

Azure Databricks – Curated

Naveed Hussain
Global CSA for Data & AI
in OCP

What are companies looking to do next?

Deep neural networks will be a standard tool for **80%** of data scientists¹

20% of companies will dedicate workers to monitor neural networks¹

30% of net new revenue growth from industry-specific solutions will include AI¹

2018

2019

2020

2021

2022

90% of modern analytics platforms will feature natural-language generation¹

More than **40%** of data science tasks will be automated¹

1 in 5 workers engaged in mostly nonroutine tasks will rely on AI to do their jobs²

¹ "100 Data and Analytics Predictions Through 2021", Gartner, 2017. ² "Predicts 2018: AI and the Future of Work", Gartner, 2018.

Hardest Part of AI isn't AI, it's Data

"Hidden Technical Debt in Machine Learning Systems," Google NIPS 2015

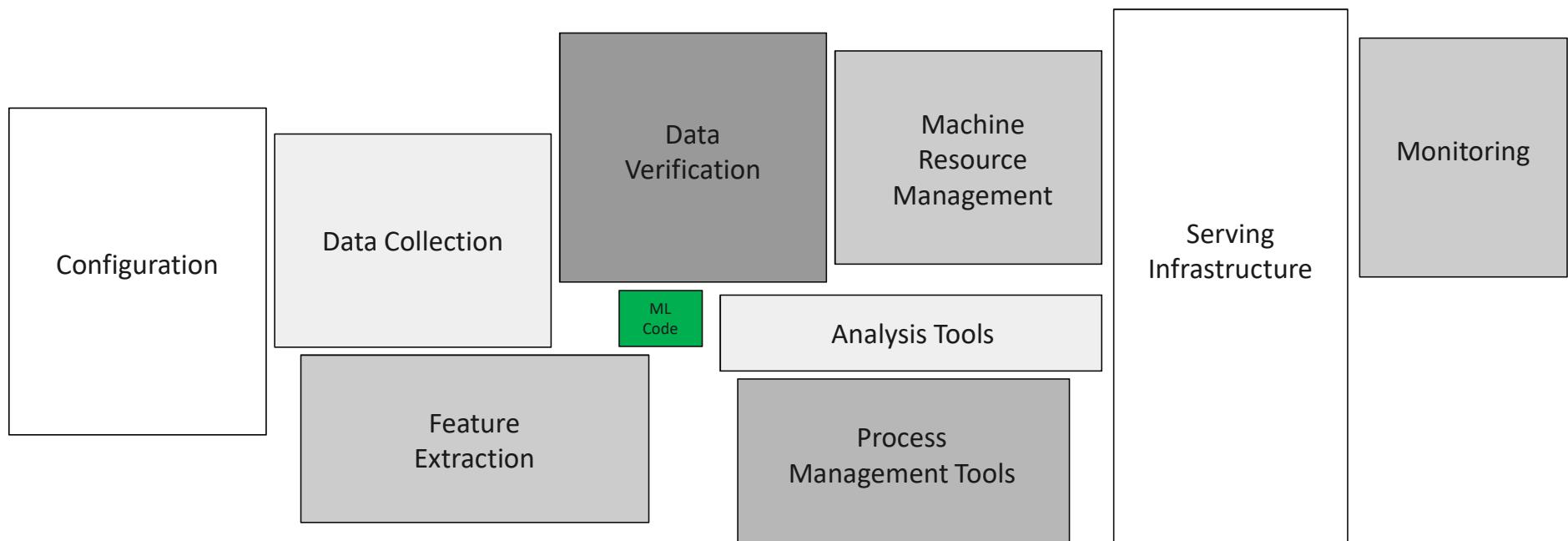
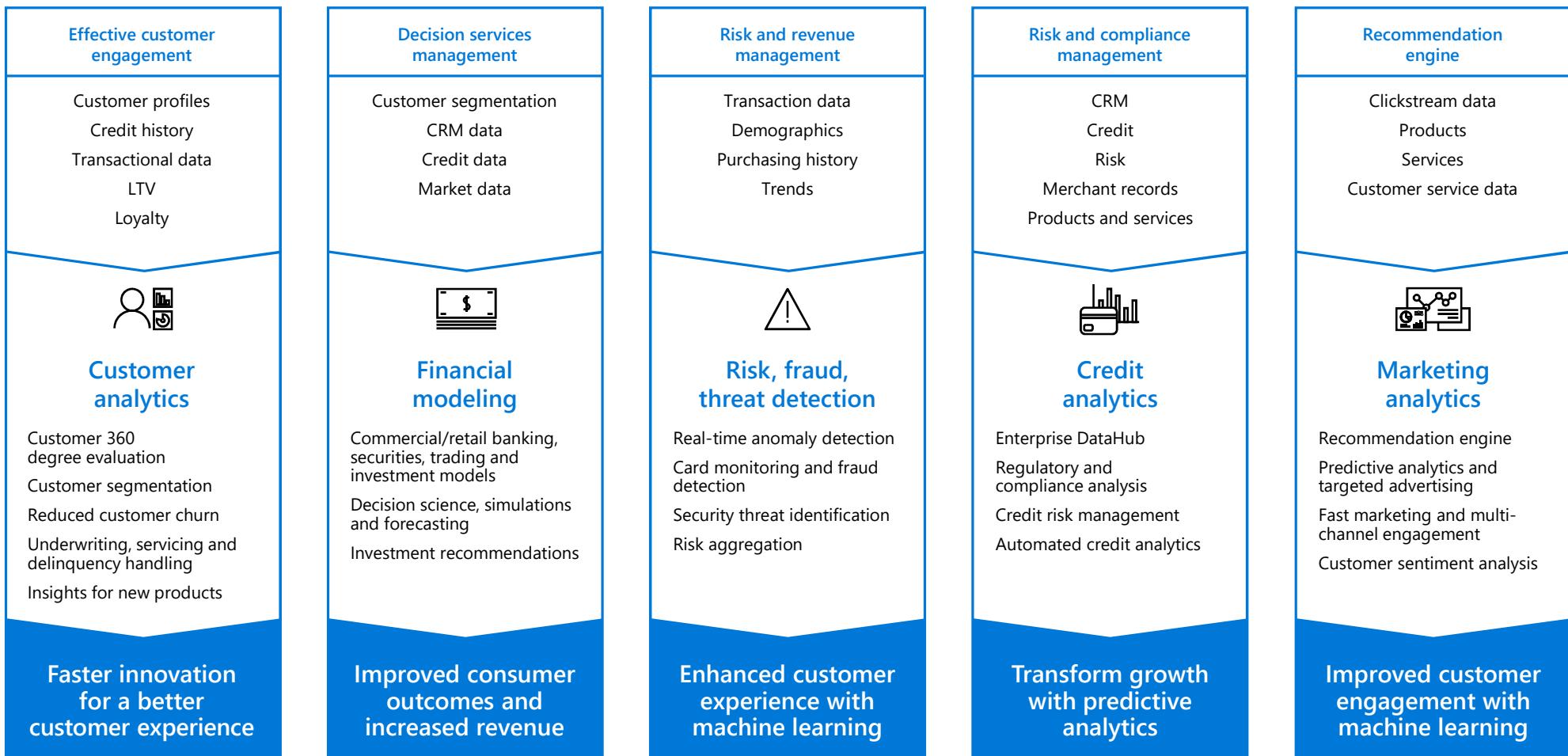
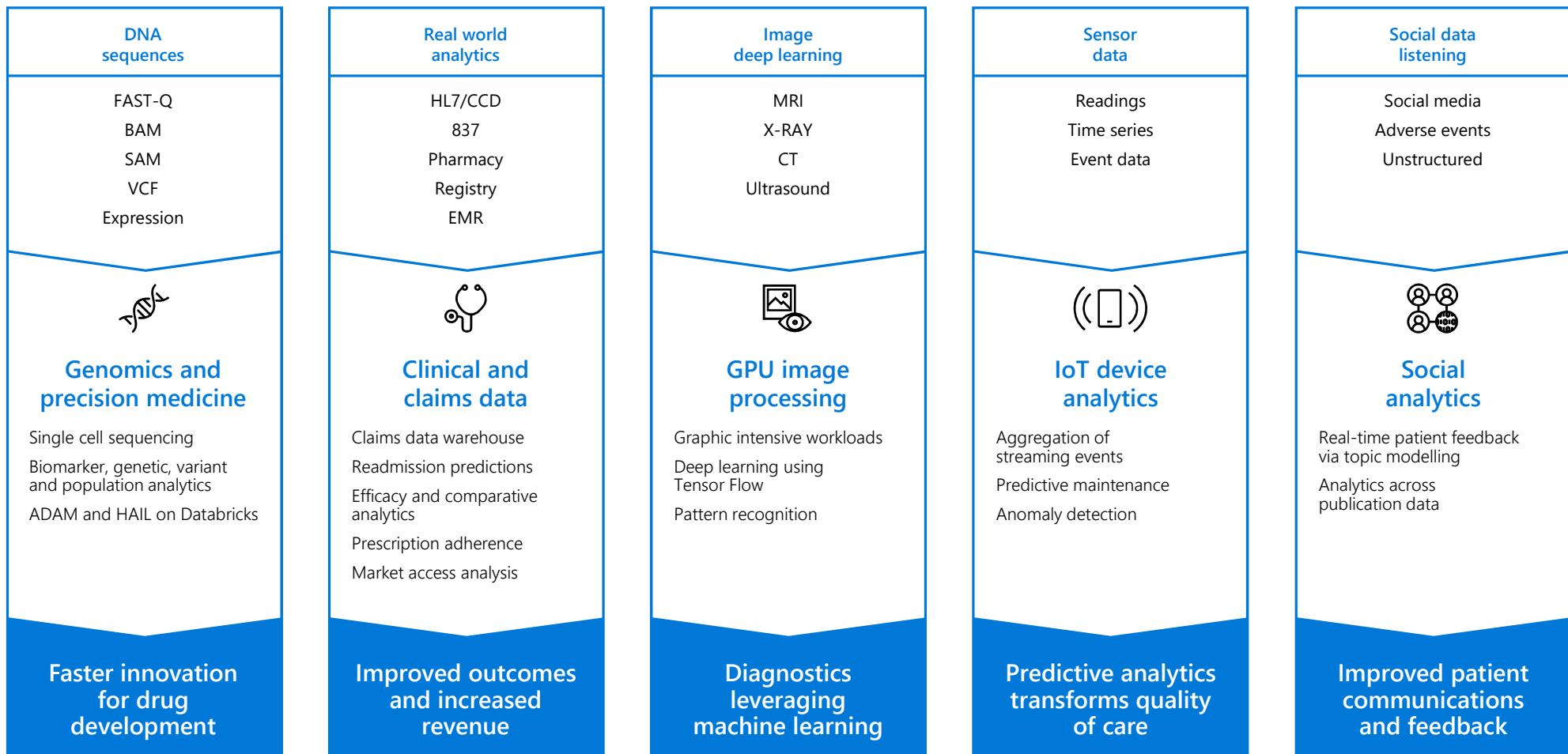


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small green box in the middle. The required surrounding infrastructure is vast and complex.

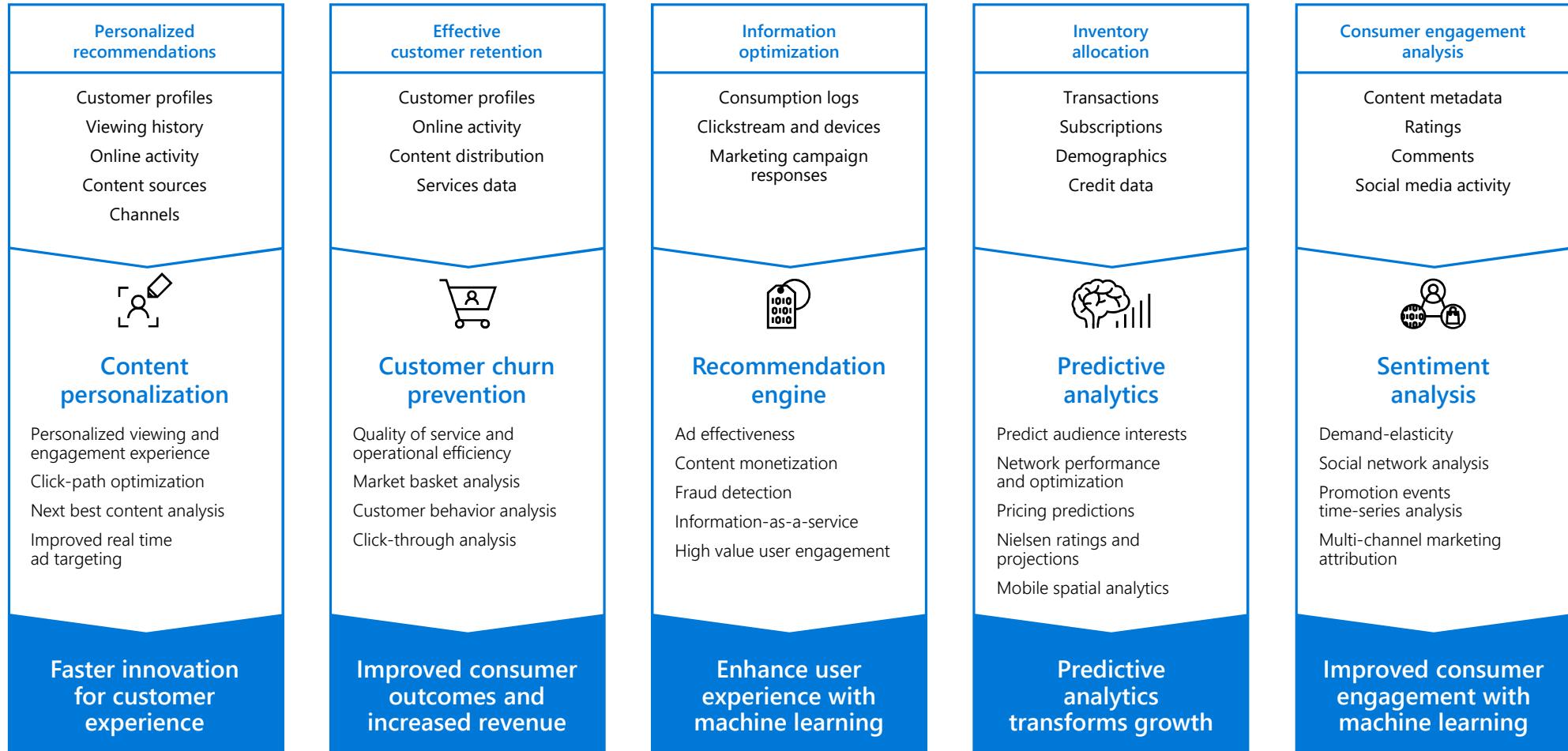
Financial services use cases



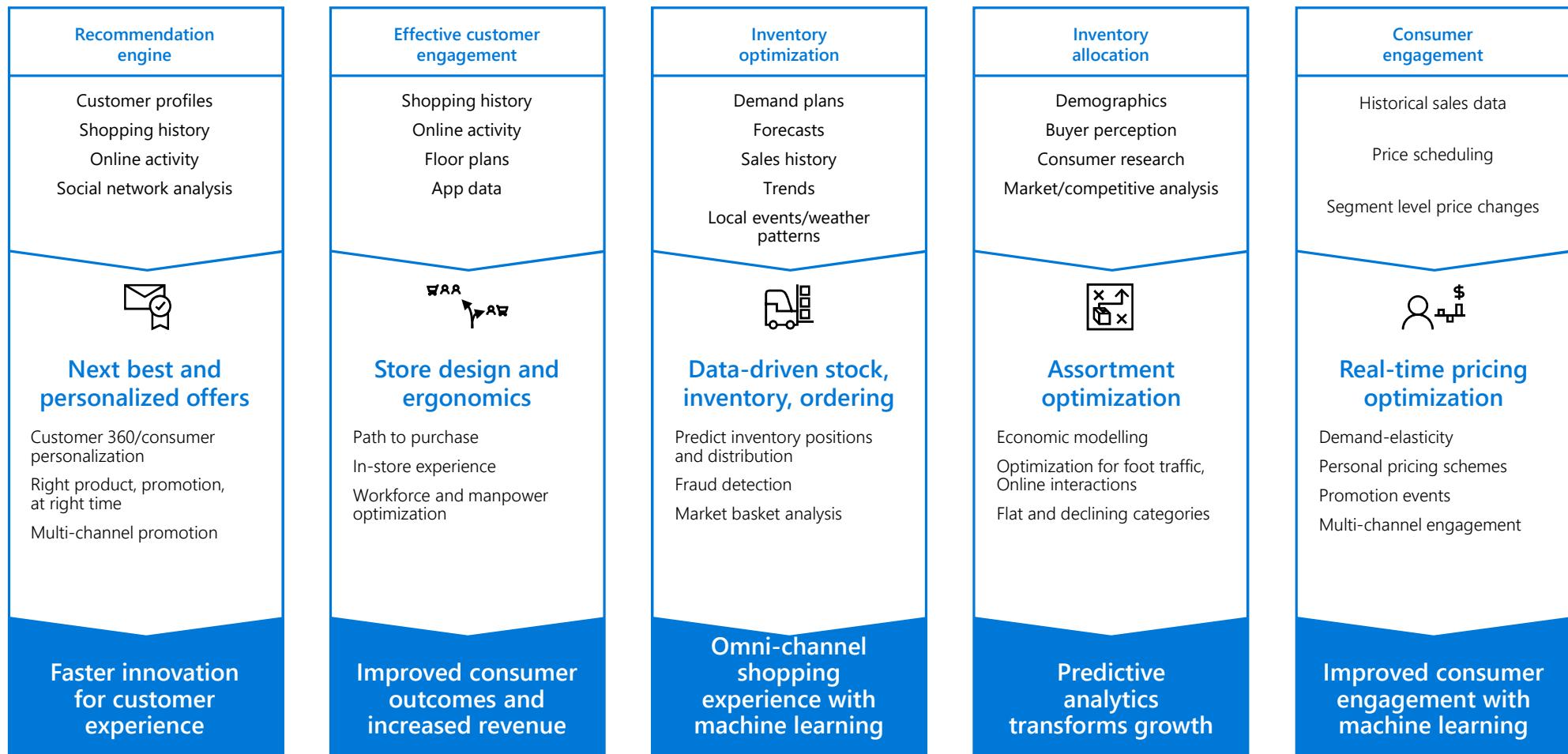
Health and life sciences use cases



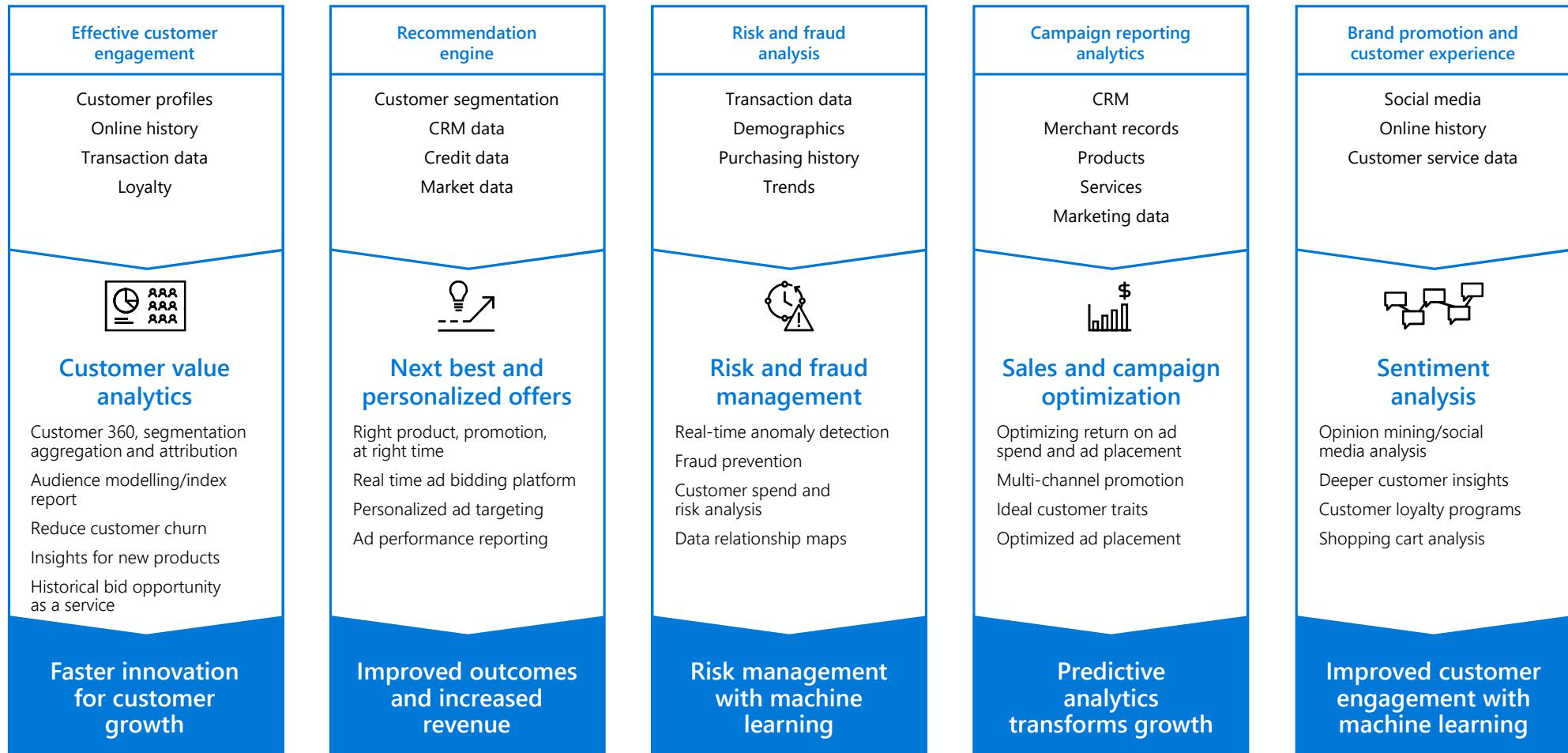
Media and entertainment use cases



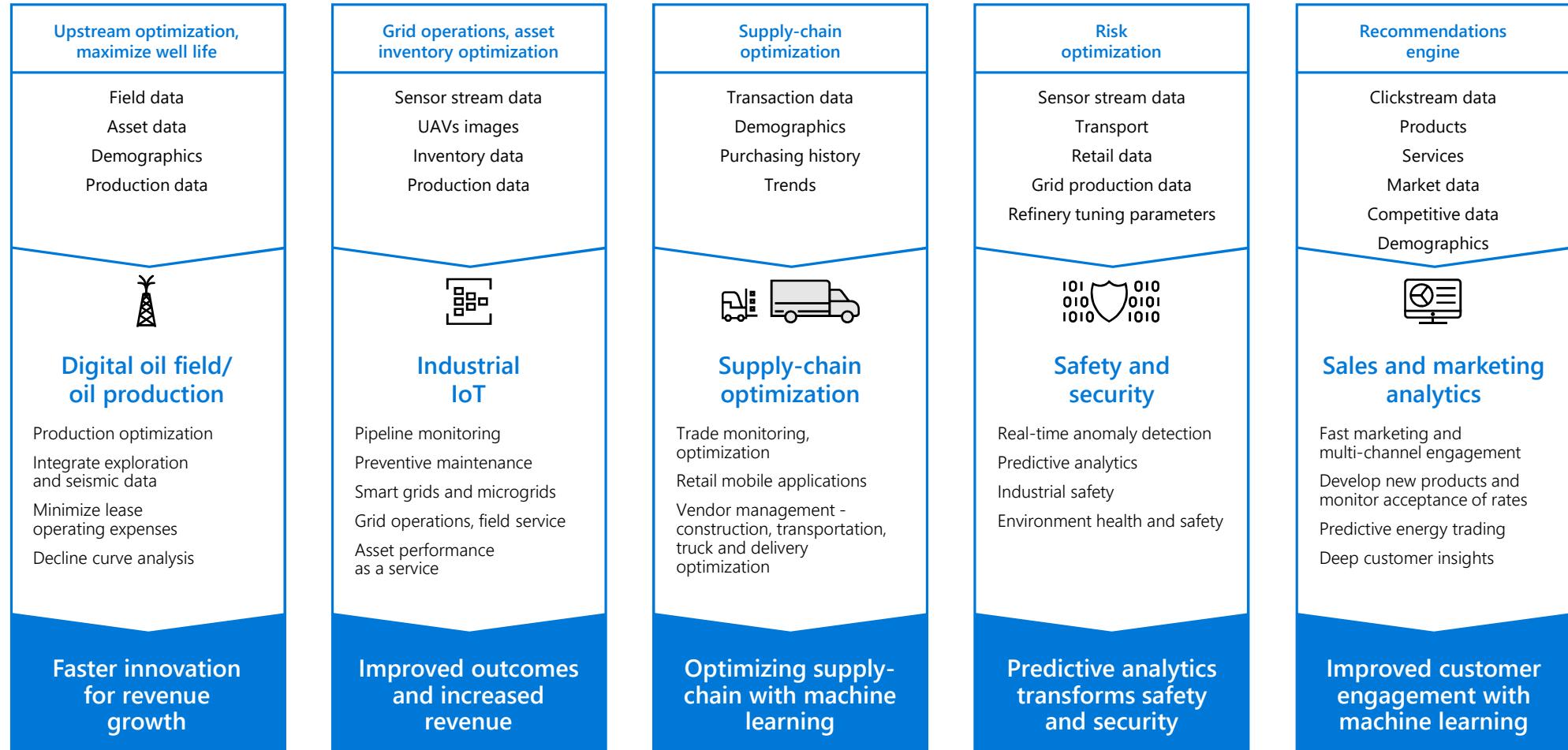
Retail use cases



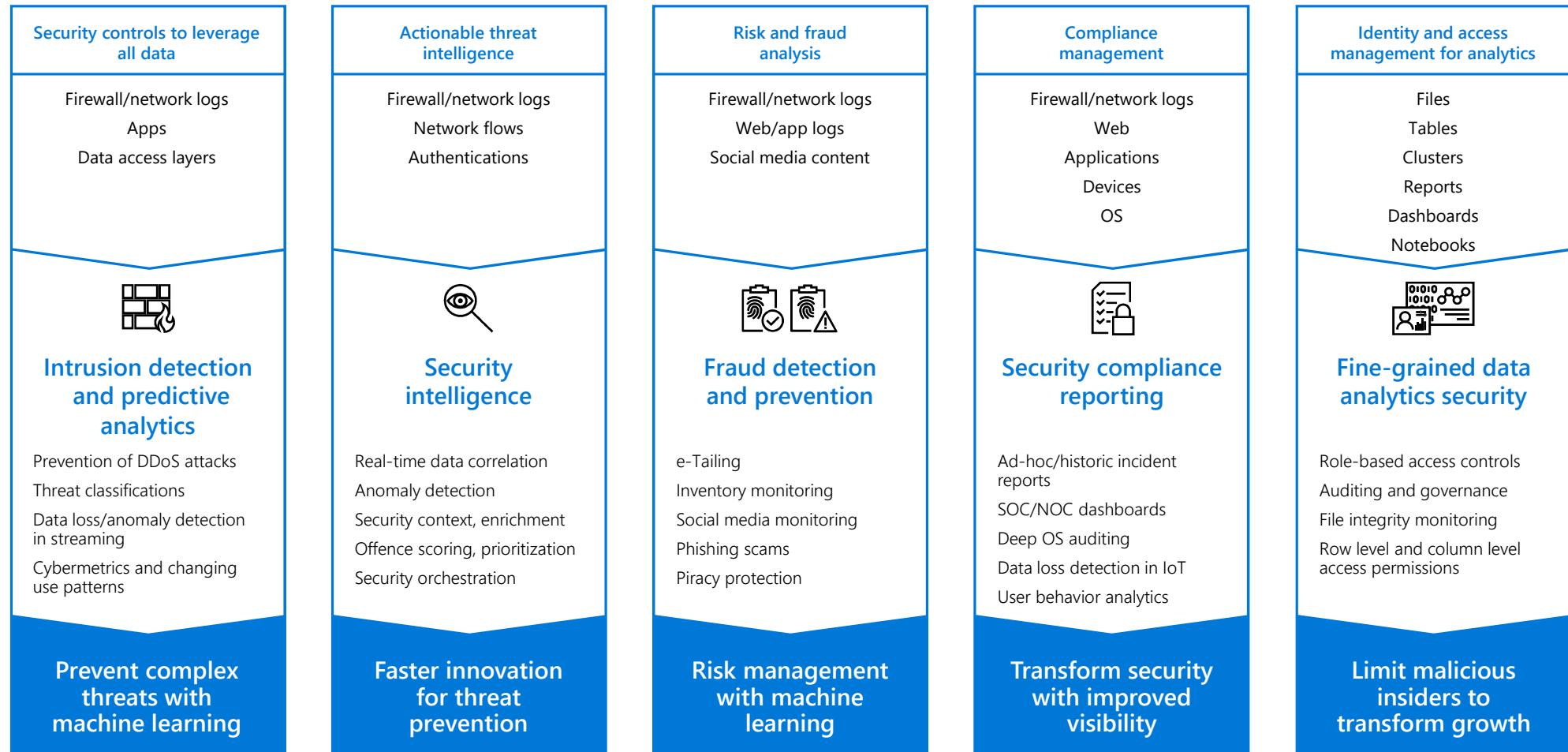
Advertising and marketing tech use cases



Oil, gas, and energy use cases



Security use cases



Our differentiated value proposition

Accelerate time to value
with agile tools and services



Pretrained AI
services



Powerful
tools



Comprehensive
platform

Innovate with AI everywhere –
in the cloud, at edge and on-premises



Cloud



Edge



On-premises

Use any language, any development
tool and any framework



python™



PYTORCH



ONNX



Benefit from industry-leading security, privacy,
compliance, transparency, and AI ethics standards

>90% of Fortune 500 companies
use Microsoft Cloud

LOOKING ACROSS THE OFFERINGS

Azure HDInsight

What It Is

- Hortonworks distribution as a first party service on Azure
- Big Data engines support – Hadoop Projects, Hive on Tez, Hive LLAP, Spark, HBase, Storm, Kafka, R Server
- Best-in-class developer tooling and Monitoring capabilities

Enterprise Features

- VNET support (join existing VNets)
- Ranger support (Kerberos based Security)
- Log Analytics via OMS
- Orchestration via Azure Data Factory
- Available in most Azure Regions (27) including Gov Cloud and Federal Clouds

Guidance

- Customer needs Hadoop technologies other than, or in addition to Spark
- Customer prefers Hortonworks Spark distribution to stay closer to OSS codebase and/or 'Lift and Shift' from on-premises deployments
- Customer has specific project requirements that are only available on HDInsight

Azure Databricks

What It Is

- Databricks' Spark service as a first party service on Azure
- Single engine for Batch, Streaming, ML and Graph
- Best-in-class notebooks experience for optimal productivity and collaboration

Enterprise Features

- Native Integration with Azure for Security via AAD (OAuth)
- Optimized engine for better performance and scalability
- RBAC for Notebooks and APIs
- Auto-scaling and cluster termination capabilities
- Native integration with SQL DW and other Azure services
- Serverless pools for easier management of resources

Guidance

- Customer needs the best option for Spark on Azure
- Customer teams are comfortable with notebooks and Spark
- Customers need Auto-scaling and
- Customer needs to build integrated and performant data pipelines
- Customer is comfortable with limited regional availability 5 in preview, 8 by GA)

Azure ML

What It Is

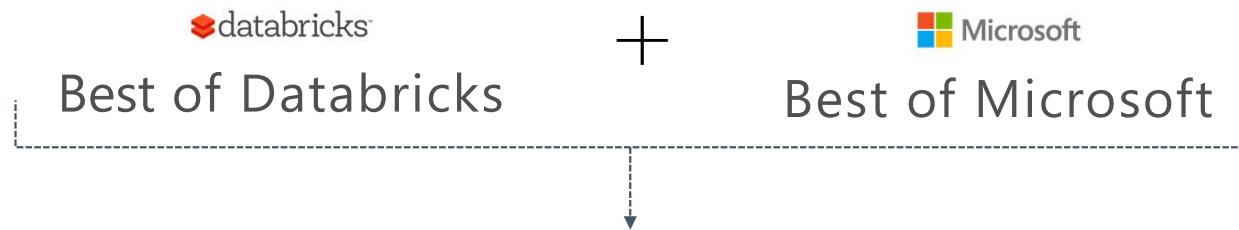
- Azure first party service for Machine Learning
- Leverage existing ML libraries or extend with Python and R
- Targets emerging data scientists with drag & drop offering
- Targets professional data scientists with
 - Experimentation service
 - Model management service
 - Works with customers IDE of choice

Guidance

- Azure Machine Learning Studio is a GUI based ML tool for emerging Data Scientists to experiment and operationalize with least friction
- Azure Machine Learning Workbench is not a compute engine & uses external engines for Compute, including SQL Server and Spark
- AML deploys models to HDI Spark currently
- AML should be able to deploy Azure Databricks in the near future

What is 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, ADLS, Azure Storage, Azure Data Factory, Azure AD, Event Hub, IoT Hub, HDInsight Kafka, SQL DB)

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

Azure Databricks

Fast, easy, and collaborative Apache Spark™-based analytics platform



Increase productivity



Build on a secure, trusted cloud



Scale without limits



Built with your needs in mind

- Role-based access controls
- Effortless autoscaling
- Live collaboration
- Enterprise-grade SLAs
- Best-in-class notebooks
- Simple job scheduling

Azure Databricks

A fast, easy, and collaborative Azure service for Apache Spark-based analytics



Productive

Set up **Spark clusters in minutes** with one click from the Azure portal

Use the **language of your choice** with Python, R, Scala, SQL

Improve collaboration through a unified workspace and **collaborate live with notebooks**



Secure

Simplify security and identity control with built-in integration with **Azure Active Directory** and role-based access

Regulate access with fine-grained user permissions to Azure Databricks' notebooks, clusters, jobs and data

Build with confidence on the **trusted cloud** backed by unmatched support, and compliance



Scalable

Autoscale effortlessly

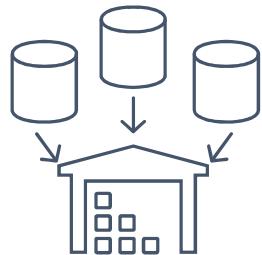
Integrate seamlessly with other **Azure data services**

Benefit from **enterprise-grade SLAs**

Operate at **massive scale**, without limits, globally

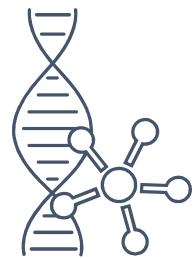
Seamlessly integrated with the Azure portfolio

Usage Objectives



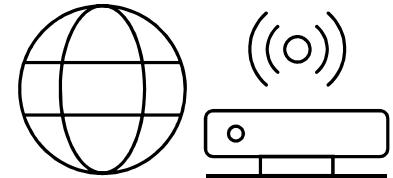
"Want to extend to untapped sources"

Modern Data Warehouse



"Want to use ML and AI to get deeper insights from our data"

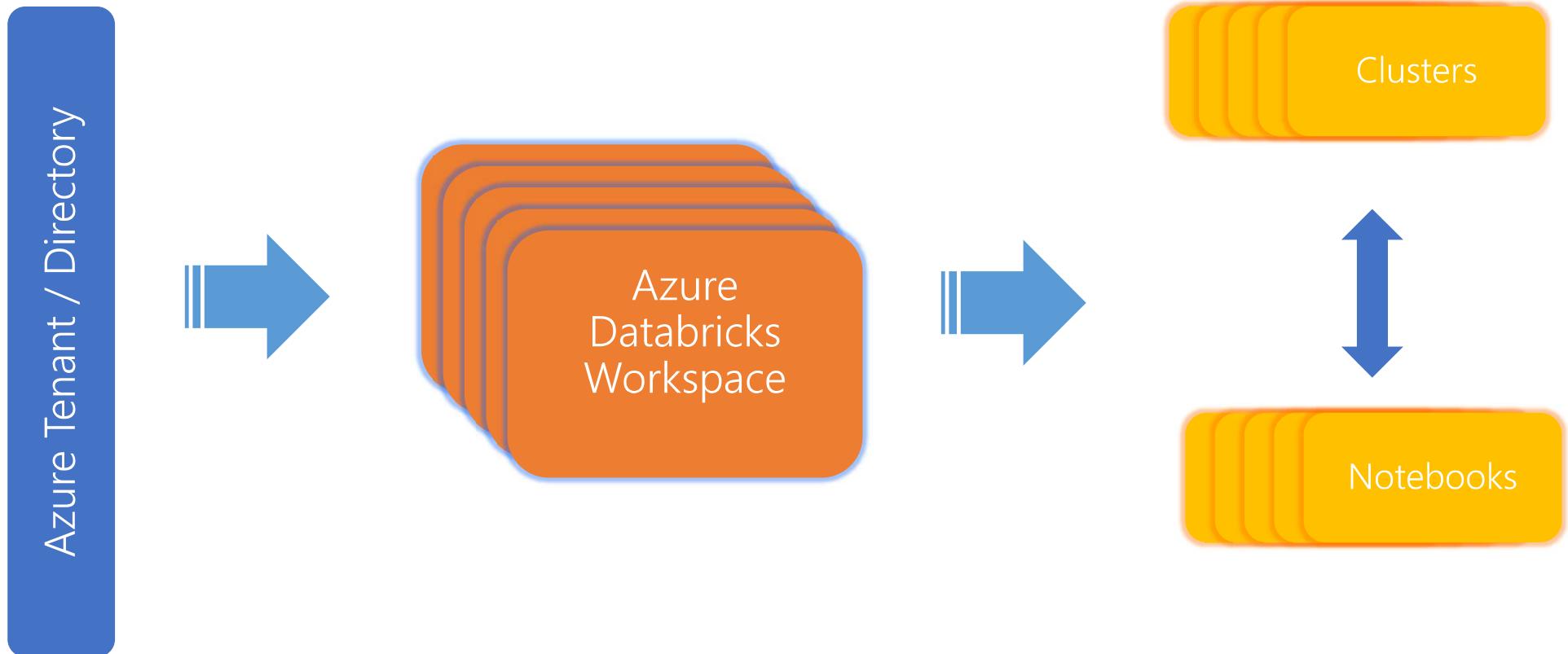
Advanced Analytics



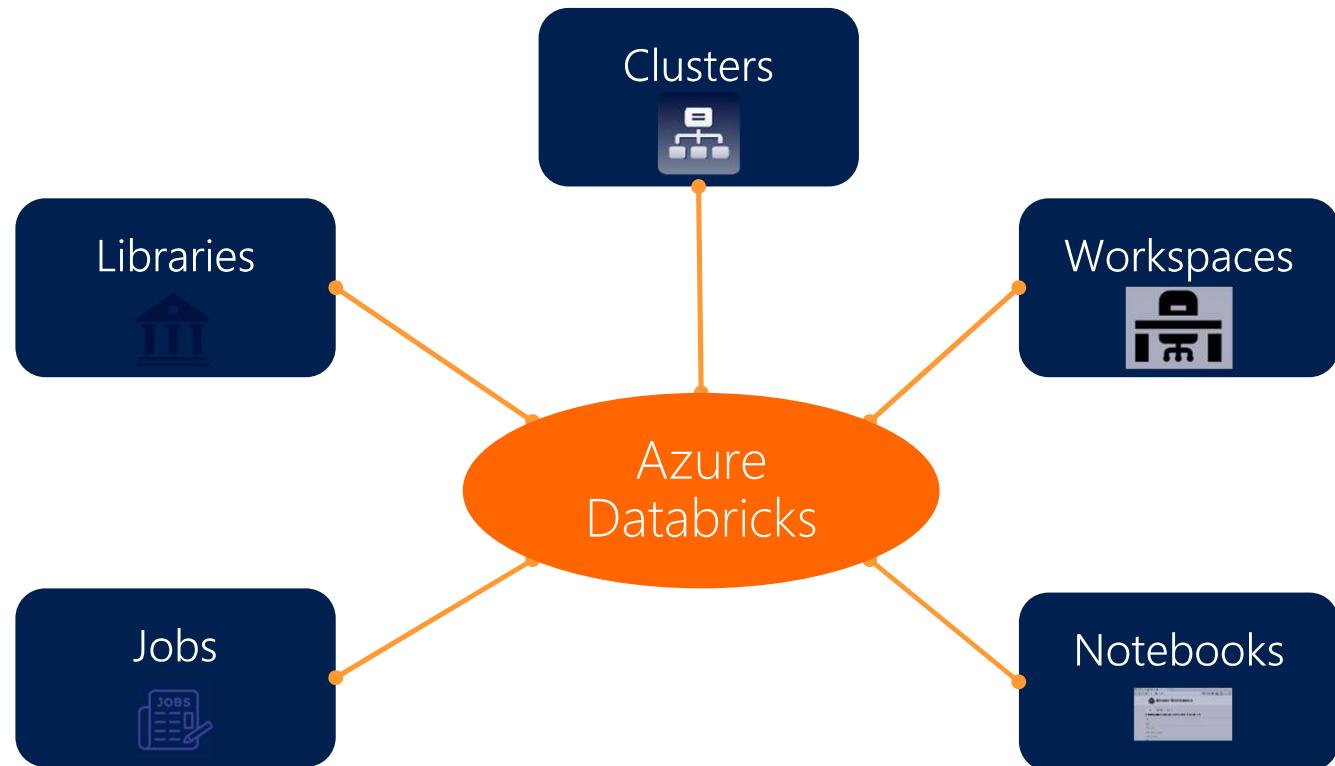
"Want to get insights from our devices in real-time"

Real-time Analytics

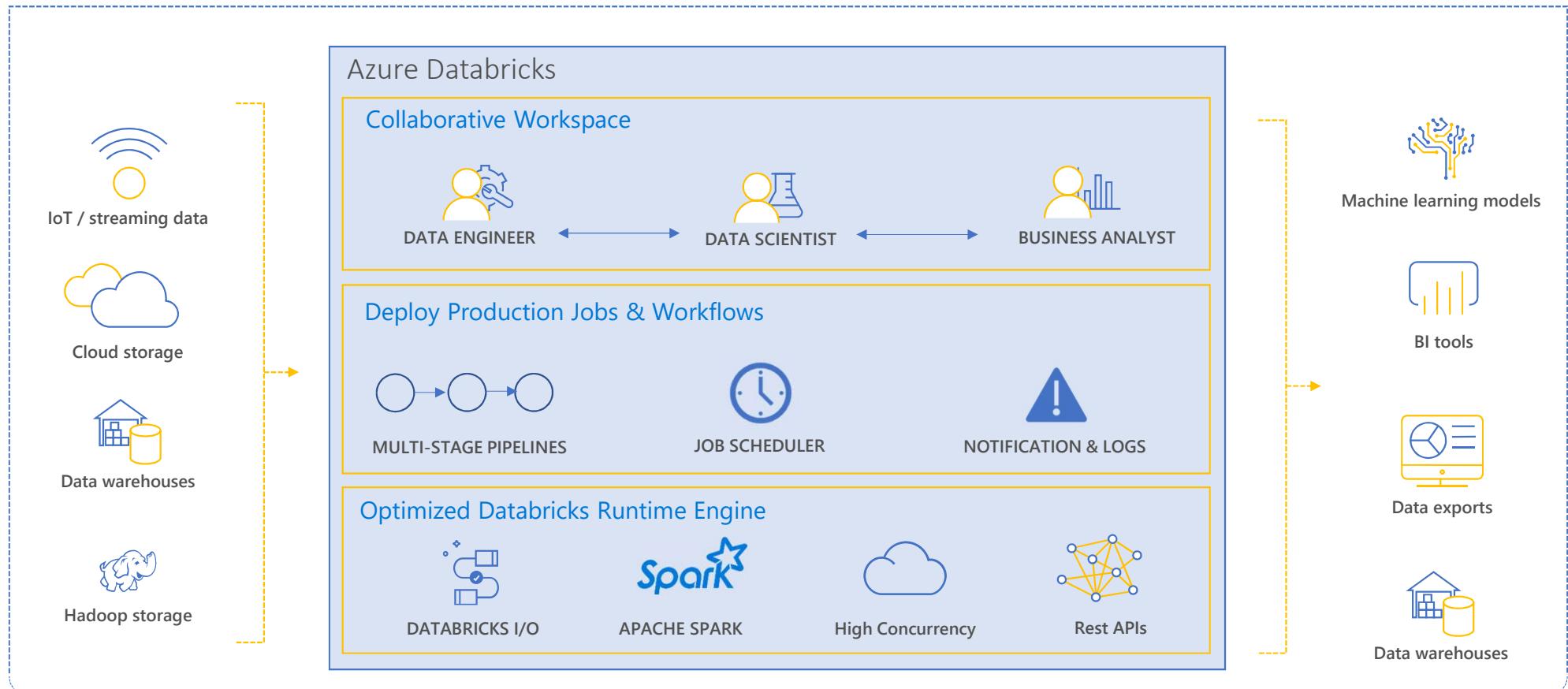
Workspaces, Clusters & Notebooks



AZURE DATABRICKS CORE ARTIFACTS



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

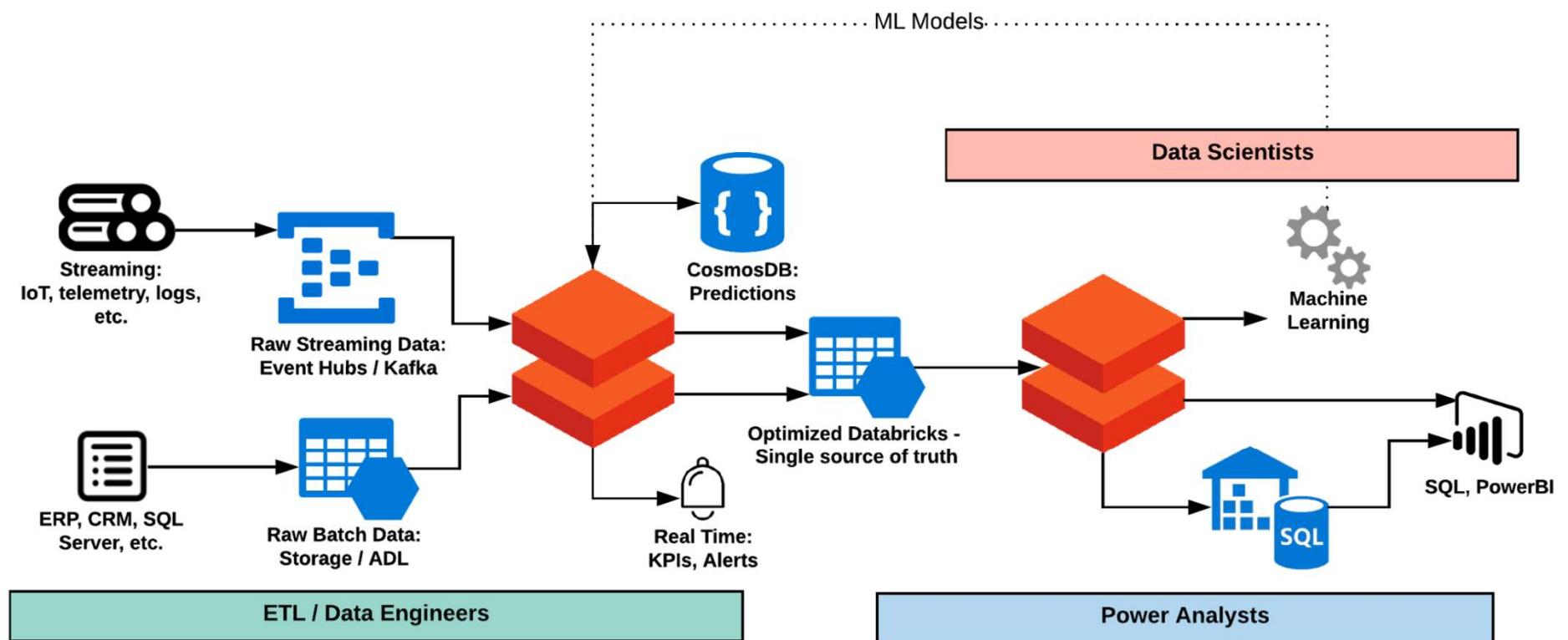


Enhance Productivity

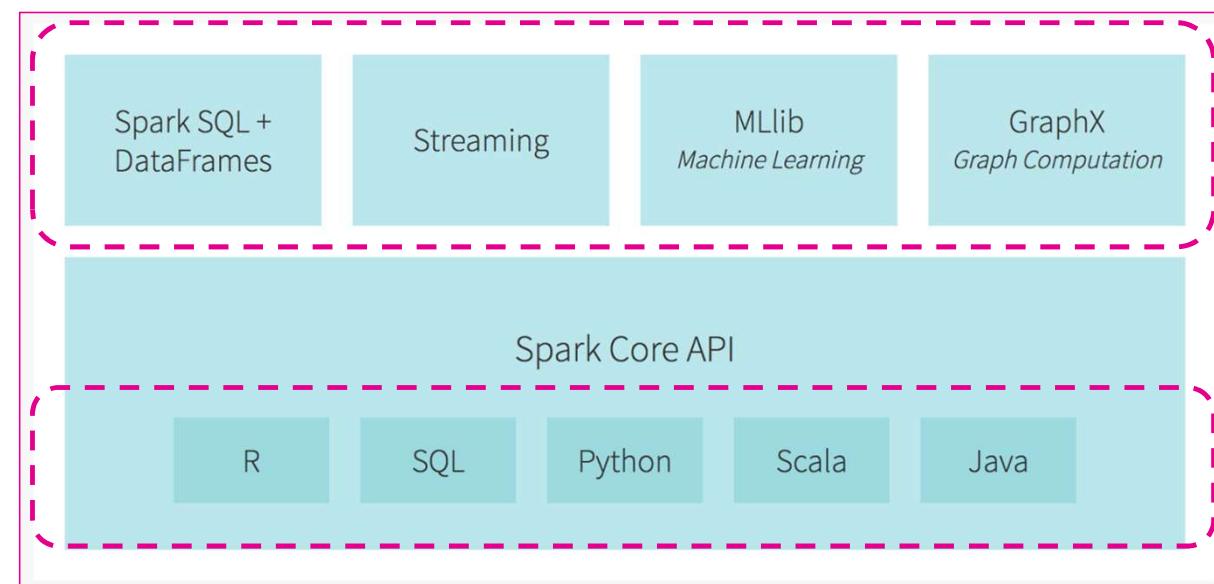
Build on secure & trusted cloud

Scale without limits

Unified Architecture

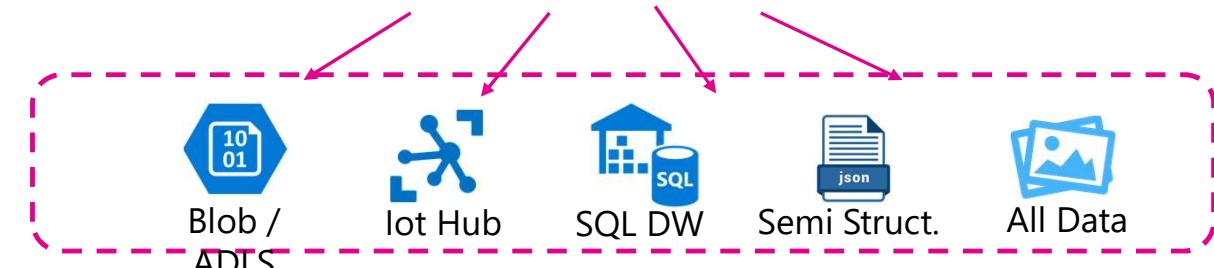


Apache Spark - Distributed Framework



Multiple Use Cases

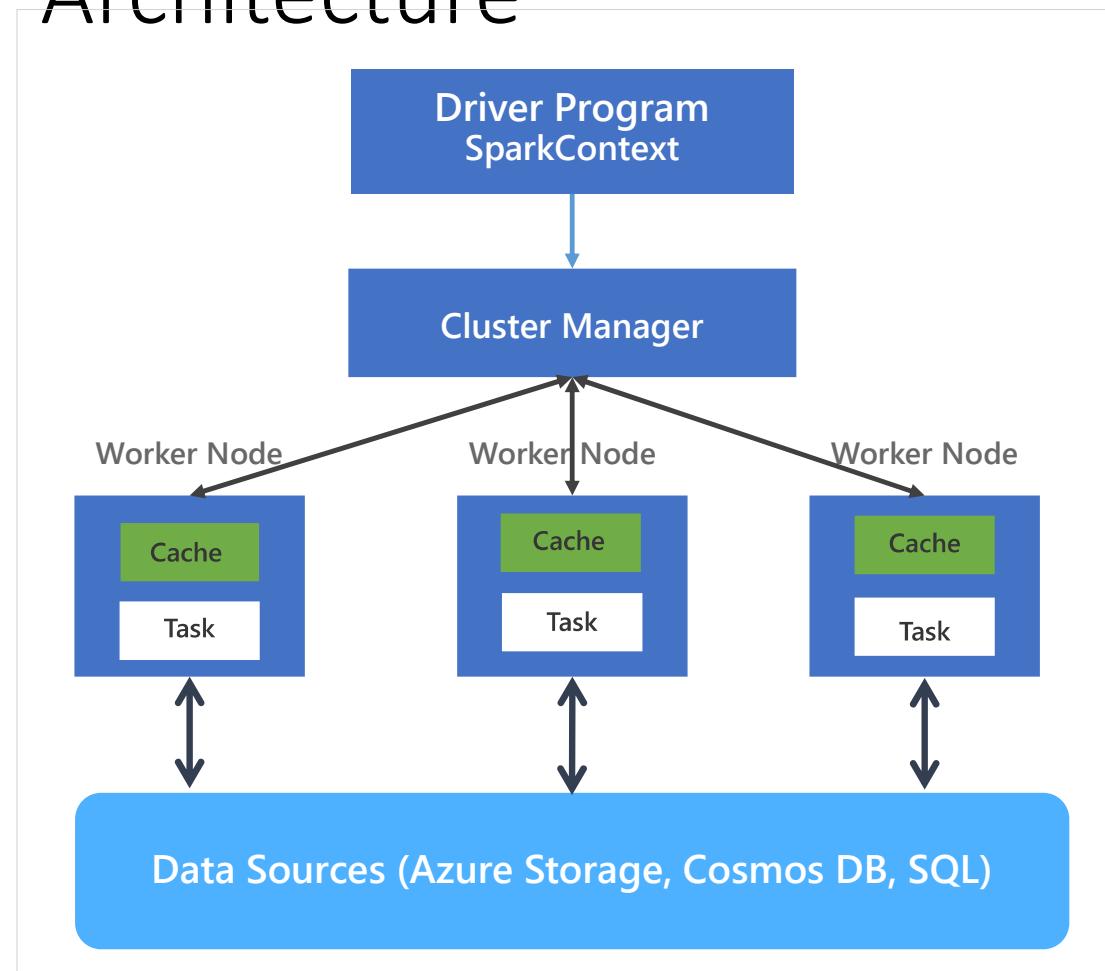
Multiple Languages



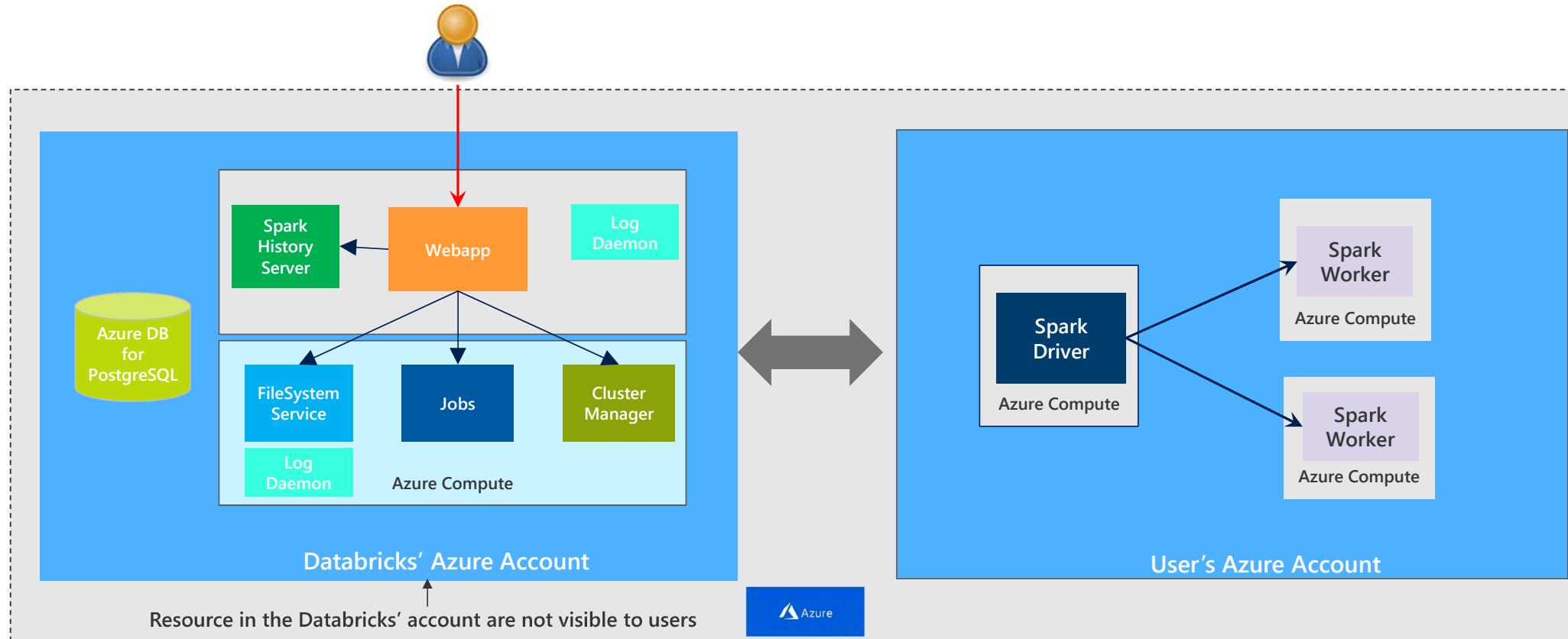
Multiple Data Sources

General Spark Cluster Architecture

- Spark is designed to run on a Cluster
- A cluster is a set of VMs
- Spark can horizontally scale, bigger workload = Add more VMs
- Azure Databricks can automatically scale up and down
- Data can read from Azure Storage or Azure Datalake Storage

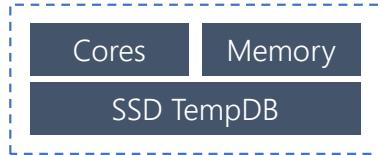


AZURE DATABRICKS CLUSTER ARCHITECTURE

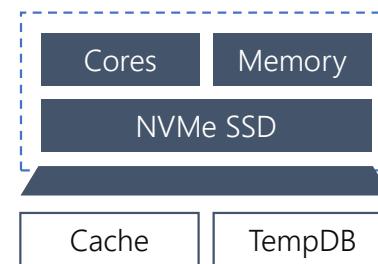
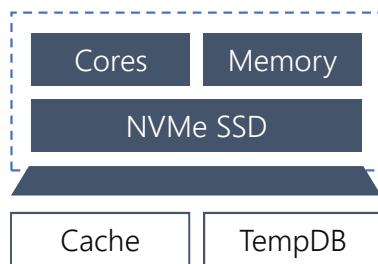
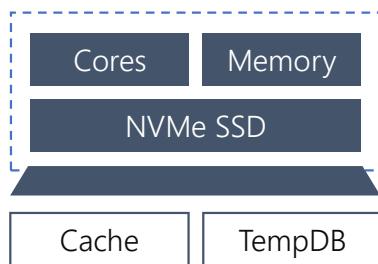


Next Generation Architecture

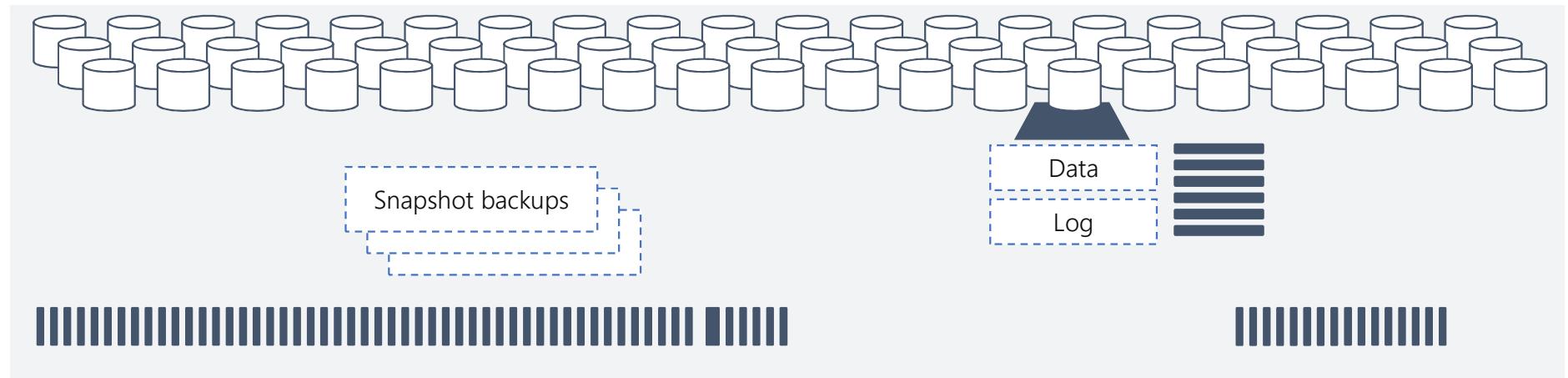
Control



Compute



Remote storage



RDDs, Data Frames and Data Sets

RDDs
(2011)

Data Frames
(2013)

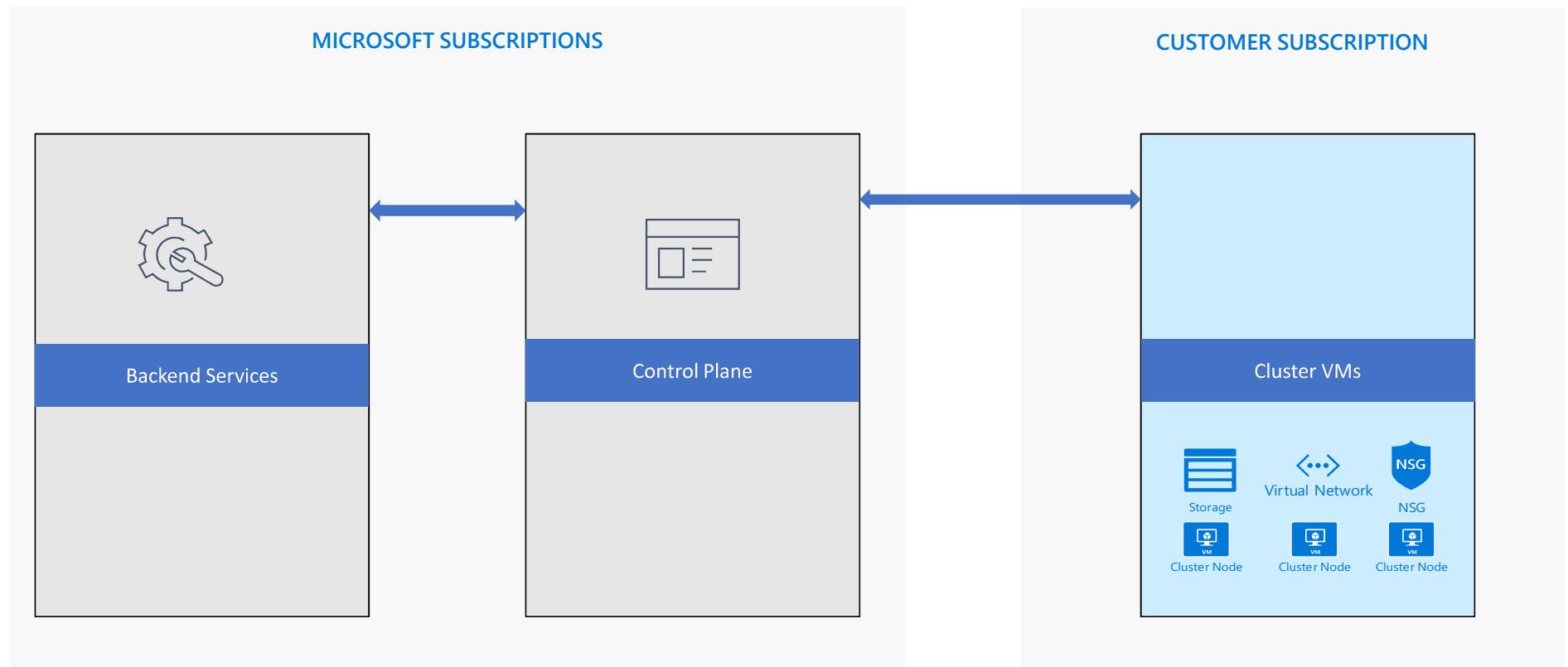
Data Sets
(2015)

- Distributed collection of JVM Objects
- Functional Operators (map, filters)

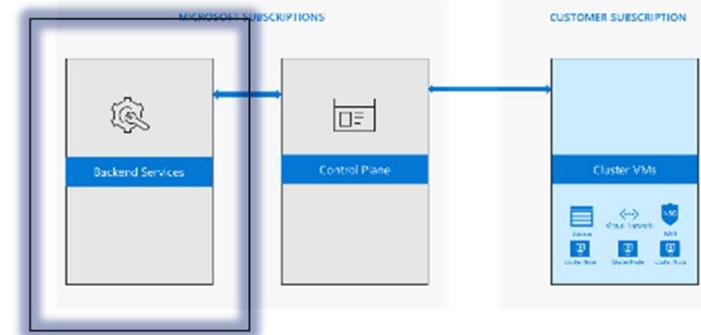
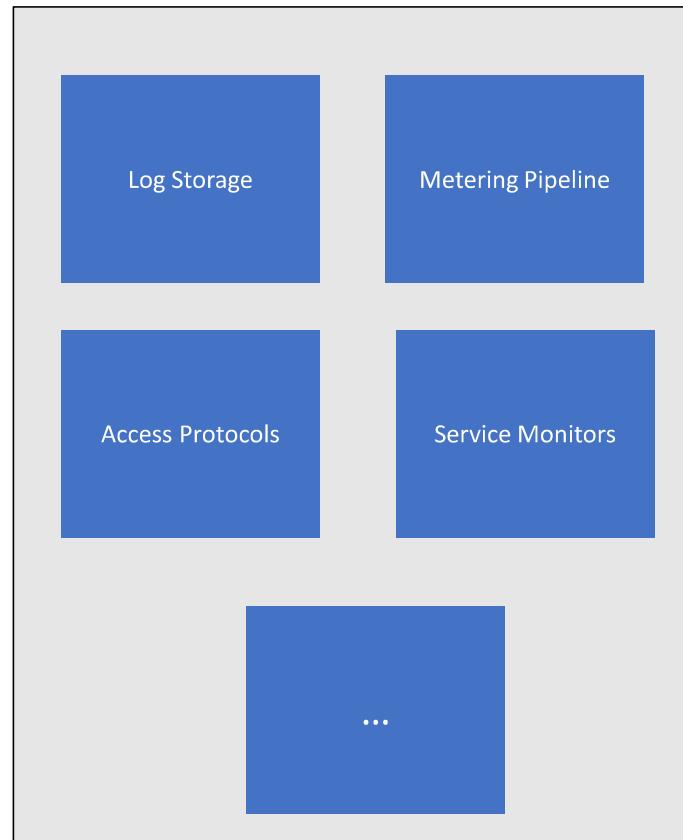
- Distributed collection of Row Objects
- Expression based operations and UDFs
- Logical plans and optimizer
- Fast/efficient internal representation

- Internally Rows, externally JVM objects
- Best of both worlds – “type safe + fast”

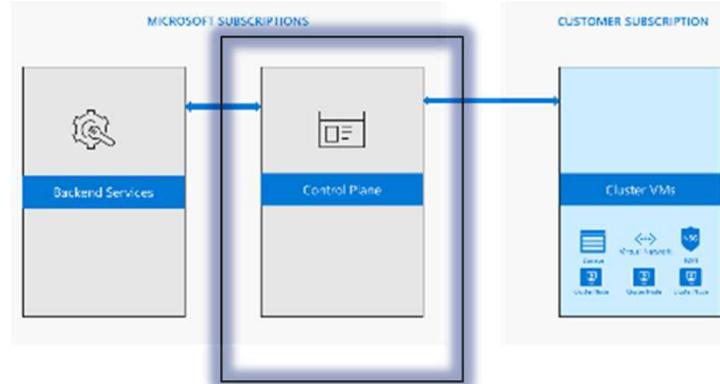
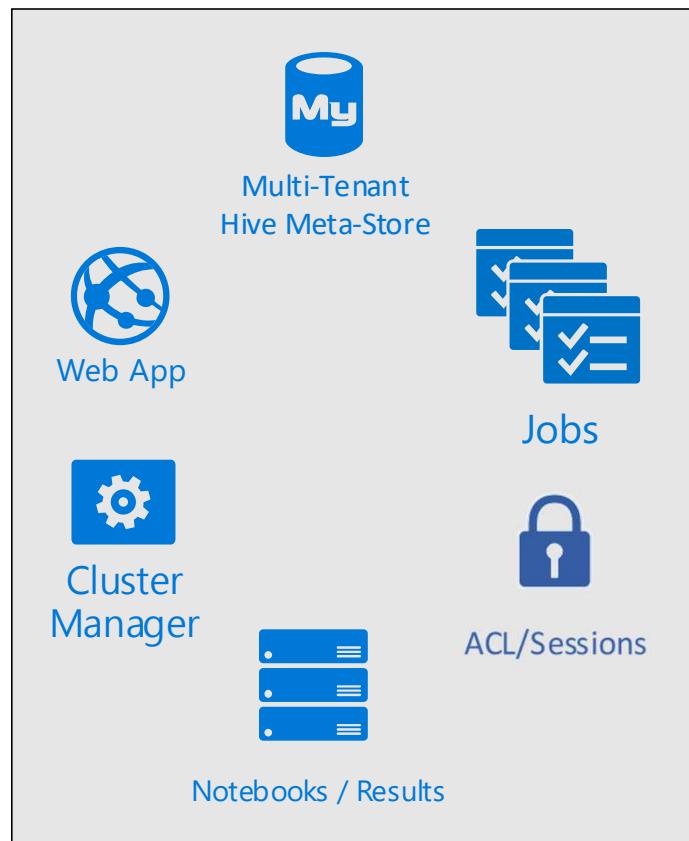
Getting the Head around



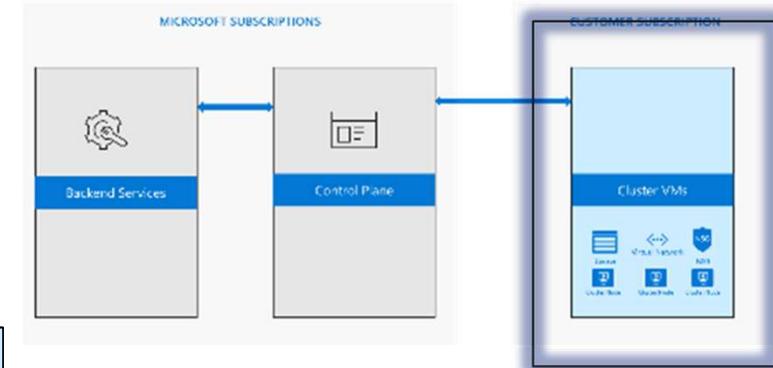
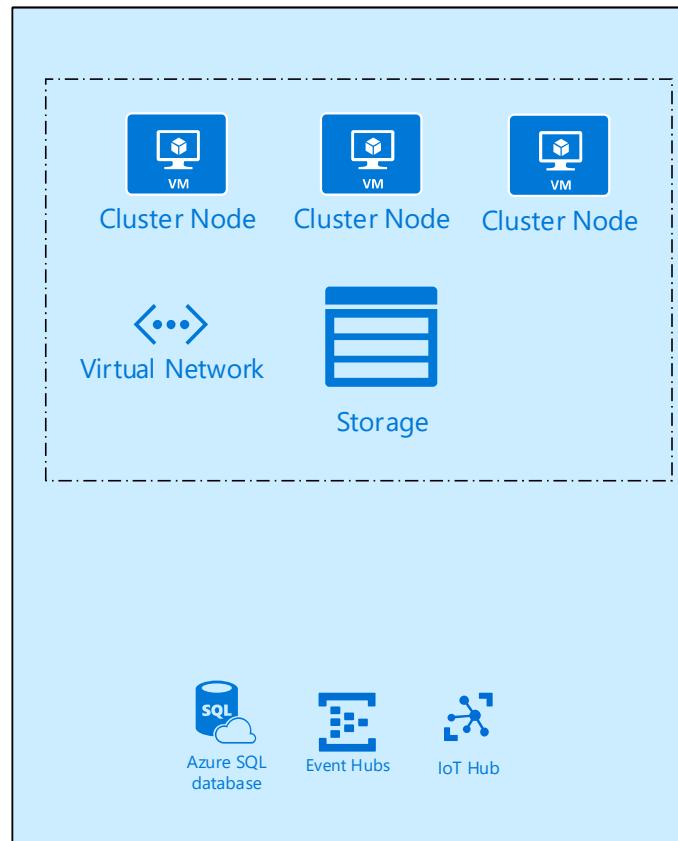
Backend Services



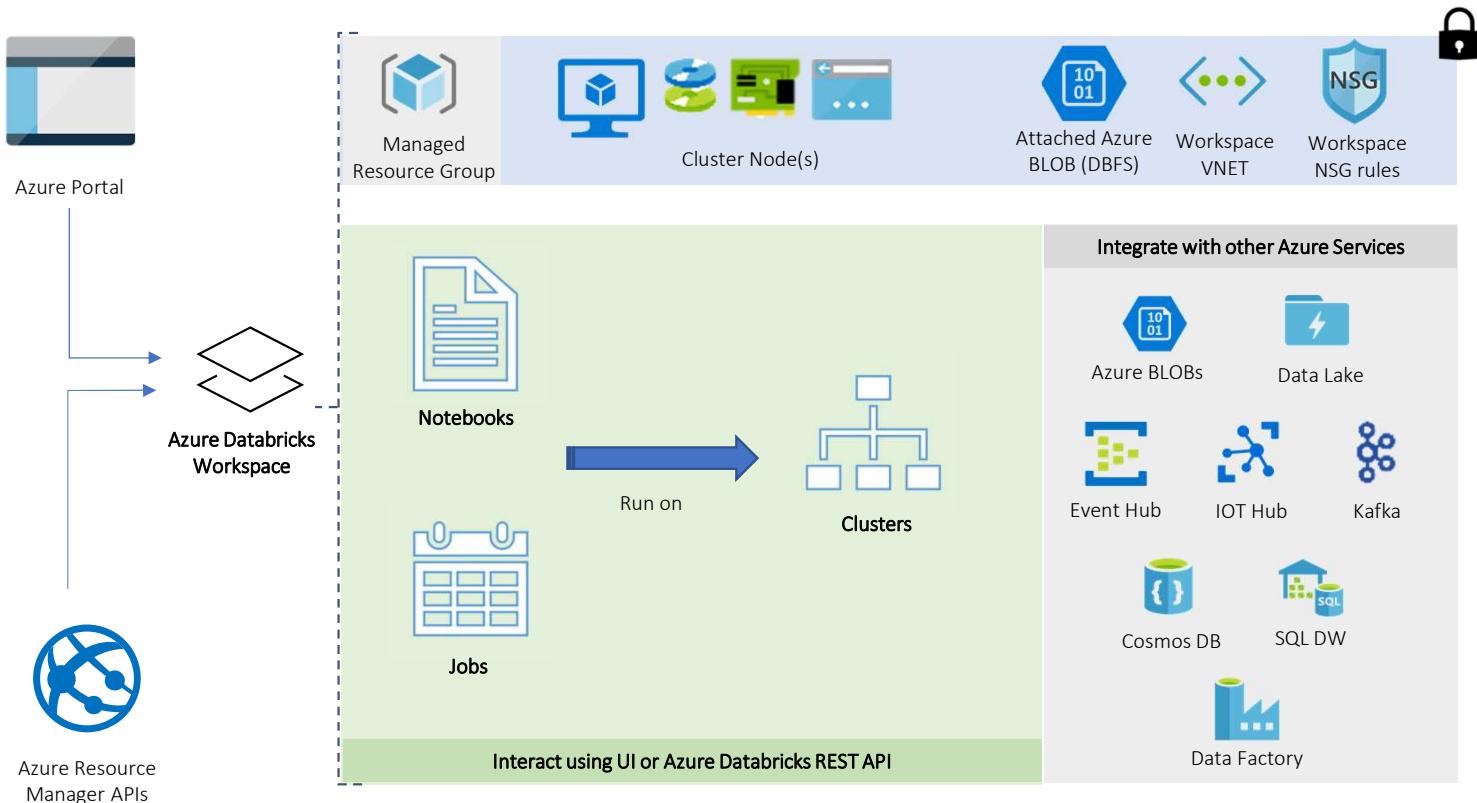
Control Plane



Customer Subscription

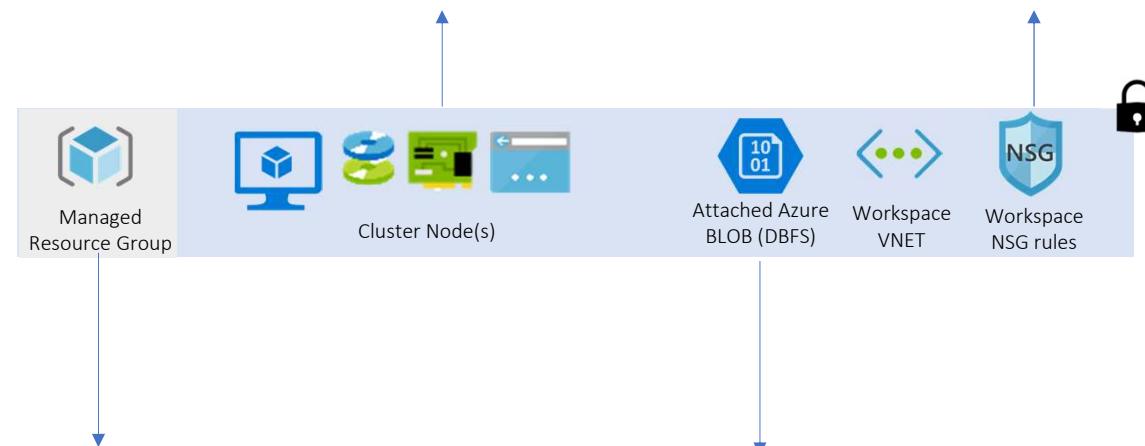


Customer view



Managed resource group

- For each driver or worker node in each cluster, a VM, SSD, Network Interface and IP address resource is created
- Each resource is tagged with
 - “Vendor : Databricks”
 - “ClusterName .” + {Cluster Name}
 - Custom Tags can be added at Cluster creation
- All resources in the RG are in a pre-configured VNET
- VNET allows communication between nodes and with the Control Plane
- NSG is locked contains the Control Plane IP addresses
- VNET & NSG are locked but customers can peer with other VNETs



- A locked RG is created in your subscription when you create the Azure Databricks Workspace
- RG Name = “databricks-rg”+ {workspace} + {random chars}

- Each workspace comes with an attached blob storage, contains workspace related configuration
- This can be access from Notebooks and Jobs as DBFS
- You can mount your own Azure Storage or ADLS to this DBFS

Regional distribution of the control plane



- Available today in 24 Regions / 6 Geographies
- Every geography has a Control Plane & Backend Services
- All dependent services run in that geography
- Data never leaves the geography
- Geographies :
<https://azure.microsoft.com/en-us/global-infrastructure/geographies/>

Who can access to Control Plane ?

Common Scenarios – When deploying a new feature, when making a fix, when adding a new region, automated jobs to read telemetry.

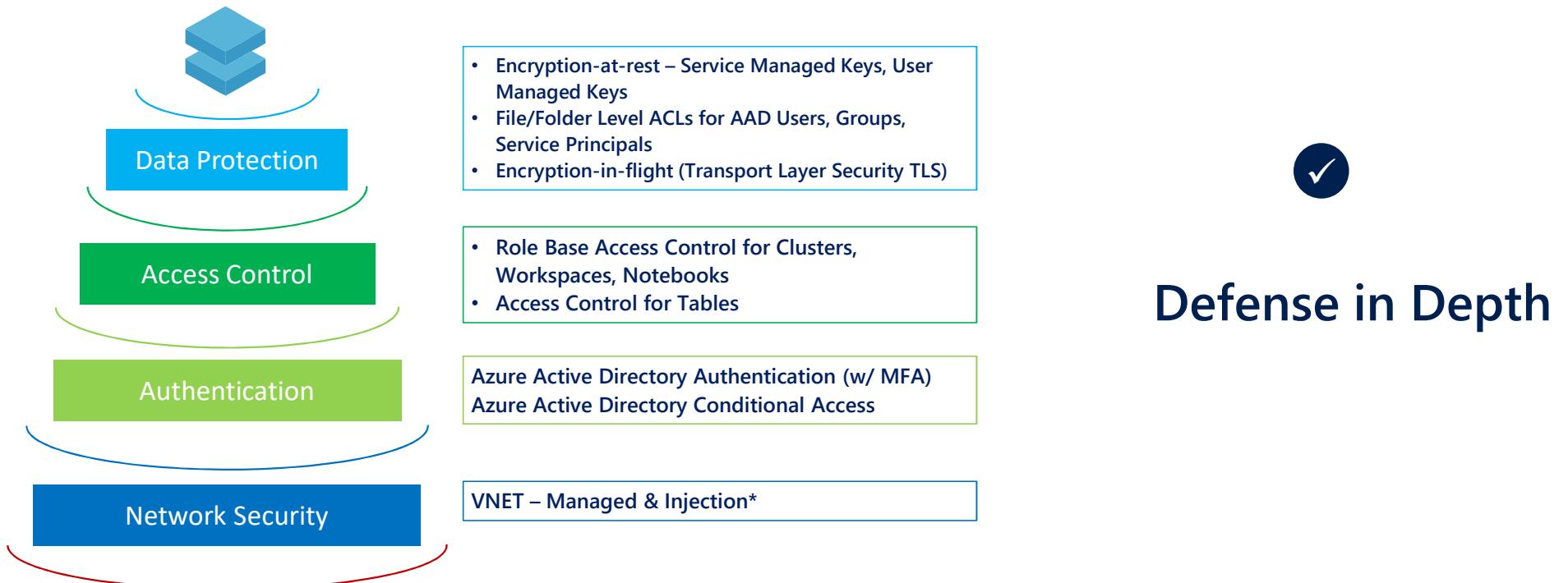
Policies and Procedures

- Follows all Azure Guidelines
- Access only allowed via secure hardware & JIT
- Logged & Audited

Architecture Summary

1. Azure Databricks is a Multi-Tenant Service
2. Control Plane is shared across all customers
3. Controls Planes are scoped to Geographies
4. Clusters are in the Customers Subscription and not shared
5. Access to Control Plane is audited, policies and procedures are enforced from the physical level to the logical level of security

Enterprise Grade Security that is Easy-to Use



* VNET Injection support in Public Preview

SECURE COLLABORATION

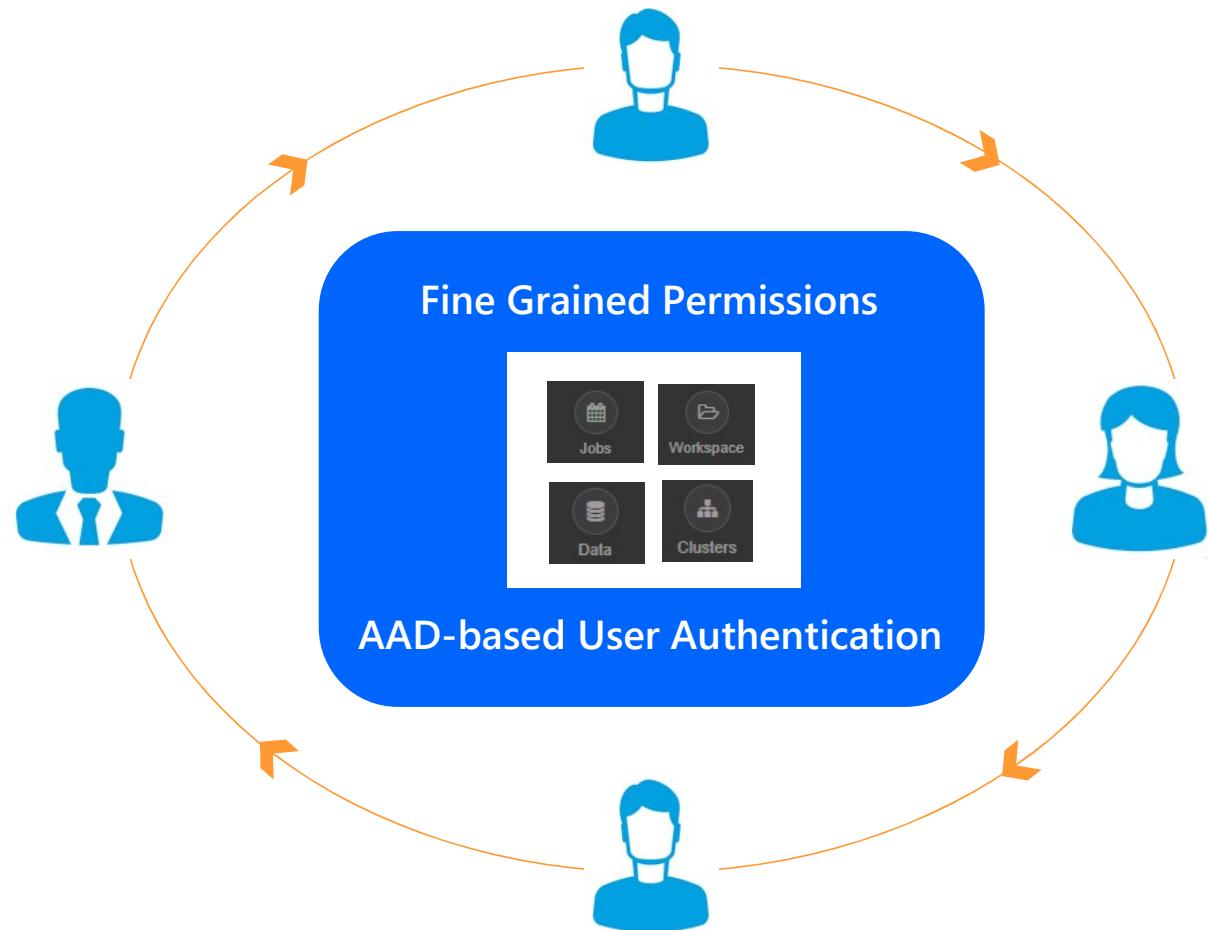
Azure Databricks enables *secure collaboration* between colleagues

- With Azure Databricks colleagues can *securely share* key artifacts such as Clusters, Notebooks, Jobs and Workspaces
- Secure collaboration is enabled through a combination of:

Fine grained permissions: Defines who can do what on which artifacts (access control)



AAD-based authentication: Ensures that users are actually who they claim to be



AZURE DATABRICKS INTEGRATION WITH AAD

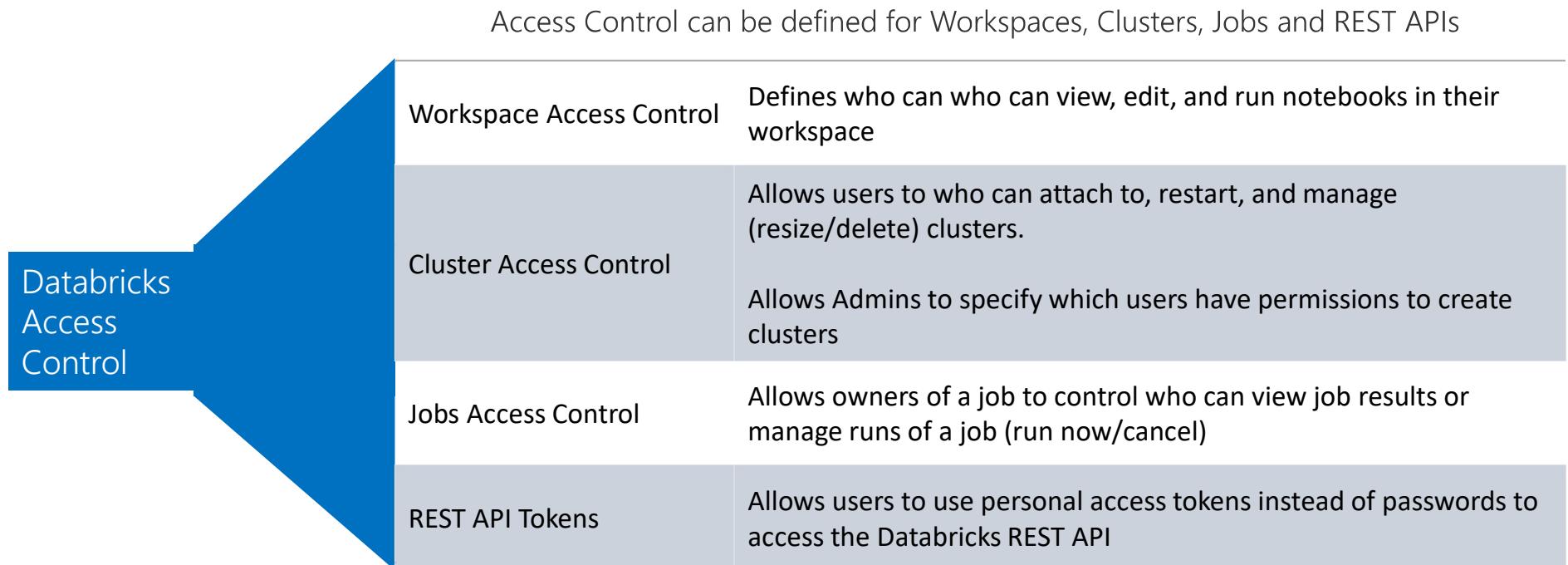
Azure Databricks is integrated with AAD—so Azure Databricks users are just regular AAD users

- There is no need to define users—and their access control—separately in Databricks.
- AAD users can be used directly in Azure Databricks for all user-based access control (Clusters, Jobs, Notebooks etc.).
- Databricks has delegated user authentication to AAD enabling single-sign on (SSO) and unified authentication.
- *Notebooks, and their outputs, are stored in the Databricks account. However, AAD-based access-control ensures that only authorized users can access them.*



DATABRICKS ACCESS CONTROL

Access control can be defined at the user level via the Admin Console



ENABLE/DISABLE ACCESS CONTROL

The screenshot shows the Microsoft Azure Databricks Settings page under the Access Control tab. It displays three sections: Workspace Access Control (Enabled), Cluster and Jobs Access Control (Enabled), and Personal Access Tokens (Enabled). Each section includes a 'What this means' link, a detailed description, and a 'Disable' button.

Workspace Access Control: Enabled

What this means >

Enabling Workspace access control will allow users to control who can attach to, restart, and manage (resize/delete) workspaces that they create. It will also allow administrators to control which users have permissions to create workspaces. In addition, jobs access control will also be turned on. Jobs access control allows owners of a workspace to control who can view job results or manage runs of a job (run now/cancel).

When workspace access control is enabled, admins will still have attach, restart and manage permissions on existing workspaces, as well as the ability to create workspaces.

If running jobs via the REST API: Before enabling workspace ACLs, users should ensure that the API user is identical to the workspace owner/creator. This will ensure seamless continued operation.

See the [Documentation](#) to learn more.

Cluster and Jobs Access Control: Enabled

What this means >

Enabling Cluster access control will allow users to control who can attach to, restart, and manage (resize/delete) clusters that they create. It will also allow administrators to control which users have permissions to create clusters. In addition, jobs access control will also be turned on. Jobs access control allows owners of a cluster to control who can view job results or manage runs of a cluster (run now/cancel).

When cluster access control is enabled, admins will still have attach, restart and manage permissions on existing clusters, as well as the ability to create clusters.

If running jobs via the REST API: Before enabling cluster ACLs, users should ensure that the API user is identical to the job owner/creator. This will ensure seamless continued operation.

See the [Documentation](#) to learn more.

Personal Access Tokens: Enabled

What this means >

Enabling Personal Access Tokens will allow users to use personal access tokens instead of passwords to access the Databricks REST API.

See the [Documentation](#) to learn more.

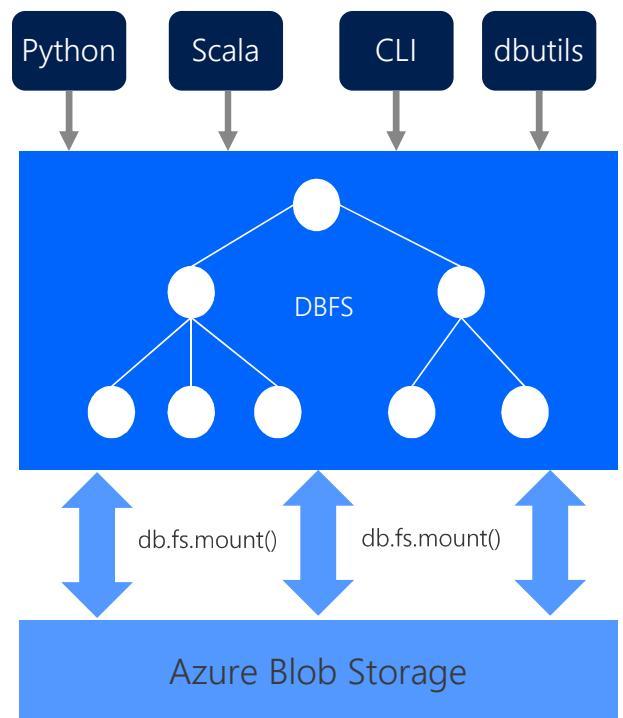
Access Control can be selectively enabled or disabled for:

- [Workspaces](#),
- [Clusters](#),
- [Jobs](#)
- [REST APIs](#)

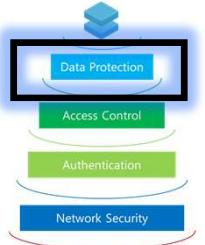
DATABRICKS FILE SYSTEM (DBFS)

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

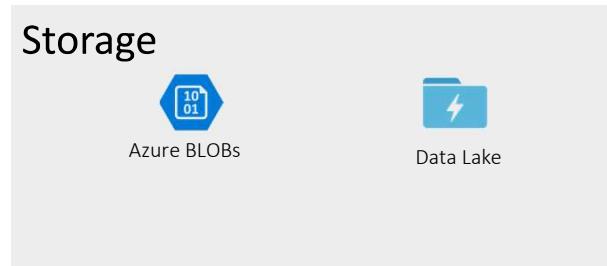
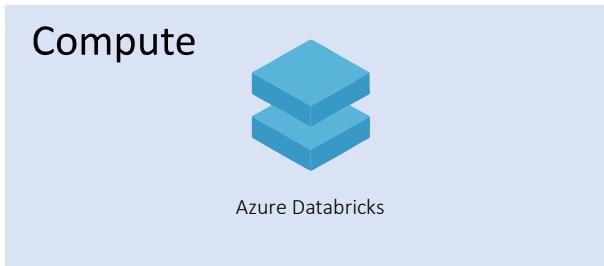
- Azure Storage buckets can be mounted in DBFS so that users can directly access them without specifying the storage keys
- DBFS mounts are created using `dbutils.fs.mount()`
- Azure Storage data can be cached locally on the SSD of the worker nodes
- Available in both Python and Scala and accessible 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



Data Protection | Encryption - Data at rest



- Azure Databricks has separation of compute and storage

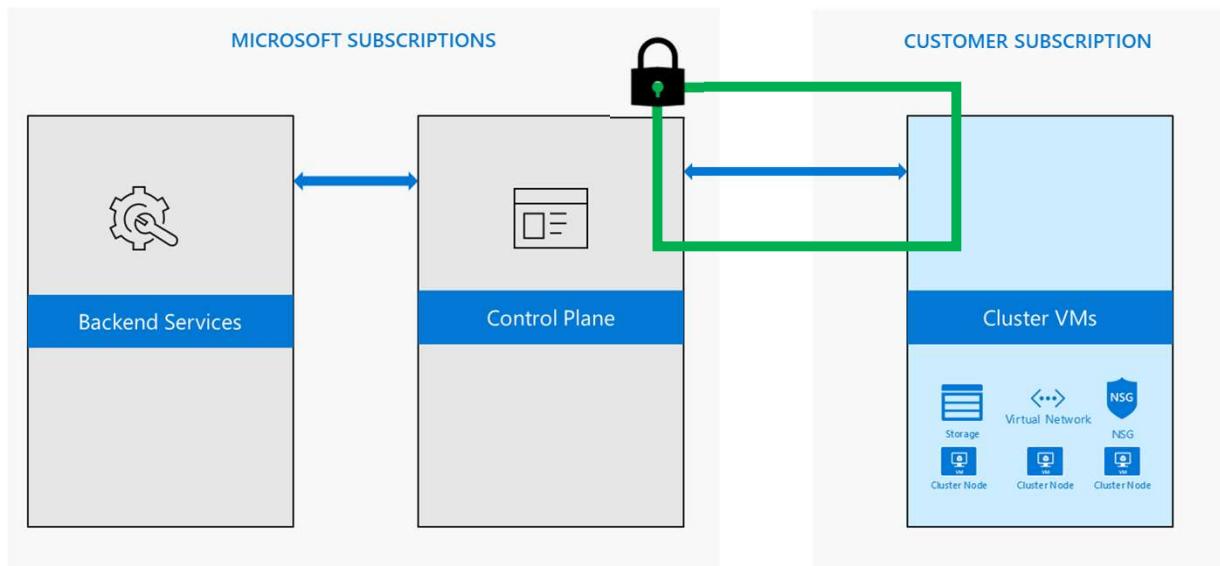


- Storage Services such as Azure Blob Store, Azure Data Lake Storage Provide
 - Encryption of Data
 - Customer Managed Keys
 - File/Folder Level ACLs (Azure Data Lake Storage)
- All Azure Databricks provided data stores are encrypted at rest

Data Protection | Encryption - Data in motion



- All the traffic from the Control Plane to the Clusters in the customer subscription is always encrypted with TLS.



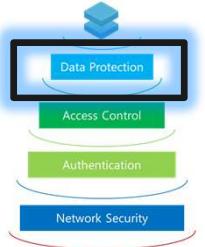
Secrets in Notebooks – Understanding the need



- Customers often connect to other Azure resources such as Azure BLOB Storage, Azure Data Lake, SQL DW from Azure Databricks
- A “Connection String” is required to connect to these services. This string may contain secrets.
- Customers don't want to store Secrets in the clear

Securing secrets in Notebooks

- Using our Secrets APIs, Secrets can be securely stored including in a Key Vault
- Authorized users can consume the secrets to access services but cannot see them.



<https://docs.azuredatabricks.net/user-guide/secrets/secret-scopes.html>

Microsoft Azure

HomePage / Create Secret Scope

Create Secret Scope | Cancel Create

A store for secrets that is identified by a name and backed by a specific store type. [Learn more](#)

Scope Name ?
key-vault-secrets

Azure Key Vault ?

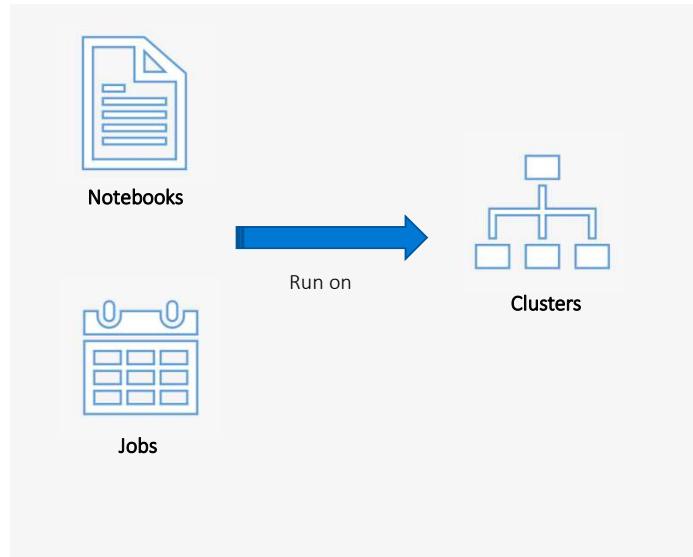
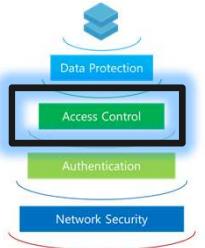
DNS Name

Resource ID

Azure Databricks Home Workspace Recent Data Clusters

Access Control

- Many users in the customers organization can use the Service
- Different users have different roles – Admin, Data Scientist, Engineers
- Access Controls lets you limit what users can do

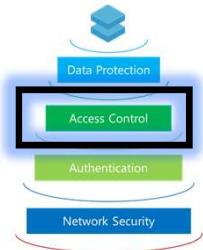


Access Control | Folders

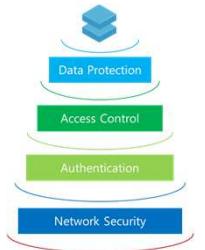


Ability	No Permissions	Read	Run	Edit	Manage
View items		X	X	X	X
Create, clone, import, export items		X	X	X	X
Run commands on notebooks			X	X	X
Attach/detach notebooks			X	X	X
Delete items				X	X
Move/rename items				X	X
Change permissions					X

Access Control | Notebooks



Ability	No Permissions	Read	Run	Edit	Manage
View cells		X	X	X	X
Comment		X	X	X	X
Run commands			X	X	X
Attach/detach notebooks			X	X	X
Edit cells				X	X
Change permissions					X



Microsoft Azure

Test (SQL)

Attached: AlwaysOnTelemetry | File | View: Code | Permissions | Run All | Clear

Cmd 1

Shift+Enter to run shortcuts

1

Permission Settings for: Test

Who has access:

admins (group)	Can Manage
Yatharth Gupta (yagupta@microsoft.com)	Can Manage

Add Users and Groups:

Can Read Add Done

This screenshot shows the Microsoft Azure Databricks interface. A modal window titled "Permission Settings for: Test" is open, displaying the current access settings for the workspace. It lists two entries: "admins (group)" and "Yatharth Gupta (yagupta@microsoft.com)", both with "Can Manage" permissions. Below this, there is a section for adding users and groups, with a dropdown menu, a "Can Read" button, and an "Add" button. A "Done" button is at the bottom right of the modal.

Access Control | Jobs



Ability	No Permissions	Can View	Can Manage Run	Is Owner	Can Manage (admin)
View job details and settings	X	X	X	X	X
View results, Spark UI, logs of a job run		X	X	X	X
Run now			X	X	X
Cancel run			X	X	X
Edit job settings				X	X
Modify permissions				X	X

Access Control | Clusters



Ability	No Permissions	Can Attach To	Can Restart	Can Manage
Attach notebook to cluster		X	X	X
View Spark UI		X	X	X
View cluster metrics		X	X	X
Terminate cluster			X	X
Start cluster			X	X
Restart cluster			X	X
Edit cluster				X
Attach library to cluster				X
Resize cluster				X
Modify permissions				X

Access Control | Tables

Objects

CATALOG | DATABASE | TABLE | VIEW | FUNCTION | ANONYMOUS FUNCTION
| ANY FILE



Privileges

- | | |
|----------------|---|
| SELECT | - read access to an object |
| CREATE | - ability to create an object (eg. Table in a Database) |
| MODIFY | - ability to add/delete/modify data in an Object |
| READ_METADATA | - ability to read Meta data about an object |
| ALL_PRIVILEGES | - all of the above |

Access Control | Tables

[GRANT | DENY]

ON [OBJECT]

TO [USER]

[PRIVILEGE_TYPE]



- Access Control on Tables limits to SQL and Python only. This ensures that low level commands cannot be used to bypass these restrictions.
- High concurrence clusters provide isolation between users.

Access Control | Tables

Microsoft Azure

Create Cluster

New Cluster

Cancel **Create Cluster** 2-8 Workers: 112.0-448.0 GB Memory, 16-64 Cores, 4-16 DBU
1 Driver: 56.0 GB Memory, 8 Cores, 2 DBU Cost \$0.55 per DBU

Cluster Name: Table Access Control

Cluster Mode:

- High Concurrency
Optimized to run concurrent SQL, Python, and R workloads.
Does not support Scala. Previously known as Serverless.
- Standard
Recommended for single-user clusters. Can run SQL, Python, R, and Scala workloads.

Databricks Runtime Version: Latest stable (Scala 2.11)

Python Version: 2

Driver Type: Same as worker

Worker Type: Standard_DS13_v2

Min Workers: 2 Max Workers: 8 Enable autoscaling

Auto Termination: Terminate after 0 minutes of inactivity

Table Access Control:

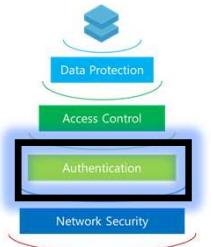
Enable table access control and only allow Python and SQL commands

Spark Tags Logging Init Scripts

Spark Configs



Authentication

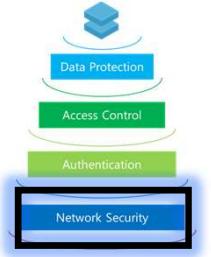


- Azure Databricks support Azure Active Directory as an Authentication provider.
- This is pre-configured with zero setup needed. It includes the ability for the organization to enable multi-factor authentication.
- Support for conditional access has been added for additional policies

<https://docs.microsoft.com/en-us/azure/active-directory/conditional-access/overview>

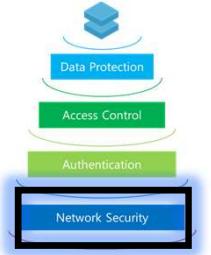
<https://docs.azuredatabricks.net/administration-guide/cloud-configurations/azure/conditional-access.html#id1>

Network Security | Managed VNETs



- Clusters (VMs) are always deployed in the customer's subscription. We deploy these in a VNET we create for the Customer.
- In this mode, the VNET and accompanying NSG rules are managed by us.
- We allow for Customer's to be able to Peer this with other VNETs

Network Security | Managed VNETs NSG Rules



Inbound security rules

PRIORITY	NAME	PORT	PROTOCOL	SOURCE	DESTINATION	ACTION	...
100	databricks-control-plane-ssh	22	Any	40.83.178.242	Any	✓ Allow	...
110	databricks-control-plane-worker-pr...	5557	Any	40.83.178.242	Any	✓ Allow	...
200	databricks-worker-to-worker	Any	Any	VirtualNetwork	Any	✓ Allow	...
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	✓ Allow	...
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	✓ Allow	...
65500	DenyAllInBound	Any	Any	Any	Any	✗ Deny	...

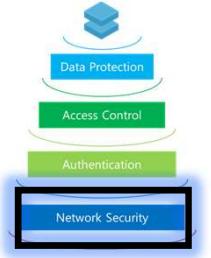
Outbound security rules

PRIORITY	NAME	PORT	PROTOCOL	SOURCE	DESTINATION	ACTION	...
65000	AllowVnetOutBound	Any	Any	VirtualNetwork	VirtualNetwork	✓ Allow	...
65001	AllowInternetOutBound	Any	Any	Any	Internet	✓ Allow	...
65500	DenyAllOutBound	Any	Any	Any	Any	✗ Deny	...

Network Security | VNET Injection

You can signup for this today !

<https://aka.ms/adbvnet>



When to chose this option ?

- You want to use your own VNET
- You want to restrict access to your other services using Service Endpoint
- You want to use User Defined Routes
- You want to use an Application Layer Firewall

Security and Networking Summary

1. Data Protection – With Azure Databricks, storage is always remote and these storage services support encryption, customer managed keys and file/folder ACLs.
2. Access Control – Rich RBAC control available in Azure Databricks for Clusters, Notebooks, Tables etc.
3. Authentication – Always AAD
4. Network Security – Two options, designed for security and flexibility.

Compliance

- ISO 27001
- ISO 27018
- HIPAA
- SOC2, Type 2



Service Level Agreement

99.95% uptime SLA

MONTHLY UPTIME PERCENTAGE	SERVICE CREDIT
< 99.95%	10%
< 99%	25%

https://azure.microsoft.com/en-us/support/legal/sla/databricks/v1_0/

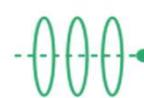
Databricks for Data Engineers

Data Engineering System Requirements

Data Management



Big Data



Fast Data



Schema Mgmt



Data Consistency



Fast Reads

Platform Management



Resource Estimation



Failure Recovery



Automatic Upgrades



Distributed Framework



Elastic Scalability



Tooling & Integration

Infrastructure Management



Managed as a Service

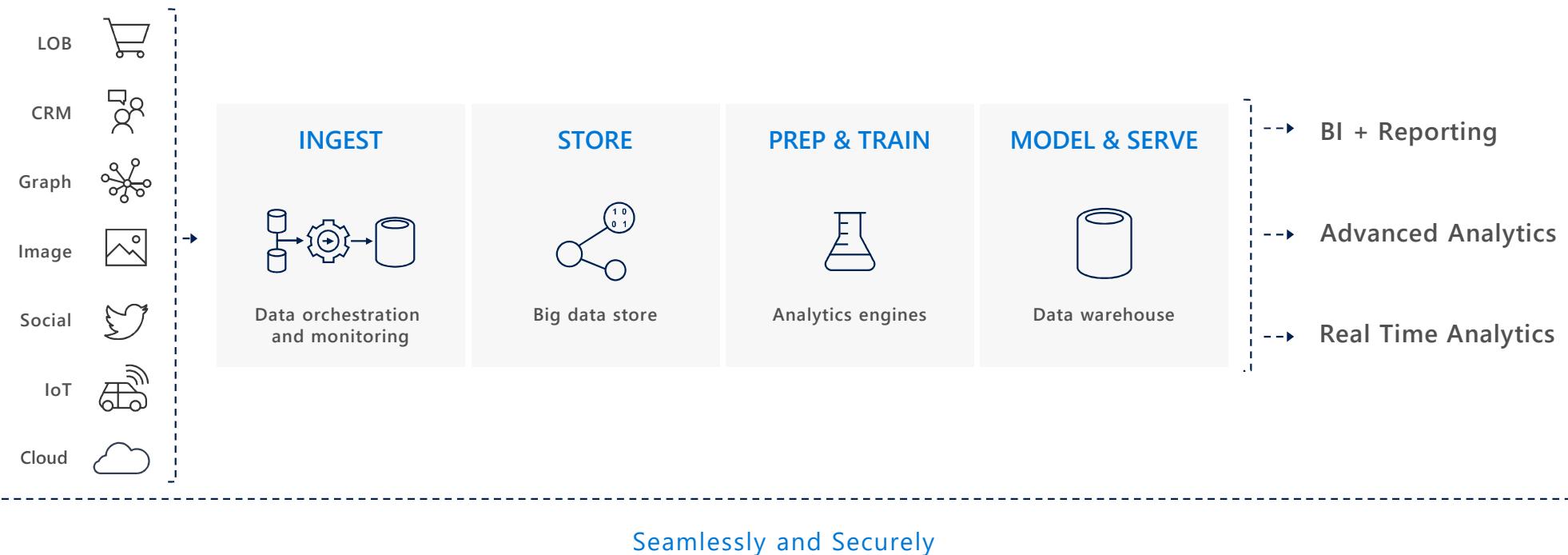
Why Data Engineering is Hard?

- Various sources/formats
- Schema mismatch
- Different representation
- Corrupted files and data
- Scalability
- Schema evolution
- Monitoring & Auditing
- Multi activity integration
- Evolve as fast as the business

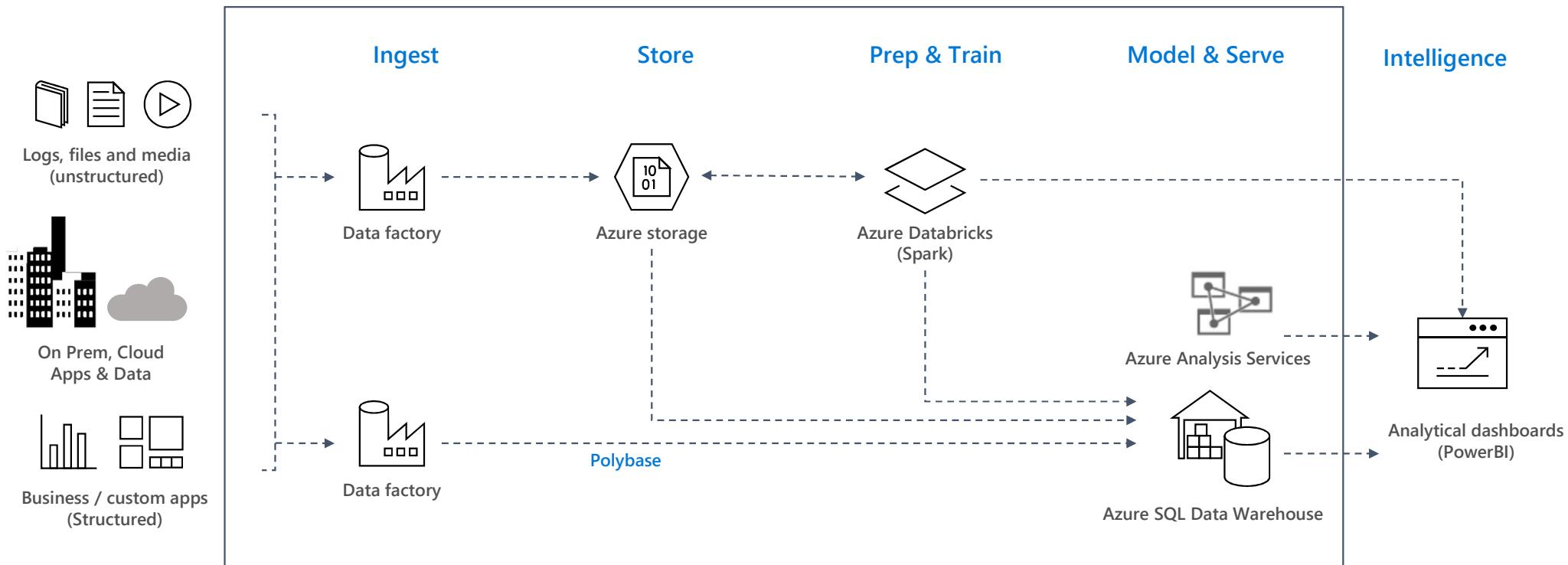
Data Engineering Scenarios

**Modern DW for BI
Advance Analytics for Apps**

Advanced analytics and AI

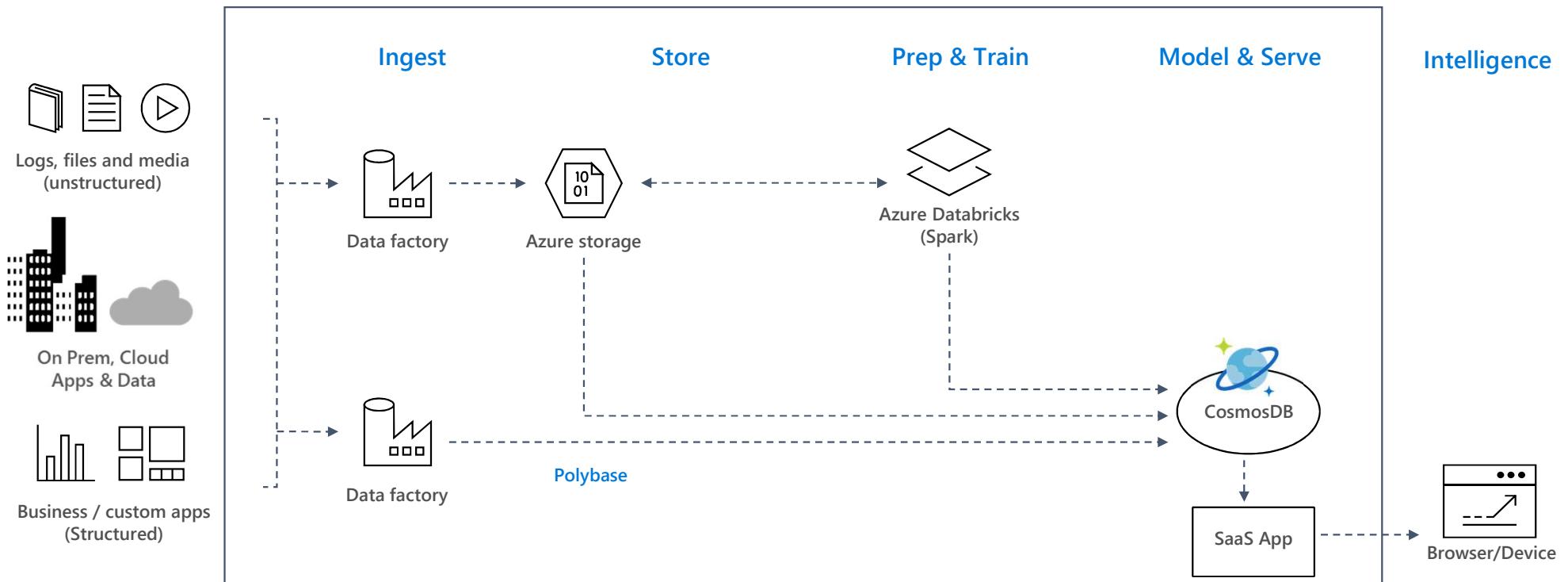


Modern DW for BI



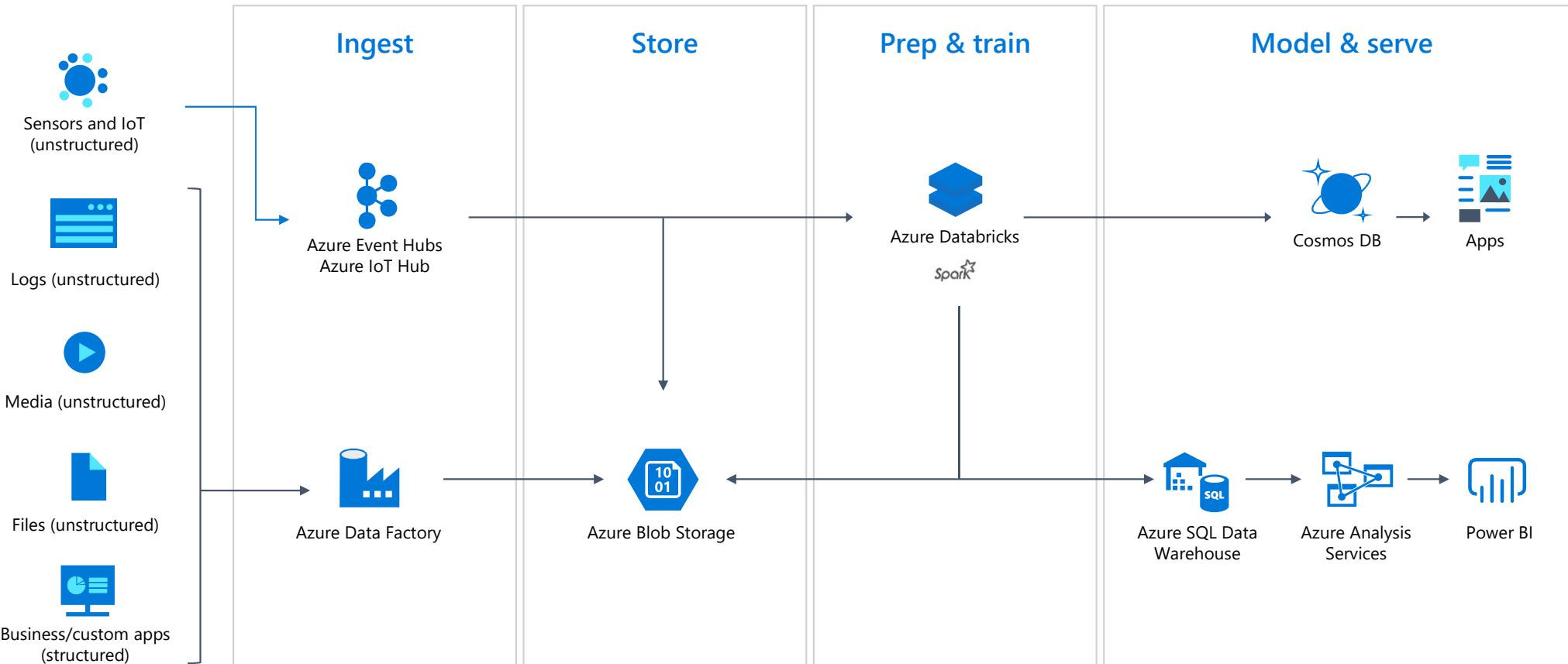
AZURE DATA FACTORY ORCHESTRATES DATA PIPELINE ACTIVITY WORKFLOW & SCHEDULING

Advance Analytics for Apps



AZURE DATA FACTORY ORCHESTRATES DATA PIPELINE ACTIVITY WORKFLOW & SCHEDULING

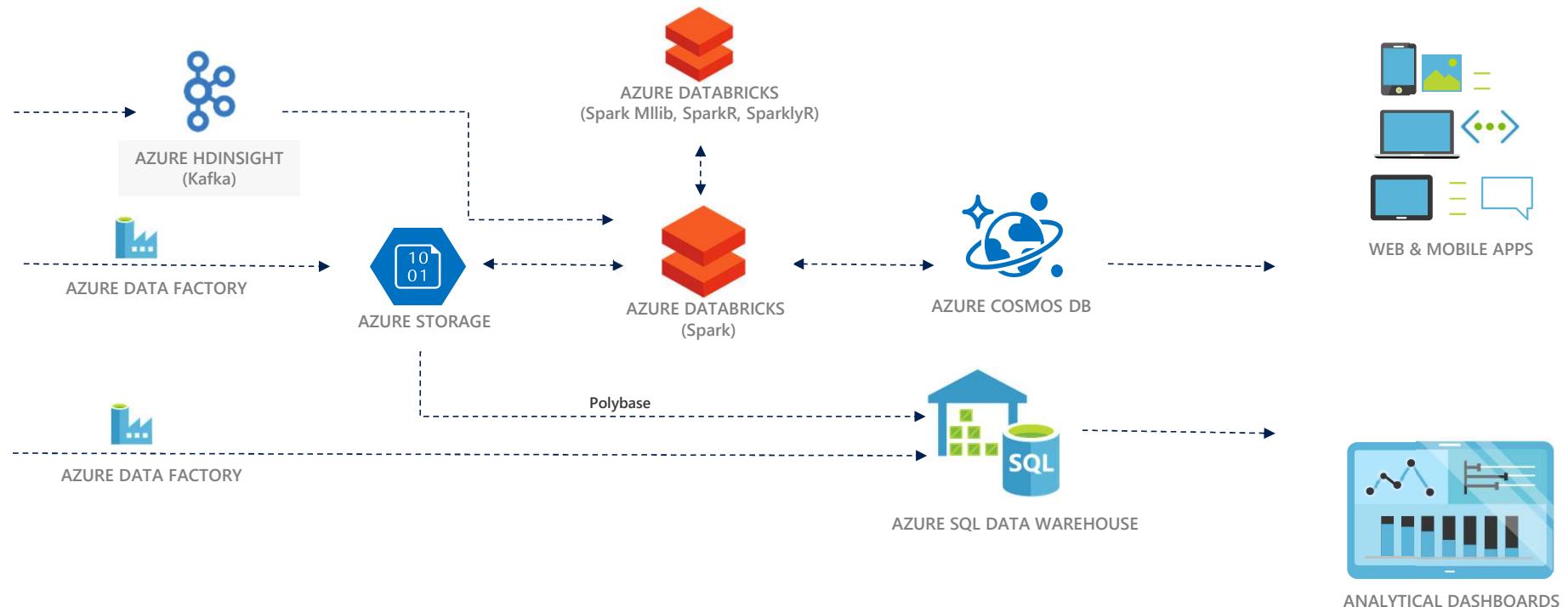
Real-Time Complex / Data Event Processing



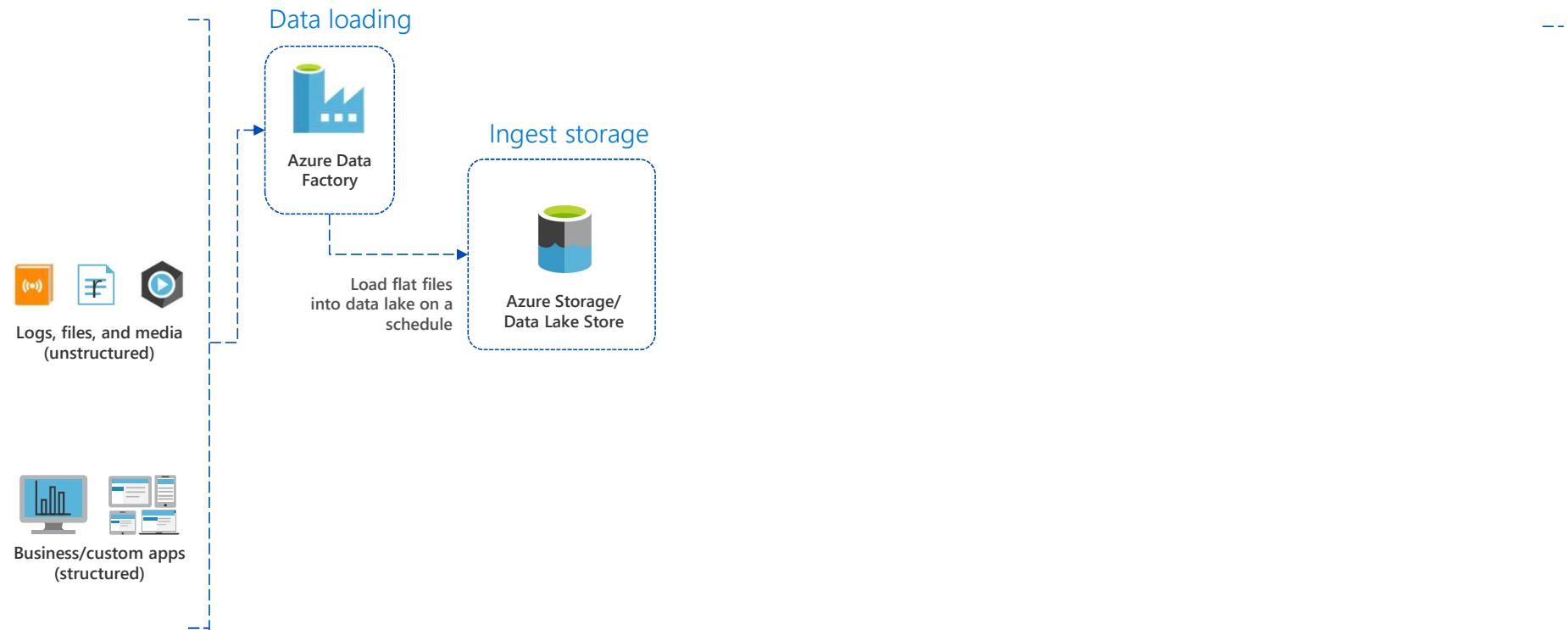
Microsoft Azure also supports other Big Data services like Azure HDInsight and Azure Data Lake to allow customers to tailor the above architecture to meet their unique needs.

Big Data Lambda Architecture

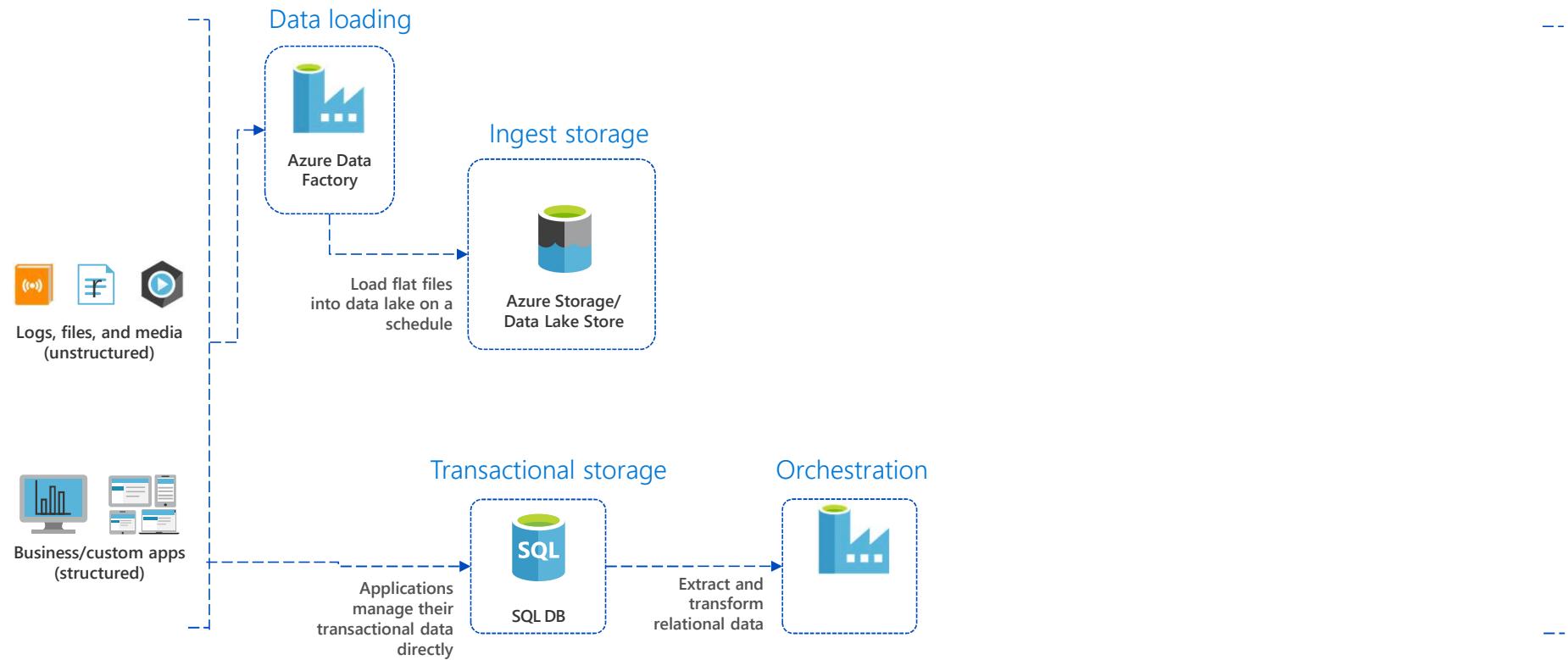
With Azure Databricks



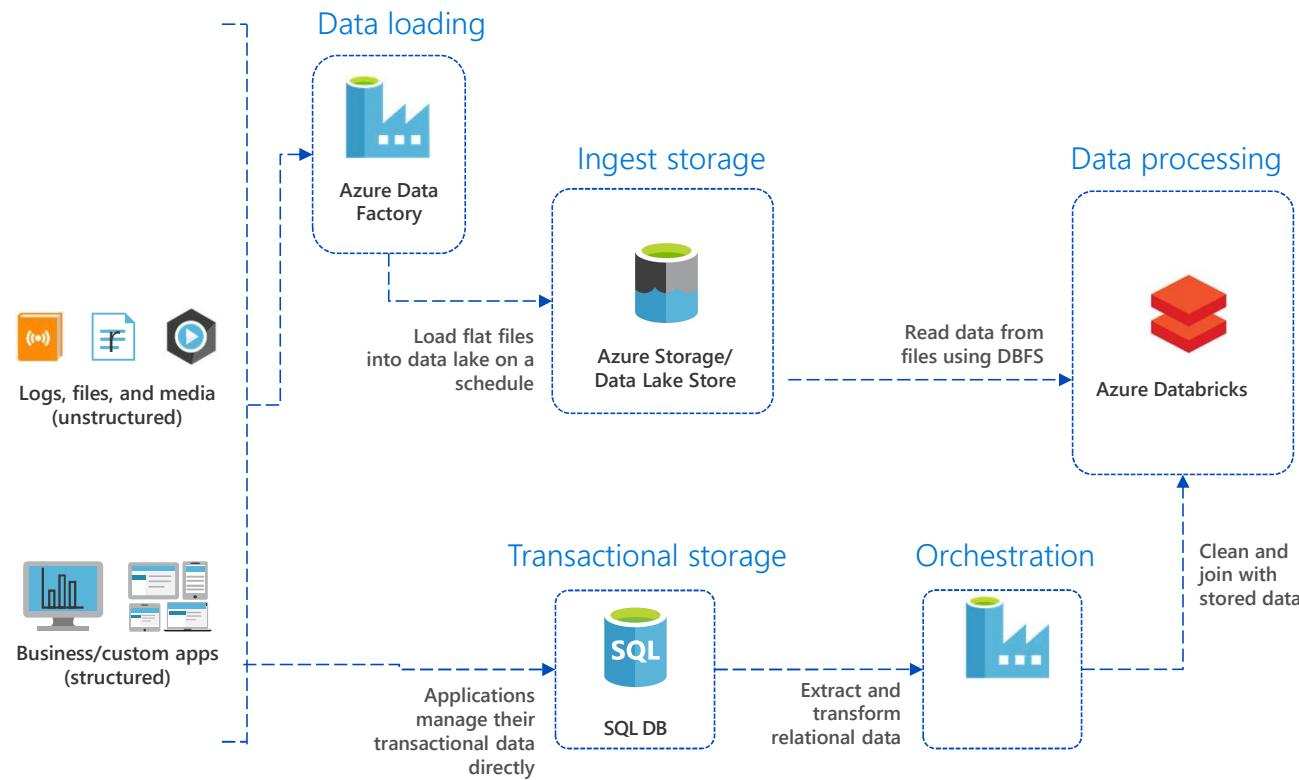
Data Ingestion



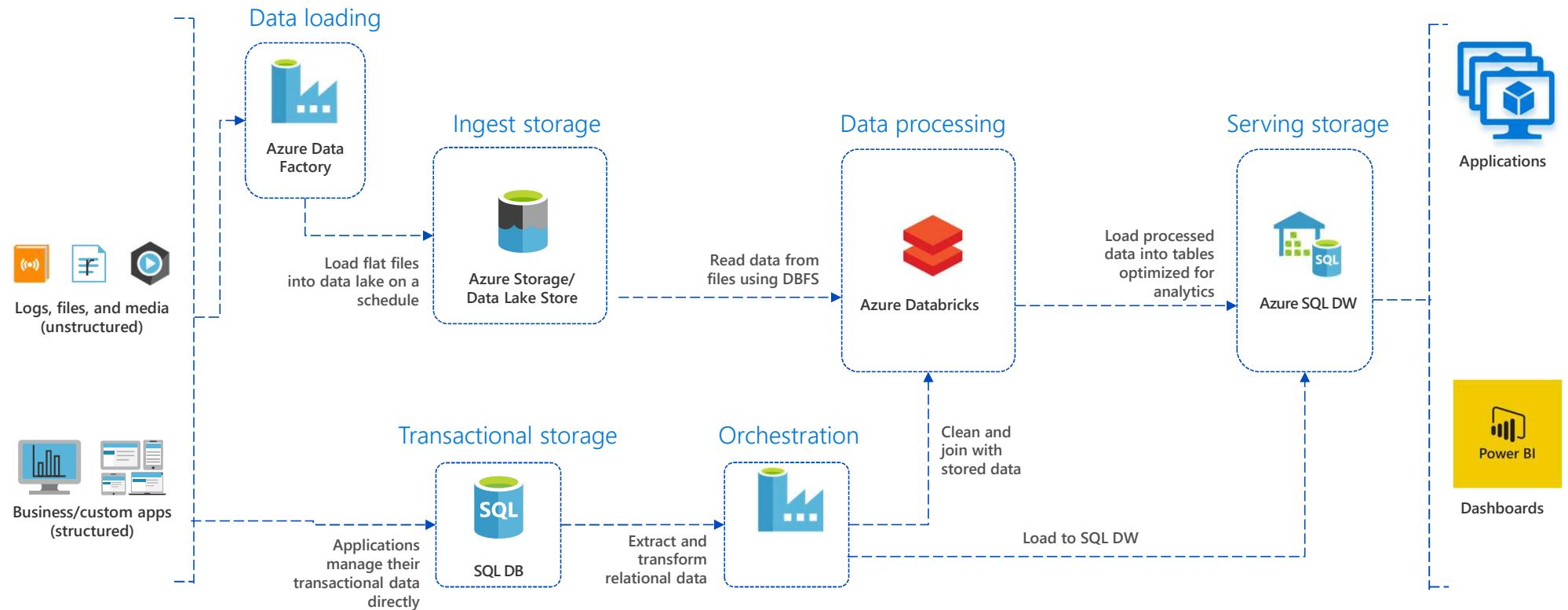
Data Ingestion



Data cleansing and prep



Analysis and Post processing



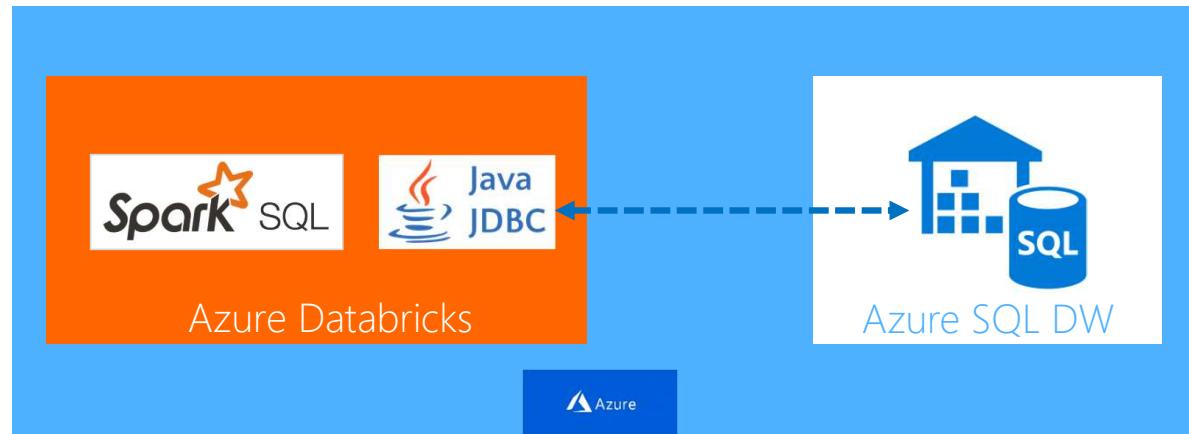
A Z U R E S Q L D W I N T E G R A T I O N

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.



COSMOS DB INTEGRATION

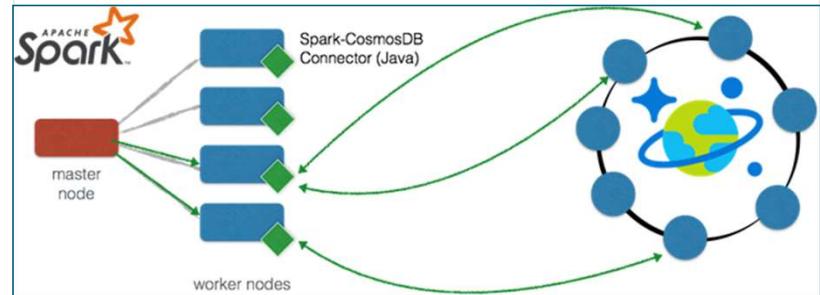
The Spark connector enables real-time analytics over globally distributed data in Azure Cosmos DB



[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 ntainer Name}@{}Your Storage Account Name}.blob.core.windows.net/{Your Directory Name}")
```

AZURE DATA LAKE INTEGRATION

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}")
```

POWER BI INTEGRATION

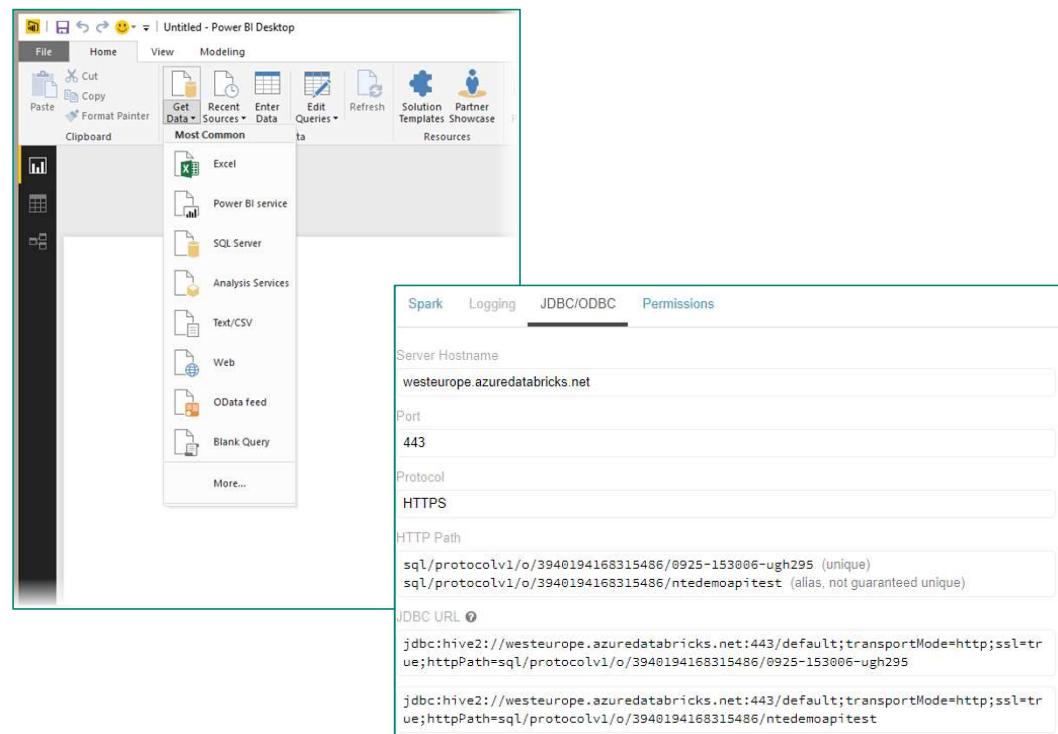
Enables powerful visualization of data in Spark with Power BI



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.

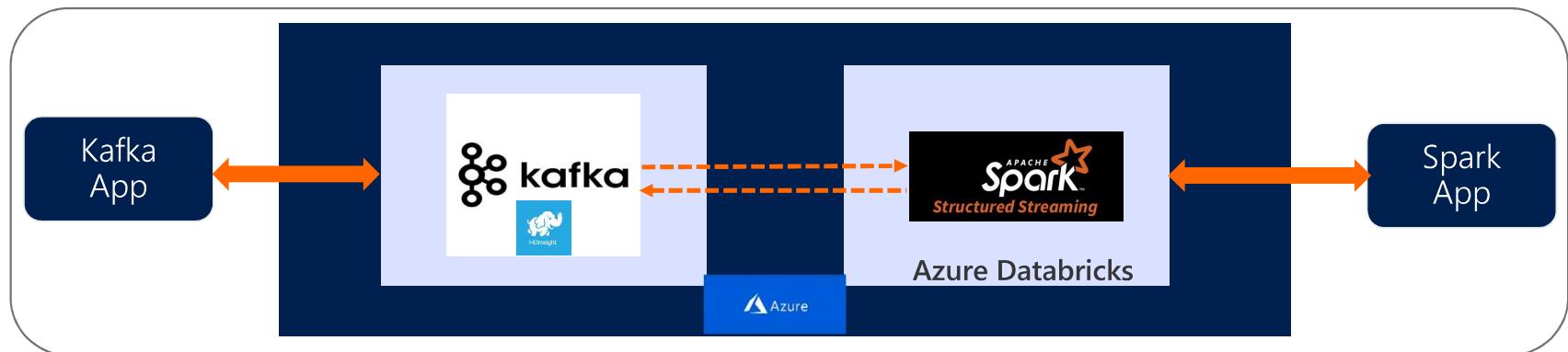
- This server listens on port 10000 and it is not accessible outside the subnet where the cluster is running.
- Azure Databricks uses a public HTTPS gateway
- The JDBC/ODBC connection information can be obtained from the Cluster UI directly as shown in the figure.
- When establishing the connection, 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.



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

ETL – Databricks Spark

```
spark.read.json("/source/path")
  .filter(...)
  .agg(...)
  .write.mode("append")
  .parquet("/output/path")
```

EXTRACT

TRANSFORM

LOAD

An ETL Query in Apache Spark

```
val csvTable = spark.read.csv("/source/path")
val jdbcTable = spark.read.format("jdbc")
    .option("url", "jdbc:postgresql:...")
    .option("dbtable", "TEST.PEOPLE")
    .load()

csvTable
    .join(jdbcTable, Seq("name"), "outer")
    .filter("id <= 2999")
    .write
    .mode("overwrite")
    .format("parquet")
    .saveAsTable("outputTableName")
```

Extract
EXTRACT

Transform

Load

Json: Dealing with Corrupt Records

```
{ "a":1, "b":2, "c":3 }  
{ "a": { , b:3 }  
{ "a":5, "b":6, "c":7 }
```

_corrupt_record	a	b	c
null	1	2	3
{"a":{, b:3}}	null	null	null
null	5	6	7

```
spark.read  
.option("mode", "PERMISSIVE")  
.option("columnNameOfCorruptRecord", "_corrupt_record")  
.json(corruptRecords)  
.show()
```

The default can be configured via
spark.sql.columnNameOfCorruptRecord

Json: Dealing with Corrupt Records

```
{ "a":1, "b":2, "c":3 }
```

```
{"a": {, b:3}}
```

```
{ "a":5, "b":6, "c":7 }
```

```
spark.read
```

```
.option("mode", "DROPMALFORMED")
```

```
.json(corruptRecords)
```

```
.show()
```

	a	b	c
	1	2	3
	5	6	7

Json: Dealing with Corrupt Records

```
{"a":1, "b":2, "c":3}  
{ "a": { , b:3 }  
{"a":5, "b":6, "c":7 }
```

```
spark.read  
.option("mode", "FAILFAST")  
.json(corruptRecords)  
.show()
```

```
org.apache.spark.sql.catalyst.json  
.SparkSQLJsonProcessingException:  
Malformed line in FAILFAST mode:  
 {"a":{ , b:3 } }
```

CSV: Dealing with Corrupt Records

```
year,make,model,comment,blank  
"2012","Tesla","S","No comment",  
1997,Ford,E350,"Go get one now they",  
2015,Chevy,Volt
```

```
spark.read.  
.option("mode", "PERMISSIVE")  
.csv(corruptRecords)  
.show()
```

_c0	_c1	_c2	_c3	_c4
year	make	model	comment	blank
2012	"Tesla"	"S"	"No comment"	null
1997	Ford	E350	"Go get one now ..."	null
2015	Chevy	Volt		null

CSV: Dealing with Corrupt Records

```
year,make,model,comment,blank  
"2012","Tesla","S","No comment",  
1997,Ford,E350,"Go get one now they",  
2015,Chevy,Volt
```

```
spark.read  
.option("header", true)  
.option("mode", "PERMISSIVE")  
.csv(corruptRecords)  
.show()
```

year	make	model	comment	blank
2012	"Tesla"	"S"	"No comment"	null
1997	Ford	E350	"Go get one now ..."	null
2015	Chevy	Volt		null

CSV: Dealing with Corrupt Records

```
val schema = "col1 INT, col2 STRING, col3 STRING, col4 STRING, " +  
  "col5 STRING, __corrupted_column_name STRING"  
spark.read  
.option("header", true)  
.option("mode", "PERMISSIVE")  
.csv(corruptRecords)  
.show()
```

col1	col2	col3	col4	col5	__corrupted_column_name
2012	"Tesla"	"S"	"No comment"	null	null
1997	Ford	E350	"Go get one now ..."	null	null
2015	Chevy	Volt	null	null	2015, Chevy, Volt

CSV: Dealing with Corrupt Records

```
year,make,model,comment,blank  
"2012","Tesla","S","No comment",  
1997,Ford,E350,"Go get one now they",  
2015,Chevy,Volt
```

```
spark.read  
.option("mode", "DROPIMALFORMED")  
.csv(corruptRecords)  
.show()
```

year	make	model	comment	blank
2012	Tesla	S	No comment	null
1997	Ford	E350	Go get one now th...	null

Schema Inference – semi-structured files

```
{ "a":1, "b":2, "c":3 }  
{ "e":2, "c":3, "b":5 }  
{ "a":5, "d":7 }
```

```
spark.read  
.json("/source/path")  
.printSchema()
```

```
root  
|-- a: long (nullable = true)  
|-- b: long (nullable = true)  
|-- c: long (nullable = true)  
|-- d: long (nullable = true)  
|-- e: long (nullable = true)
```

Schema Inference – semi-structured files

```
{"a":1, "b":2, "c":3.1}
```

```
{"e":2, "c":3, "b":5}
```

```
{"a":"5", "d":7}
```

```
spark.read  
.json("/source/path")  
.printSchema()
```

root

```
|-- a: string (nullable = true)  
|-- b: long (nullable = true)  
|-- c: double (nullable = true)  
|-- d: long (nullable = true)  
|-- e: long (nullable = true)
```

User-specified Schema

```
{ "a":1, "b":2, "c":3 }  
{ "e":2, "c":3, "b":5 }  
{ "a":5, "d":7 }
```

	a	b
	1	2
	null	5
	5	null

```
val schema = new StructType()  
  .add("a", "int")  
  .add("b", "int")
```

```
spark.read  
  .json("/source/path")  
  .schema(schema)  
  .show()
```

Traditional ETL



Raw, dirty, un/semi-structured is data dumped as files

Periodic jobs run every few hours to convert raw data to structured data ready for further analytics

Traditional ETL



Hours of delay before taking decisions on latest data

Unacceptable when time is of essence
[intrusion detection, anomaly detection, etc.]

Streaming ETL w/ Structured Streaming



Structured Streaming enables raw data to be available as structured data as soon as possible

Azure Databricks Delta

Public Preview

Optimize Spark Queries for
faster analytics,
better data reliability guarantees,
and simplified data pipelines

Why Databricks Delta?

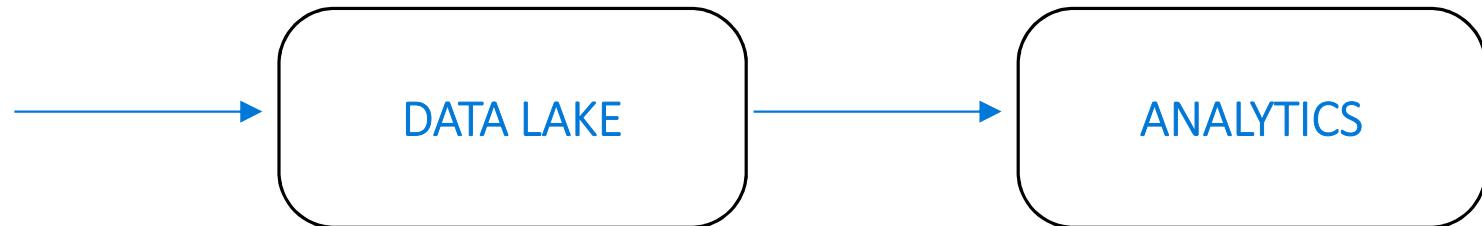
Data Engineering Challenges

1. Data Reliability Issues

Inconsistent Data
Evolving Schema

LOTS OF NEW DATA

User Behavior Data
Click Streams
Sensor Data (IoT)
Video/Speech
Usage/Billing Data
Machine Telemetry
Commerce Data



2. Performance Challenges

Slow and Costly
Worse at Scale

3. System Complexity

Separate batch & streaming
Low-level code for pipelines

Why Databricks Delta?

Data Engineering Challenges

1. Data Reliability

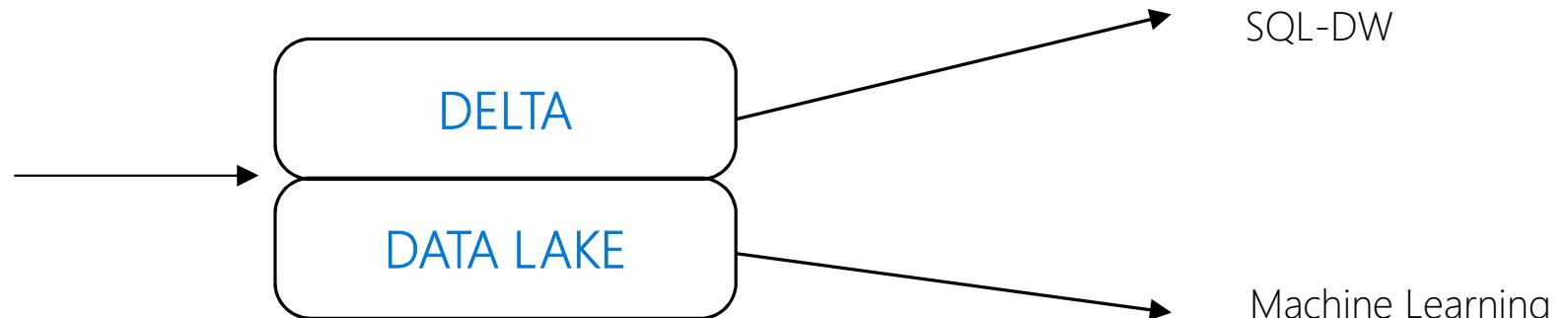
ACID Compliant Transactions
Schema Enforcement & Evolution

LOTS OF NEW DATA

User Behavior Data
Click Streams
Sensor data (IoT)
Video/Speech
Usage/Billing data
Machine Telemetry
Commerce Data

2. Spark Query Performance

Fast at Scale (10-100x Faster)
Cheaper to Operate
Indexing, Statistics & Caching



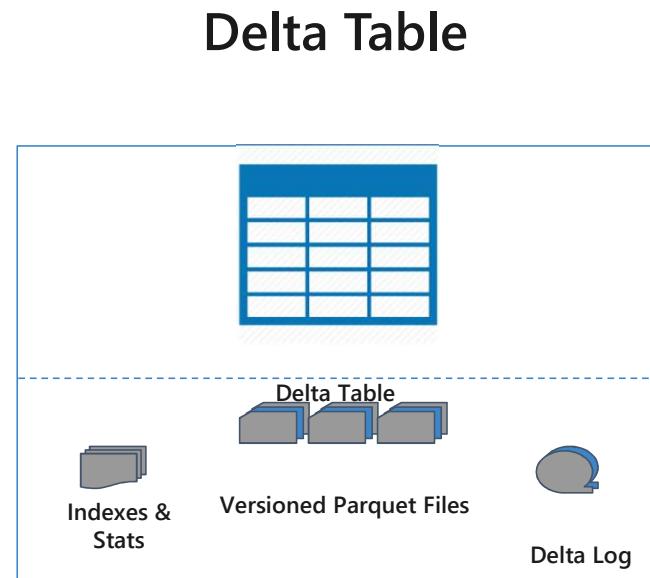
3. Simplified Architecture

Unify batch & streaming
Early data availability for analytics

Azure Databricks Delta Architecture

Delta Table = Parquet +
Transaction Log

- Linear history of atomic changes
- Optimistic Concurrency Control
- Log checkpoint is stored as Parquet
- Lazy GC = Free Snapshot Isolation



Azure Databricks Delta



Handle terabytes & petabytes of data



Low latency streaming ingestion



Avoid corrupt & messy data while reading & writing



Control on how to adapt to changing schema



Enable scientists & analysts to read data quickly for interactive analysis - Indexing

Azure Databricks Delta – Fast Reads



Data format



Compaction



Partitioning



Indexing



Caching

An ETL Query In Spark

```
val csvTable = spark.read.csv("/source/path")
```

Read Table 1

```
val jdbcTable = spark.read.format("jdbc")  
  .option("url", "jdbc:postgresql:...")  
  .option("dbtable", "TEST.PEOPLE")  
  .load()
```

Read Table 2

```
csvTable  
  .join(jdbcTable, Seq("name"), "outer")  
  .filter("id <= 2999")  
  .write  
  .mode("overwrite")  
  .format("delta")  
  .saveAsTable("outputTableName")
```

Join

Filter

Write

Performance Tips

- Land data in Blob Store/ADLS partitioned into separate directory
 - Avoid high list cost on large directories
- Parallelization of Azure Databricks streaming is driven by number of partitions in Eventhub
- For best query performance use Delta table. Alternatively, use regular Spark table backed by Parquet
 - ORC OK if customer has a preference
- Avoid small files. File size 100s MB – 1GB preferred
 - Delta supports compaction

ETL – Spark Databricks

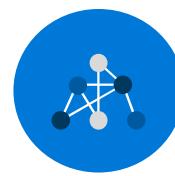
Azure AI

AI apps & agents



Azure Bot Service
Azure Cognitive Services

Machine learning



Azure Databricks
Azure Machine Learning

Knowledge mining



Azure Cognitive Search

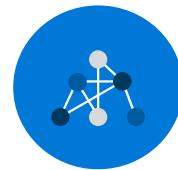
Azure AI

AI apps & agents



Azure Bot Service
Azure Cognitive Services

Machine learning



Azure Databricks
Azure Machine Learning

Knowledge mining



Azure Cognitive Search

Machine Learning on Azure

Sophisticated pretrained models
To simplify solution development



Vision



Speech



Language



Search

Popular frameworks
To build advanced deep learning solutions



Pytorch



TensorFlow



Keras



ONNX

Productive services
To empower data science and development teams



Azure
Databricks

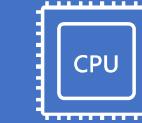


Azure
Machine Learning

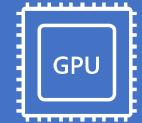


Machine Learning
VMs

Powerful infrastructure
To accelerate deep learning



CPU



GPU



FPGA

Flexible deployment
To deploy and manage models on intelligent cloud and edge



On-premises



Cloud

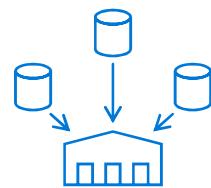


Edge

How companies are transforming



Serving business users and end users with **intelligent** and **dynamic** applications



Build a unified and usable data pipeline

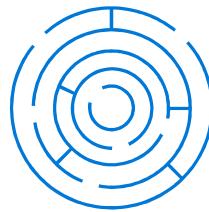


Train ML and DL models to derive insights



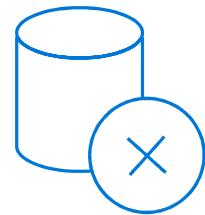
Operationalize models and distribute insights at scale

But there's a lot to consider



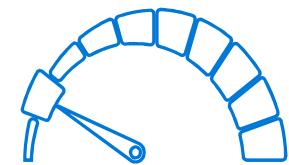
Complexity of solutions

Many options in the marketplace



Data silos

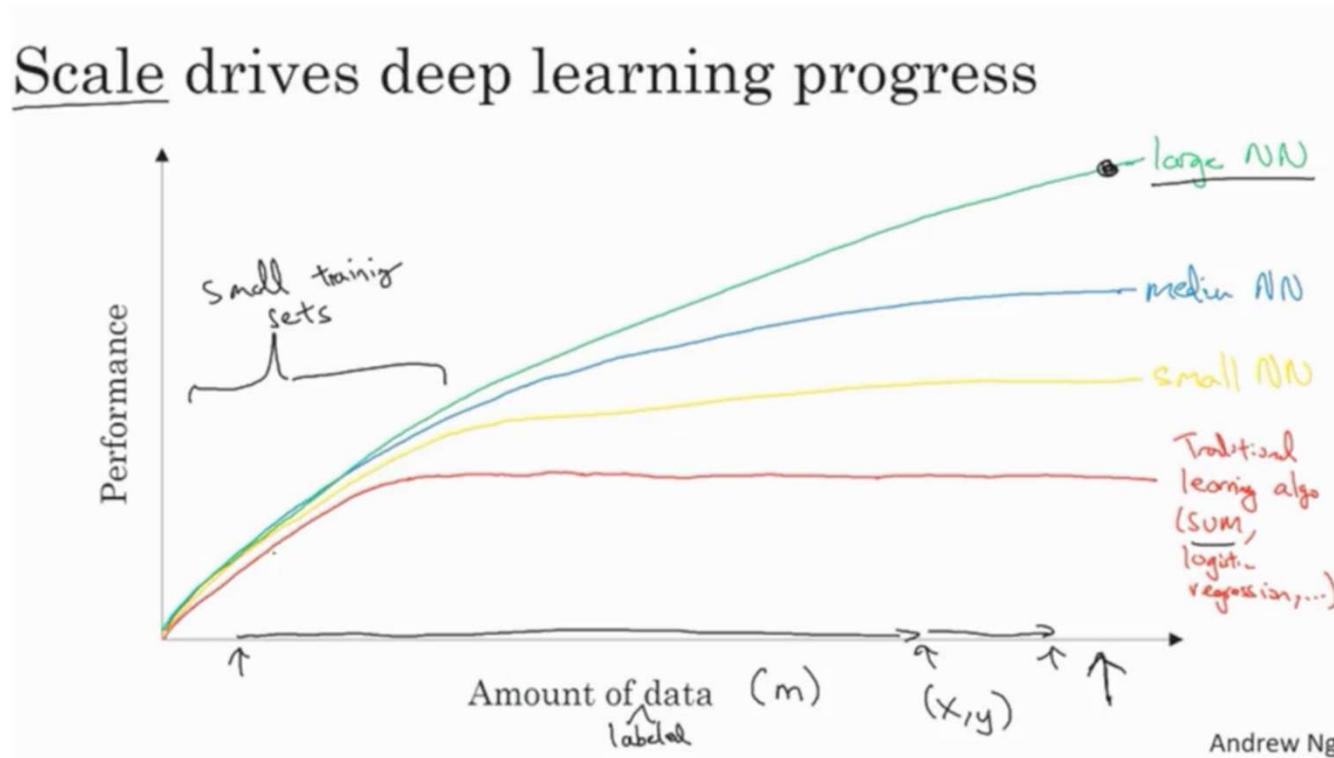
Incongruent data types



Difficult to scale effectively

Performance constraints

DL model accuracy goes up w/ data size!



From Andrew Ng's Coursera course

Challenges for Data Scientists

- Infrastructure management
- Data exploration and visualization at scale
- Time to value - From model iterations to intelligence
- Integrating with various ML tools to stitch a solution together
- Operationalize ML models to integrate them into applications

Typical Data Science Projects



**DATA
EXPLORATION**



**ANALYSIS AND
MODELING**



**COMMUNICATE
RESULTS**



**OPERATIONALIZE
MODELS**

What do Data Scientists care about?



**RAPID
EXPERIMENTATI
ON**



**DATA
VISUALIZATION**

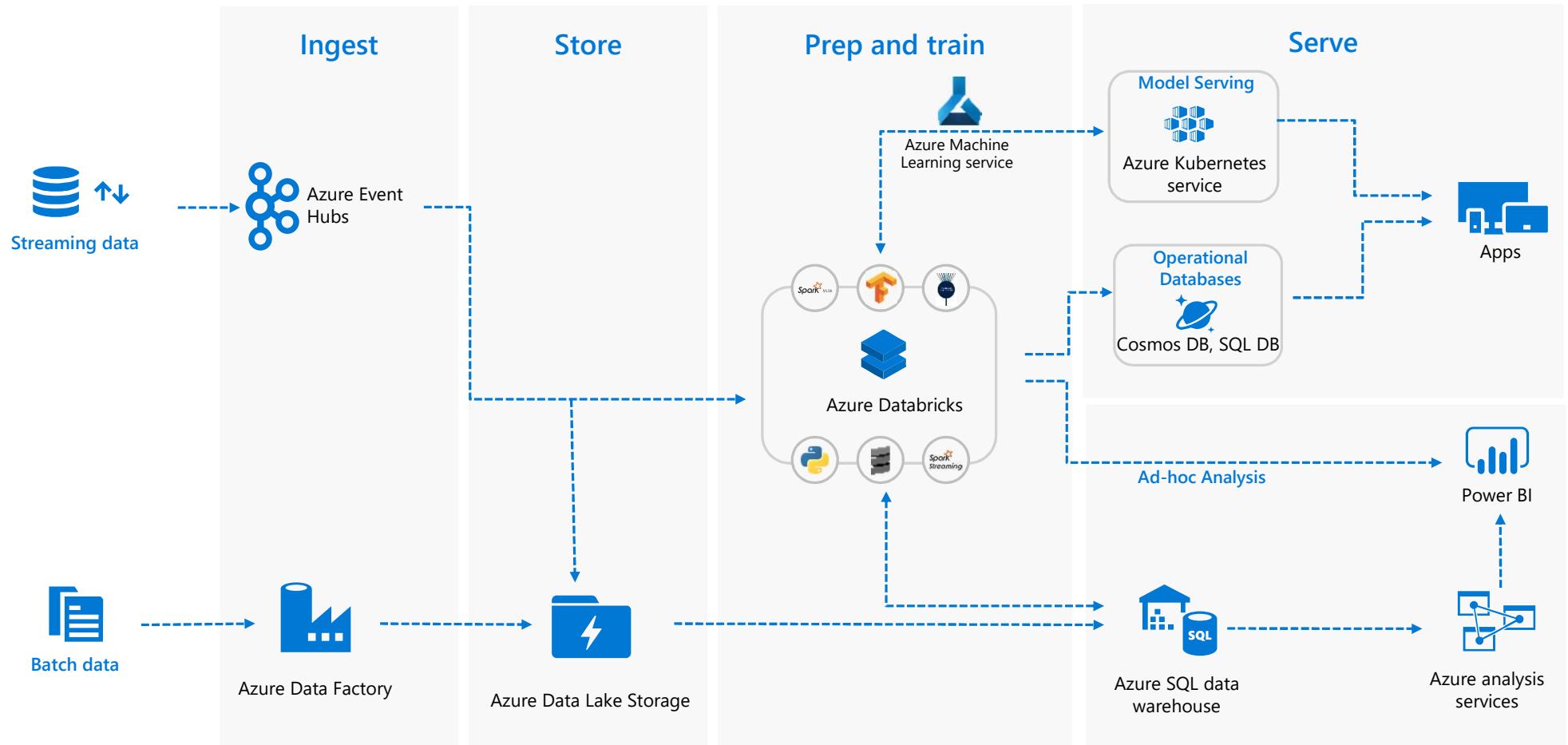


**CROSS-TEAM
COLLABORATION**

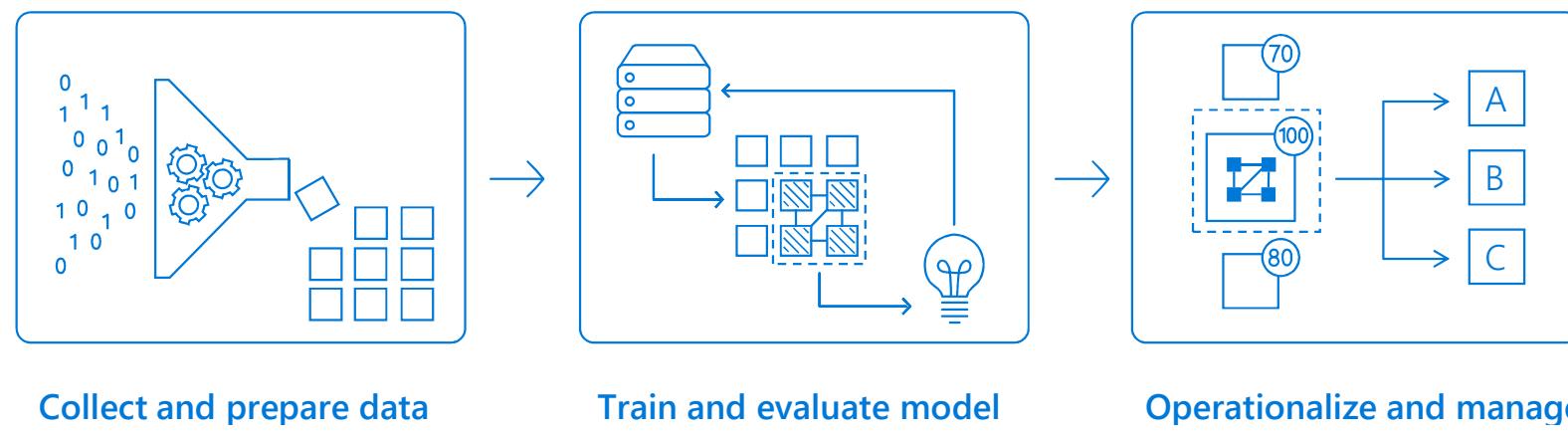


**EASY SHARING
OF INSIGHTS**

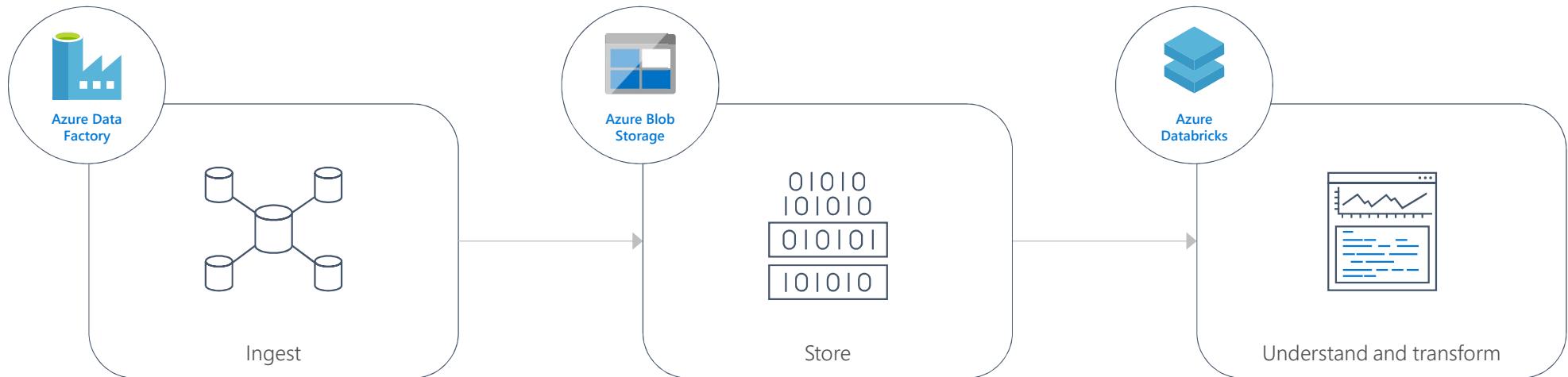
Recommended architecture to build e2e ML solutions



PREP & TRAIN



Collect and prepare all of your data at scale



Connect to data from any source

- Integrate with all of your data sources
- Create hybrid pipelines
- Orchestrate in a code-free environment



Leverage best-in-class analytics capabilities

- Leverage open source technologies
- Collaborate within teams
- Use ML on batch streams

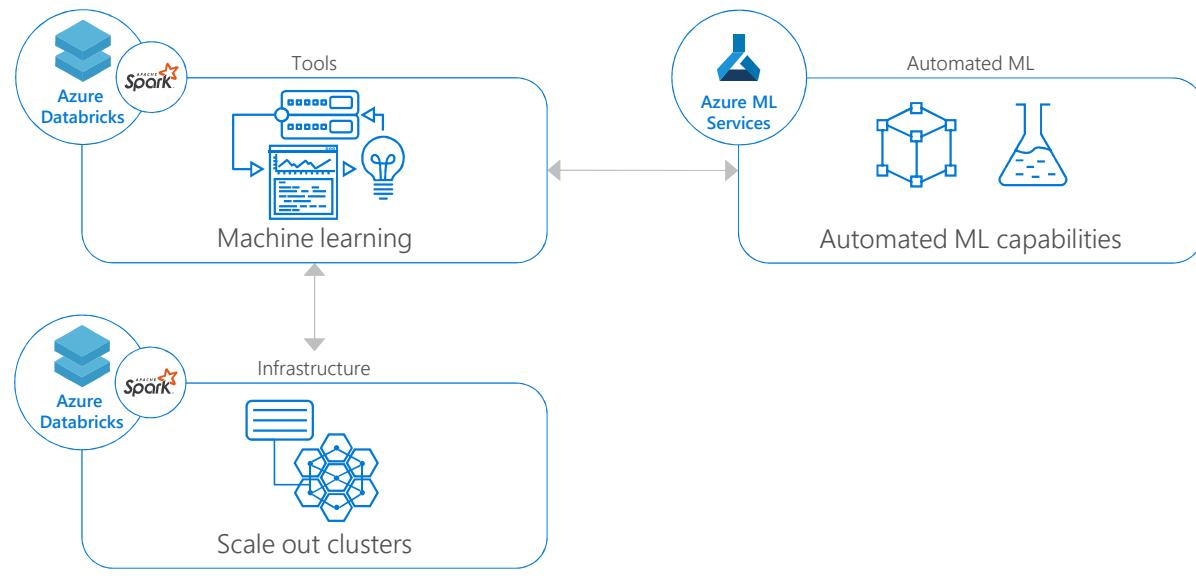


Scale without limits

- Build in the language of your choice
- Leverage scale out topology
- Scale compute and storage separately

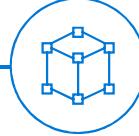


Train and evaluate machine learning models



Simplify model development

- Collaborate in interactive workspaces
- Access a library of battle-tested models
- Automate job execution



Scale compute resources to meet your needs

- Easily scale up or scale out
- Autoscale on serverless infrastructure
- Leverage commodity hardware



Quickly determine the right model for your data

- Determine the best algorithm
- Tune hyperparameters to optimize models
- Rapidly prototype in agile environments



3 Ways for Machine Learning

#1 Scalable Machine Learning with Spark MLlib

- Goal is to make practical machine learning extremely *scalable* and *easy*
- Common Algorithms, Featurization, Pipelines, and Utilities need for ML
- Subset of all ML techniques, but *extremely scalable*

#2 Single Machine Data Science on Big Data with Azure Databricks

- Use ADB to query “Big Data” stored on ADLS or Blob
- Use Spark to Aggregate, Sample “Big Data” to make it “small data”
- Collect this “small data” back to the driver for normal smaller data ML tools, R, Scikit-learn, etc

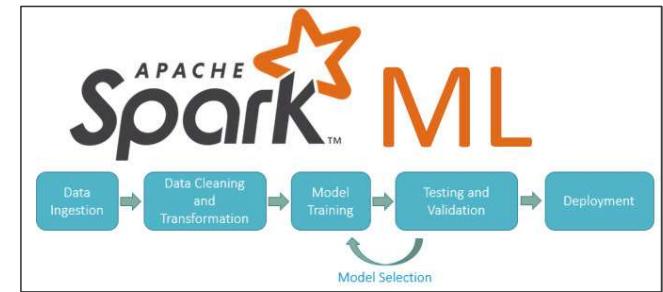
#3 Scale Out / Parallelization for Single Machine Data Science

- Combination of the above two
- Use Databricks for cross validation, training a bunch of small models, etc
- Apply user defined functions from R and Python

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](#)



Interactive Data Science

Sales Table Analysis (Scala)

Attached: SHARED AUTOSCALING

Jobs

Recent

Tables

Clusters

Jobs

Apps

Search

databricks

Home

Workspace

?

User icon

File icon

Lock icon

Help icon

Job status icon

Calendar icon

Comment icon

Refresh icon

SQL command: %sql select * from sales_long

(2) Spark Jobs Cancel

- Job 1306 View (1 stages)
- Job 1307 View (3 stages)
 - Stage 1965: 2/2 (0 running)
 - Stage 1966: 48/200 (4 running)
 - Stage 1967: 0/1 (0 running)

Bar chart showing sales by country and product type:

Country	MacBook	Power Adapter	iPhone	Total Sales
BRA	20,000,000	8,000,000	38,000,000	66,000,000
FRA	18,000,000	7,000,000	32,000,000	57,000,000
RUS	24,000,000	12,000,000	44,000,000	80,000,000
USA	34,000,000	17,000,000	62,000,000	113,000,000

product

- MacBook
- Power Adapter
- iPhone

Plot Options...

Chat interface:

- Chaoyu Yang 8/8/2016, 11:58:06 AM: Can you break down the revenue by type and country?
- David 8/8/2016, 11:58:35 AM: Sure. Here you go.

SPARK ML ALGORITHMS

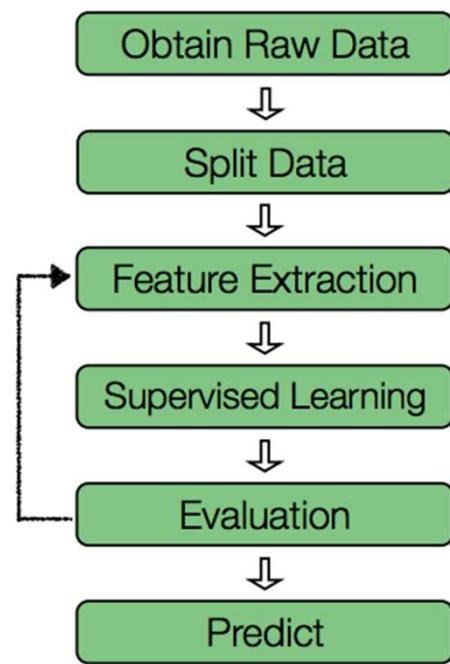
Spark ML Algorithms

Classification and Regression	<ul style="list-style-type: none">• Linear Models (SVMs, logistic regression, linear regression)• Naïve Bayes• Decision Trees• Ensembles of trees (Random Forest, Gradient-Boosted Trees)• Isotonic regression
Clustering	<ul style="list-style-type: none">• k-means and streaming k-means• Gaussian mixture• Power iteration clustering (PIC)• Latent Dirichlet allocation (LDA)
Collaborative Filtering	<ul style="list-style-type: none">• Alternating least squares (ALS)
Dimensionality Reduction	<ul style="list-style-type: none">• SVD• PCA
Frequent Pattern Mining	<ul style="list-style-type: none">• FP-growth• Association rules
Basic Statistics	<ul style="list-style-type: none">• Summary statistics• Correlations• Stratified sampling• Hypothesis testing• Random data generation

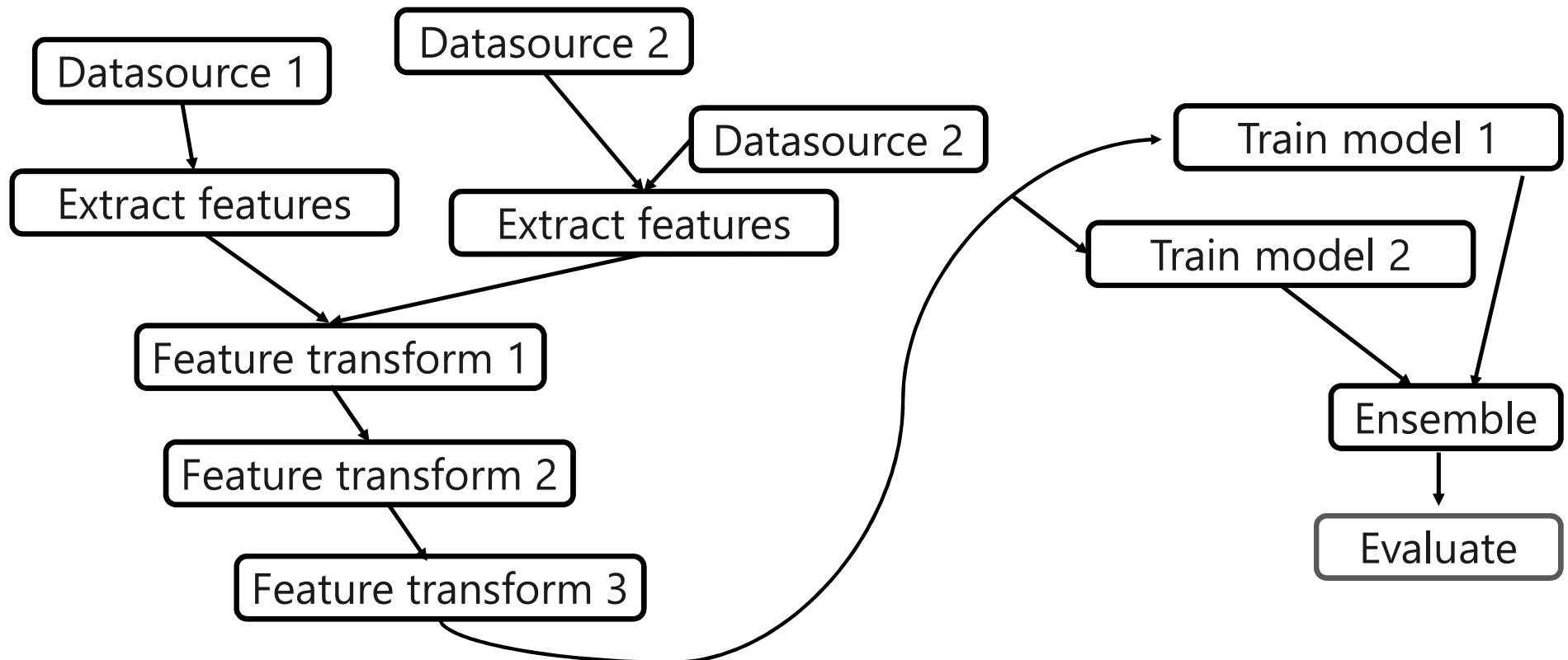
Why use Azure Databricks for Machine learning?

- Complete platform in one (Data ingestion, exploration, transformation, featurization, model building, model tuning, and even model serving).
- No need to copy the data in our system to do ML on it.
- DataScientists like the ease of use of our platform.
- Deep learning algorithms are now available!
- Productionization Features built in.

ML Pipelines



ML Pipelines



Estimator

- More complex feature processing, and/or prediction
- Transformer that is initialized with data (eg – convert a column into %tile rep and would require to initialize it)
- Actual models that be trained and tuned so we can use for prediction
 - Logistic Regression (classification)
 - Decision Tree
 - Random Forest

Evaluator

- Evaluate the models
- Evaluate how a given estimator performs according to criteria like ROC curve, RMSE
- Select best model based on the test

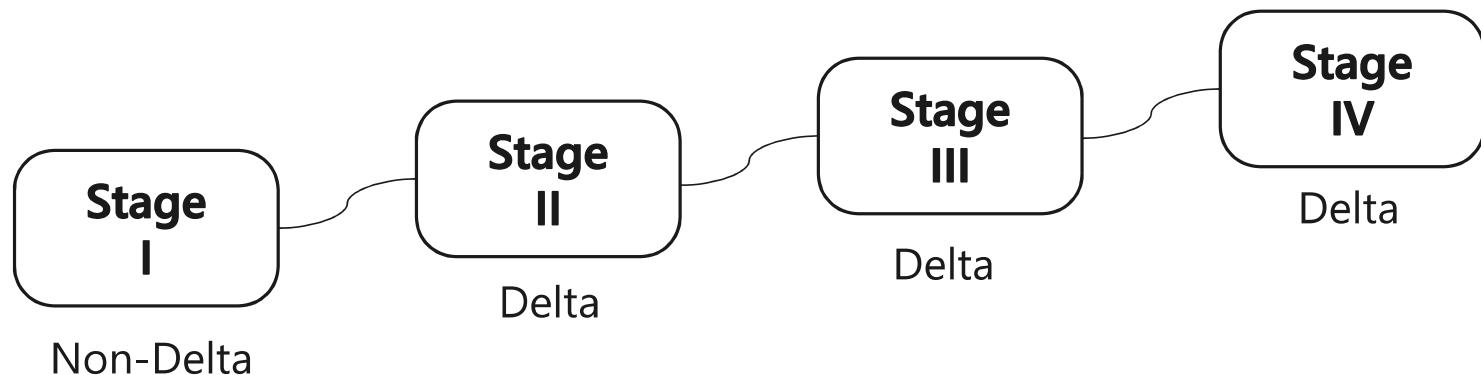
Pipeline

- Represents composition of
 - various Transformers' .transform methods
 - various Estimators' .fit and result's .transform methods
- Transformations and Estimators are specified together
- Pipeline allows to set up a dataflow of the relevant transformations, ending with an estimator that is automatically tuned according to your specifications resulting in a tuned model ready for a production use case

Multi Hop Data Pipelines

LOTS OF NEW DATA

User Behavior Data
Click Streams
Sensor Data (IoT)
Video/Speech
Usage/Billing Data
Machine Telemetry
Commerce Data
...



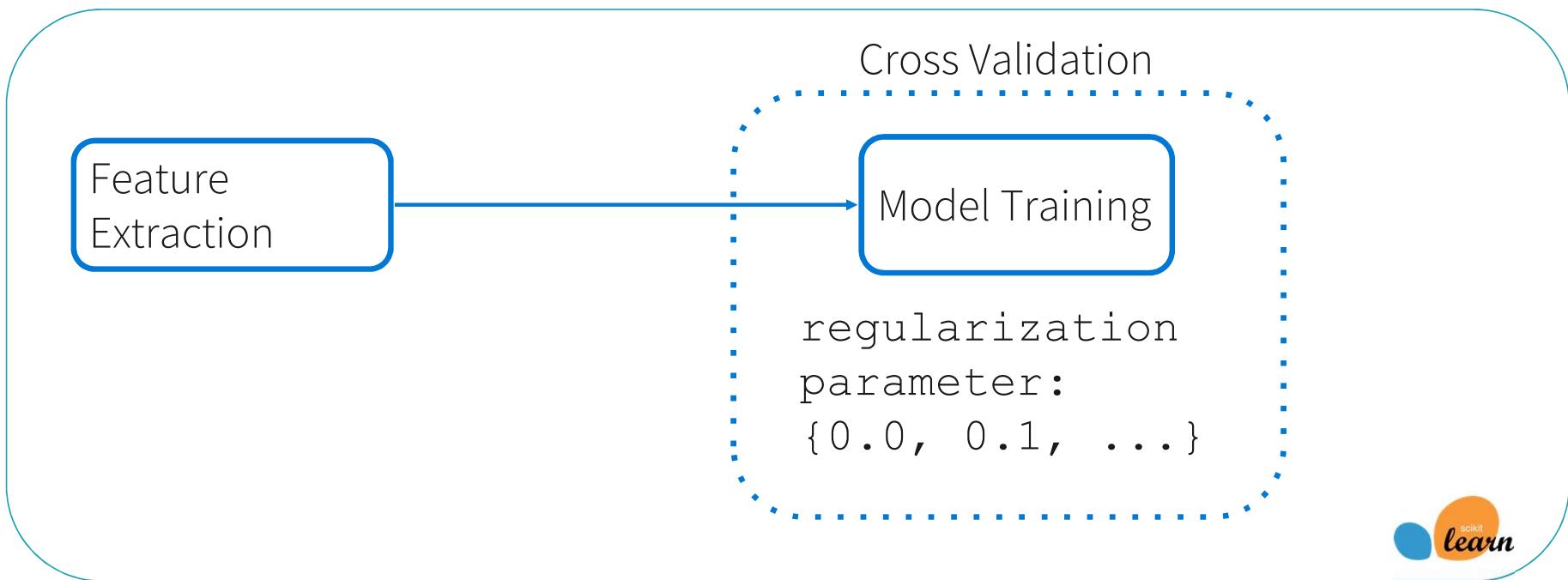
Stage I - Raw events from many different parts of the organization

Stage II - Normalized and enriched with dimension information

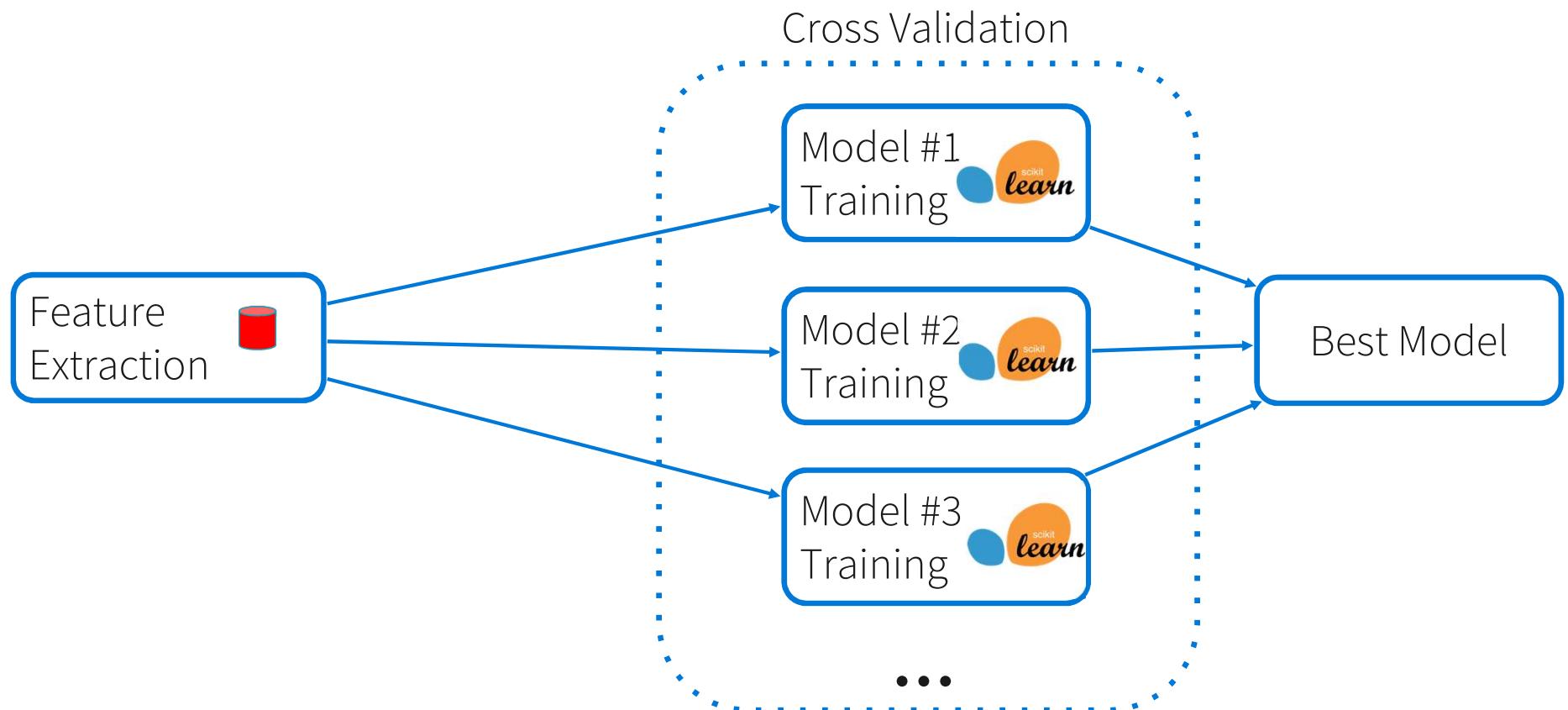
Stage III - Filtered down and aggregated for particular business objective.

Stage IV - High-level summaries of key business metrics.

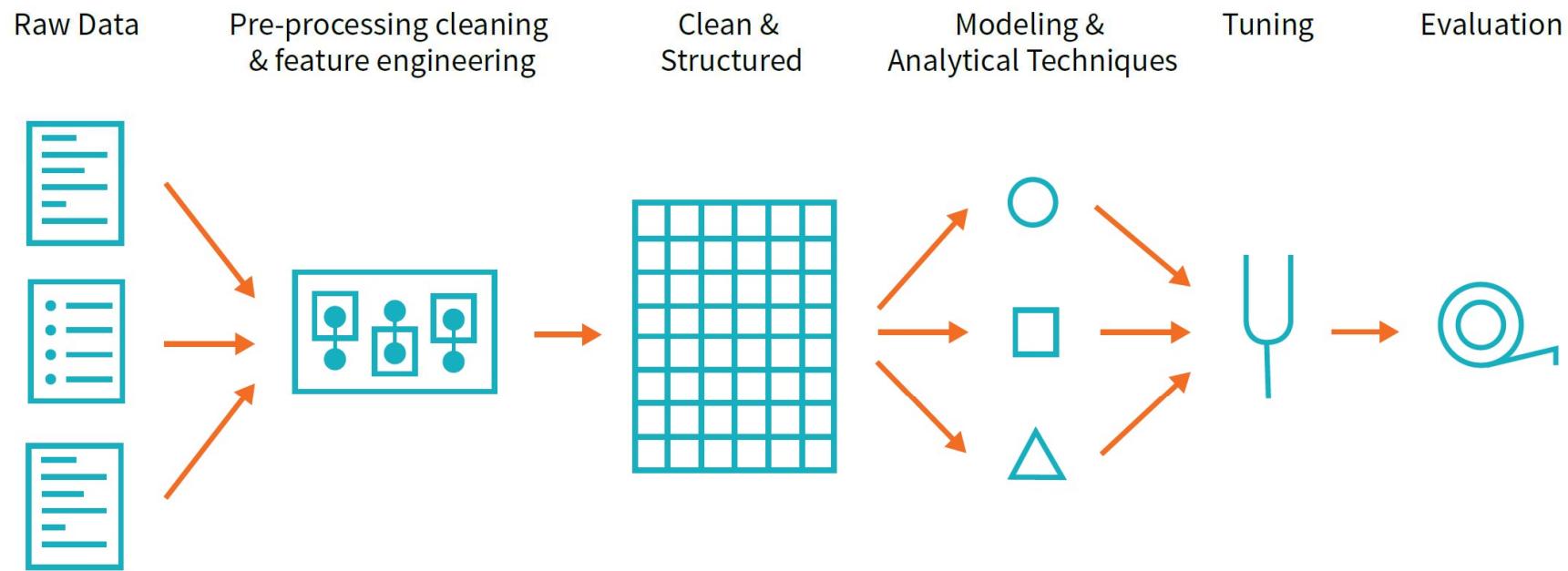
Cross Validation and Tuning



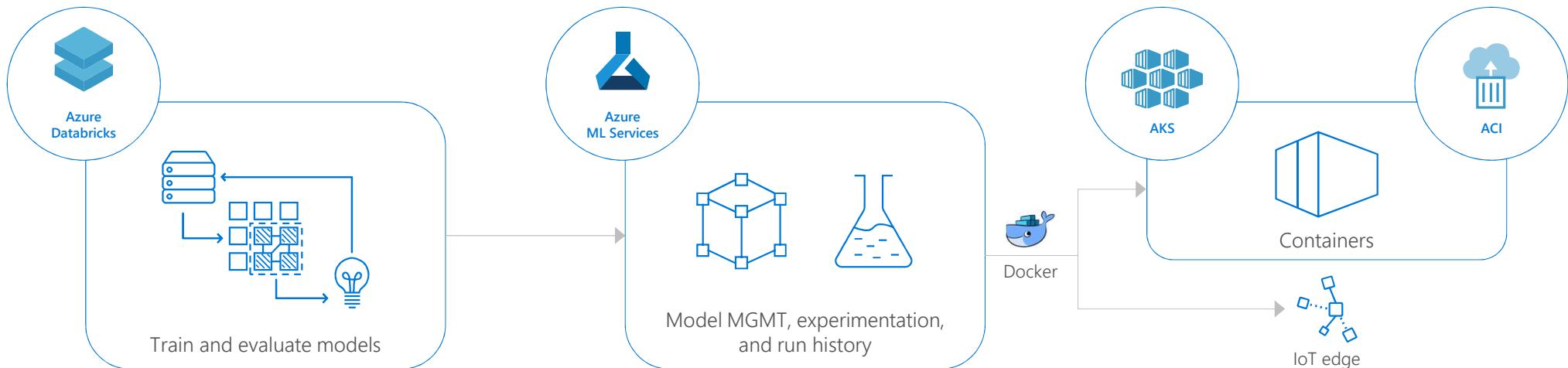
Model Selection



Advanced Analytics: Pipeline

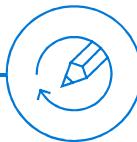


Operationalize and manage models with ease



Bring models to life quickly

- Build and deploy models in minutes
- Iterate quickly on serverless infrastructure
- Easily change environments



Proactively manage model performance

- Identify and promote your best models
- Capture model telemetry
- Retrain models with APIs

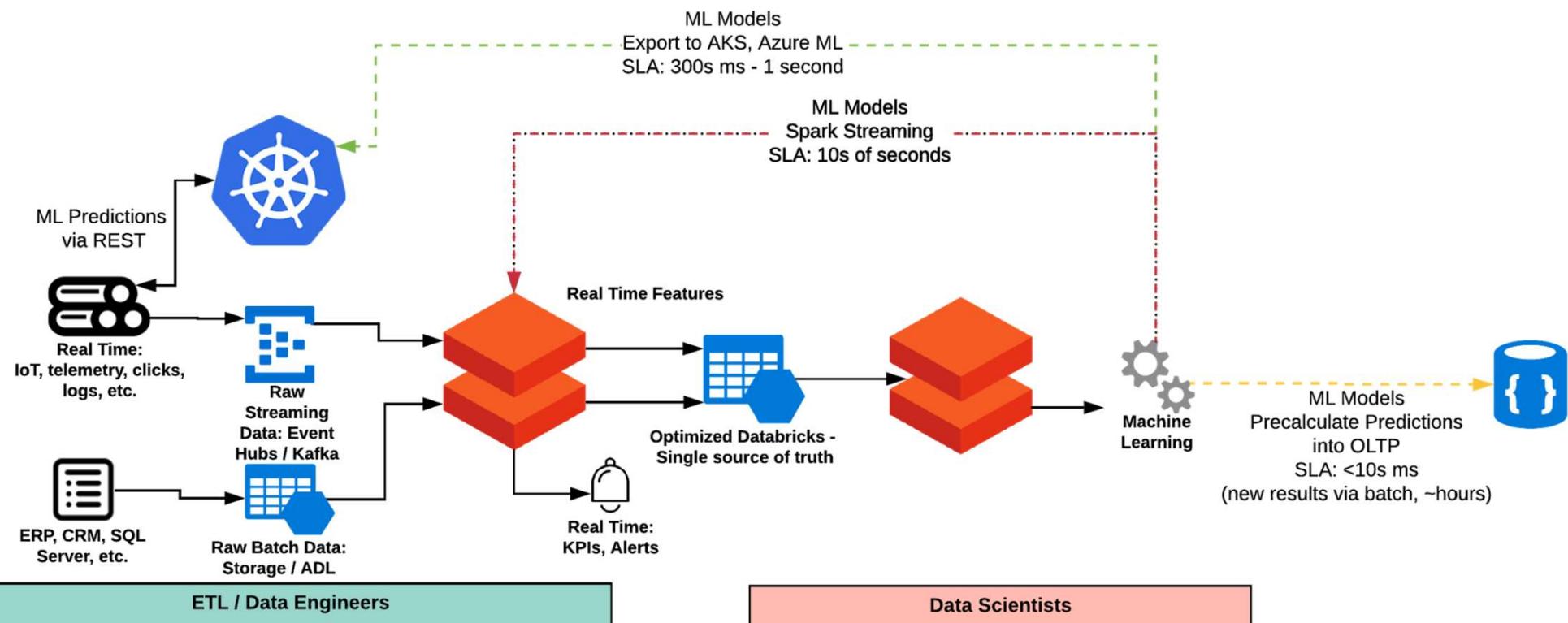


Deploy models closer to your data

- Deploy models anywhere
- Scale out to containers
- Infuse intelligence into the IoT edge



Operationalize Models





Azure Machine Learning service

Bring AI to everyone with an end-to-end, scalable, trusted platform



Boost your data science productivity



Increase your rate of experimentation



Deploy and manage your models everywhere



Built with your needs in mind

- GPU-enabled virtual machines
- Low-latency predictions at scale
- Integration with popular Python IDEs
- Role-based access controls
- Model versioning
- Automated model retraining

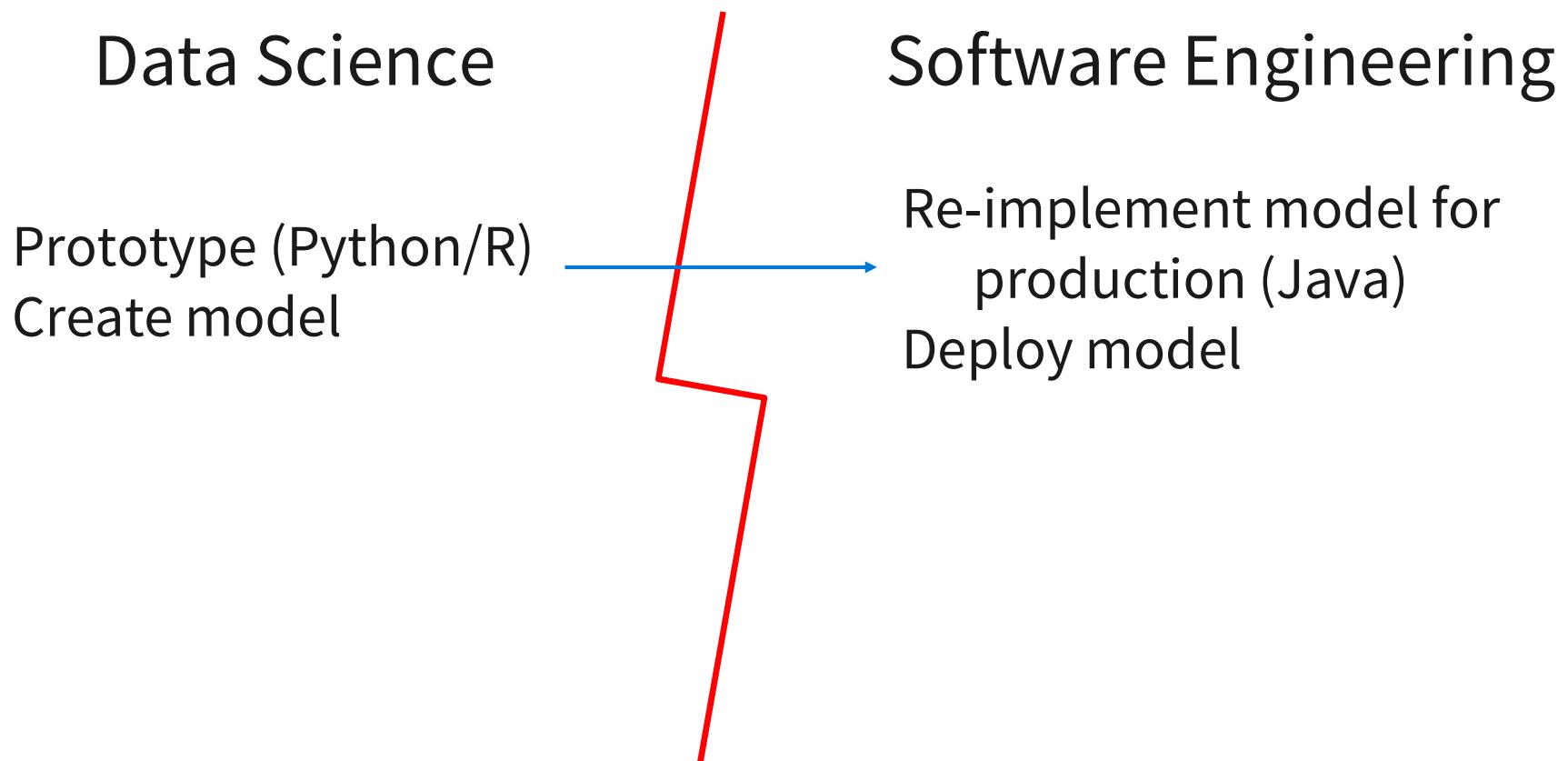


Seamlessly integrated with the Azure Portfolio

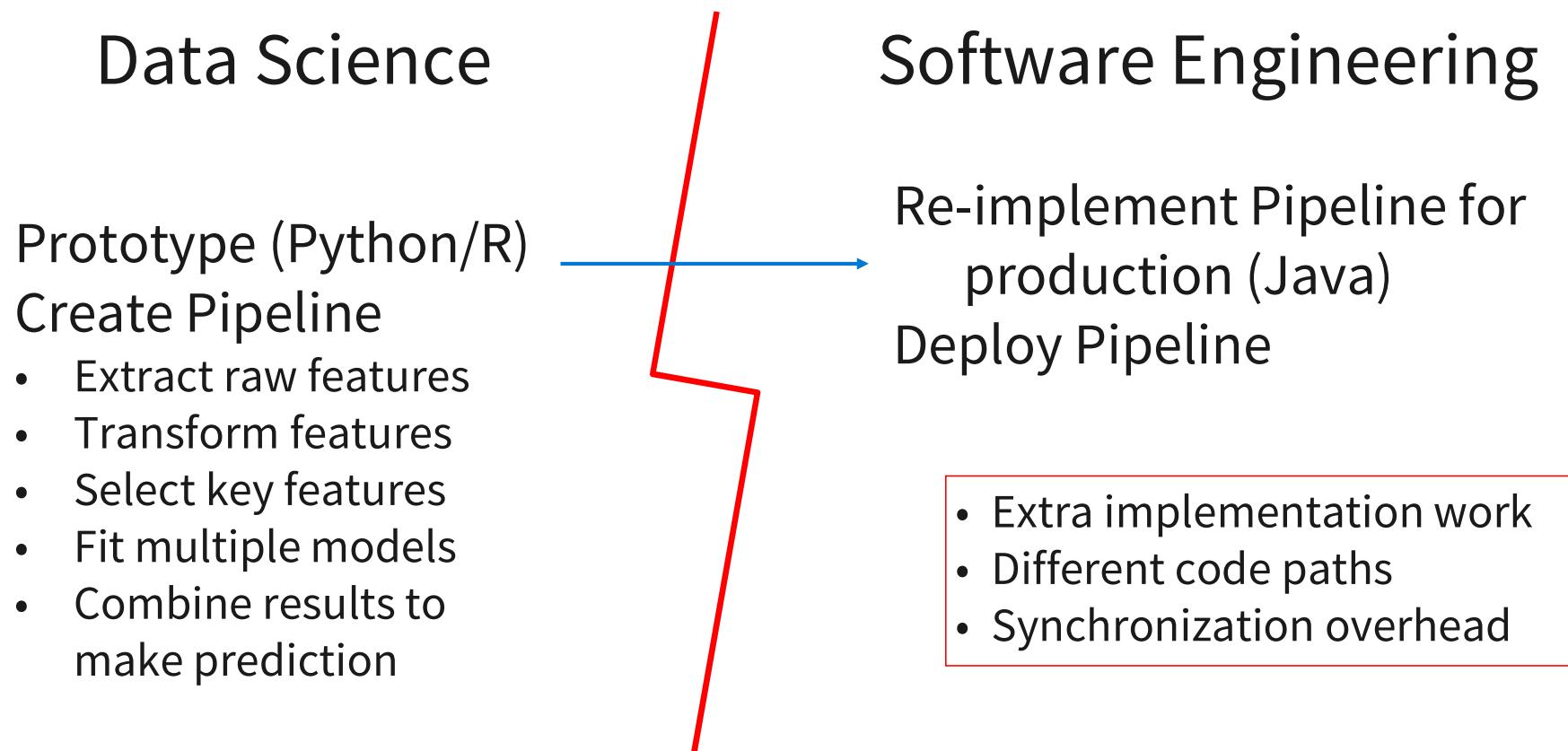
ML Export

- ML Model Export allows you to export models and full ML pipelines
 - Then imported into Spark and non-Spark platforms to do scoring, make predictions
 - Targeted at low-latency, lightweight ML-powered applications
- We recommend using MLeap, an open source solution for ML Model Export that works well in Azure Databricks

Why ML persistence?



Why ML persistence?



With ML persistence...

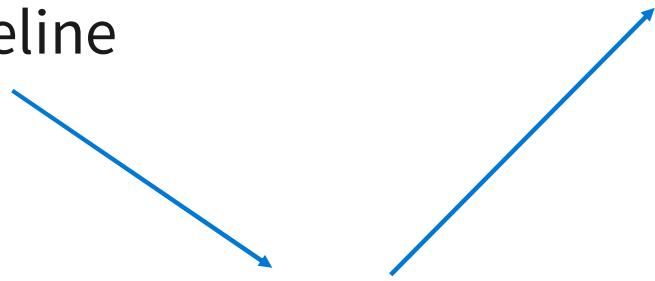
Data Science

Prototype (Python/R)
Create Pipeline

Persist model or Pipeline:
`model.writeBundle.save()`

Software Engineering

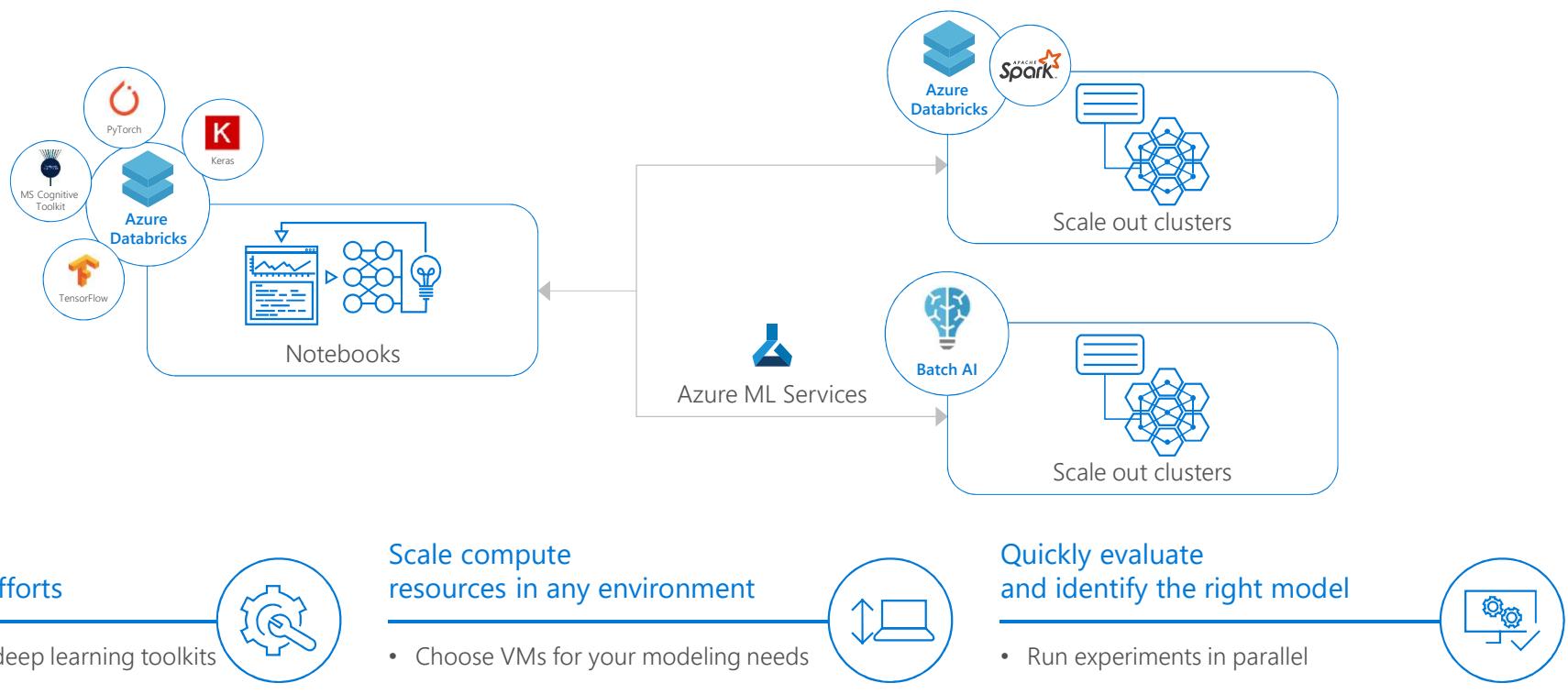
Load Pipeline (Scala/Java)
`bundle.loadSparkBundle()`
Deploy in production



Workflow

- 1) Fit ML model in Databricks using Apache Spark MLlib.
- 2) Export model in Databricks
 - MLeap calls these "bundles"
- 3) Deploy model in external system
 - Load model from "bundle" at runtime
 - Make predictions in real time

Build and deploy deep learning models



Azure Databricks for deep learning modeling

Ready-to-use clusters with Azure Databricks Runtime for ML



Tools

Use HorovodEstimator via a native runtime to enable build deep learning models with a few lines of code

Load images natively in Spark DataFrames to automatically decode them for manipulation at scale with distributed DNN training on Spark

Simultaneously collaborate within notebooks environments to streamline model development



Frameworks

Full Python and Scala support for transfer learning on images

Seamlessly use TensorFlow, Microsoft Cognitive Toolkit, Caffe2, Keras, and more

Use built-in hyperparameter tuning via Spark MLLib to quickly optimize the model



Infrastructure

Leverage powerful GPU-enabled VMs pre-configured for deep neural network training

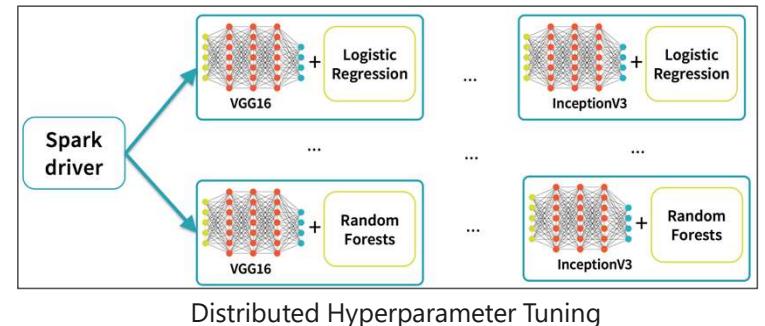
Automatically store metadata in Azure Database with geo-replication for fault tolerance

Improve performance 10x-100x over traditional Spark deployments with an optimized environment

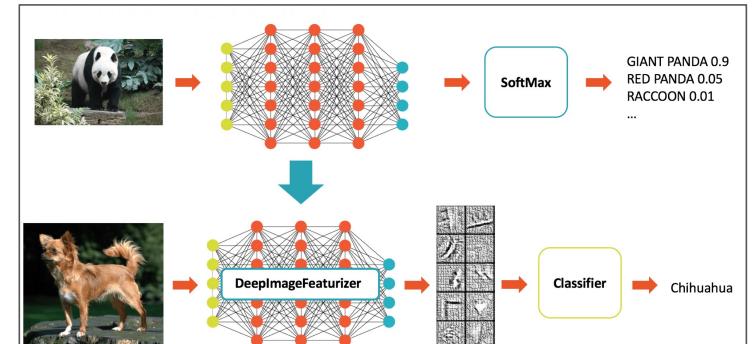
DEEP LEARNING

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



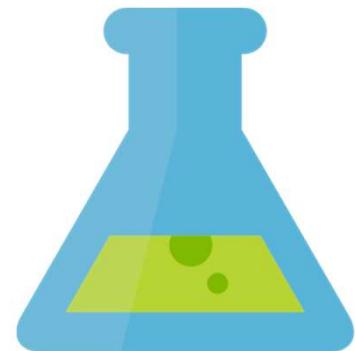
Transfer Learning

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



Leverage deep learning services and frameworks



AZURE DATABRICKS



Accelerate processing with the fastest Spark engine



Integrate natively with Azure services



Access enterprise-grade Azure security



AZURE ML service



Bring AI to the edge



Increase your rate of experimentation



Deploy and manage your models everywhere

Leverage your favorite deep learning frameworks



TensorFlow



MS Cognitive Toolkit



PyTorch



Scikit-Learn



ONNX



Caffe2



MXNet



Chainer

Powerful infrastructure

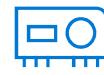
Accelerate deep learning



CPUs

General purpose machine
learning

D, F, L, M, H Series



GPUs

Deep learning

N Series



FPGAs

Specialized hardware
accelerated deep learning

Project Brainwave

Optimized for flexibility

Optimized for performance

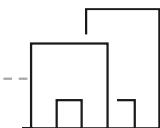
Flexible deployment

Deploy and manage models on intelligent cloud and edge

Train & deploy



Train & deploy



Track models in production
Capture model telemetry
Retrain models automatically



Deploy



paypal-churn-sparkml.pdf

Use Case – Churn Analytics

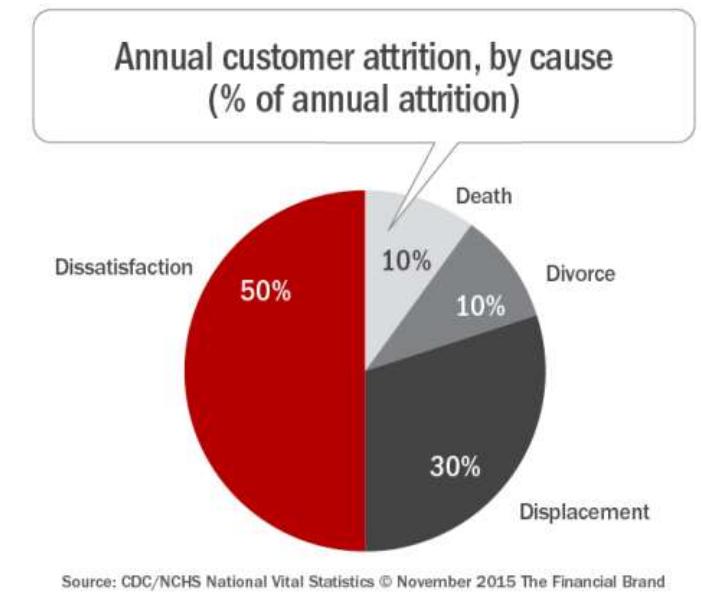


Customer Churn: A Key Performance Indicator for Banks

- Customer churn and engagement is one of the top issues for most banks
- Costs significantly more to acquire new customers than retain existing ones
- and it costs far more to re-acquire deflected customers.

The key issue: knowing the customer and predicting churn

Building a churn prediction model

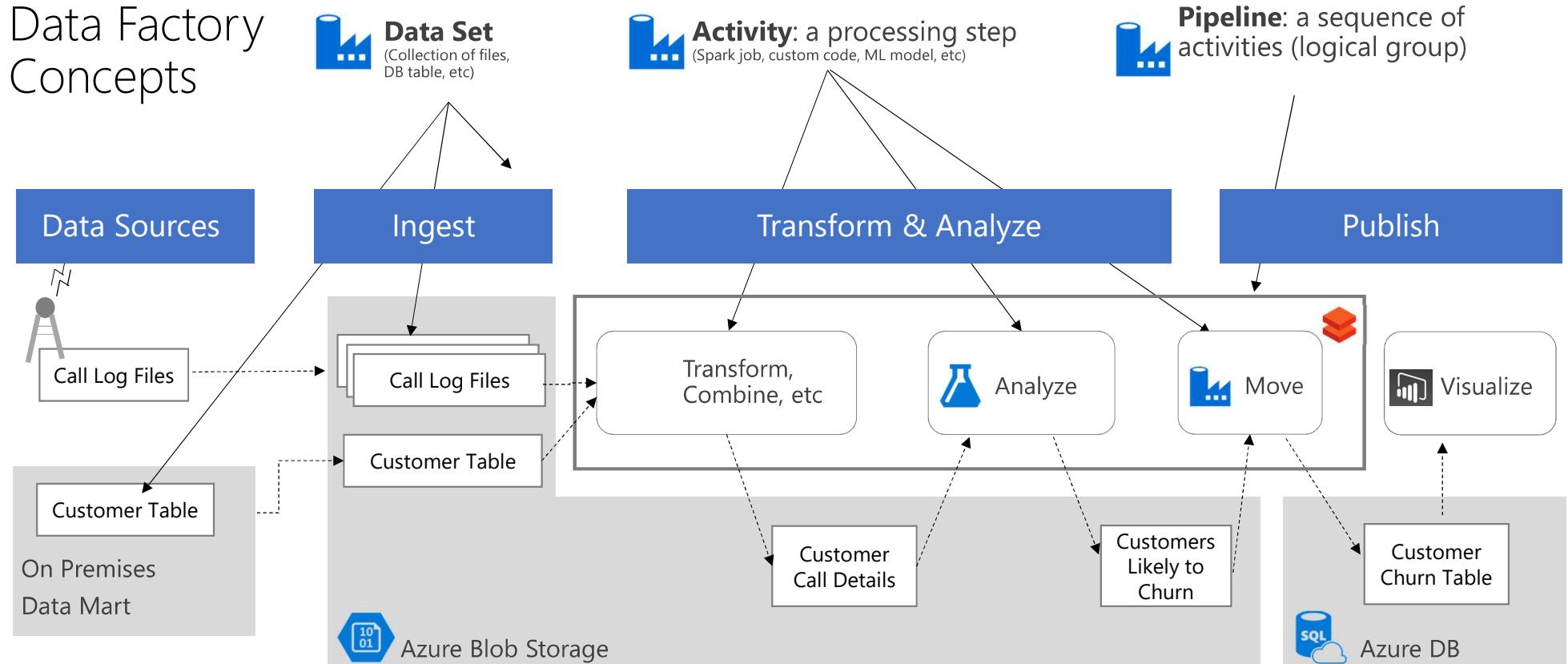


What to look for – Opportunities?

- **Death** – One way to reduce the impact of death on your portfolio is to influence the age distribution of your customer base
- **Divorce** – One way to stem attrition from divorce is to provide education on account ownership options and to ensure staff members are properly trained
- **Displacement** – Strong mobile and online banking capabilities will help you keep customers who are comfortable using mobile or online banking even if they move out of your market area
- **Dissatisfaction** – Sentiment and QoS Analysis

Conceptual Architecture

Data Factory Concepts



Resources



Democratization-of-AI-and-Deep-Learning.pdf



Data-Scientists-Guide-to-Apache-Spark.pdf



The-Data-Engineers-Guide-to-Apache-Spark.pdf



A-Gentle-Introduction-to-Apache-Spark.pdf



FY18_AA_Databricks_e-book_FINAL_032018.pdf



Three practical use cases with Azure Databrick - Aug 2018.pdf

