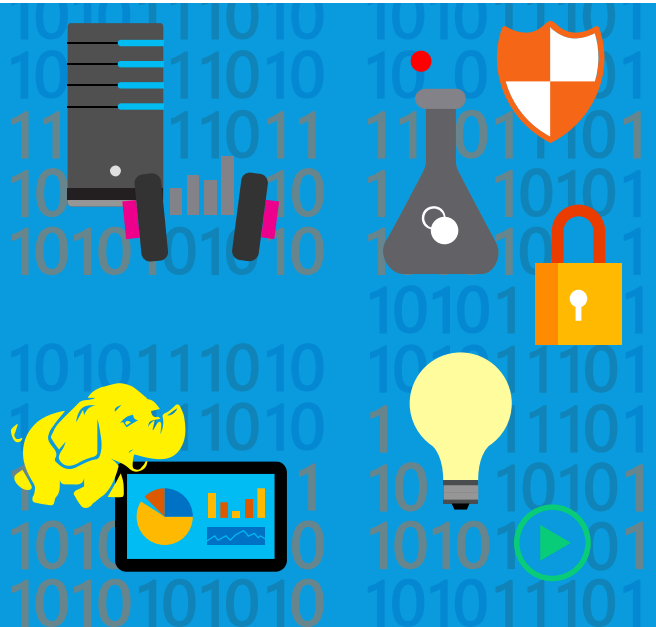
AZURE DATA LAKE

Azure Machine Learning Track

Marek Chmel

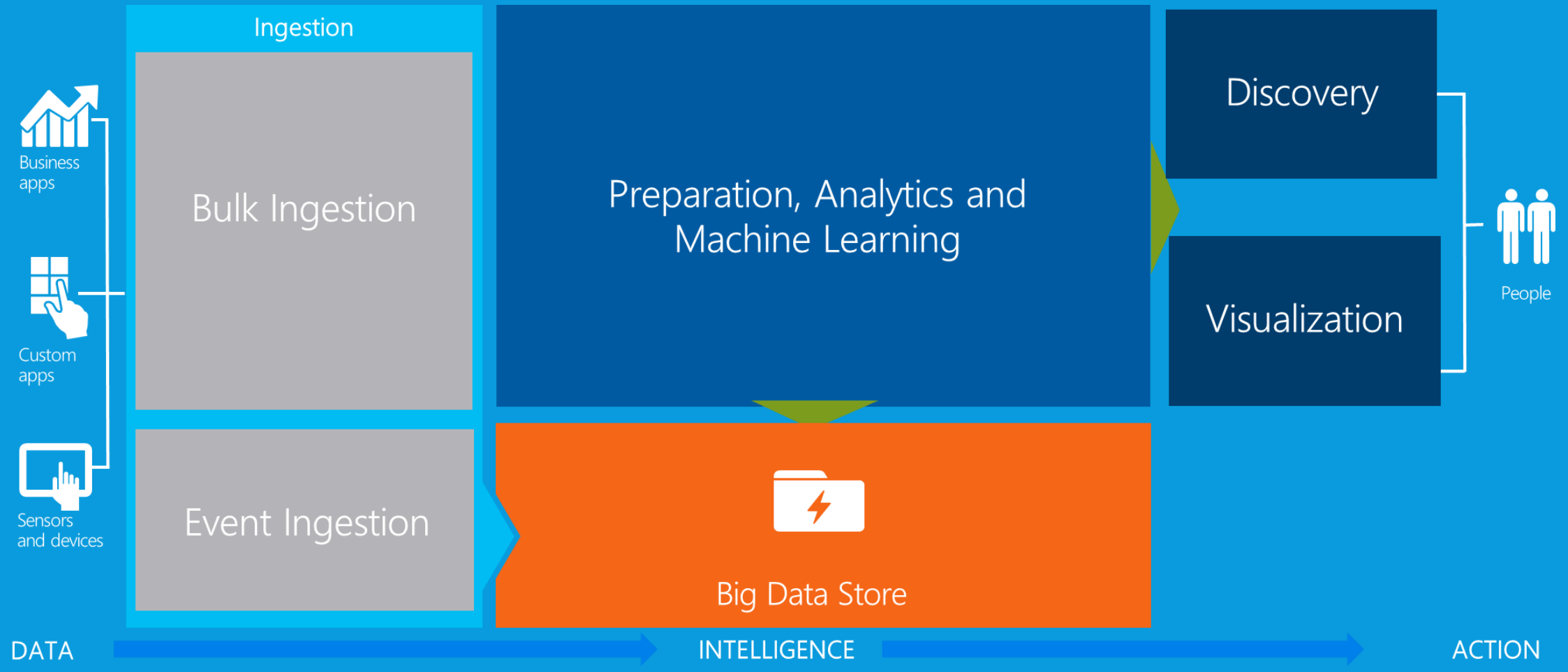
2017 Global Azure BOOTCAMP



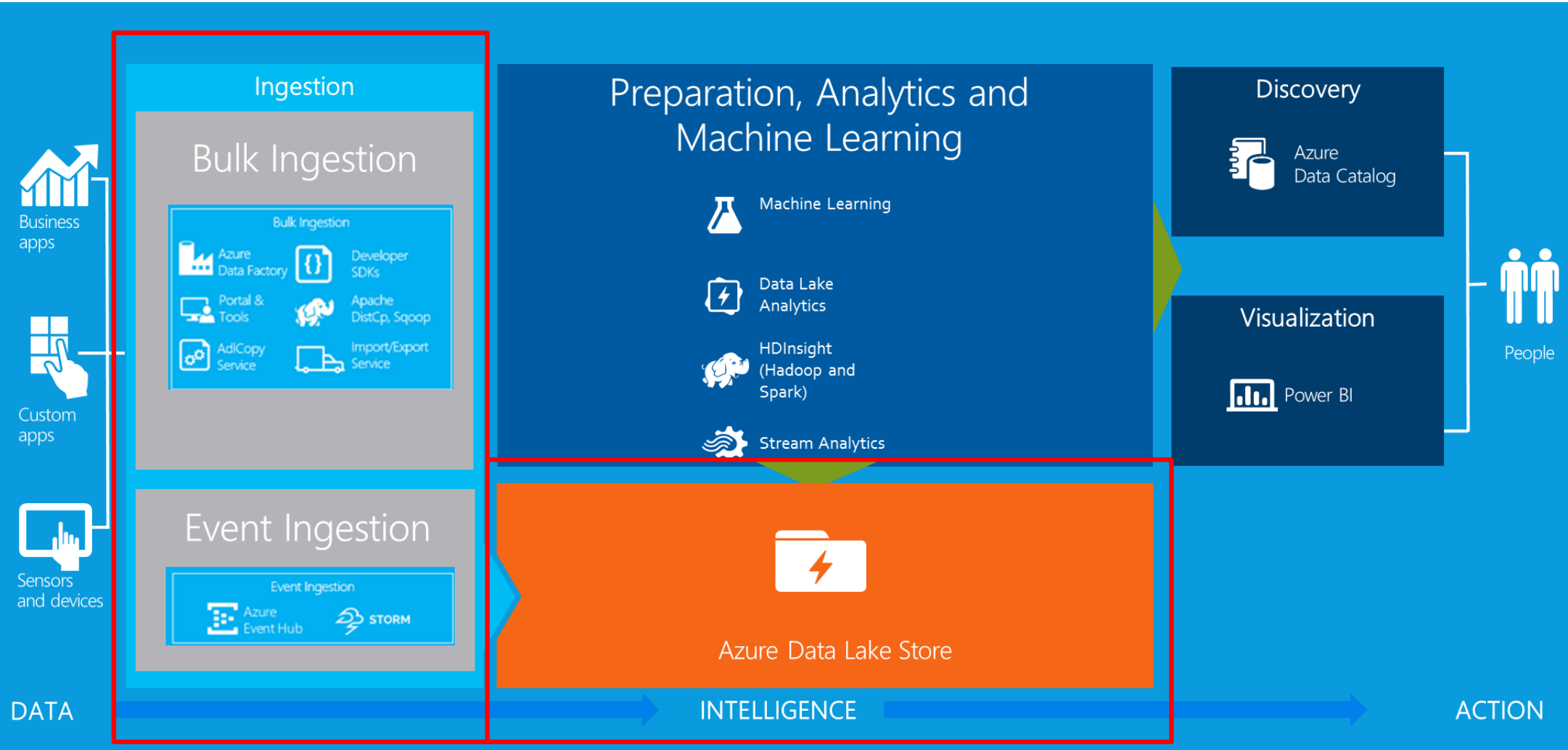
SESSION AGENDA

- Big Data Pipeline and Workflow
- Azure Data Lake Overview
- Data Ingestion
- Azure Data Lake Analytics
- U-SQL

BIG DATA PIPELINE AND WORKFLOW

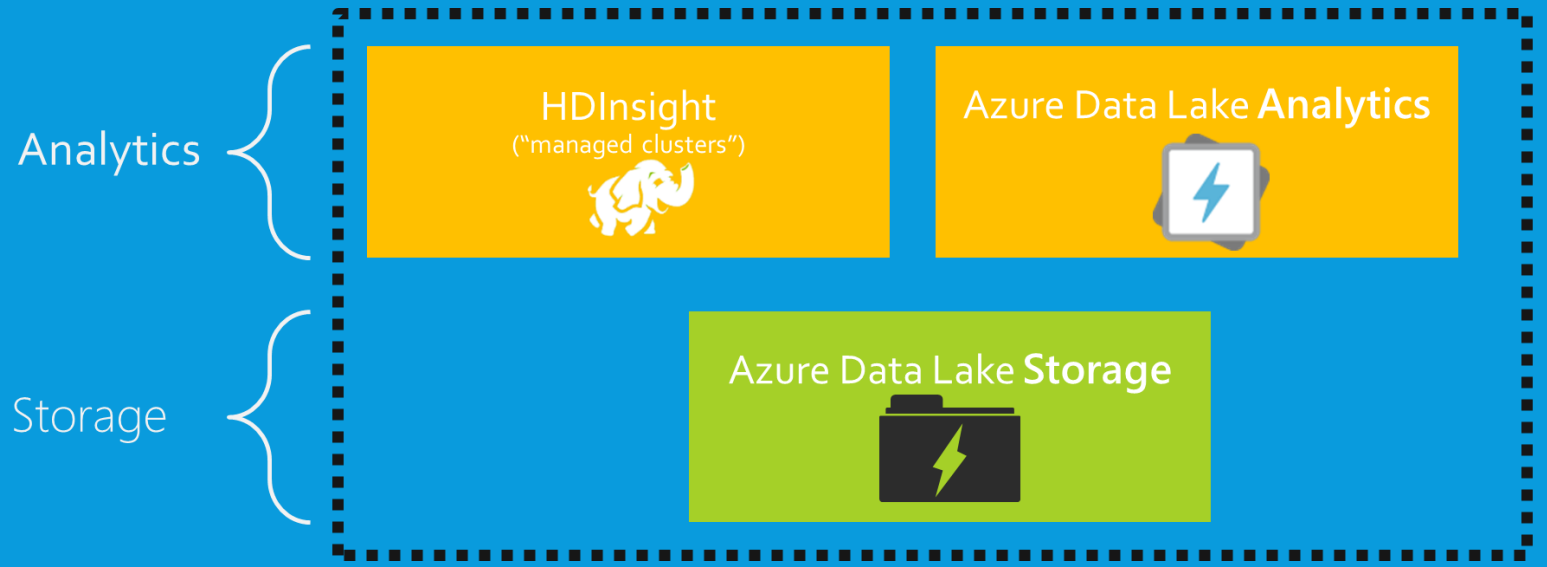


BIG DATA PIPELINE AND DATA FLOW IN AZURE

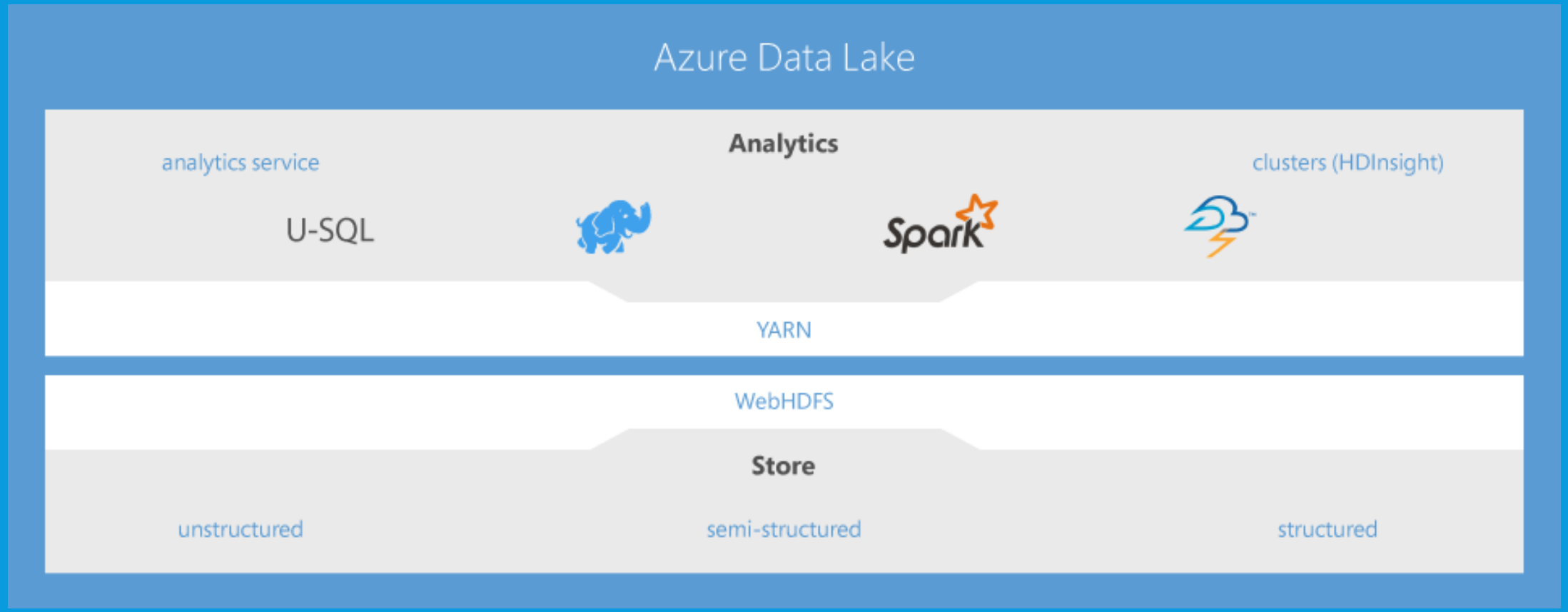


DATA LAKE OVERVIEW

- "A single store of all data… ranging from raw data (which implies exact copy of source system data) to transformed data which is used for various forms including reporting, visualization, analytics, and machine learning"



AZURE DATA LAKE



ADL STORE

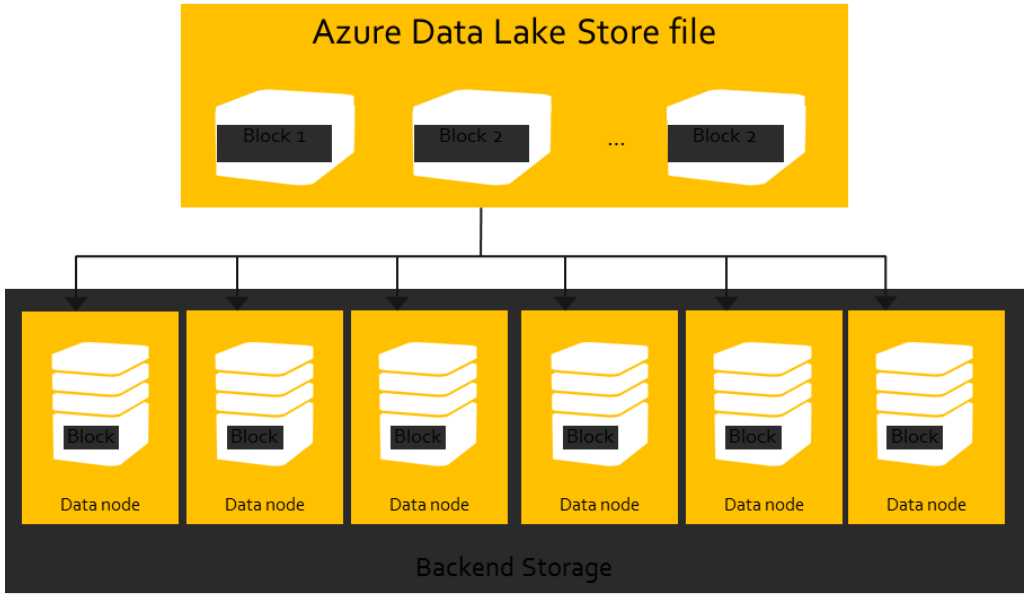
- HDFS-as-a-service
- Durable, redundant storage
- A variety of data scenarios
  - High capacity
  - High frequency
  - High throughput
- Store data in its native format
  - Structured, semi-structured, unstructured storage formats

NO LIMITS TO SCALE

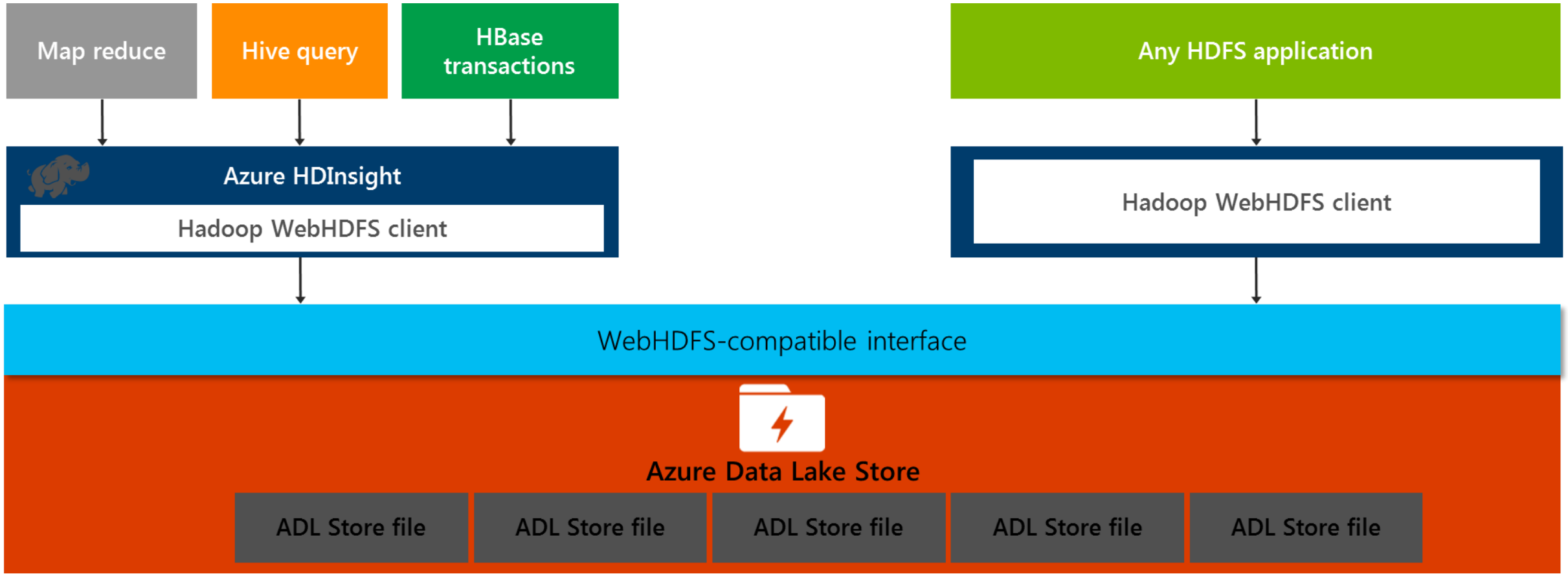
- No fixed limits on:
- Amount of data stored
- How long data can be stored
- Number of files
- Size of the individual files
- Ingestion/egress throughput

Seamlessly scales from a few KBs to several PBs

NO LIMITS TO STORAGE



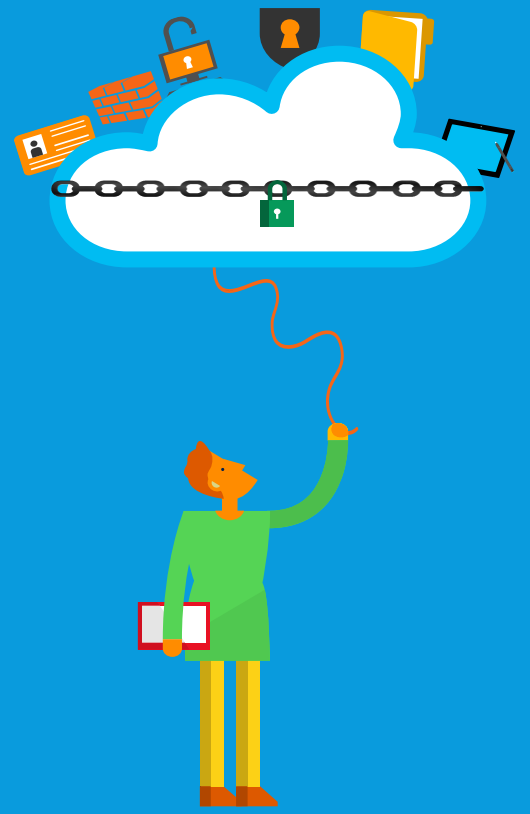
- Each file in ADL Store is sliced into blocks
- Blocks are distributed across multiple data nodes in the backend storage system
- With sufficient number of backend storage data nodes, files of any size can be stored
- Backend storage runs in the Azure cloud which has virtually unlimited resources
- Metadata is stored about each file
  No limit to metadata either.



HDFS-COMPATIBLE

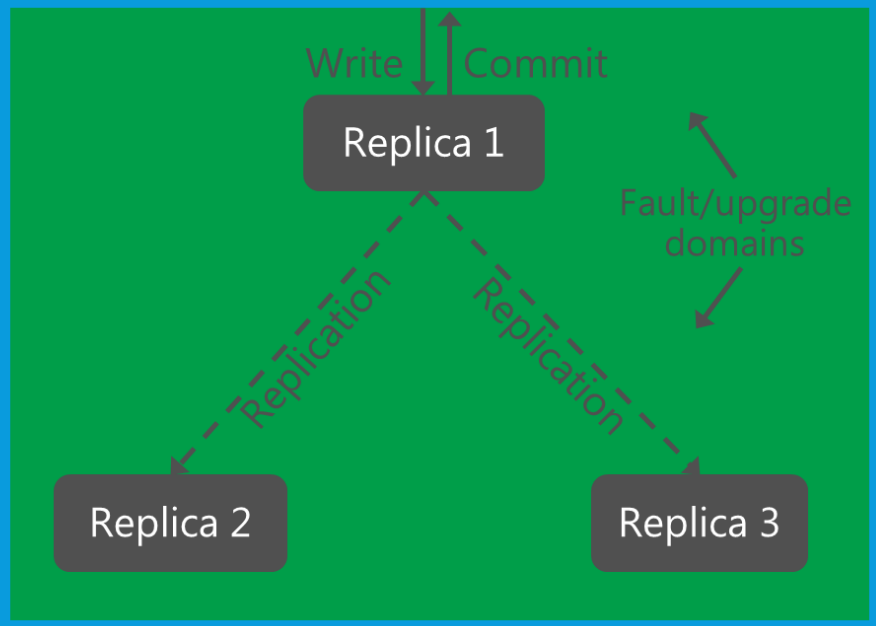
ENTERPRISE GRADE SECURITY

- Enterprise-grade security permits even sensitive data to be stored securely
- Regulatory compliance can be enforced
- Integrates with Azure Active Directory for authentication
- Data is encrypted at rest and in flight
- POSIX-style permissions on files and directories
- Audit logs for all operations



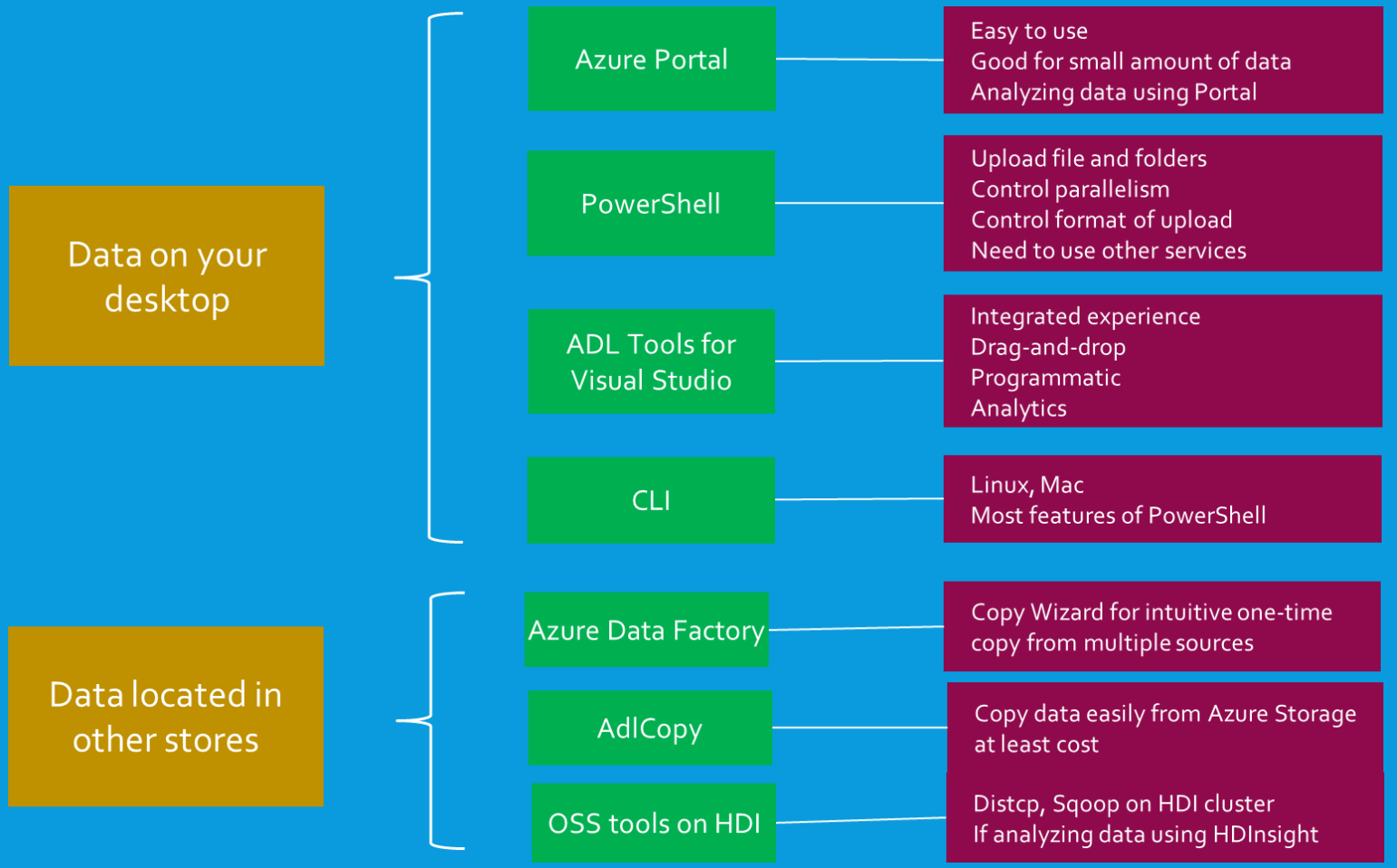
ENTERPRISE GRADE AVAILABILITY AND RELIABILITY

- Azure maintains 3 replicas of each data object per region across three fault and upgrade domains
- Each create or append operation on a replica is replicated to other two
- Writes are committed to application only after all replicas are successfully updated
- Read operations can go against any replica
- Provides 'read-after-write' consistency



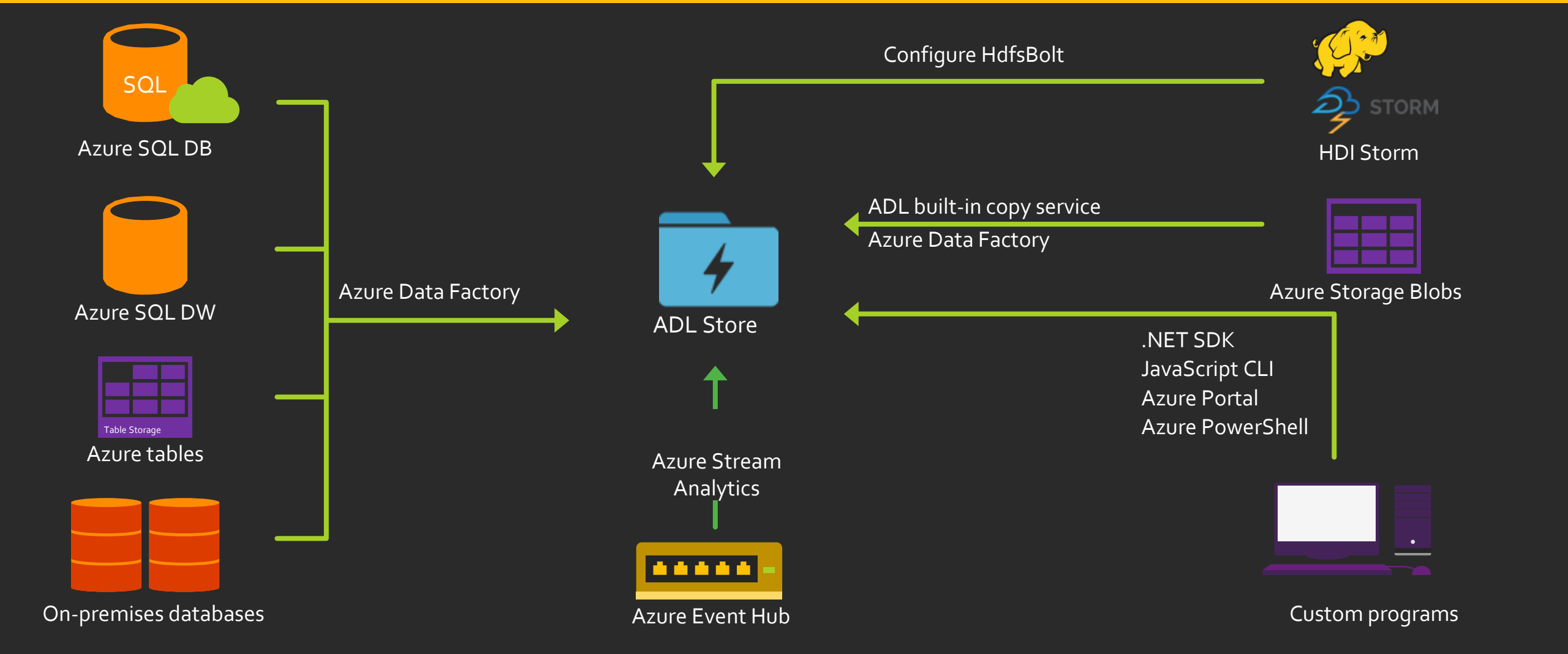
Data is never lost or unavailable even under failures

ADLS: TOOLS FOR DATA INGESTION



ADLS: INGRESS

Data can be ingested into Azure Data Lake Store from a variety of sources



ADLS: MOVE REALLY LARGE DATASETS

- Azure ExpressRoute
  - Dedicated private connections
  - Supported bandwidth up to 10Gbps
- "Offline"
  - Azure Import/Export service
  - Data is first uploaded to Azure Storage Blobs
  - Use Azure Data Factory or AdlCopy to copy data from Azure Storage Blobs to Data Lake Store

ADLS: ADLCOPY

- A command line tool
- Copy data
  - Azure Storage Blobs <==>Azure Data Lake Store
  - Azure Data Lake Store <==>Azure Data Lake Store
- Run in two ways:
  - Standalone
  - Using a Data Lake Analytics account

```
AdlCopy /Source <Blob source> /Dest <ADLS destination> /SourceKey
<Key for Blob account> /Account <ADLA account> /Units <Number of
Analytics units>
```

ADLS: DISTCP

- Copy data
  - M/R Hadoop job
  - HDInsight cluster storage <==> Data Lake Store account

```
hadoop distcp
wasb://<container_name>@<storage_account_name>.blob.core.windows.net/example/dat
a/gutenberg adl://<data_lake_store_account>.azuredatalakestore.net:443/myfolder
```

ADLS: SQOOP

- Apache Sqoop is a tool designed to transfer data between relational databases and a big data repository, such as Data Lake Store.
- You can use Sqoop to copy data **to and from** Azure SQL database into a Data Lake Store account, in addition to other other relational DBs.
- More details are here.

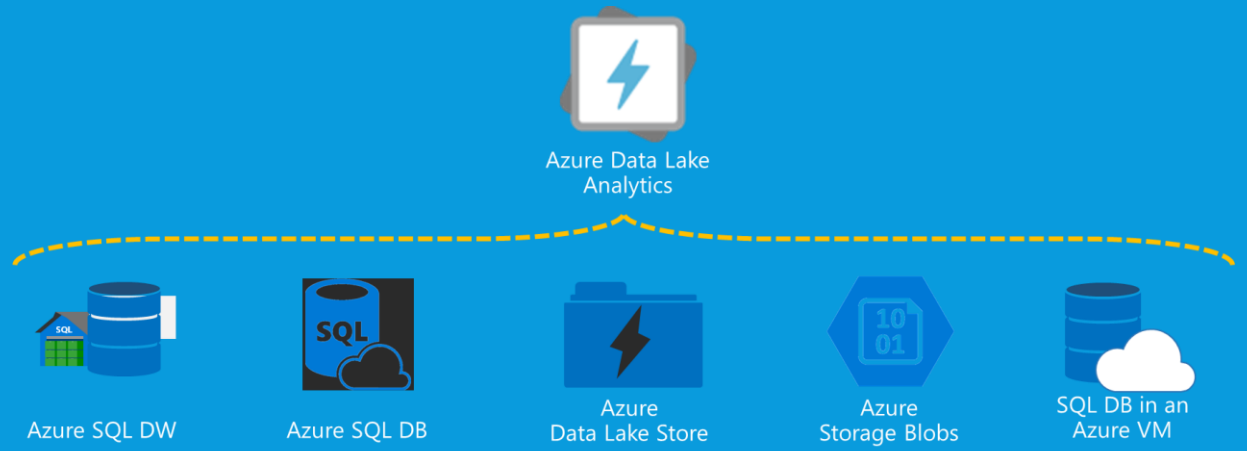
```
sqoop-import --connect "jdbc:sqlserver://<sql-database-server-name>.database
.windows.net:1433;username=<username>@<sql-database-server-name>;password=
<password>;database=<sql-database-name>“ --table Table1 --target-dir adl://
<data-lake-store-name>.azuredatalakestore.net/Sqoop/SqoopImportTable1
```

BIG DATA IS DRIVING TRANSFORMATIVE CHANGES

	Traditional	Big Data
Data Characteristics	Relational (with highly modeled schema)	
Cost	Expensive (storage and compute capacity)	Commodity (storage and compute capacity)
Culture	Rear-view reporting (using relational algebra)	Intelligent action (using relational algebra AND ML, graph, streaming, image processing)

ADL ANALYTICS

- Built on Apache YARN
- Scales dynamically with the turn of a dial
- Pay by the query
- Supports Azure AD for access control, roles, and integration with on-prem identity systems
- Built with U-SQL to unify the benefits of SQL with the power of C#
- Processes data across Azure



ADLA VS HDINSIGHT

ADLA COMPLEMENTS HDINSIGHT

HDInsight (Cluster as a Service)

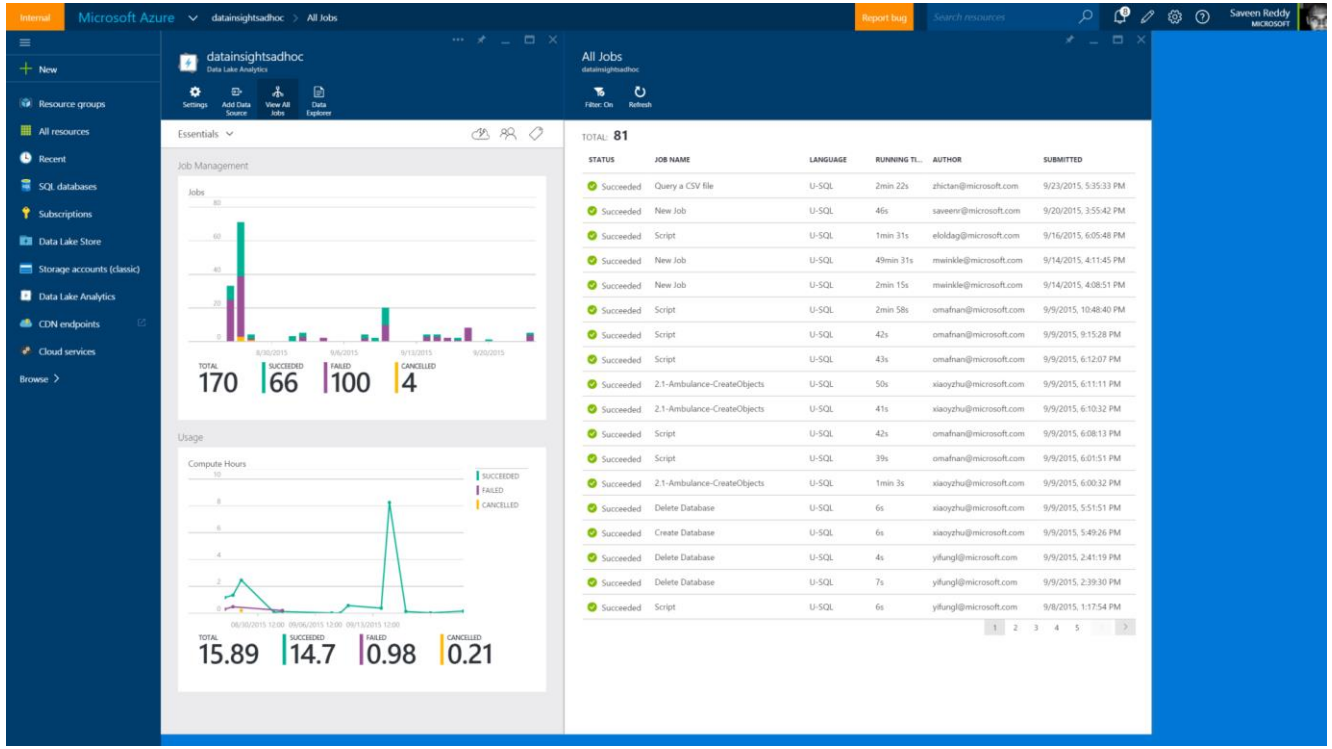
- Provision cluster of n nodes
- Run your queries
- Delete cluster
- (Repeat)
- For developers familiar with the Open Source: Java, Eclipse, Hive, etc.
- Clusters offer customization, control, and flexibility in a managed Hadoop cluster

ADLA (Query as a service)

- Don't provision anything
- Specify node count (parallelism) at job submission time
- Pay per query
- Enables customers to leverage existing experience with C#, SQL & PowerShell
- Offers convenience, efficiency, automatic scale, and management in a "job service" form factor

SIMPLIFIED MANAGEMENT AND ADMINISTRATION

- Web-based management in Azure Portal
- Automate tasks using PowerShell
- Role-based access control with Azure AD
- Monitor service operations and activity



CHARACTERISTICS OF BIG DATA ANALYTICS

- Requires processing of any type of data
- Allow use of custom algorithms
- Scale to any size and be efficient

Some sample use cases

Digital Crime Unit – Analyze complex attack patterns to understand BotNets and to predict and mitigate future attacks by **analyzing log records with complex custom algorithms**

Image Processing – **Large-scale** image feature extraction and classification using **custom code**

Shopping Recommendation – Complex pattern analysis and prediction over shopping records using **proprietary algorithms**

CHARACTERISTICS OF BIG DATA ANALYTICS

- Requires processing of any type of data
- Allow use of custom algorithms
- Scale to any size and be efficient

Some sample use cases

Digital Crime Unit – Analyze complex attack patterns to understand BotNets and to predict and mitigate future attacks by **analyzing log records with complex custom algorithms**

Image Processing – **Large-scale** image feature extraction and classification using **custom code**

Shopping Recommendation – Complex pattern analysis and prediction over shopping records using **proprietary algorithms**

STATUS QUO: PROGRAMMING LANGUAGES FOR BIG DATA

- Requires processing of any type of data
- Allow use of custom algorithms
- Scale to any size and be efficient

☺ **Extensibility through custom code is "native"**

☹ **Declarativity is bolted on and not "native"**

- ☹ User often has to care about scale and performance
- ☹ SQL is 2nd class within string
- ☹ Often no code reuse/ sharing across queries

WHY U-SQL?

- Requires processing of any type of data
- Allow use of custom algorithms
- Scale to any size and be efficient

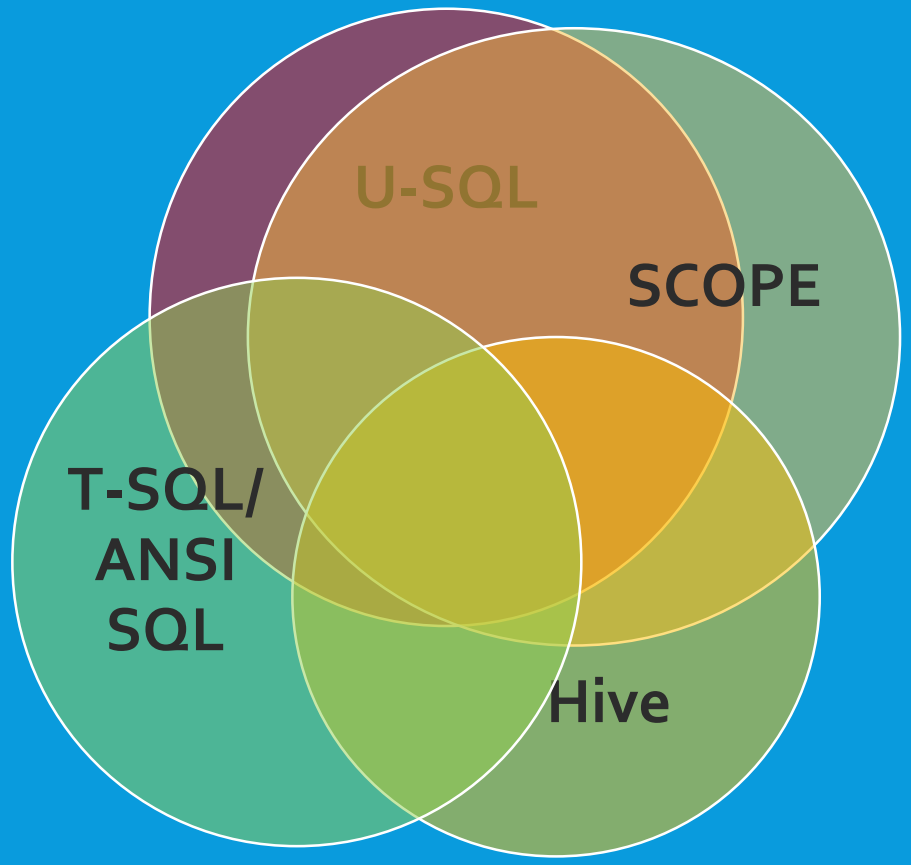
☺☺ Declarativity and Extensibility are equally native to the language!

Get benefits of both!

Makes it **easy** for you by **unifying**:

- Unstructured and structured data processing
- Declarative SQL and custom imperative Code
- Local and remote Queries
- Increase productivity and agility from Day 1 and at Day 100 for **YOU**!

THE ORIGINS OF U-SQL



SCOPE – Microsoft's internal Big Data language

- SQL and C# integration model
- Optimization and Scaling model
- Runs 100'000s of jobs daily

Hive

- Complex data types (Maps, Arrays)
- Data format alignment for text files

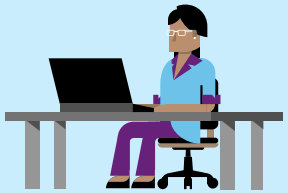
T-SQL/ANSI SQL

- Many of the SQL capabilities (windowing functions, meta data model etc.)

U-SQL EXTENSIBILITY

Built-in operators, function, aggregates

C# expressions (in SELECT expressions)



User-defined operators (UDOs)

User-defined functions (UDFs)

User-defined aggregates (UDAGGs)

U-SQL Language Philosophy

Declarative Query and Transformation Language:

- Uses SQL's SELECT FROM WHERE with GROUP BY/Aggregation, Joins, SQL Analytics functions
- Optimizable, Scalable

Expression-flow programming style:

- Easy to use functional lambda composition
- Composable, globally optimizable

Operates on Unstructured & Structured Data

- Schema on read over files
- Relational metadata objects (e.g. database, table)

Extensible from ground up:

- Type system is based on C#
- Expression language IS C#
- User-defined functions (U-SQL and C#)
- User-defined Aggregators (C#)
- User-defined Operators (UDO) (C#)

U-SQL provides the Parallelization and Scale-out Framework for Usercode

- EXTRACTOR, OUTPUTTER, PROCESSOR, REDUCER, COMBINER, APPLIER

Federated query across distributed data sources

```
REFERENCE MyDB.MyAssembly;

CREATE TABLE T( cid int, first_order DateTime
              , last_order DateTime, order_count int
              , order_amount float );

@o = EXTRACT oid int, cid int, odate DateTime, amount float
     FROM "/input/orders.txt"
     USING Extractors.Csv();

@c = EXTRACT cid int, name string, city string
     FROM "/input/customers.txt"
     USING Extractors.Csv();

@j = SELECT c.cid, MIN(o.odate) AS firstorder
           , MAX(o.date) AS lastorder, COUNT(o.oid) AS ordercnt
           , AGG<MyAgg.MySum>(c.amount) AS totalamount
     FROM @c AS c LEFT OUTER JOIN @o AS o ON c.cid == o.cid
     WHERE c.city.StartsWith("New")
       && MyNamespace.MyFunction(o.odate) > 10
     GROUP BY c.cid;

OUTPUT @j TO "/output/result.txt"
USING new MyData.Write();

INSERT INTO T SELECT * FROM @j;
```