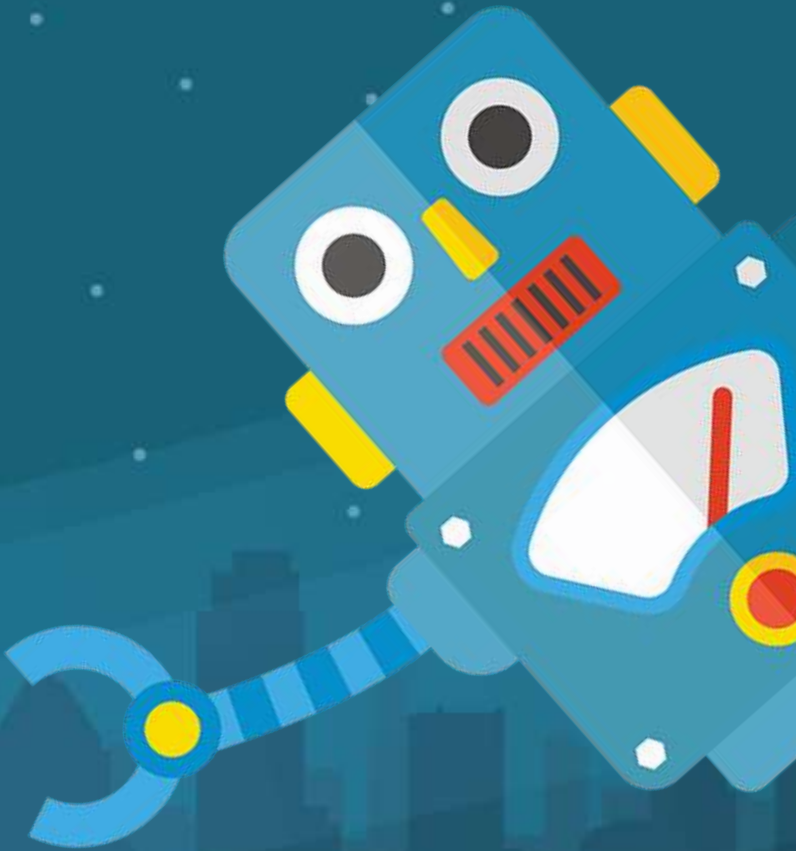
TECHORAMA

DEEP KNOWLEDGE IT CONFERENCE

October 1-3 | 2018

Ede, The Netherlands



betabit

Rick van den Bosch

Cloud Solutions Architect

@rickvdbosch

rickvandenbosch.net

r.van.den.bosch@betabit.nl

AN INTRO TO AZURE DATA LAKE

TECHORAMA

DEEP KNOWLEDGE IT CONFERENCE

October 1-3 | 2018

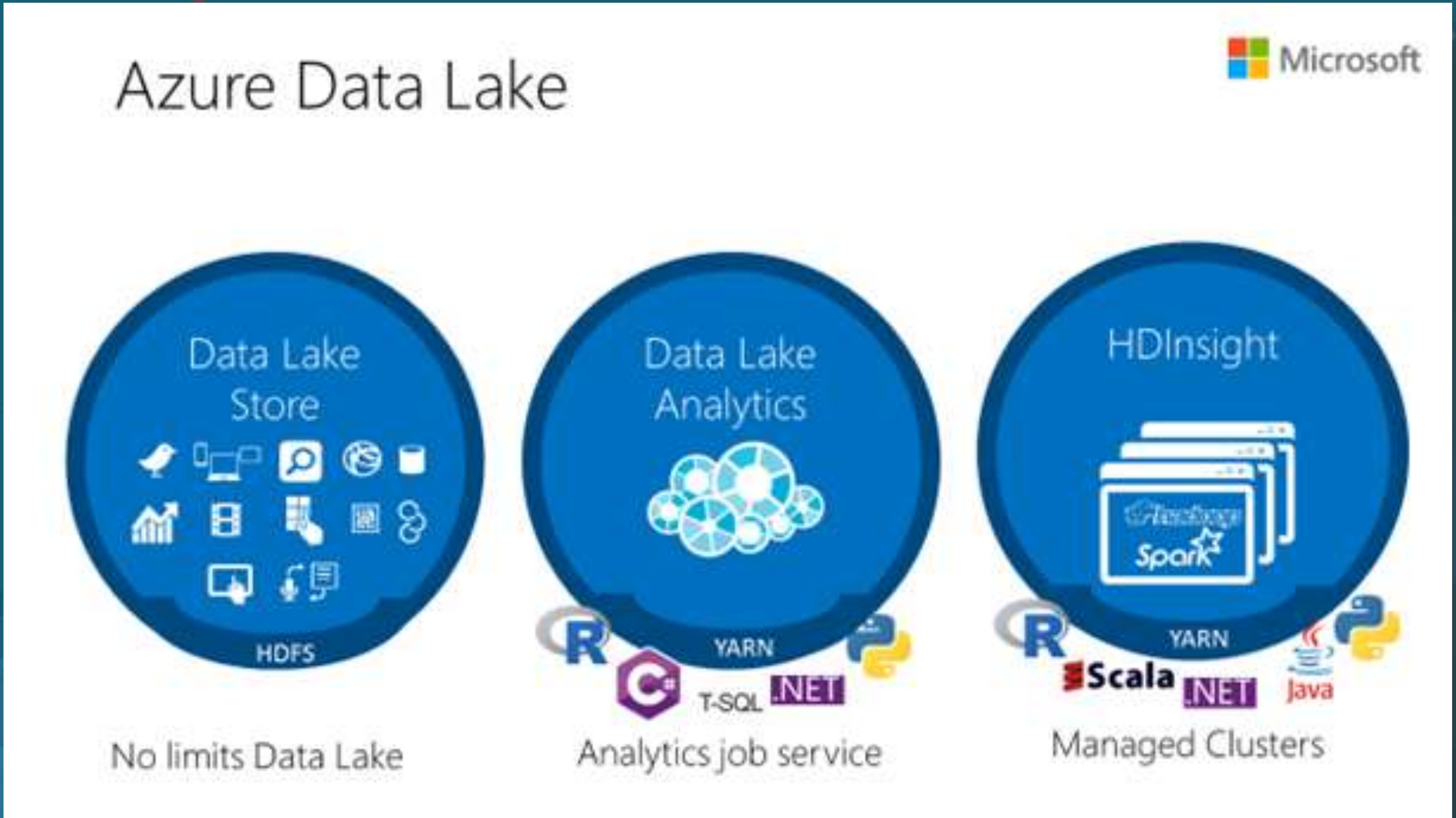
Ede, The Netherlands

Calendar

- About Azure Data Lake
- Azure Data Lake Store
  - Demo
- Azure Data Lake HDInsight
- Azure Data Lake Analytics
  - Demo
- Power BI
- Resources

AZURE DATA LAKE

Azure Data Lake



Azure Data Lake

Microsoft

Data Lake Store

HDFS

No limits Data Lake

Data Lake Analytics

YARN

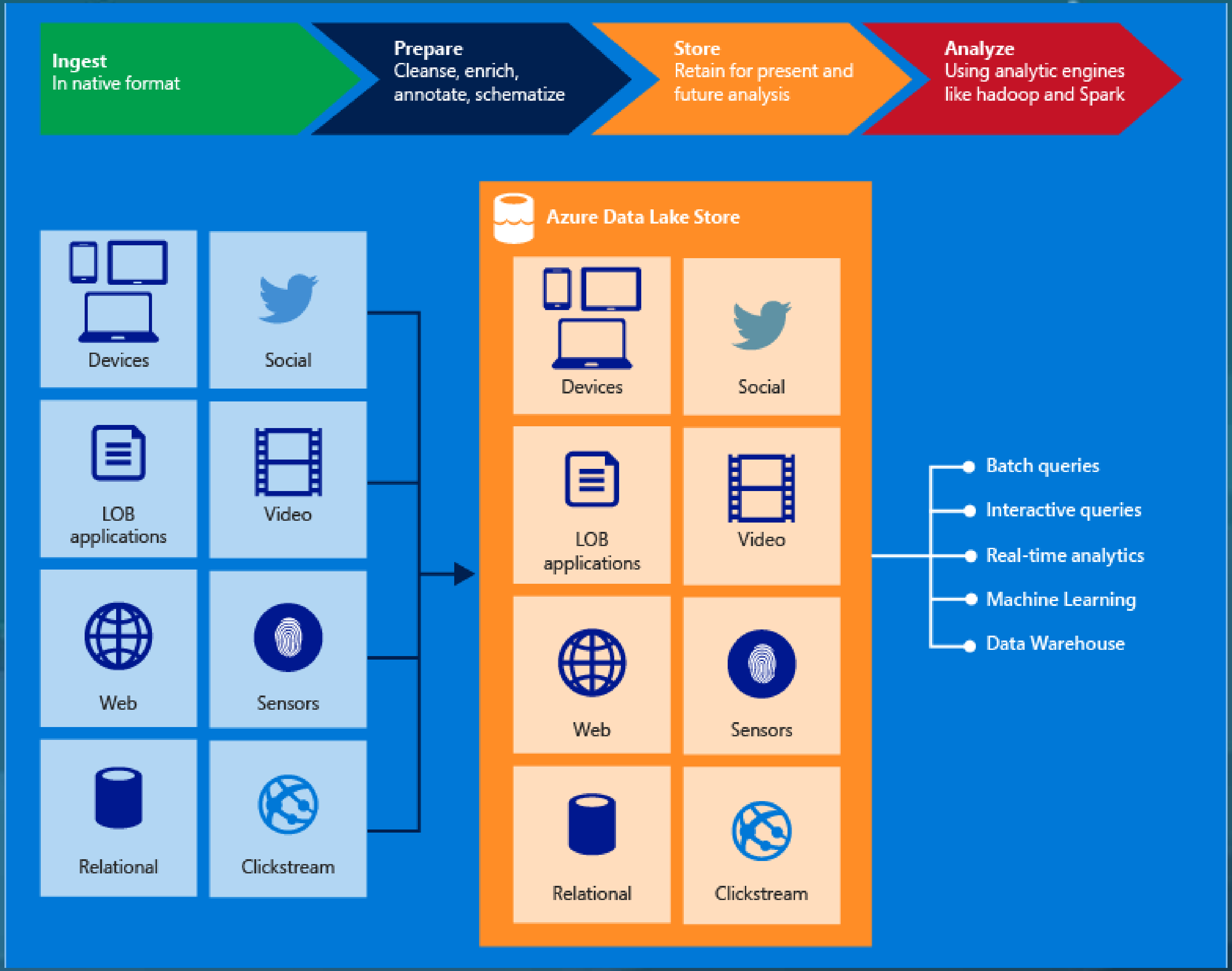
Analytics job service

HDInsight

YARN

Managed Clusters

Example



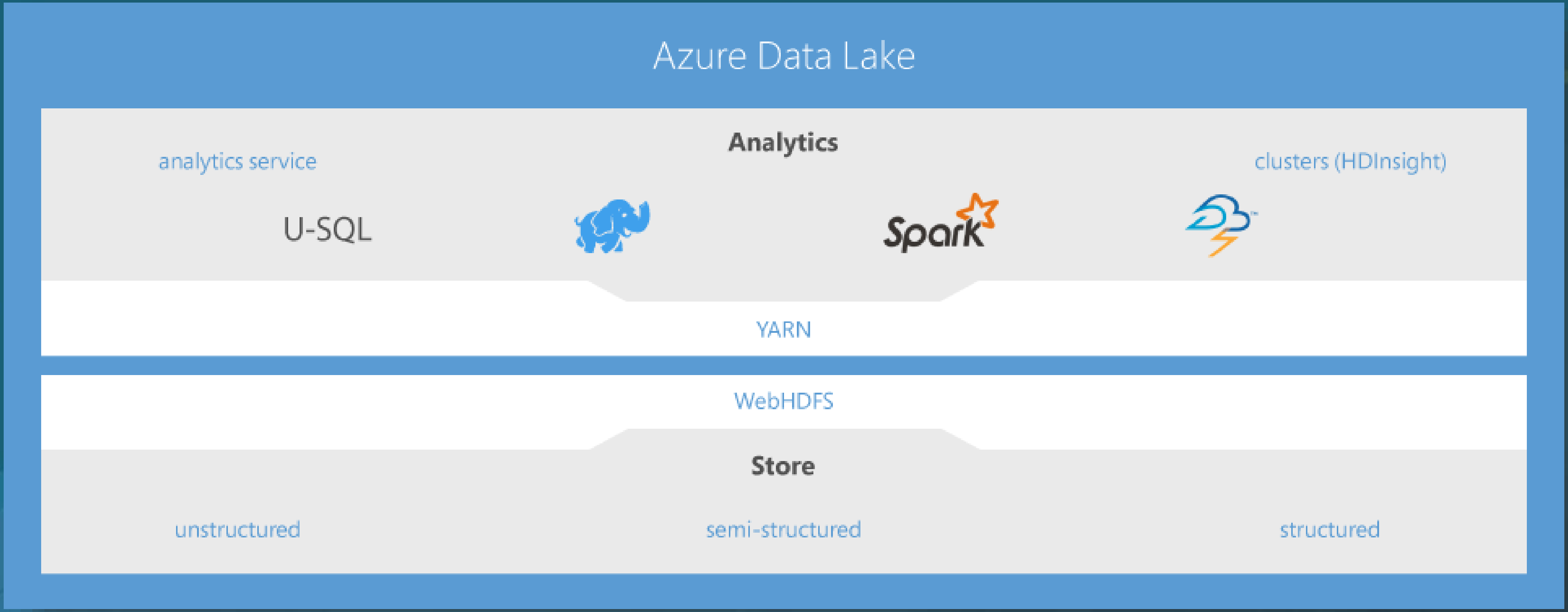
Ingest
In native format

Prepare
Cleanse, enrich, annotate, schematize

Store
Retain for present and future analysis

Analyze
Using analytic engines like hadoop and Spark

Azure Data Lake



Azure Data Lake

Analytics

analytics service

clusters (HDInsight)

U-SQL

Spark

YARN

WebHDFS

Store

unstructured

semi-structured

structured

AZURE DATA LAKE *STORE*

Store

- Enterprise-wide hyper-scale repository
- Data of any size, type and ingestion speed
- Operational and exploratory analytics

- WebHDFS-compatible API
- Specifically designed to enable analytics
- Tuned for (data analytics scenario) performance

- Out of the box:
  security, manageability, scalability, reliability, and availability

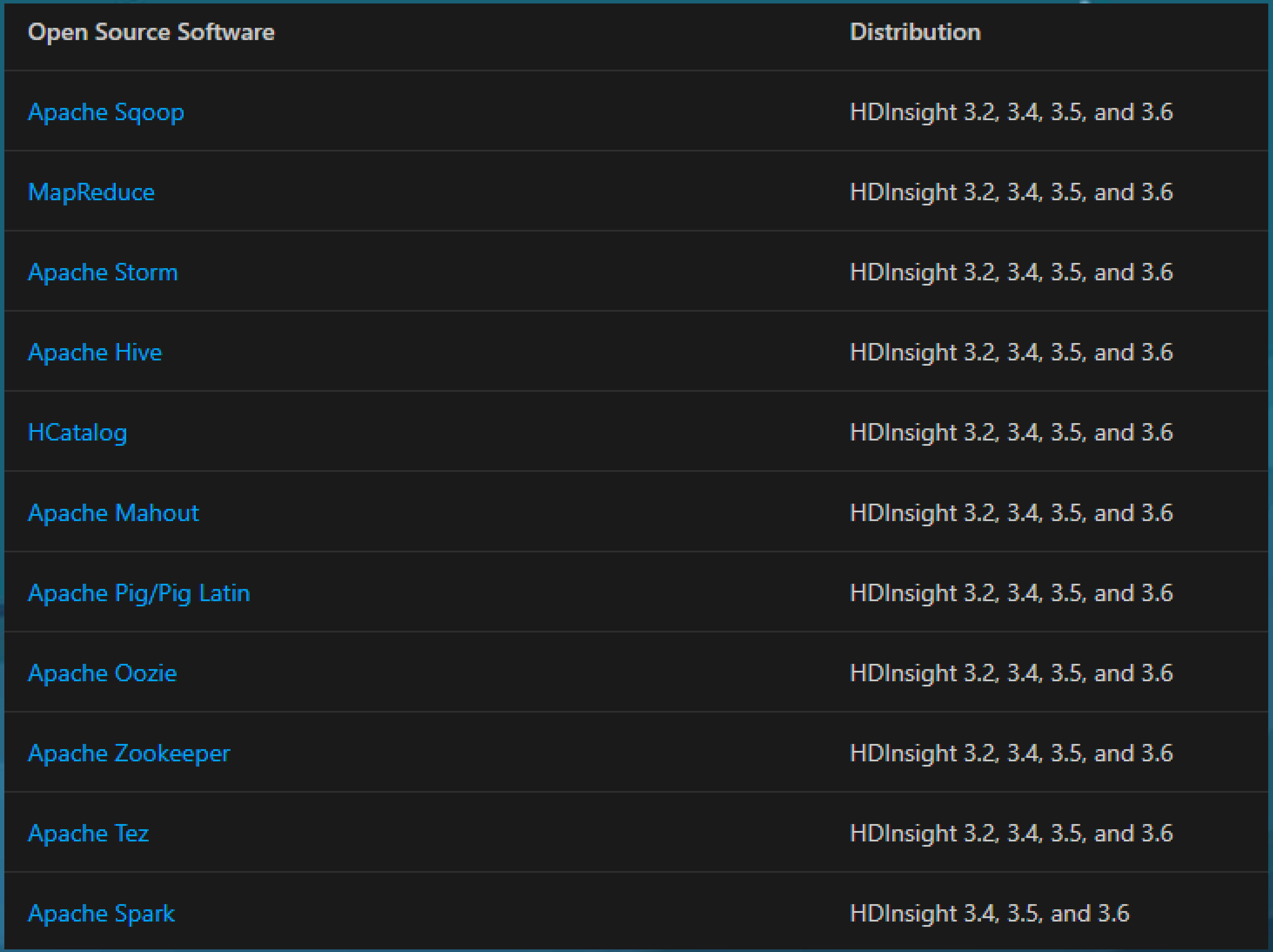
Key capabilities

- Built for Hadoop
- Unlimited storage, petabyte files
- Performance-tuned for big data analytics
- Enterprise-ready: Highly-available and secure
- All data

Security

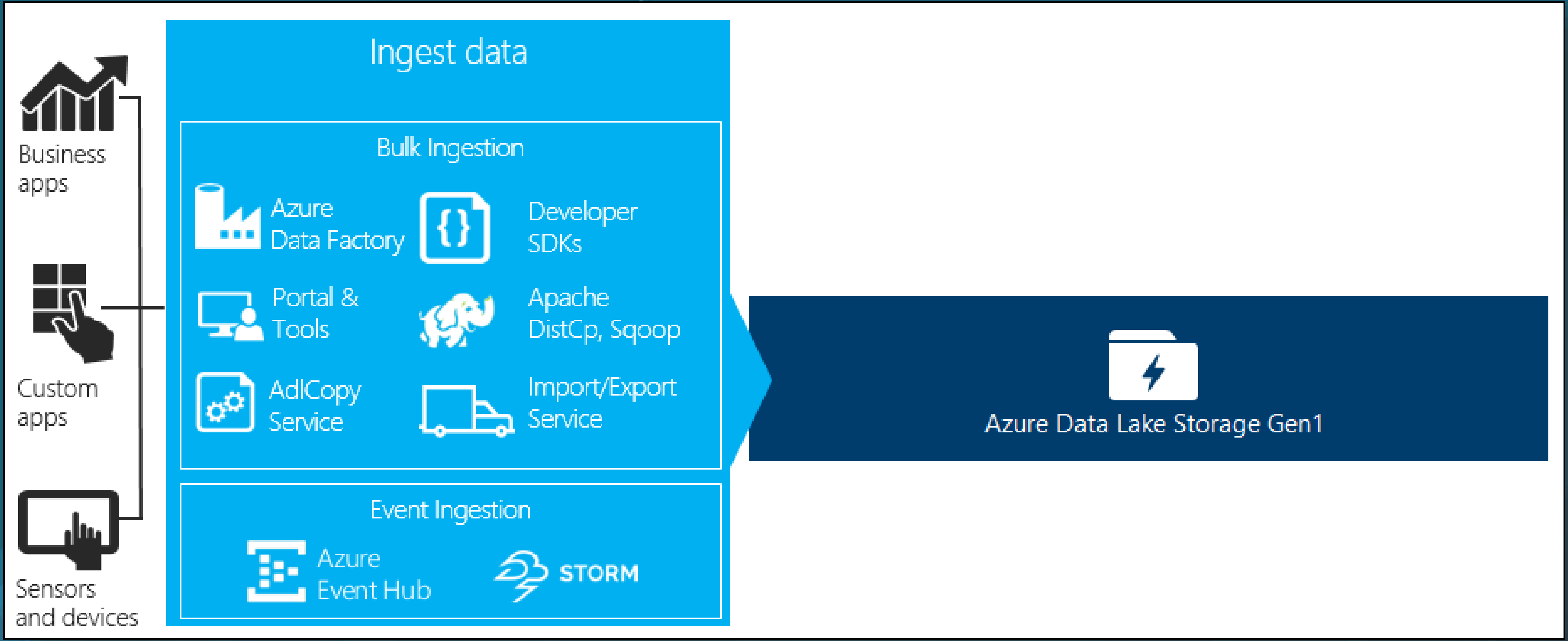
- Authentication
  - Azure Active Directory integration
  - Oauth 2.0 support for REST interface
- Access control
  - Supports POSIX-style permissions (exposed by WebHDFS)
  - ACLs on root, subfolders and individual files
- Encryption

Compatibility



Open Source Software	Distribution
Apache Sqoop	HDInsight 3.2, 3.4, 3.5, and 3.6
MapReduce	HDInsight 3.2, 3.4, 3.5, and 3.6
Apache Storm	HDInsight 3.2, 3.4, 3.5, and 3.6
Apache Hive	HDInsight 3.2, 3.4, 3.5, and 3.6
HCatalog	HDInsight 3.2, 3.4, 3.5, and 3.6
Apache Mahout	HDInsight 3.2, 3.4, 3.5, and 3.6
Apache Pig/Pig Latin	HDInsight 3.2, 3.4, 3.5, and 3.6
Apache Oozie	HDInsight 3.2, 3.4, 3.5, and 3.6
Apache Zookeeper	HDInsight 3.2, 3.4, 3.5, and 3.6
Apache Tez	HDInsight 3.2, 3.4, 3.5, and 3.6
Apache Spark	HDInsight 3.4, 3.5, and 3.6

Ingest data



Business apps
Custom apps
Sensors and devices
Ingest data
Bulk Ingestion
Azure Data Factory
Developer SDKs
Portal & Tools
Apache DistCp, Sqoop
AdlCopy Service
Import/Export Service
Event Ingestion
Azure Event Hub
STORM
Azure Data Lake Storage Gen1

Ingest data – Ad hoc

- Local computer
  - Azure Portal
  - Azure PowerShell
  - Azure CLI
  - Using Data Lake Tools for Visual Studio

- Azure Storage Blob
  - Azure Data Factory
  - AdlCopy tool
  - DistCp running on HDInsight cluster

Ingest data – Streamed data

- Azure Stream Analytics
- Azure HDInsight Storm
- EventProcessorHost

Ingest data – Relational data

- Apache Sqoop
- Azure Data Factory

Ingest data – Web server log data

Upload using custom applications

- Azure CLI
- Azure PowerShell
- Azure Data Lake Storage Gen1 .NET SDK
- Azure Data Factory

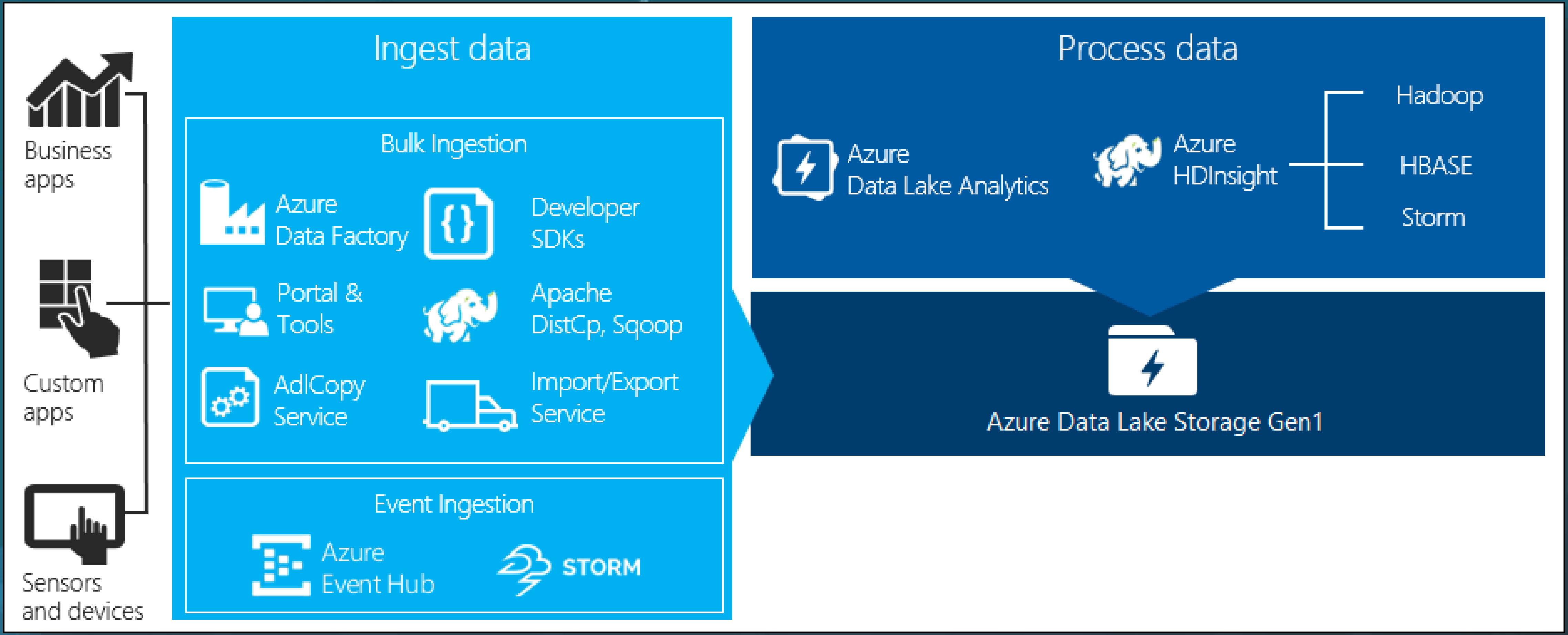
Ingest data - Data associated with Azure HDInsight clusters

- Apache DistCp
- AdlCopy service
- Azure Data Factory

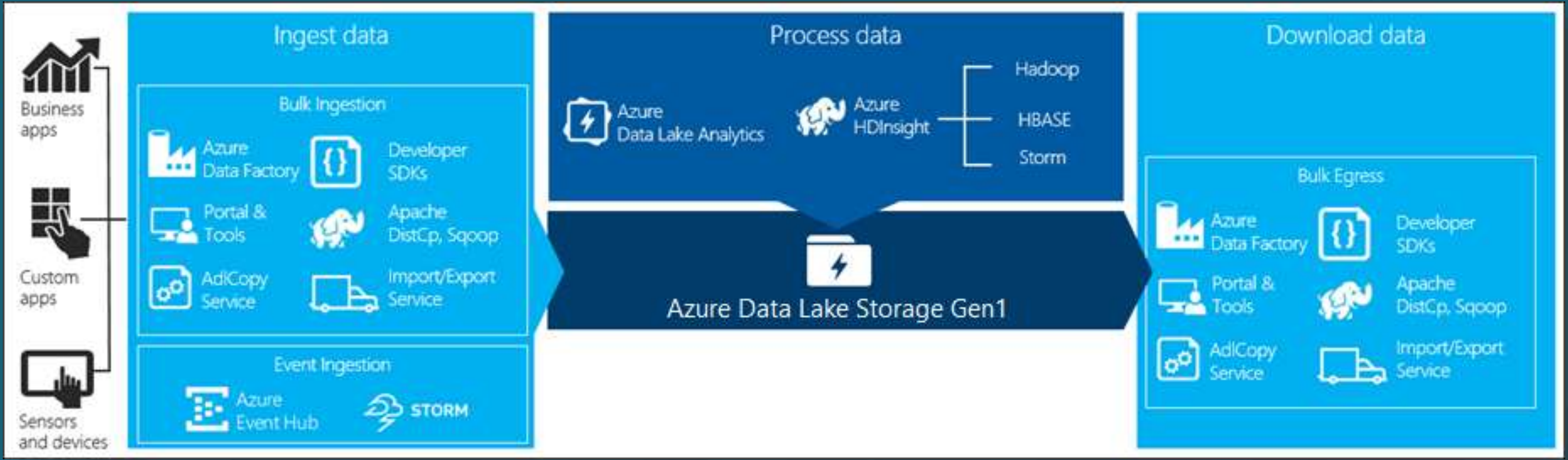
Ingest data – Really large datasets

- ExpressRoute
- “Offline” upload of data
  - Azure Import/Export service

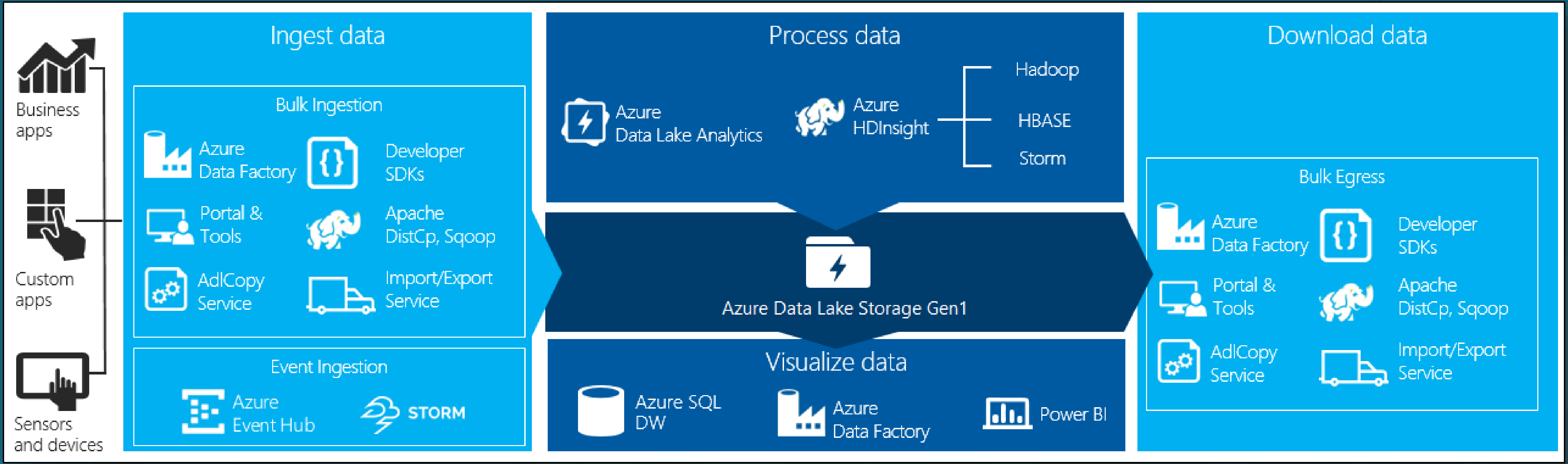
Process data



Download data



Visualize data



DEMO

Storage Gen2 (Preview)

- Dedicated to big data analytics
- Built on top of Azure Storage
- The only cloud-based multi-modal storage service

"In Data Lake Storage Gen2, all the qualities of object storage remain while adding the advantages of a file system interface optimized for analytics workloads."

Store Gen2 (Preview)

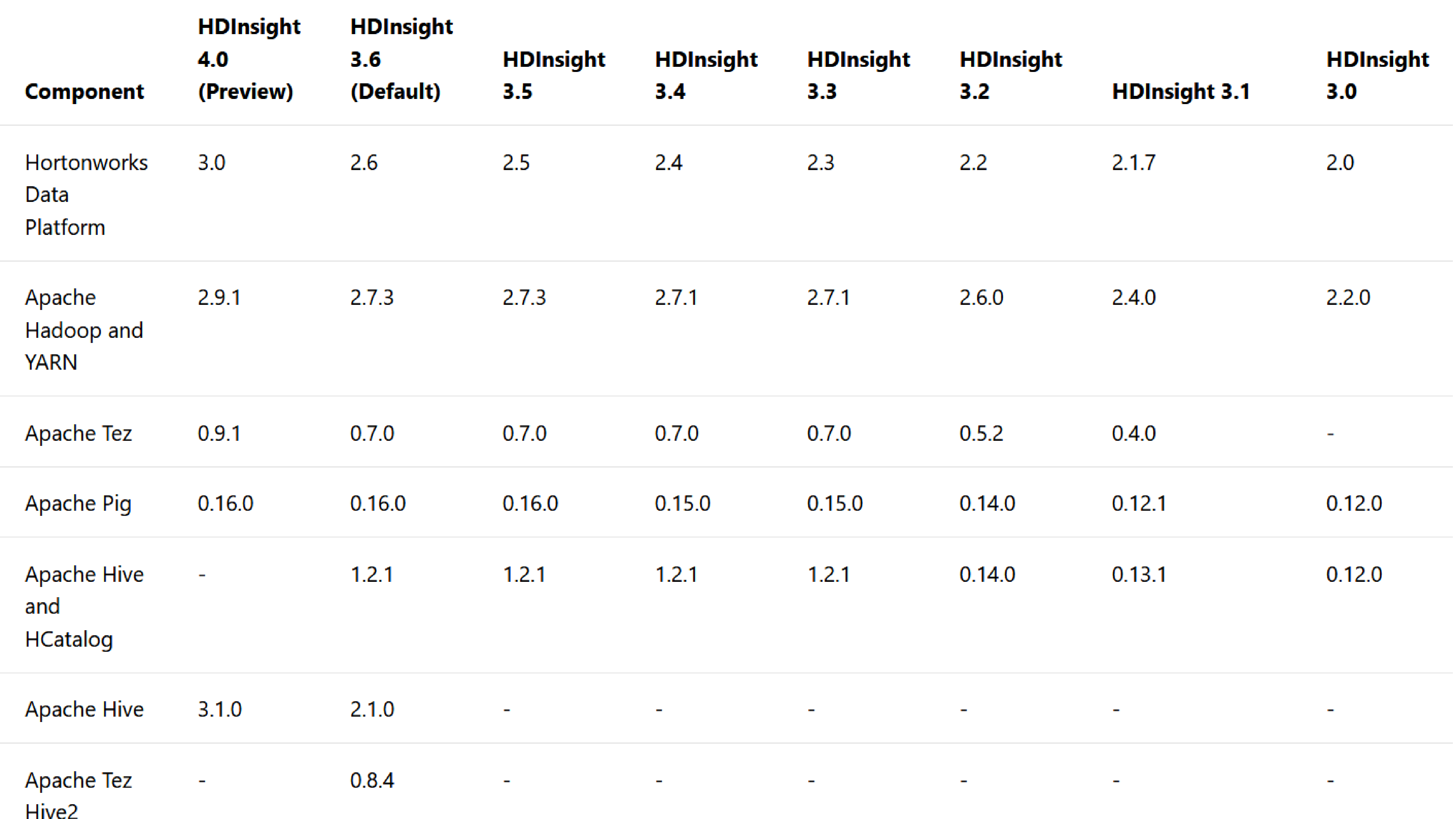
- Optimized performance
  - No need to copy or transform data
- Easier management
  - Organize and manipulate files through directories and subdirectories
- Enforceable security
  - POSIX permissions on folders or individual files
- Cost effectiveness
  - Built on top of the low-cost Azure Blob storage

AZURE DATA LAKE HDINSIGHT

HDInsight

- Cloud distribution of the (Hortonworks) Hadoop components
- Supports multiple Hadoop cluster versions (can be deployed any time)

- Hadoop
  - YARN for job scheduling & resource management
  - MapReduce for parallel processing
  - HDFS



Component	HDInsight 4.0 (Preview)	HDInsight 3.6 (Default)	HDInsight 3.5	HDInsight 3.4	HDInsight 3.3	HDInsight 3.2	HDInsight 3.1	HDInsight 3.0
Hortonworks Data Platform	3.0	2.6	2.5	2.4	2.3	2.2	2.1.7	2.0
Apache Hadoop and YARN	2.9.1	2.7.3	2.7.3	2.7.1	2.7.1	2.6.0	2.4.0	2.2.0
Apache Tez	0.9.1	0.7.0	0.7.0	0.7.0	0.7.0	0.5.2	0.4.0	-
Apache Pig	0.16.0	0.16.0	0.16.0	0.15.0	0.15.0	0.14.0	0.12.1	0.12.0
Apache Hive and HCatalog	-	1.2.1	1.2.1	1.2.1	1.2.1	0.14.0	0.13.1	0.12.0
Apache Hive	3.1.0	2.1.0	-	-	-	-	-	-
Apache Tez Hive2	-	0.8.4	-	-	-	-	-	-

Cluster types

- Apache Hadoop
- Apache Spark
- Apache Kafka
- Apache Interactive Query (AKA: Live Long and Process)
- Apache Storm
- Microsoft Machine Learning Services (R Server)

Component & utilities

- Ambari
- Avro
- Hive & HCatalog
- Mahout
- MapReduce
- Oozie
- Phoenix
- Pig
- Sqoop
- Tez
- YARN
- ZooKeeper

Languages - Default

- Java
  - Clojure
  - Jython
  - Scala

- Python

- Pig Latin (for Pig jobs)

- HiveQL for Hive jobs and SparkSQL

AZURE DATA LAKE *ANALYTICS*

Analytics

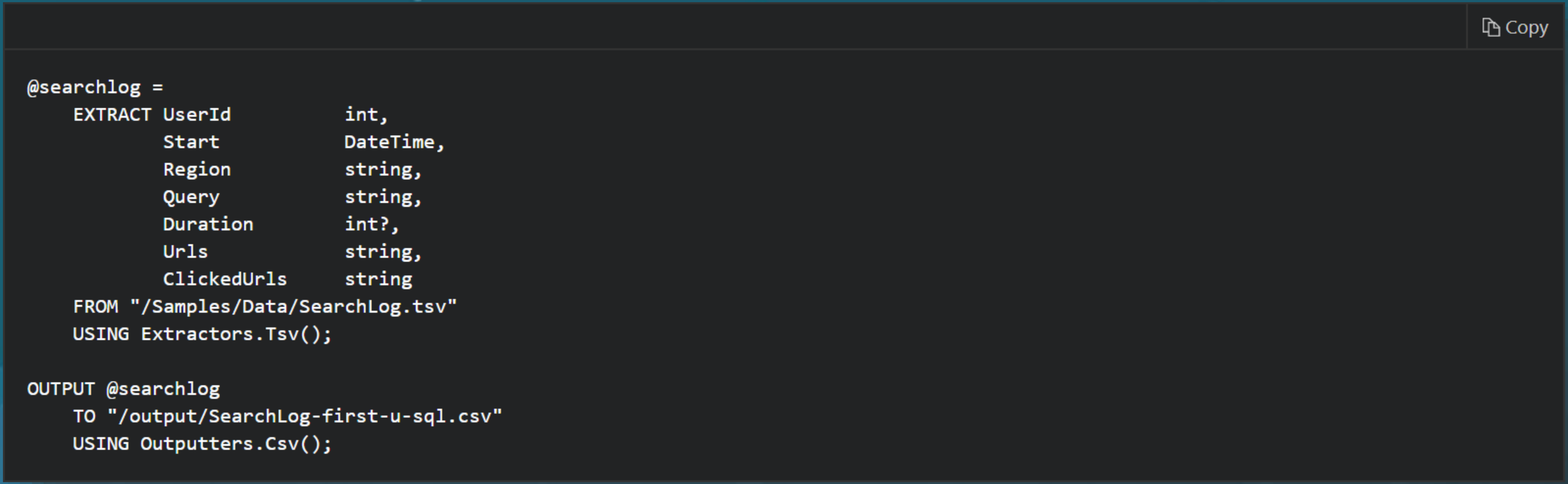
- Dynamic scaling
- Develop faster, debug and optimize smarter using familiar tools
- U-SQL: simple and familiar, powerful, and extensible
- Integrates seamlessly with your IT investments
- Affordable and cost effective
- Works with all your Azure data

Analytics

- on-demand analytics job service to simplify big data analytics
- can handle jobs of any scale instantly
- Azure Active Directory integration
- U-SQL

U-SQL

- language that combines declarative SQL with imperative C#



```
@searchlog =
    EXTRACT UserId          int,
            Start           DateTime,
            Region          string,
            Query           string,
            Duration        int?,
            Urls            string,
            ClickedUrls     string
    FROM "/Samples/Data/SearchLog.tsv"
    USING Extractors.Tsv();

OUTPUT @searchlog
    TO "/output/SearchLog-first-u-sql.csv"
    USING Outputters.Csv();
```

U-SQL – Key concepts

- Rowset variables
  - Each query expression that produces a rowset can be assigned to a variable.
- EXTRACT
  - Reads data from a file and defines the schema on read *
- OUTPUT
  - Writes data from a rowset to a file *

TECHORAMA

U-SQL – Scalar variables

```
DECLARE @in  string = "/Samples/Data/SearchLog.tsv";
DECLARE @out string = "/output/SearchLog-scalar-variables.csv";

@searchlog =
    EXTRACT    UserId        int,
               ClickedUrls   string
    FROM @in
    USING Extractors.Tsv();

OUTPUT @searchlog
    TO @out
    USING Outputters.Csv();
```

U-SQL – Transform rowsets

```
@searchlog =
    EXTRACT UserId    int,
            Region    string
    FROM "/Samples/Data/SearchLog.tsv"
    USING Extractors.Tsv();

@rs1 =
    SELECT UserId, Region
    FROM @searchlog
WHERE Region == "en-gb";

OUTPUT @rs1
    TO "/output/SearchLog-transform-rowsets.csv"
    USING Outputters.Csv();
```

U-SQL – Extractor parameters

- delimiter
- encoding
- escapeCharacter
- nullEscape
- quoting
- rowDelimiter
- silent
- skipFirstNRows
- charFormat

U-SQL – Outputter parameters

- delimiter
- dateTimeFormat
- encoding
- escapeCharacter
- nullEscape
- quoting
- rowDelimeter
- charFormat
- outputHeader

U-SQL

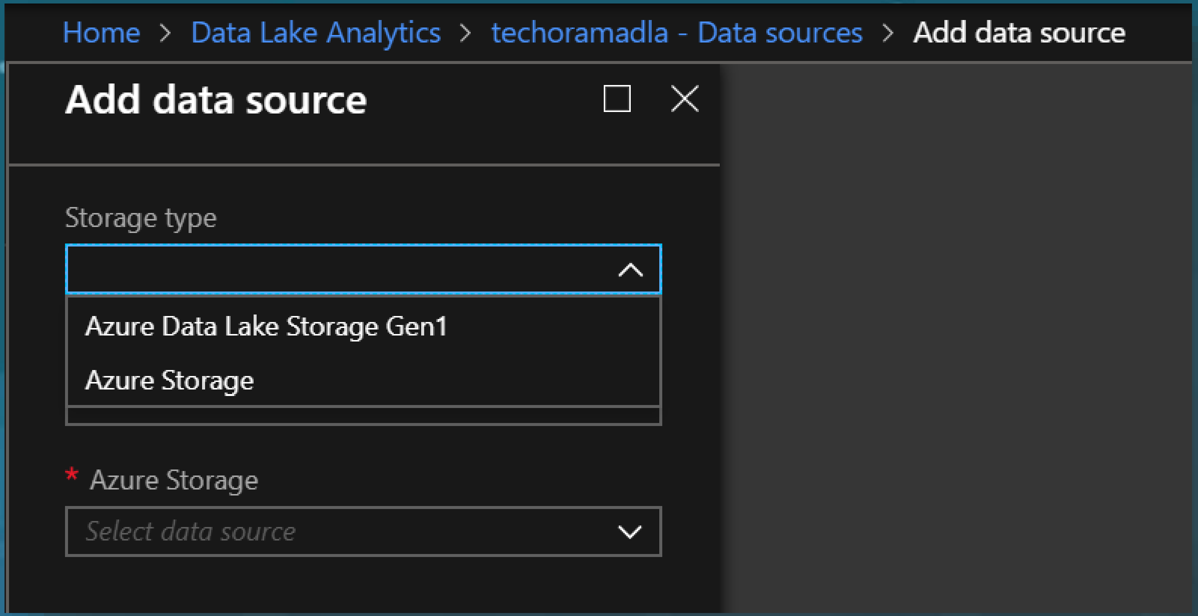
Built-in extractors and outputters:

- Text
- Csv
- Tsv

A (for instance) CSV Extractor or Outputter is **EXACTLY THAT**

Data sources

- Options in the Azure Portal:
  - Data Lake Storage Gen1
  - Azure Storage



Home > Data Lake Analytics > techoramadla - Data sources > Add data source

Add data source

Storage type

Azure Data Lake Storage Gen1

Azure Storage

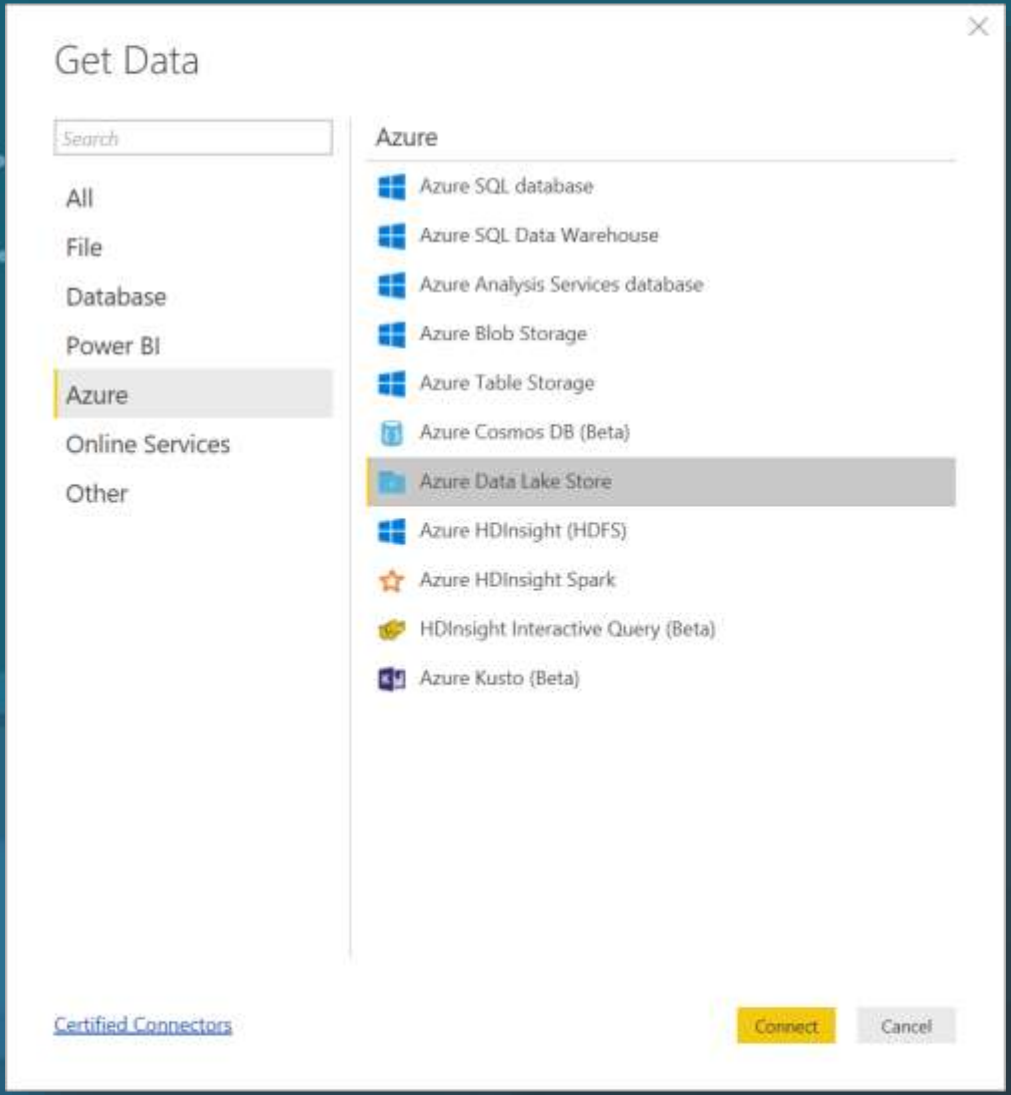
* Azure Storage

Select data source

DEMO

POWERBI

Power BI


Get Data
Search
All
File
Database
Power BI
Azure
Online Services
Other
Azure
Azure SQL database
Azure SQL Data Warehouse
Azure Analysis Services database
Azure Blob Storage
Azure Table Storage
Azure Cosmos DB (Beta)
Azure Data Lake Store
Azure HDInsight (HDFS)
Azure HDInsight Spark
HDInsight Interactive Query (Beta)
Azure Kusto (Beta)
Certified Connectors
Connect
Cancel

DEMO

USING AZURE SQL IN DATA LAKE ANALYTICS

TECHORAMA

DEEP KNOWLEDGE IT CONFERENCE

October 1-3 | 2018

Ede, The Netherlands

Data Sources

CREATE DATA SOURCE –statement

- Azure SQL Database
- Azure SQL Datawarehouse
- SQL Server 2012 and up in an Azure VM

Create Azure SQL Data Source

1. Make sure your SQL Server firewall settings allow Azure Services to connect
2. Create a ‘database’ in the Data Lake Analytics account
3. Create a Data Lake Analytics Catalog Credential
4. Create a Data Lake Analytics Data Source
5. Query your Azure SQL Database from Data Lake Analytics

Create ‘database’ in DLA (U-SQL)

```
CREATE DATABASE <YourDatabaseName>;
```

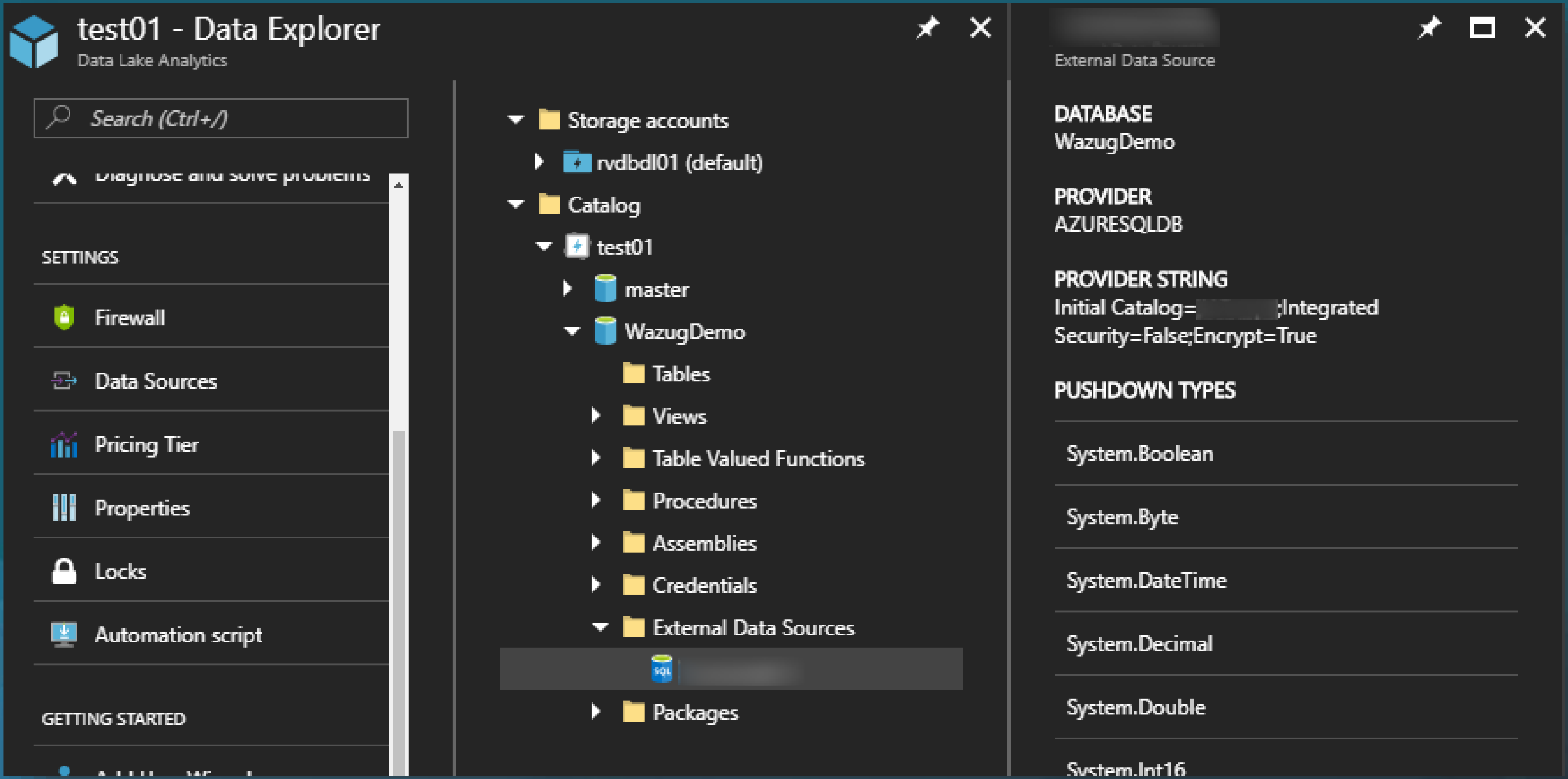
Create credential (PowerShell)

```
Login-AzureRmAccount;
Set-AzureRMContext -SubscriptionId <YourSubscriptionId>;

New-AzureRmDataLakeAnalyticsCatalogCredential
       -AccountName "<YourDLAAccount>"
       -DatabaseName "<YourDatabaseName>"
       -CredentialName "YourCredentialName"
       -Credential (Get-Credential)
       -DatabaseHost "<YourAzureSqlServer>.database.windows.net"
       -Port 1433;
```

Create Data Source (U-SQL)

```
USE DATABASE <YourDatabaseName>;

CREATE DATA SOURCE <YourDataSourceName>
FROM AZURESQLDB
WITH
(
        PROVIDER_STRING =
                "Database=<YourAzureSQLDatabaseName>;Trusted_Connection=False;
                 Encrypt=True",
        CREDENTIAL = <YourCredentialName>,
        REMOTABLE_TYPES = (bool, byte, sbyte, int, string, …)
);
```

Data Source (under Data Explorer)



test01 - Data Explorer
Data Lake Analytics

Search (Ctrl+/)

Diagnose and solve problems

SETTINGS

- Firewall
- Data Sources
- Pricing Tier
- Properties
- Locks
- Automation script

GETTING STARTED

- Storage accounts
  - rvdbdl01 (default)
- Catalog
  - test01
    - master
    - WazugDemo
      - Tables
      - Views
      - Table Valued Functions
      - Procedures
      - Assemblies
      - Credentials
      - External Data Sources
        - [redacted]
      - Packages

[redacted]
External Data Source

DATABASE
WazugDemo

PROVIDER
AZURESQLDB

PROVIDER STRING
Initial Catalog=[redacted];Integrated Security=False;Encrypt=True

PUSHDOWN TYPES

- System.Boolean
- System.Byte
- System.DateTime
- System.Decimal
- System.Double
- System.Int16

TECHORAMA

Query your Azure SQL Database (U-SQL)

```
USE DATABASE <YourDatabaseName>;


@results =
        SELECT *
        FROM EXTERNAL <YourDataSourceName> EXECUTE
                @"<QueryForYourSQLDatabase>";

OUTPUT @results
TO "<OutputFileName>"
USING Outputters.Csv(outputHeader: true);
```

Resources

- Basic example
- Advanced example
- Create Database (U-SQL) & Create Data Source (U-SQL)

- This example

- Azure blog
- Azure roadmap

- rickvandenbosch.net

THANK YOU

TECHORAMA

DEEP KNOWLEDGE IT CONFERENCE

October 1-3 | 2018 Ede, The Netherlands

TECHORAMA

DEEP KNOWLEDGE IT CONFERENCE

October 1-3 | 2018

Ede, The Netherlands