep Learning Libraries/frameworks including:
 - [Microsoft Cognitive Toolkit \(CNTK\)](#)
 - [Article](#) explains how to install CNTK on Azure Databricks.
 - [TensorFlowOnSpark](#)
 - [BigDL](#)
- Offers [Spark Deep Learning Pipelines](#), a suite of tools for working with and processing images using deep learning using [transfer learning](#). It includes high-level APIs for common aspects of deep learning so they can be done efficiently in a few lines of code:
 - Image loading
 - Applying pre-trained models as transformers in a Spark ML pipeline
 - Transfer learning
 - Distributed hyperparameter tuning
 - Deploying models in DataFrames and SQL



Distributed Hyperparameter Tuning



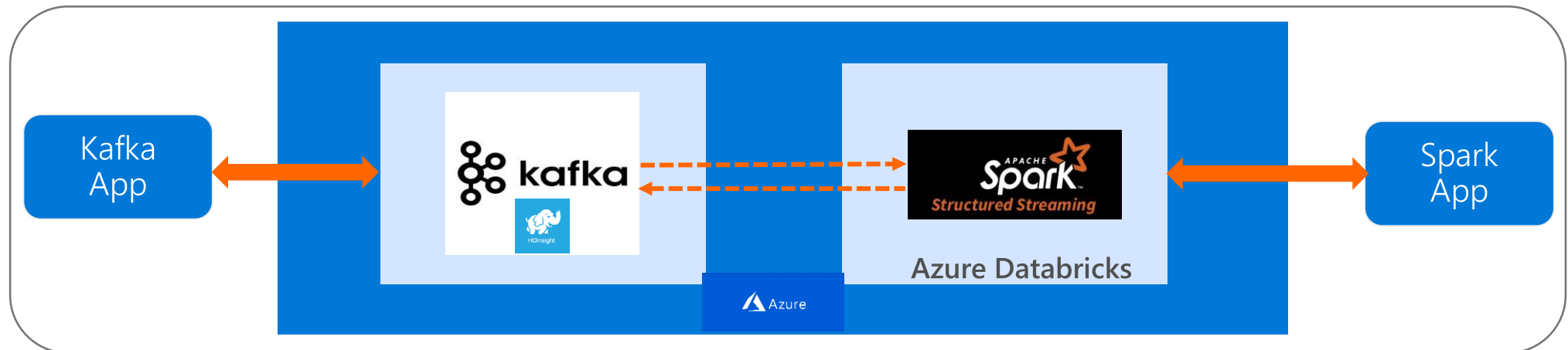
Transfer Learning

Stream Analytics & Graph Processing

APACHE KAFKA FOR HDINSIGHT INTEGRATION

Azure Databricks Structured Streaming integrates with Apache Kafka for HDInsight

- Apache Kafka for Azure HDInsight is an enterprise grade streaming ingestion service running in Azure.
- Azure Databricks Structured Streaming applications can use Apache Kafka for HDInsight as a data source or sink.
- No additional software (gateways or connectors) are required.
- Setup: Apache Kafka on HDInsight does not provide access to the Kafka brokers over the public internet. So the Kafka clusters and the Azure Databricks cluster must be located in the same Azure Virtual Network.



Note: Azure Databricks Structured Streaming integration with **Azure Event Hubs** is forthcoming



© 2017 Microsoft Corporation. All rights reserved. Microsoft, Windows, and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries. The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.