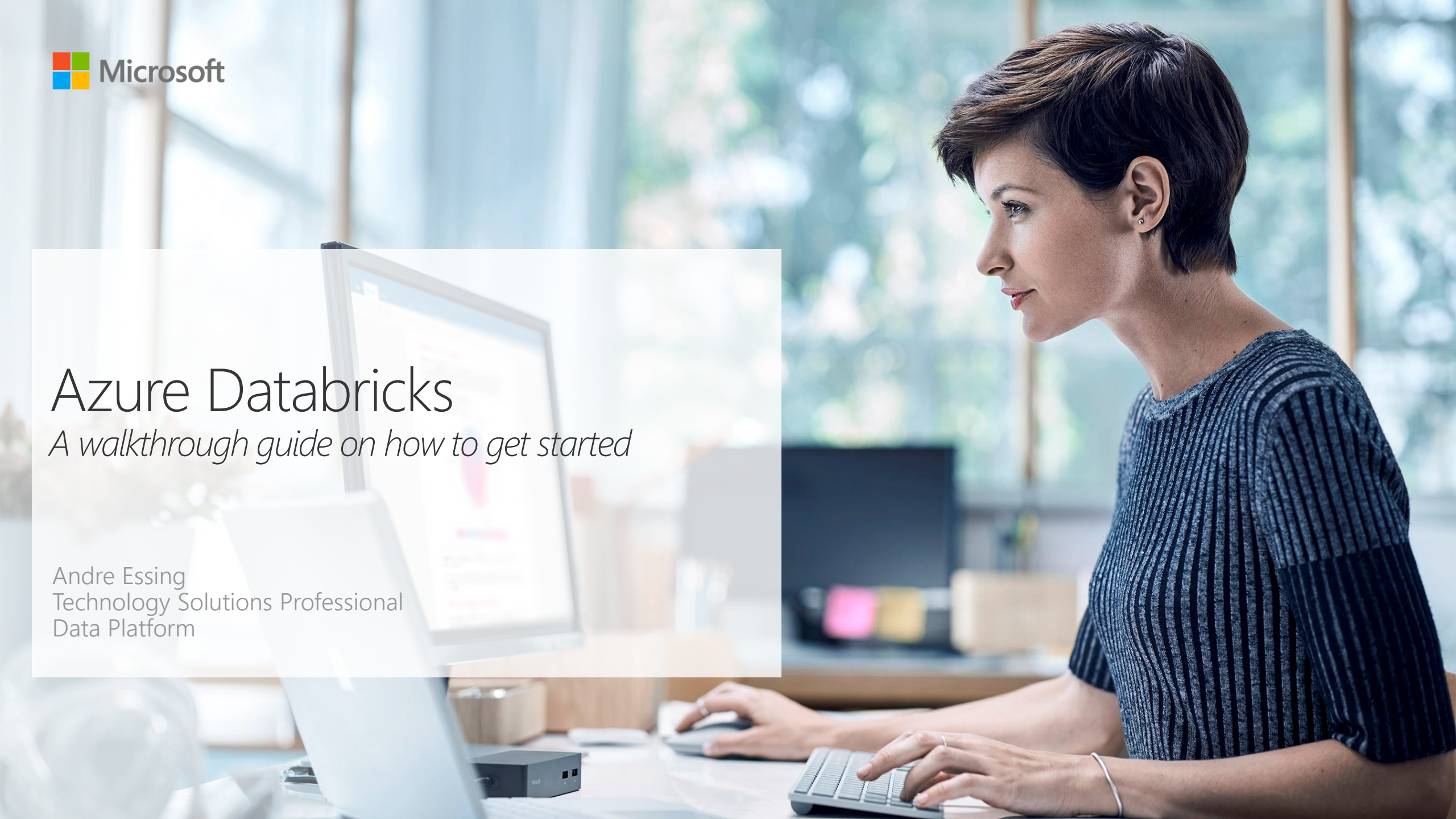# Azure Databricks

A walkthrough guide on how to get started

Andre Essing
Technology Solutions Professional
Data Platform

Andre Essing

Technology Solutions Professional
Microsoft Deutschland GmbH

Andre advises customers in topics all around the Microsoft Data Platform. Since version 7.0, Andre gathering experience with the SQL Server product family. Today Andre concentrates on working with data in the cloud, like Modern Data Warehouse architectures, Artificial Intelligence and new scalable database systems like Azure Cosmos DB.
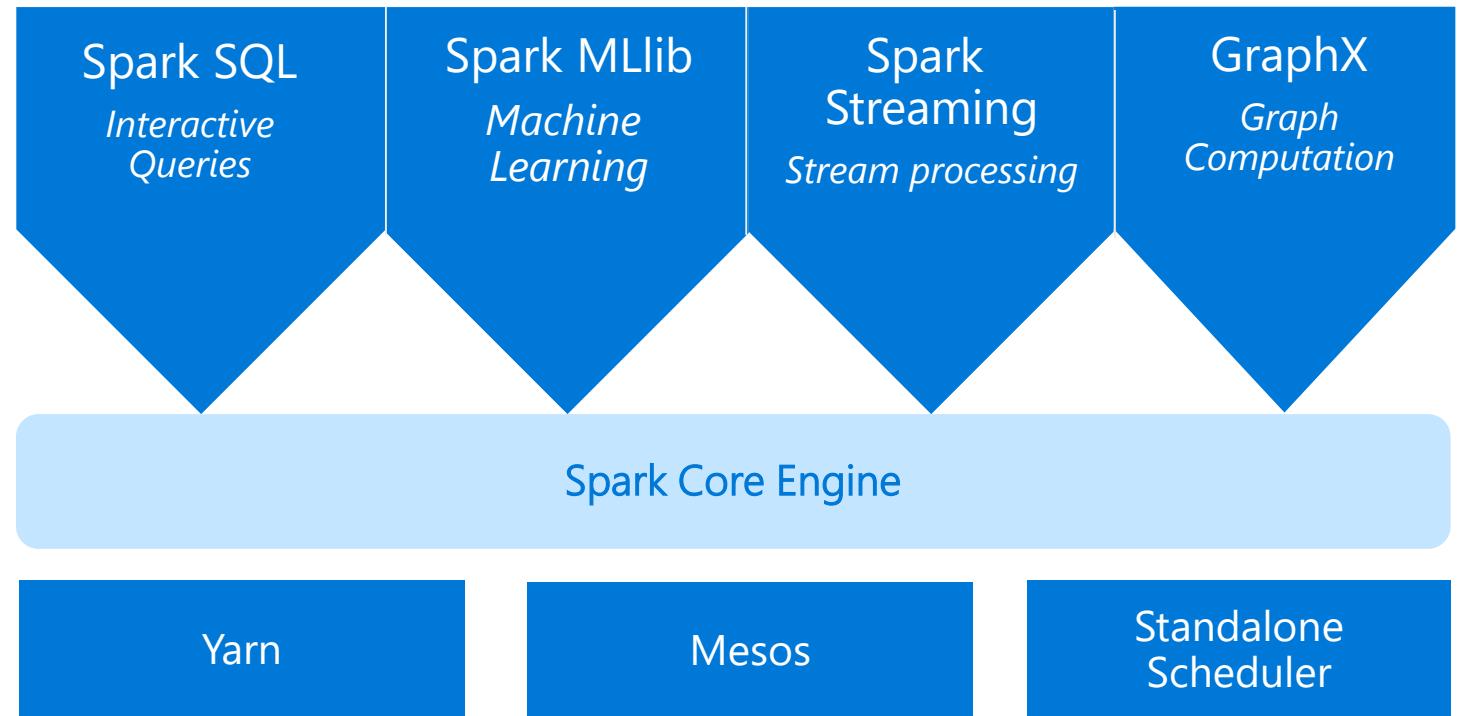
✉ andre.essing@microsoft.com   🌐 andreessing.de   in aessing   🐦 @aessing   aessing
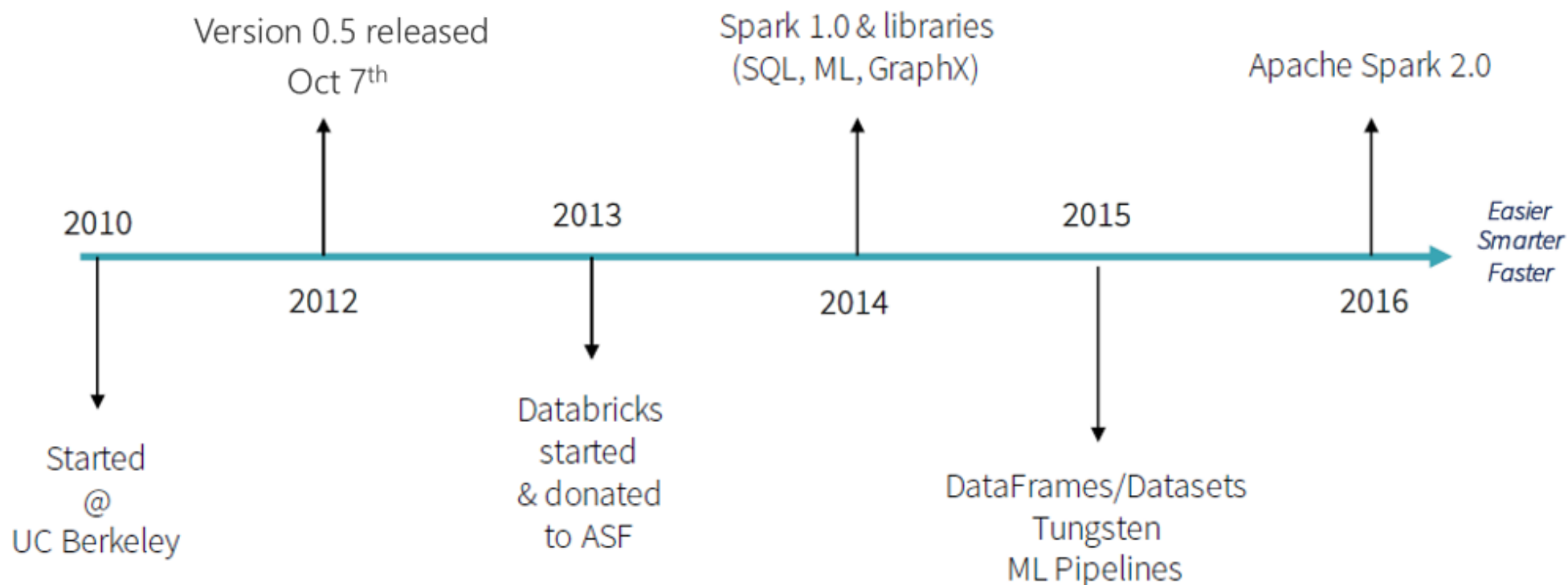
APACHE SPARK

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

Spark SQL *Interactive Queries*	Spark MLlib *Machine Learning*	Spark Streaming *Stream processing*	GraphX *Graph Computation*

Spark Core Engine

| Yarn	Mesos	Standalone Scheduler

SPARK: A BRIEF HISTORY

DATABRICKS - COMPANY OVERVIEW

- Founded in late 2013

- By the creators of Apache Spark, original team from UC Berkeley AMPLab

- Largest code contributor code to Apache Spark

- Level 2/3 support partnership with
  - Hortonworks
  - MapR
  - DataStax

- Provides certifications such as Databricks Certified Application, Databricks Certified Distribution and Databricks Certified Developer

- Main Product: The Unified Analytics Platform

- In Oct 2017, introduced Databricks Delta (currently in private preview).
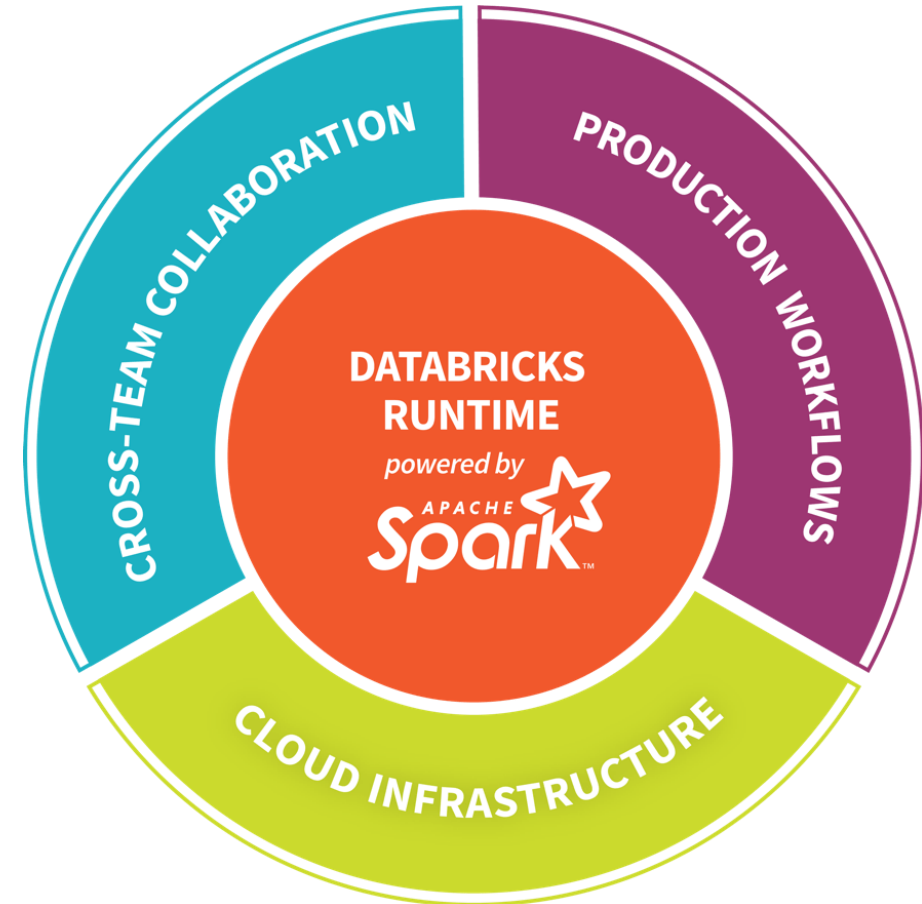
AZURE DATABRICKS

- 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 an 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 uses 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

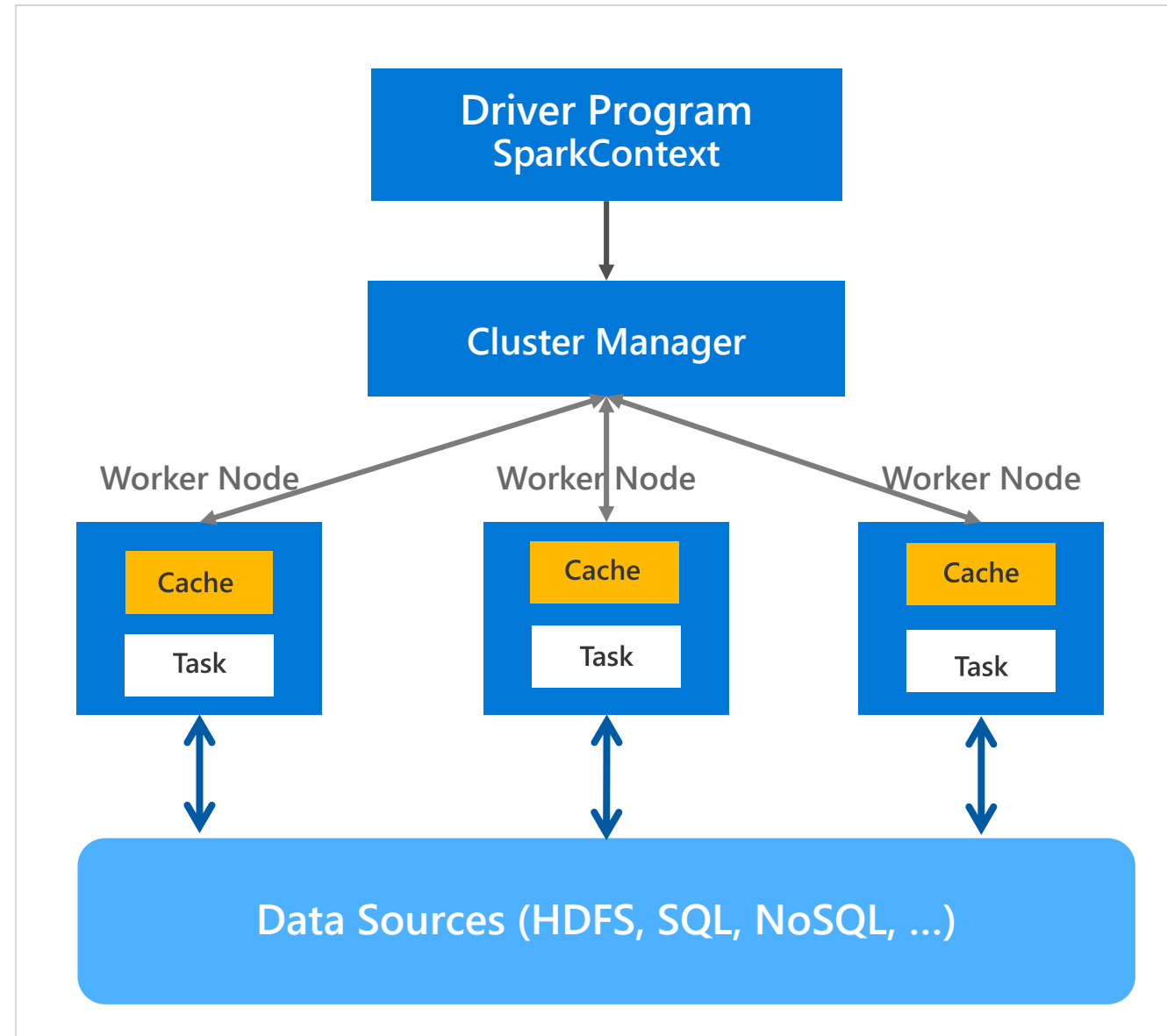- Eliminates need to create a separate account with Databricks.
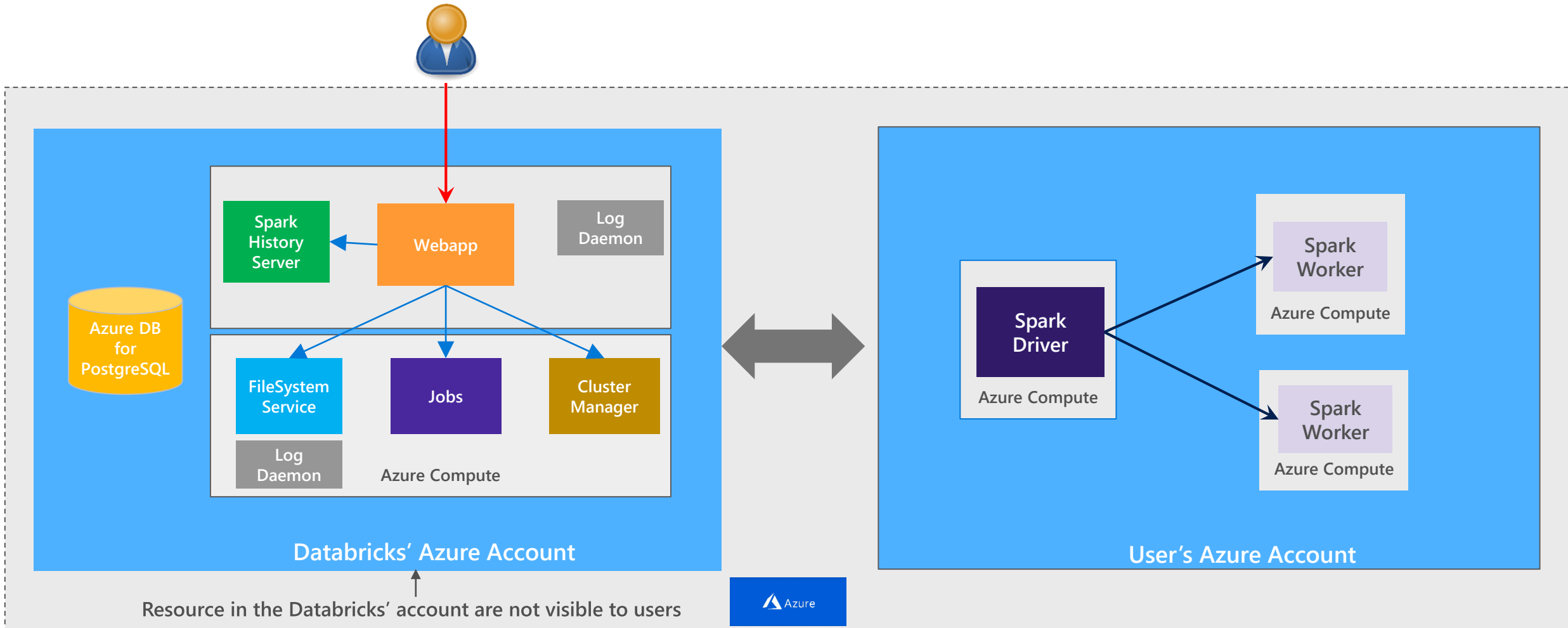
Use Cases

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

Spark History Server

Webapp

Log Daemon

Azure DB for PostgreSQL

FileSystem Service

Jobs

Cluster Manager

Log Daemon

Azure Compute

Databricks' Azure Account

Resource in the Databricks' account are not visible to users

Azure

Spark Driver

Azure Compute

Spark Worker

Azure Compute

Spark Worker

Azure Compute

User's Azure Account

AZURE DATABRICKS NOTEBOOKS OVERVIEW

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
- **%run**   Allows you to run another notebook from within another

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

Python | Scala | CLI | dbutils

DBFS

db.fs.mount()   db.fs.mount()

Azure Blob Storage
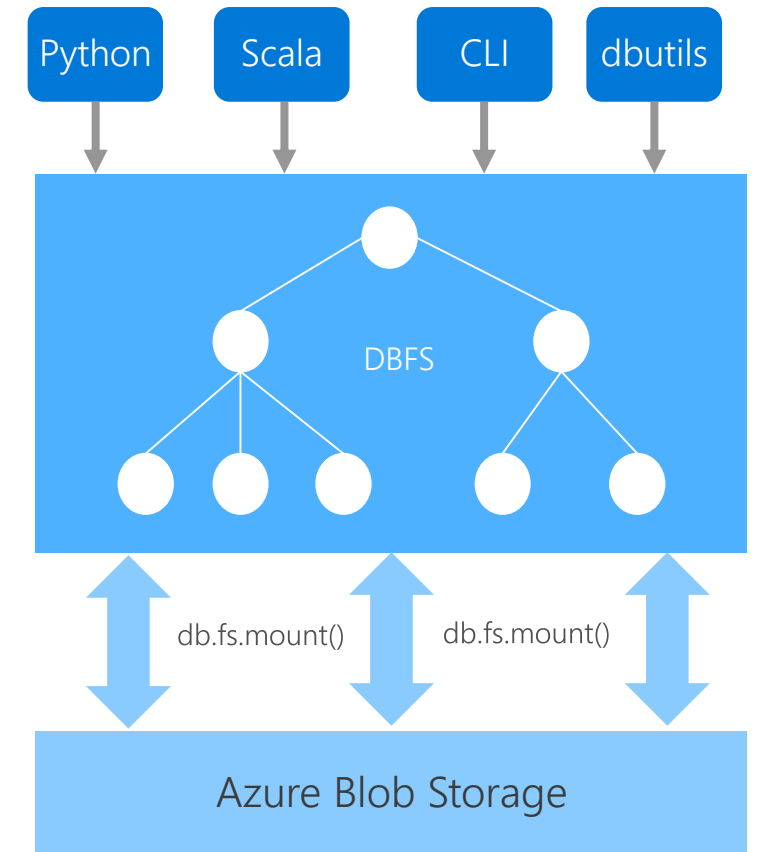
DEMO

Have a look at Azure Databricks