

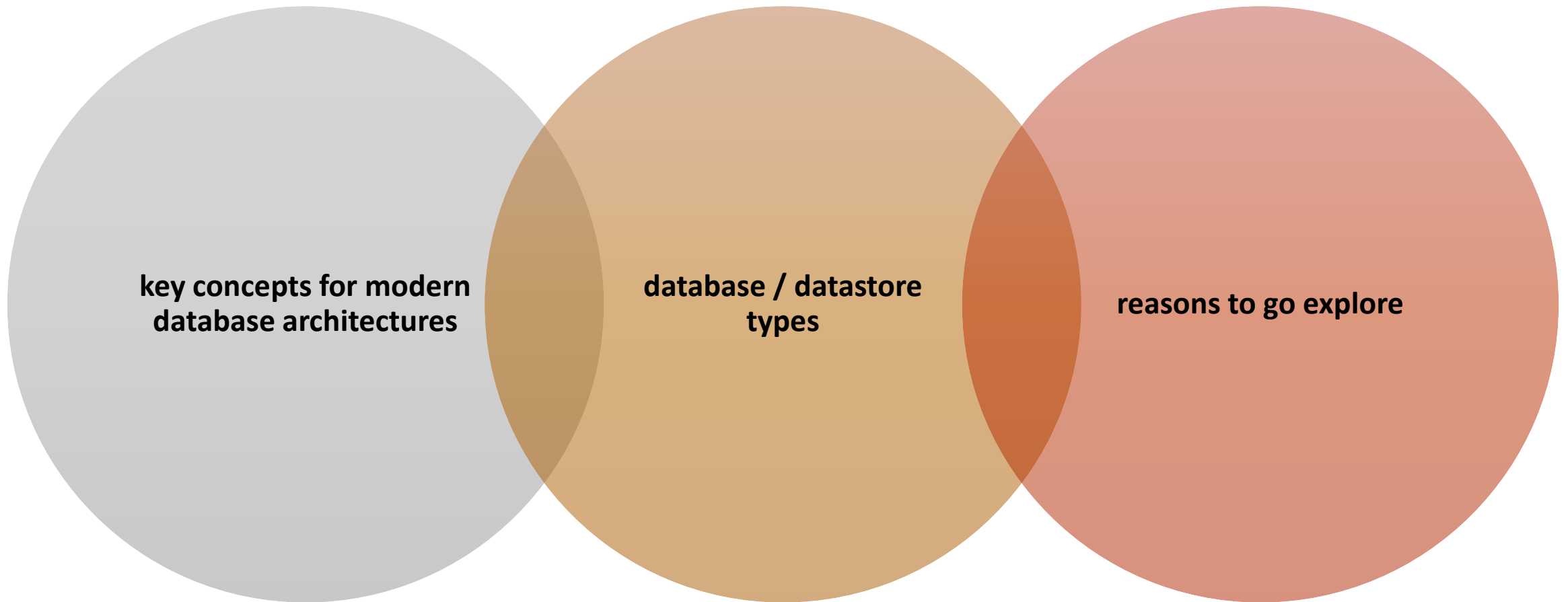
Using Azure Data Services for Modern Data Applications

Lara Rubbelke | Principal SDE | Microsoft

Allan Mitchell | Cloud Architect | elastabytes

Outcomes

We want you to leave here understanding:



Agenda

- Context of Lambda Architecture
- Ingestion
- Hot Path
 - Processing
- Cold Path
 - Processing
 - Staging
- Enrichment and Serving

4 Questions

What Happened?



What's Happening?



Why?

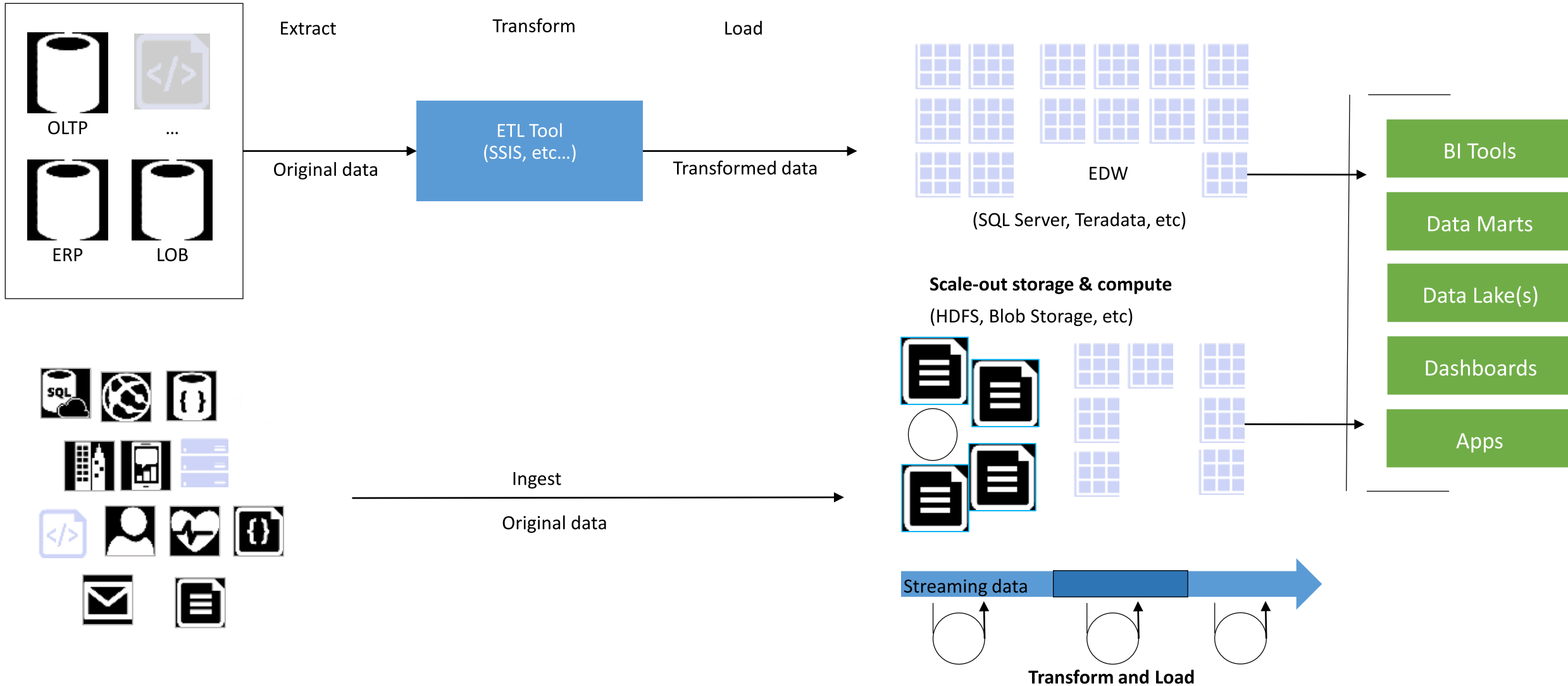




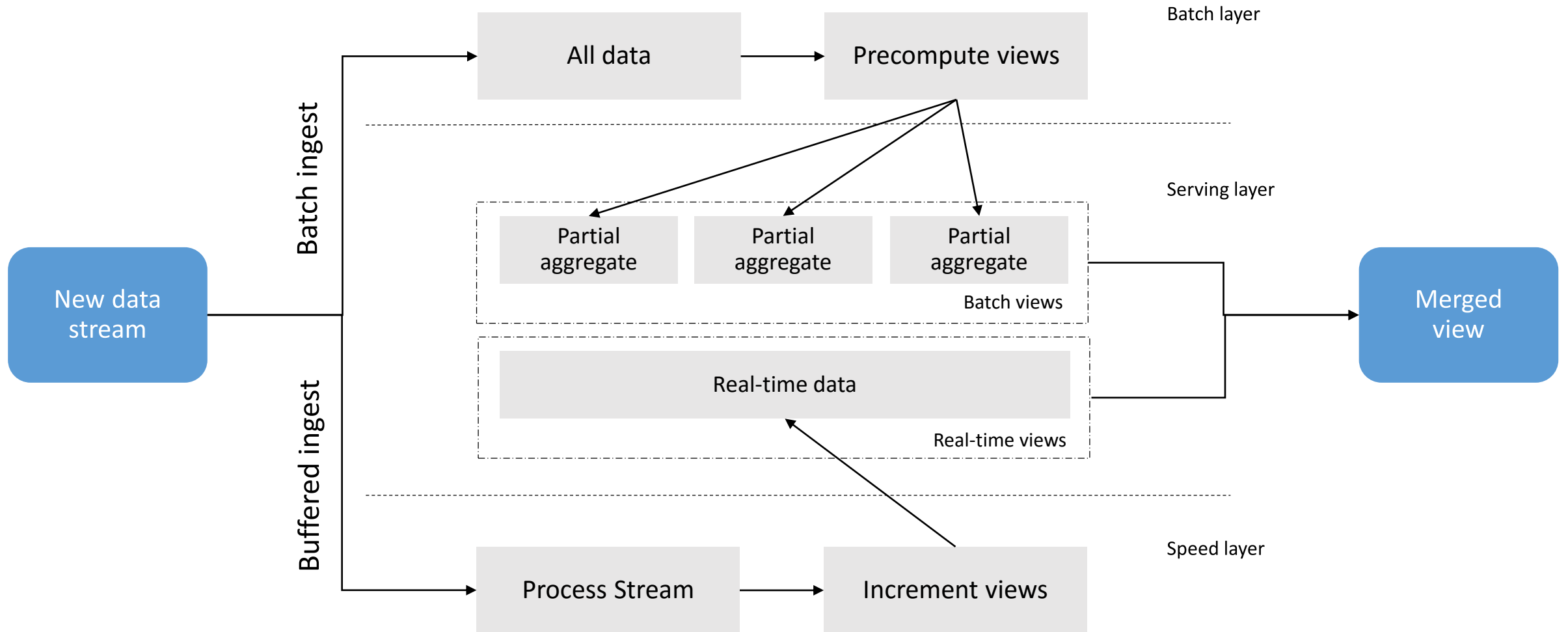
What Will Happen?

Lambda Architecture

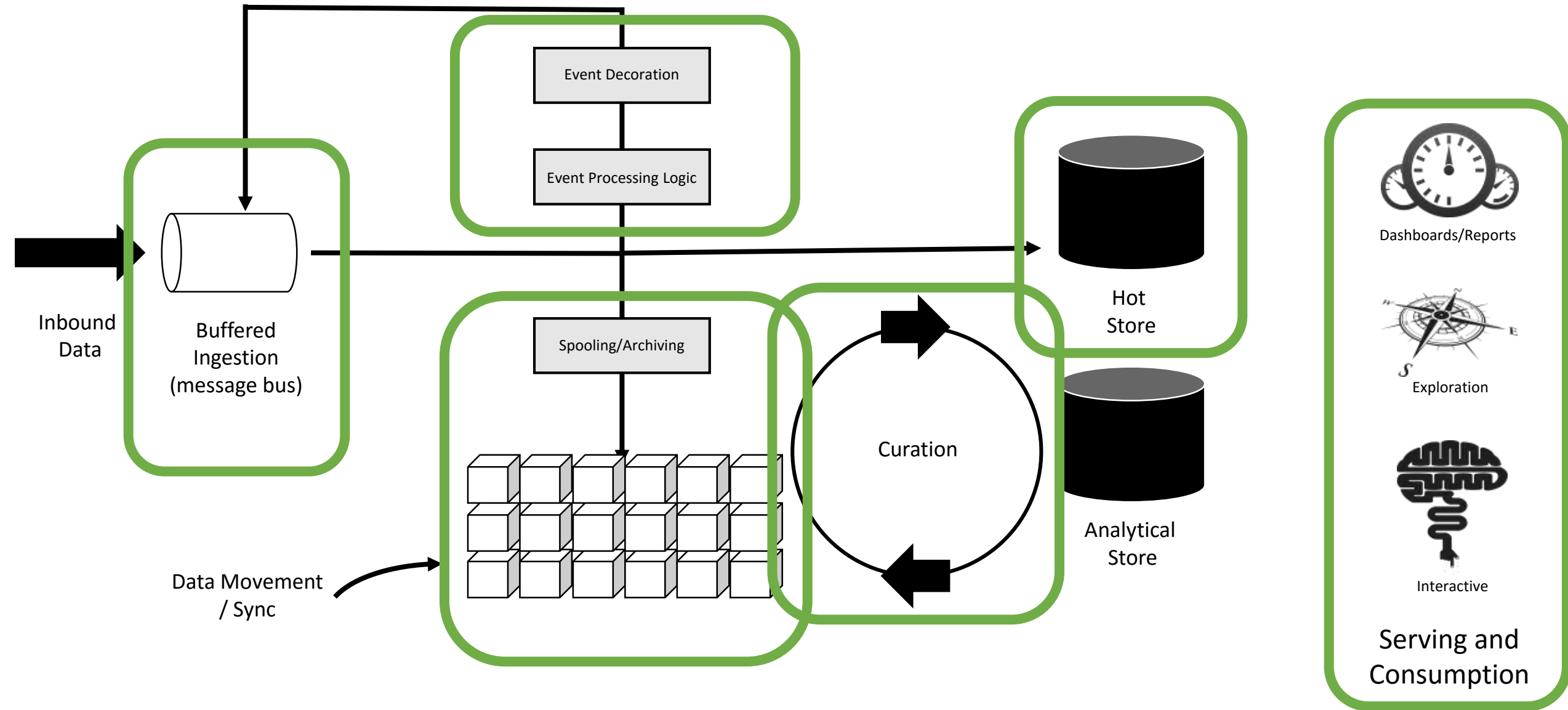
The Old and the New Data Processing



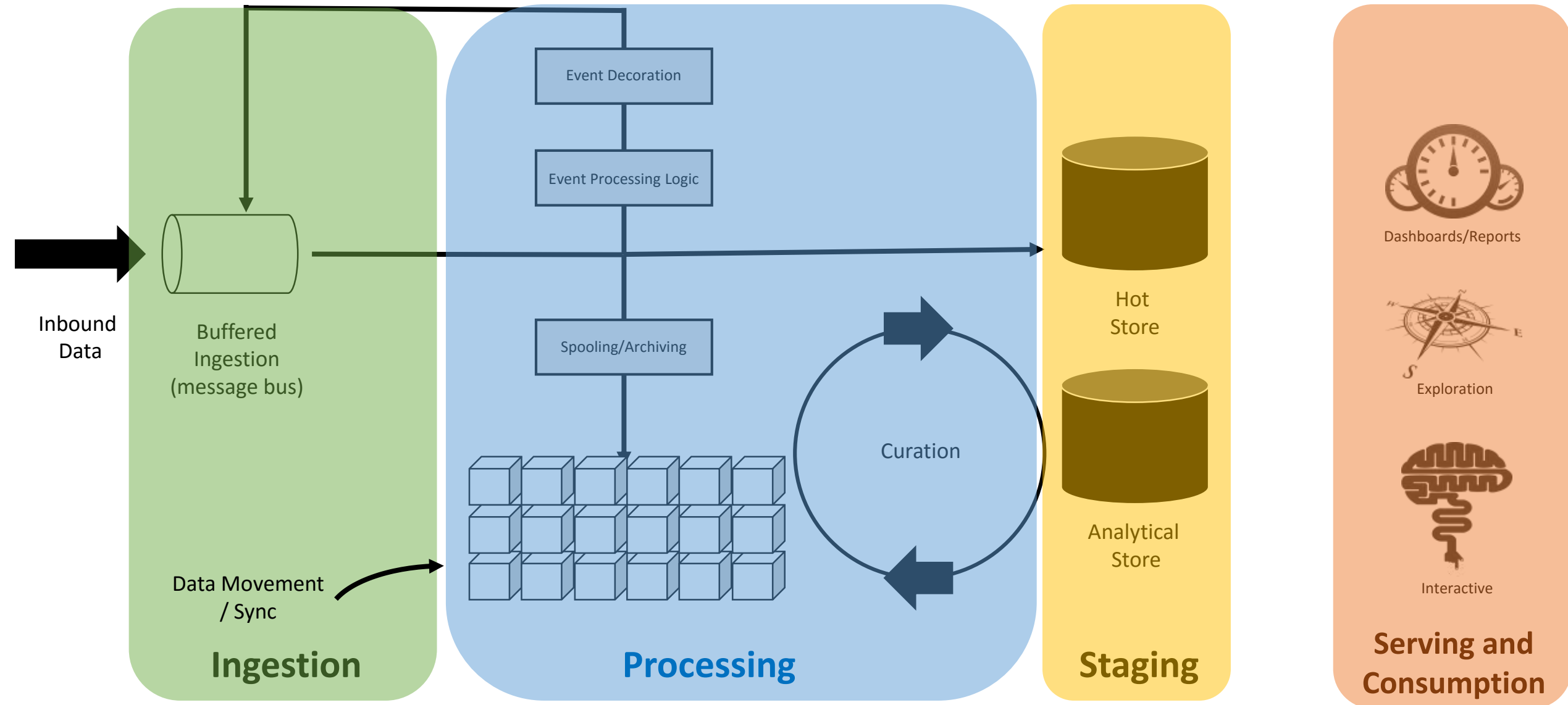
Lambda Architecture – High Level View



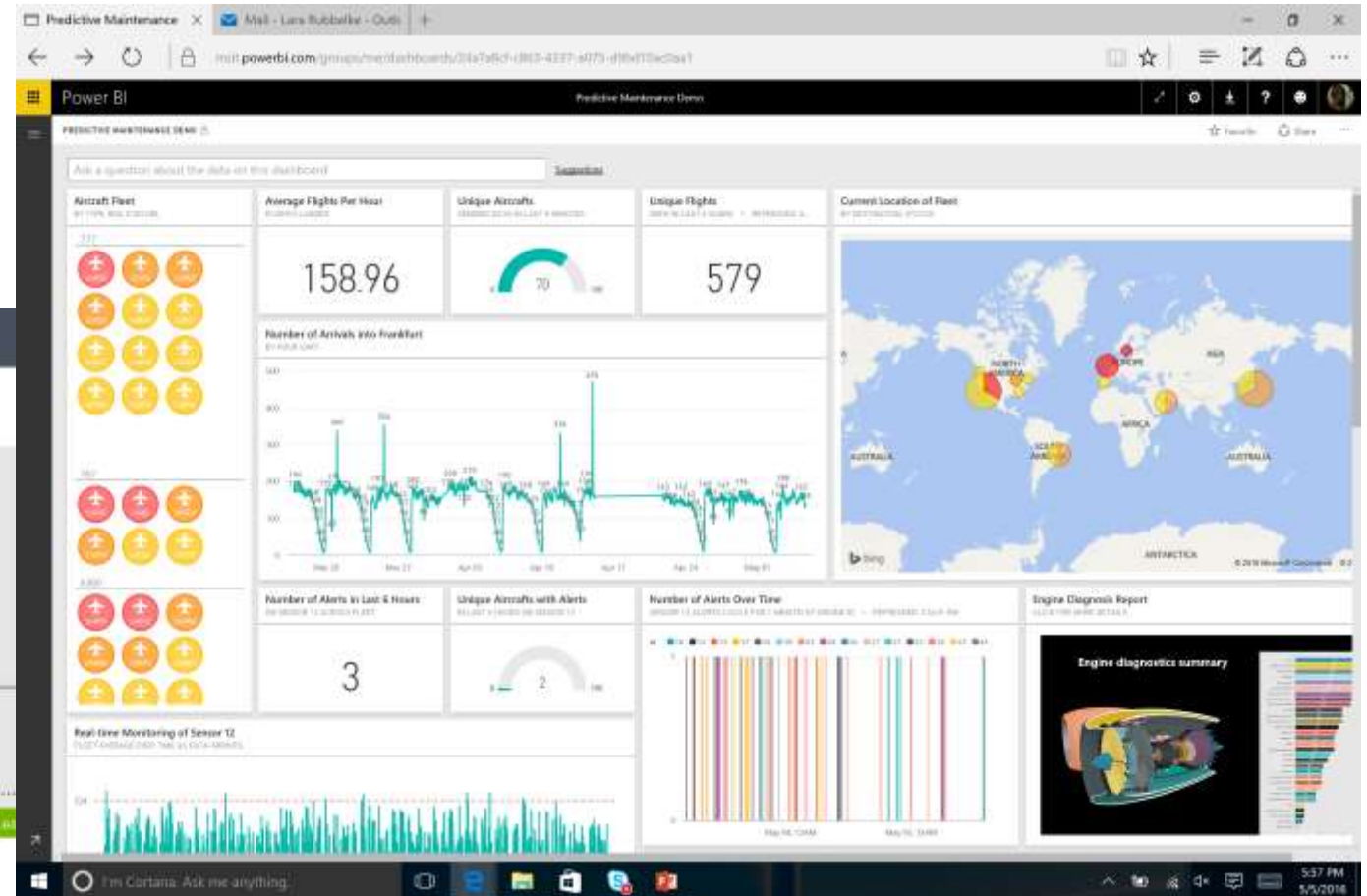
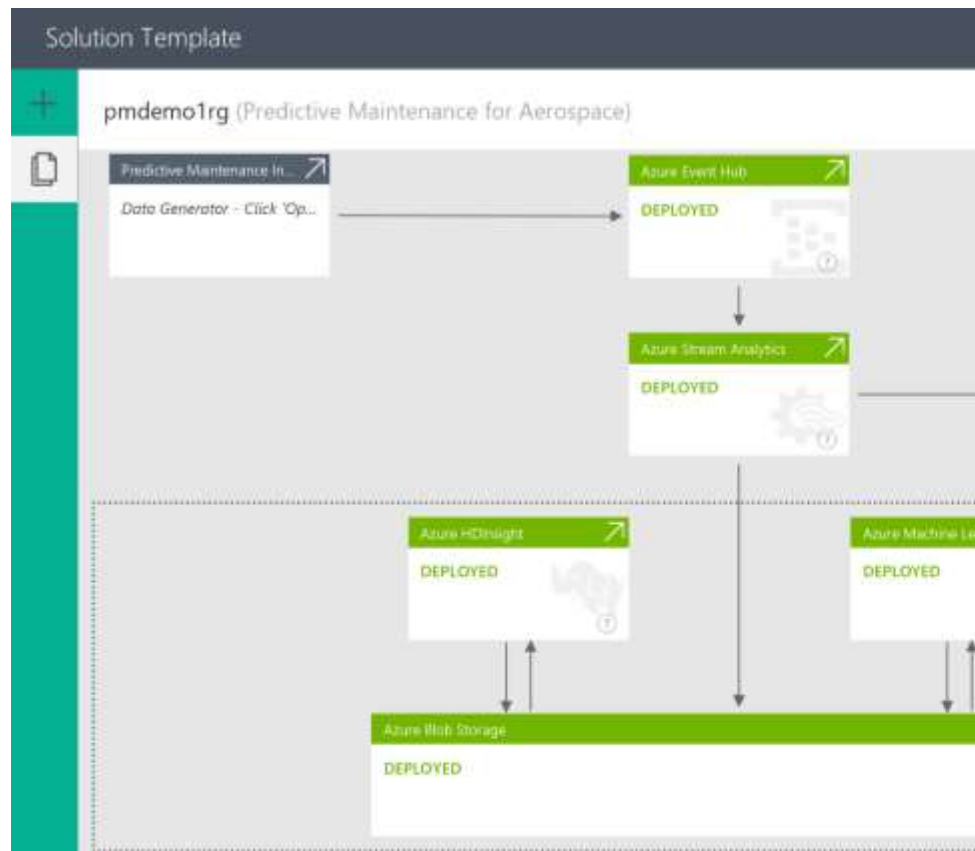
Lambda Architecture – Detailed View



Lambda Architecture – Detailed View



Cortana Intelligence Gallery Demo



airline industry. The green nodes on the left indicate the Azure Services that are deployed to your subscription as part of this solution. If you wish to remove this solution from your subscription, click on your solution name in the left panel in this UI and use the **Delete** option from the popup menu.

Ingestion



Processing



Staging



Serving




Enrichment and Curation

Modern Data Lifecycle

Ingestion

Event Hubs
IoT Hubs
Service Bus
Kafka




Processing

HDInsight
ADLA
Storm
Spark
Stream Analytics



Staging

ADLS
Azure Storage
Azure SQL DB



Serving

ADLS
Azure DW
Azure SQL DB
Hbase
Cassandra
Azure Storage
Power BI



Enrichment and Curation

Azure Data Factory

Azure ML

Modern Data Lifecycle

Ingestion

Event Hubs
IoT Hubs
Service Bus
Kafka



Processing



Staging



Serving

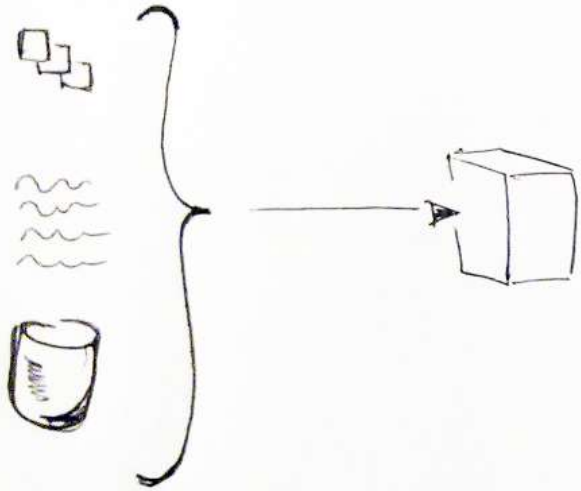


Enrichment and Curation

Modern Data Lifecycle

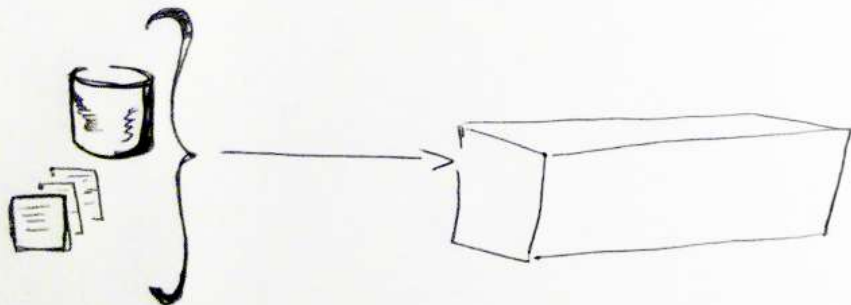


Ingestion



*Data pushed to a
broker for further
processing*

Typically indicative of streaming approaches.



*Data pushed to
storage for further
processing*

*Best alignment with many existing
systems.*

*Use scheduled jobs to synchronize or
move data.*

Typical “batch” processing.

Ingestion Options

- IoT Hub
 - Bi-Directional Communication
- Event Hub
 - Device to cloud mass ingestion
- Service Bus
 - Complex filters and processing rules
- Kafka
 - Common OSS integration
- RabbitMQ
 - More common OSS integration, run in IaaS

IoT Hub

Azure IoT Suite: IoT Hub

Designed for IoT

Connect millions of devices to a partitioned application back-end

Service assisted communications

Devices are not servers

Use IoT Hub to enable secure bi-directional comms

Cloud-scale messaging

Device-to-cloud and Cloud-to-device

Durable messages (*at least once* semantics)

Cloud-facing feedback

Delivery receipts, expired messages

Device communication errors

Per-device authentication

Individual device identities and credentials

Connection multiplexing

Single device-cloud connection for all communications (C2D, D2C)

Multi-protocol

Natively supports AMQP, HTTP

Designed for extensibility to custom protocols

Multi-platform

Device SDKs available for multiple platforms (e.g. RTOS, Linux, Windows)

Multi-platform Service SDK.

Setup

- Retention Period
- Event Hub Endpoint
- C2D settings

The screenshot displays the Azure IoT Hub 'Messaging' settings for a device named 'myloTHub'. The interface is divided into two main sections: 'Cloud-to-device settings' and 'Device-to-cloud settings'. In the 'Device-to-cloud settings' section, which is highlighted with a red box, the 'Partitions' are set to 4, the 'Event Hub-compatible name' is 'myloTHub', and the 'Event Hub-compatible endpoint' is 'sb://iothub-ns-myiothub-35-c12891486d'. The 'Cloud-to-device settings' section shows the 'Default cloud-to-device TTL' and 'Feedback retention time' both set to 1 hour. The left sidebar shows the 'Messaging' tab selected, with other options like 'Shared access policies', 'Pricing and scale', and 'Users' visible.

Settings
myloTHub

Search settings

- Shared access policies
- Messaging**
- Pricing and scale
- Users

Messaging
myloTHub

Save Discard

Cloud-to-device settings

Default cloud-to-device TTL ⓘ

1 hr

Feedback retention time ⓘ

1 hr

Device-to-cloud settings

Partitions ⓘ

4

Event Hub-compatible name ⓘ

myloTHub

Event Hub-compatible endpoint ⓘ

sb://iothub-ns-myiothub-35-c12891486d

Retention time ⓘ

1 day

Consumer groups ⓘ

\$Default

testgroup

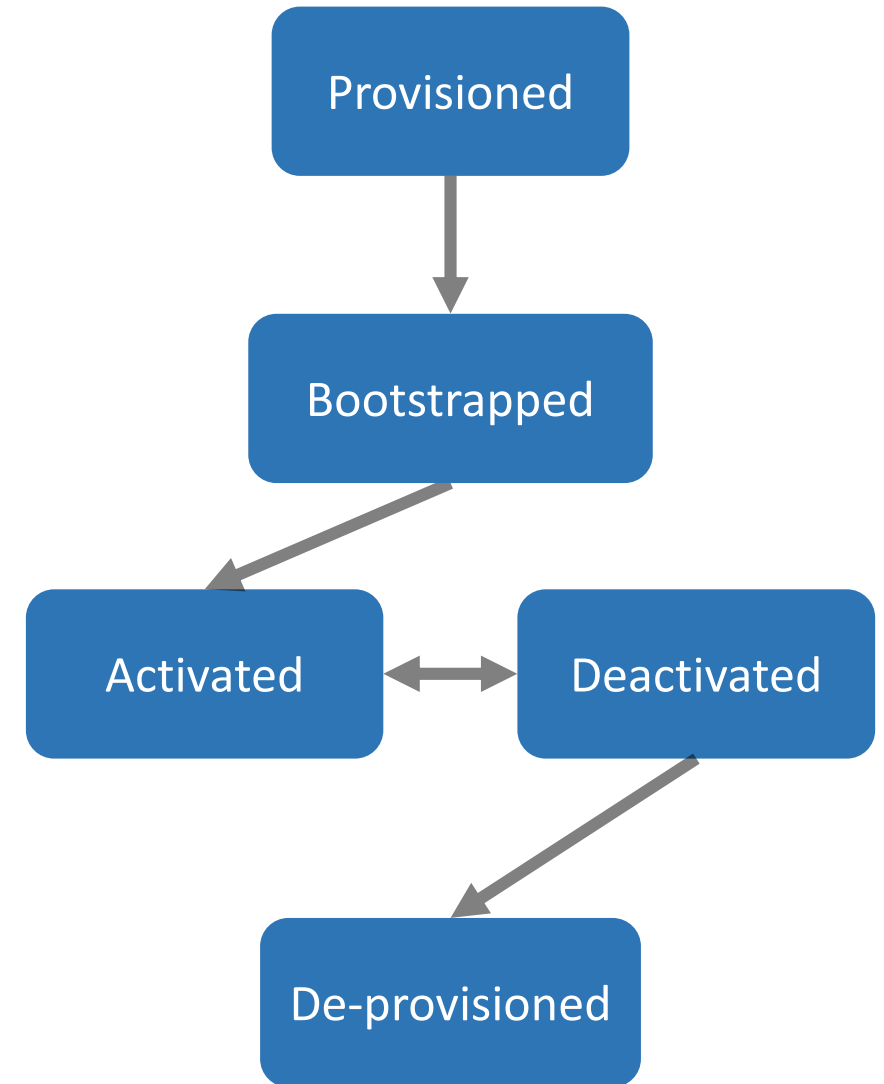
Device provisioning

Making devices known to your system

- Many systems involved (IoT Hub, device registry, ERPs, ...)
- Device identity (composite devices, many concerns)

Sample provisioning

1. Device **provisioned** at manufacturing into system
2. Device connects for the first time and gets associated to its regional data center (**bootstrapped**)
3. As a result of customer interactions the device is **activated**
4. Devices can be **deactivated** for security and other reasons
5. A device can also be **de-provisioned** at end-of-life or decommission.



IoT Hub

Opening up the channels

Ingestion



Processing

Stream Analytics
Storm
Spark Streaming



Staging



Serving

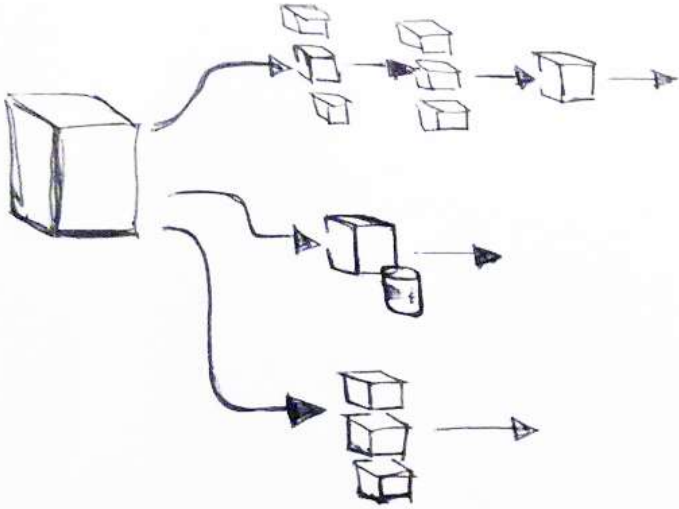


Enrichment and Curation

Modern Data Lifecycle



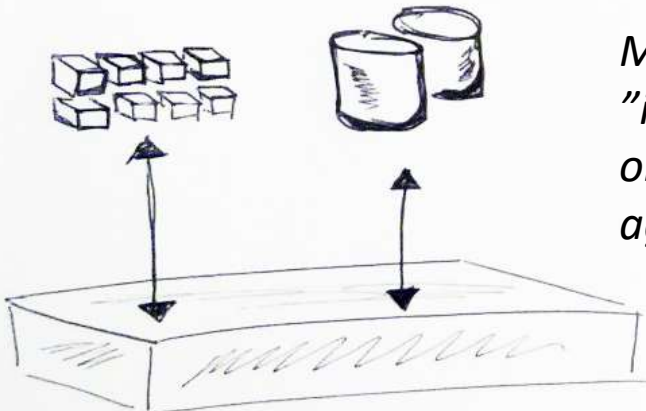
Processing



*Multiple processing
"instances" can work
off of the broker.*

Must think about the end use:

- *Indexing for operational dashboards*
- *Transformation/Curation for persistence*
- *Spooling to an alternate store*



*Multiple processing
"instances" can work
on top of dis-
aggregated storage*

Lot's of options abound depending on the need:

- *Relational engines*
- *Big Data solutions / in-mem or not*
- *Other*



Bounded vs. Unbounded Processing

Stream Options

- Azure Stream Analytics
- Spark Streaming
- Storm

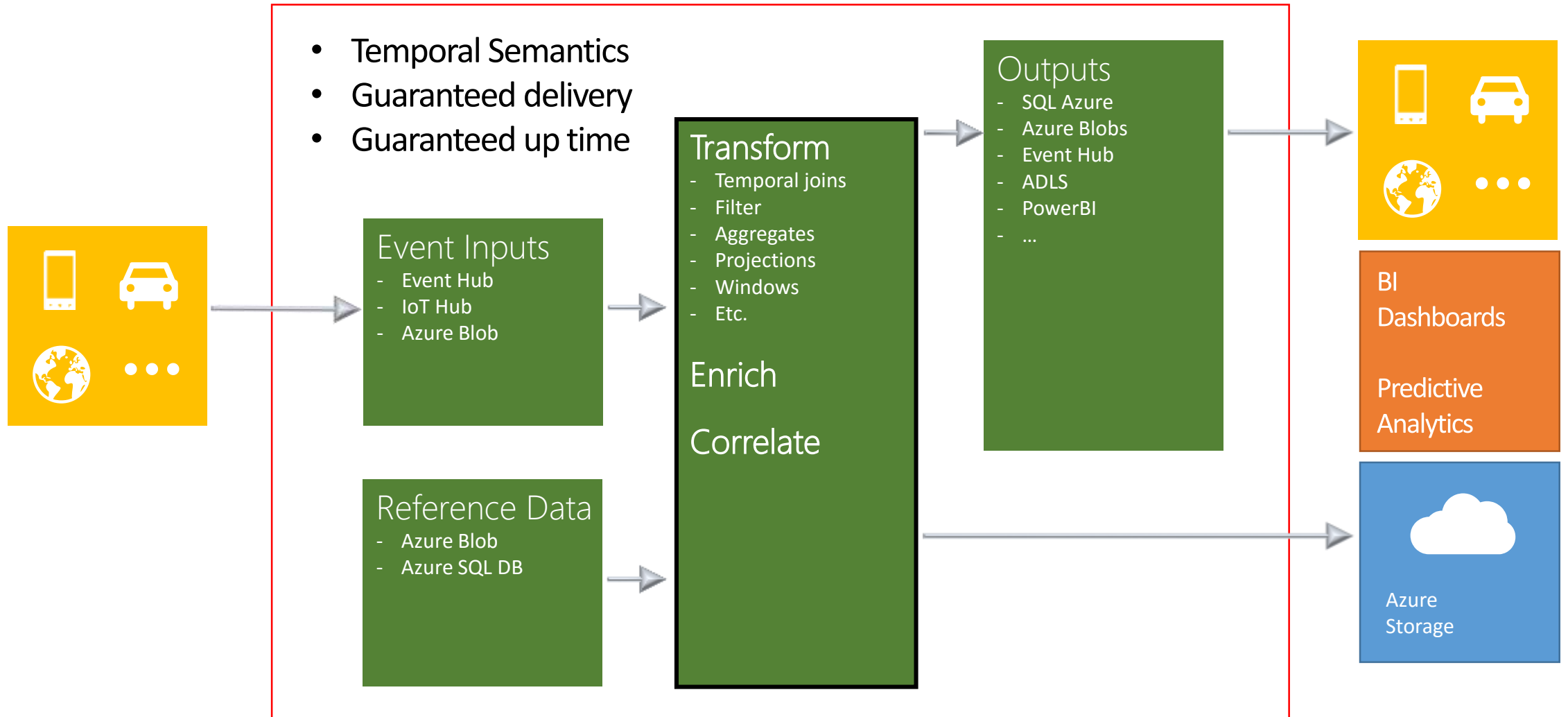
Azure Stream Analytics

What is Azure Stream Analytics?

- Cost effective event processing engine
- SQL-like syntax
- Naturally integrated with Azure IoT Hub and Event Hubs

Azure Stream Analytics

End-to-End Architecture Overview

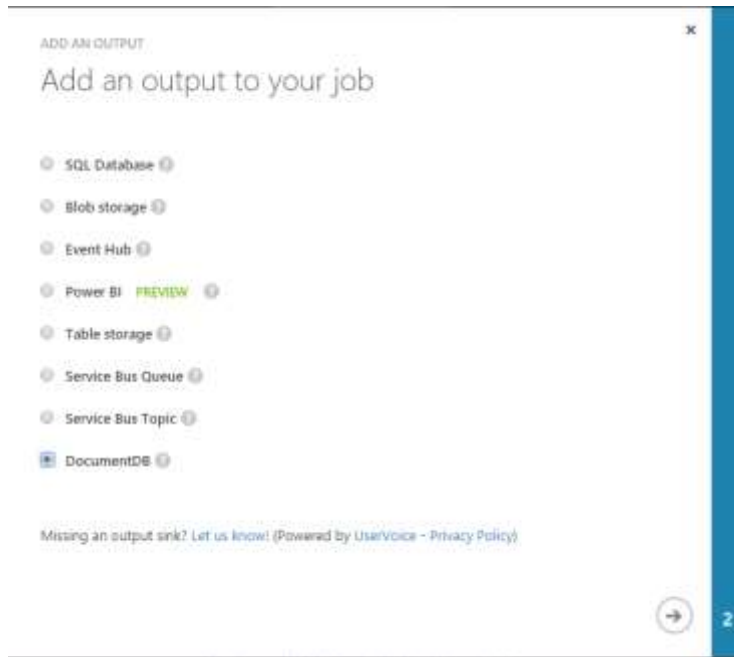


Inputs sources for a Stream Analytics Job

The image shows two overlapping screenshots from the Azure Stream Analytics portal. The top screenshot, titled 'ADD AN INPUT', instructs the user to 'Add an input to your job' and notes that 'Each job requires at least one data stream input. Adding reference data input is optional.' It lists two options: 'Data stream' (selected with a radio button) and 'Reference data'. The 'Data stream' option is described as 'A continuous sequence of data or events to be consumed and transformed by a Stream Analytics job.' The 'Reference data' option is described as 'Auxiliary data used for correlation and lookups. Reference data is static or slow-changing.' The bottom screenshot, titled 'ADD A DATA STREAM', instructs the user to 'Add a data stream to your job'. It lists three options: 'IoT Hub' (marked with a green 'PREVIEW' label and a question mark), 'Event Hub' (with a question mark), and 'Blob storage' (with a question mark). A callout box points to the 'IoT Hub' option, stating: 'IOT HUB Choose IoT Hub for real-time workflows at high scale with low latency and high reliability.'

- Currently supported input Data Streams are **Azure Event Hub**, **Azure IoT Hub** and **Azure Blob Storage**. Multiple input Data Streams are supported.
- Advanced options lets you configure how the Job will read data from the input blob (which folders to read from, when a blob is ready to be read, etc).
- Reference data is usually static or changes very slowly over time.
 - Must be stored in **Azure Blob Storage**.
 - Cached for performance

Output for Stream Analytics Jobs



Currently data stores supported as outputs

Azure Blob storage: creates log files with temporal query results
Ideal for archiving

Azure Table storage:
More structured than blob storage, easier to setup than SQL database and durable (in contrast to event hub)

SQL database: Stores results in Azure SQL Database table
Ideal as source for traditional reporting and analysis

Event hub: Sends an event to an event hub
Ideal to generate actionable events such as alerts or notifications

Service Bus Queue: sends an event on a queue
Ideal for sending events sequentially

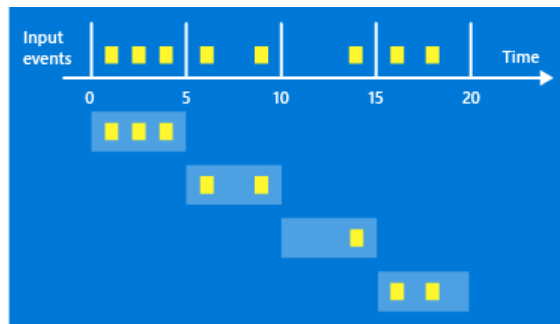
Service Bus Topics: sends an event to subscribers
Ideal for sending events to many consumers

PowerBI.com:
Ideal for near real time reporting!

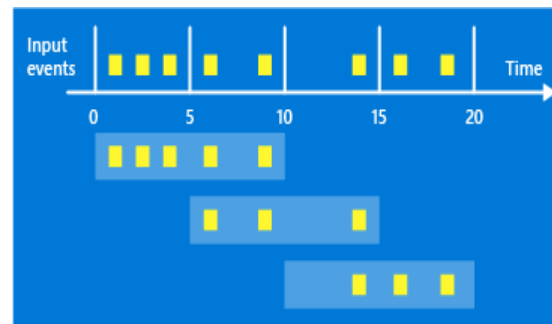
DocumentDb:
Ideal if you work with json and object graphs

ASA: Three Types of Windows

- Every window operation outputs events at the end of the window
 - The output of the window will be single event based on the aggregate function used. The event will have the time stamp of the window
- All windows have a fixed length



Tumbling window
Aggregate per time interval



Hopping window
Schedule overlapping windows



Sliding window
Windows constant re-evaluated

Multiple Steps, Multiple Outputs

- A query can have multiple steps to enable pipeline execution
 - A step is a sub-query defined using WITH (“common table expression”)
- The only query outside of the WITH keyword is also counted as a step
- Can be used to develop complex queries more elegantly by creating a intermediary named result
 - Each step’s output can be sent to multiple output targets using INTO

```
WITH Step1 AS (  
    SELECT Count(*) AS CountTweets, Topic  
    FROM TwitterStream PARTITION BY  
        PartitionId  
    GROUP BY TumblingWindow(second, 3),  
        Topic, PartitionId  
)  
Step2 AS (  
    SELECT Avg(CountTweets)  
    FROM Step1  
    GROUP BY TumblingWindow(minute, 3)  
)  
SELECT * INTO Output1 FROM Step1  
SELECT * INTO Output2 FROM Step2  
SELECT * INTO Output3 FROM Step2
```


Azure Streaming Analytics

where the smarts happen

Ingestion



Processing



Staging



Serving



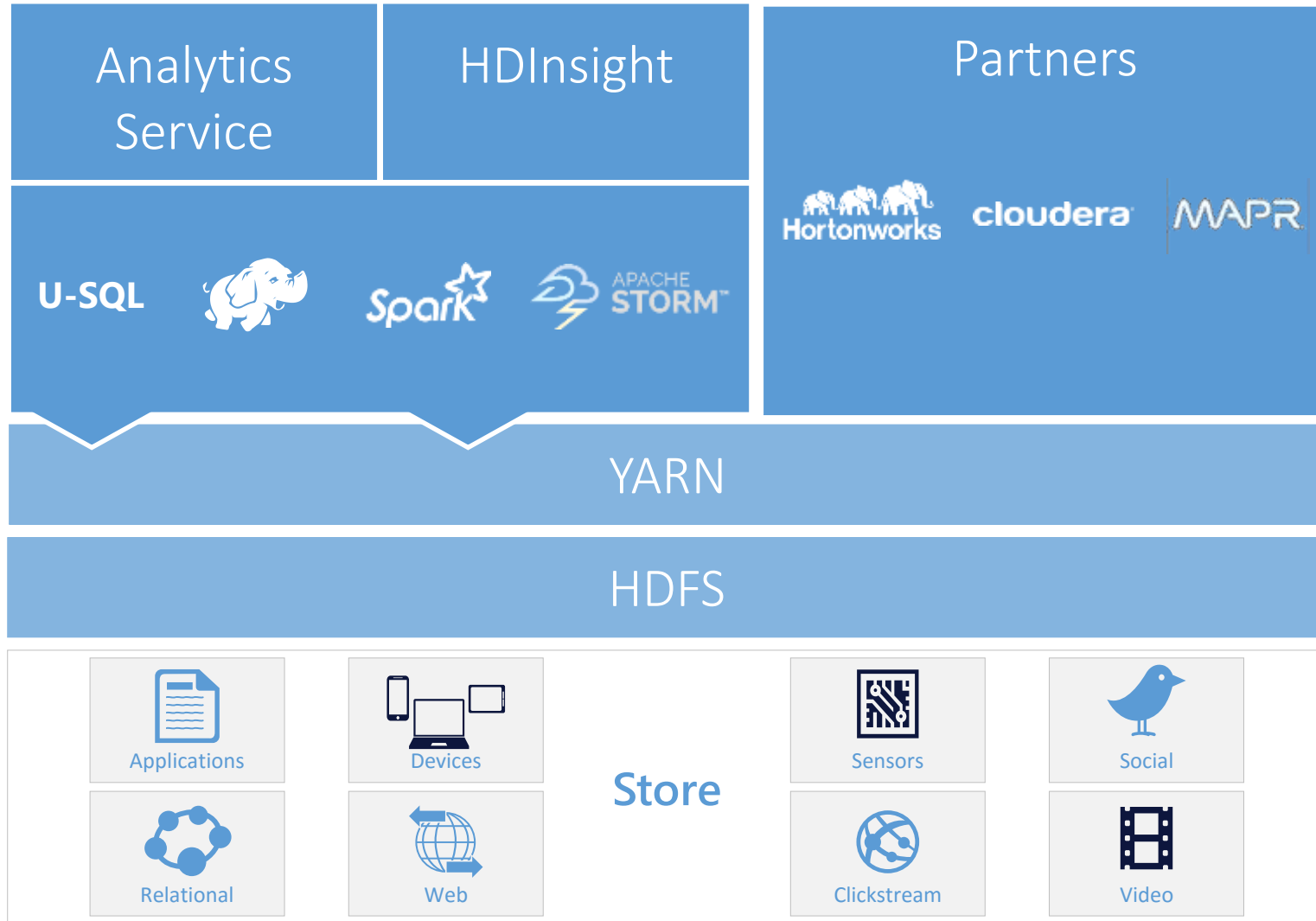
Enrichment and Curation

Modern Data Lifecycle

Batch Processing

- Azure Data Lake
- HDInsight
- Spark

Azure Data Lake



- Integrated analytics and storage
- Fully managed
- Easy to use—“dial for scale”
- Proven at scale
- Analyze data of any size, shape or speed
- Open-standards based

HDInsight

• Patterns/What Works

- Batch processing
- Map...and reduce
- Lots of aggregation
- Multiple schemas on same data
- Fast

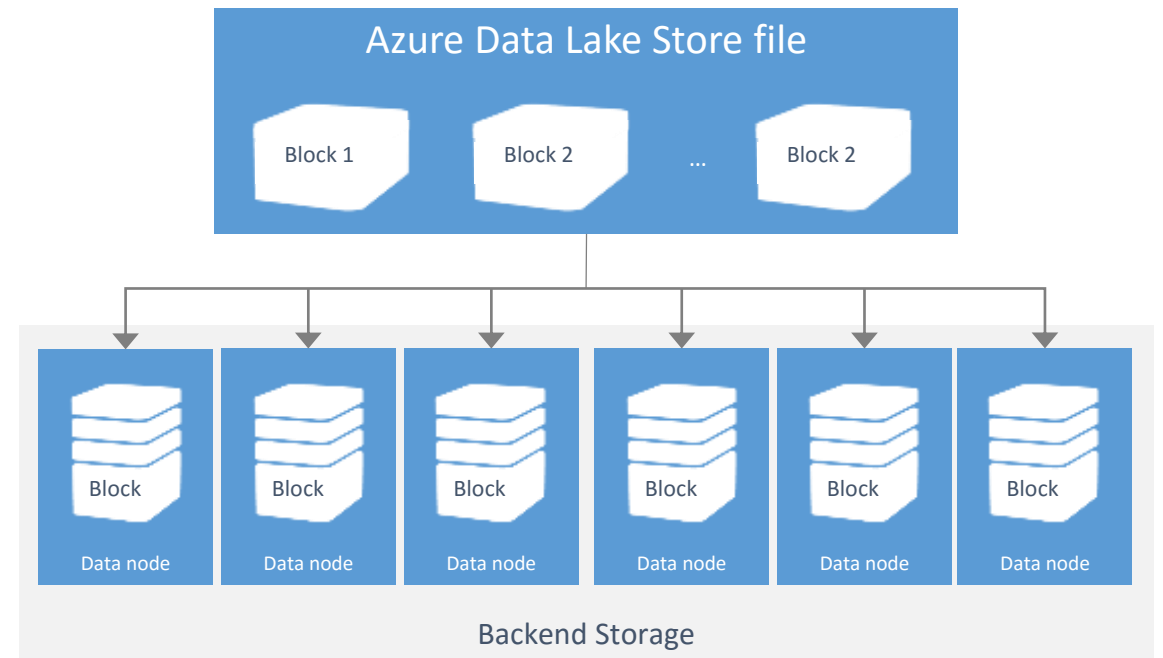
• Anti-Pattern/Danger

Anything that requires:

- Joins
- Complex transactional needs
- Granular security requirements
- Not a relational database replacement
- Not fast

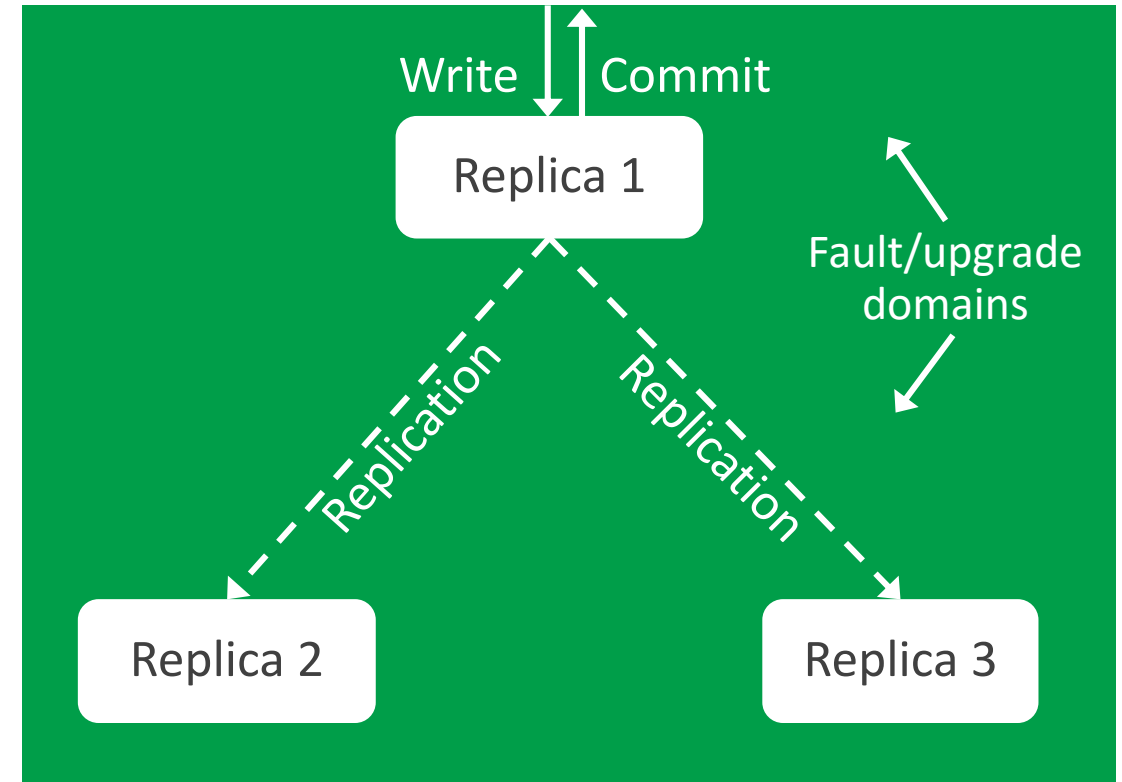
ADL Store Unlimited Scale

- Optimized for analytics and IoT systems.
- Each file in ADL Store is sliced into blocks, 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.



ADL Store: High 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

ADL Store: Enterprise Grade Security

- Auditing, alerting, access control - all from within a single web-based portal
- Azure Active Directory integration for identity and access management

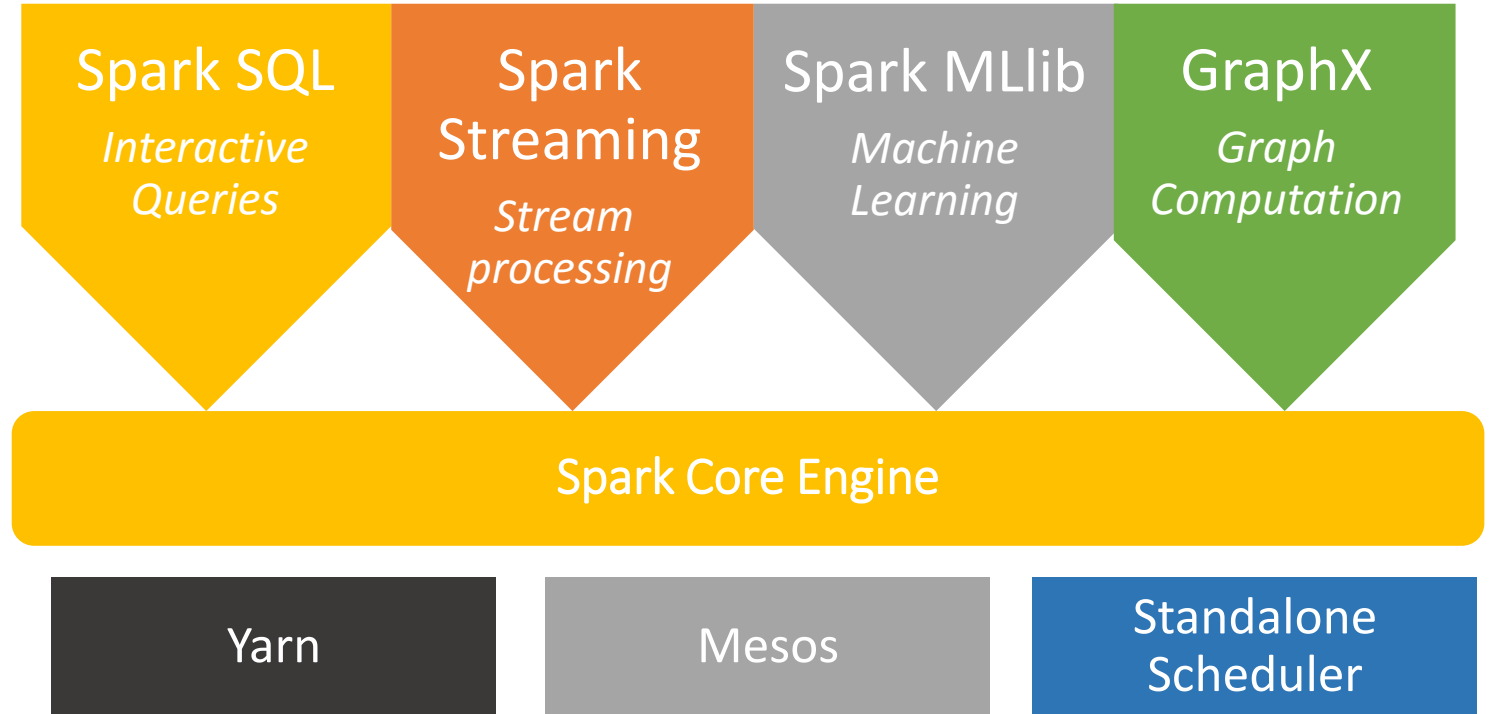


Apache Spark – An Unified Framework

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

Spark Unifies:

- Batch Processing
- Real-time processing
- Stream Analytics
- Machine Learning
- Interactive SQL



[Intro to Apache Spark \(Brain-Friendly Tutorial\): https://www.youtube.com/watch?v=rvDpBTV89AM](https://www.youtube.com/watch?v=rvDpBTV89AM)

What makes Spark fast?

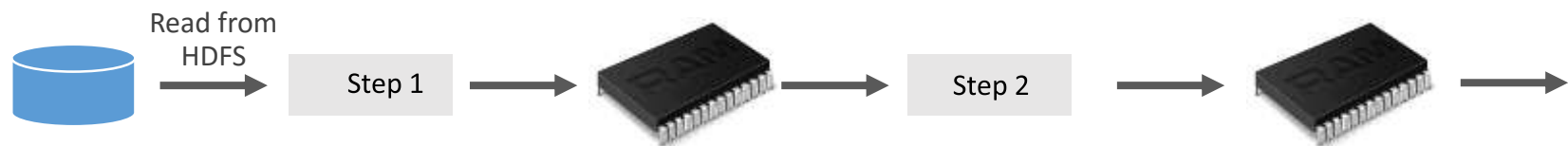
- Provides primitives for *in-memory* cluster computing. A Spark job can *load and cache* data into memory and query it repeatedly (iteratively) much quicker than disk-based systems.
- Spark integrates into the [Scala](#) programming language, letting you manipulate distributed datasets like local collections. No need to structure everything as map and reduce operations
- *Data-sharing* between operations is faster as data is in-memory:
 - In Hadoop data is shared through HDFS which is expensive. HDFS maintains three replicas.
 - Spark stores data in-memory *without any replication*.

Data Sharing between Steps of a Job

In Traditional MapReduce



In Spark



Spark (Preview) on Azure HDInsight

- Fully Managed Service
 - 100% open source Apache Spark and Hadoop bits
 - Latest releases of Spark
 - Fully supported by Microsoft and Hortonworks
 - 99.9% Azure Cloud SLA
- Coming Soon: Advanced Enterprise Features
 - Integration with Azure Data Lake Store
 - Role based security and audit
 - Encryption at rest and in transit
 - Certifications: PCI in addition to existing ISO 27018, SOC, HIPAA, EU-MC

Optimized for Data Scientist Productivity

- On-demand compute
 - Dynamically scale cluster to 1000s of cores to compress time of the ML job
 - Coming Soon: Auto-scale during job execution or time-based
- Tools for experimentation and development
 - Jupyter Notebooks (scala, python, automatic data visualizations)
 - IntelliJ plugin (integrated job submission, remote debugging)
 - ODBC connector for Power BI, Tableau, Excel, etcon Spark
 - ML algorithms in R parallelized using Spark
 - R Studio

Basic building blocks

Resilient Distributed Datasets (RDDs)

Lowest level set of object representing data, can be stored in memory or disk across a cluster.

DataFrame

Higher level abstraction API.

A distributed collection of rows organized into named columns.

RDD with schema and optimizations

DataSet APIs

Extension of Spark's DataFrame API that supports static typing and user functions that run directly on existing JVM types (such as user classes).

Compile time type safety with optimizations.

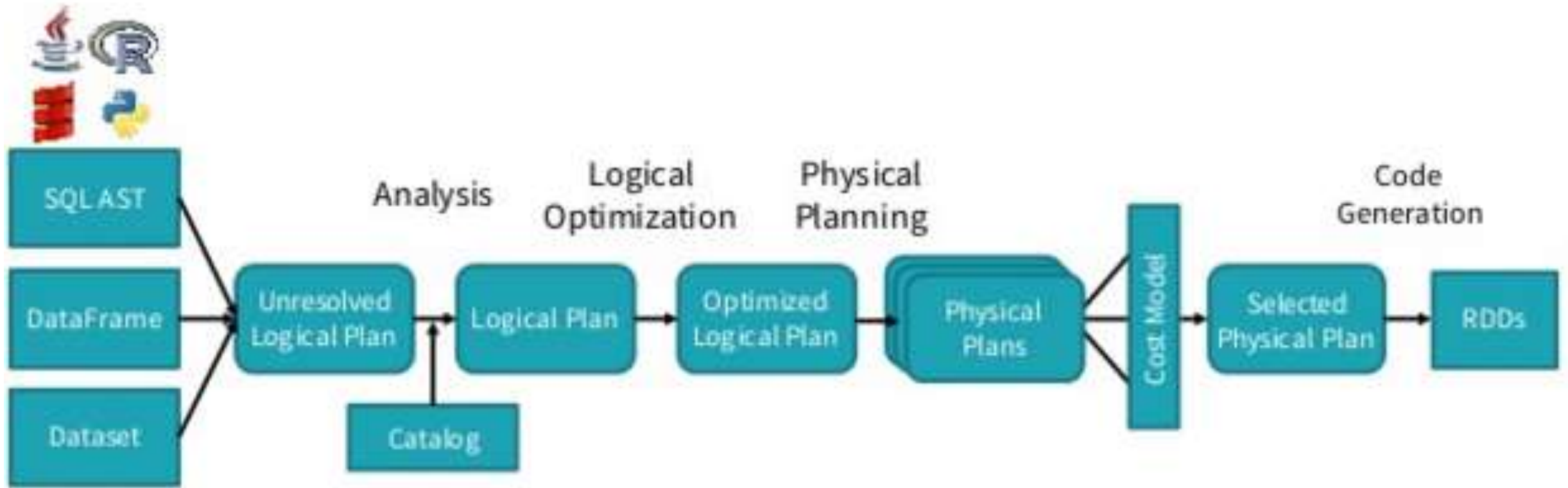
“Preview”.

Spark 2.0 will unify these APIs

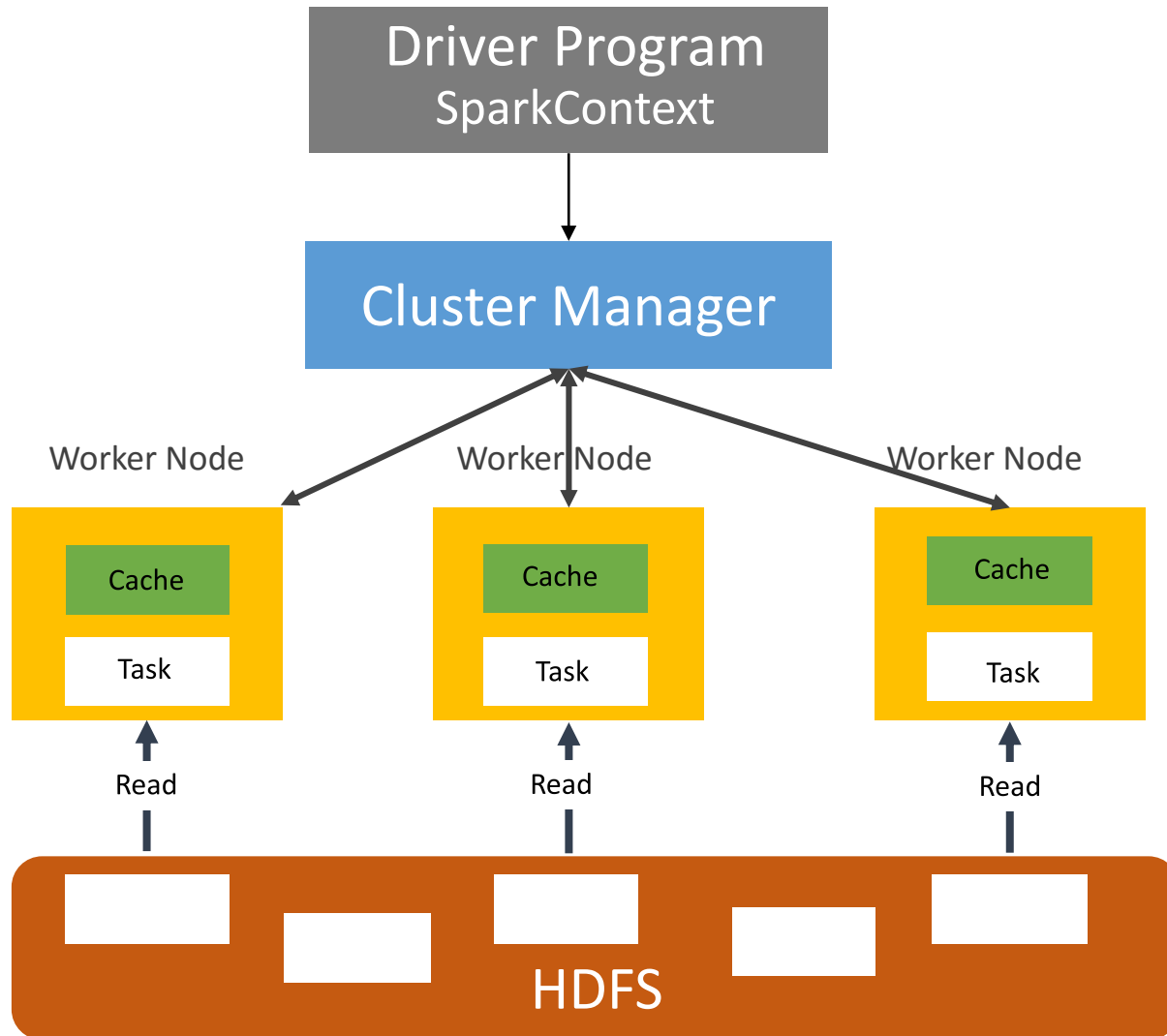
Structuring Spark: DataFrames, Datasets, and Streaming: <https://www.youtube.com/watch?v=i7l3JQRx7Qw&feature=youtu.be>

RDDs vs DataFrames vs DataSets

Which one to use?



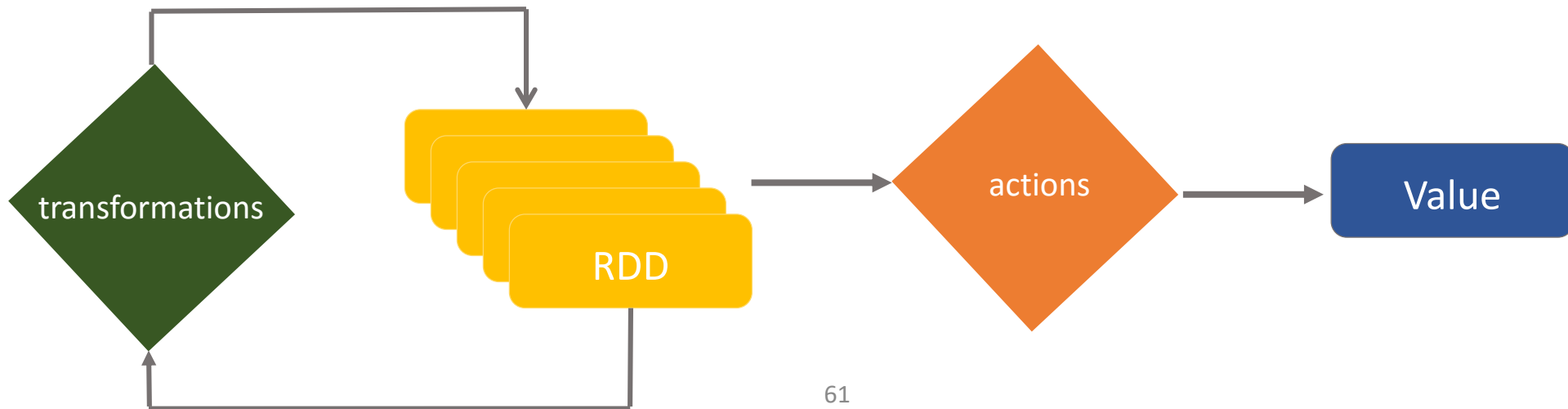
Spark Cluster Architecture



- 'Driver' runs the user's 'main' function and executes the various parallel operations on the worker nodes.
- The results of the operations are collected by the driver
- The worker nodes read and write data from/to HDFS.
- Worker node also cache transformed data in memory as RDDs

RDDs: Transformations and Actions

- RDDs support two types of operations: Transformation and Actions
- Transformations create a new dataset from an existing dataset
 - All transformations are lazy: they do not compute their results right away.
 - The transformations are only computed when an action requires a result to be returned to the driver program. Obviously *does not apply to persistent RDDs*.
 - map, filter, sample, union, ...
- Actions return a value to the driver program after running a computation on the dataset.
 - Example: reduce, collect, count, first, foreach, etc.

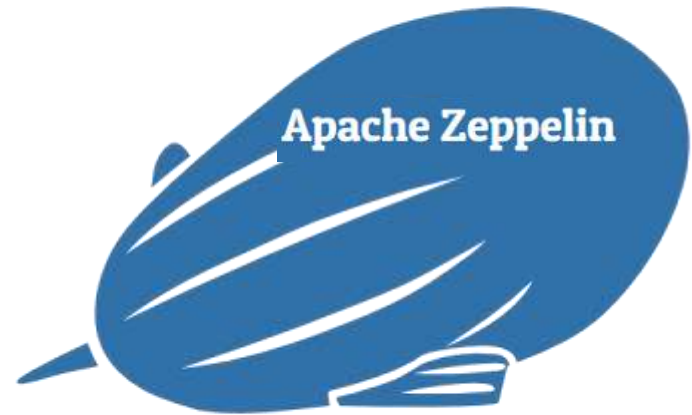


Developing Spark Apps with Notebooks

Jupyter and Zeppelin are two Notebooks that work with Apache Spark

Notebooks:

- Are web-based, interactive servers for REPL (Read-Evalute-Print-Loop) style programming.
- Are well-suited for prototyping, rapid development, exploration, discovery and iterative development
- Typically consist of code, data, visualization, comments and notes
- Enable collaboration with team members



Jupyter

- Names inspired by scientific and statistical languages (Julia, Python and R)
- Is based on, and an evolution of IPython [Shortly IPython will not ship separately]
- Is language agnostic. All languages are equal class citizens. Languages are supported through 'kernels' (a program that runs and introspects the users code)
- Supports a rich REPL protocol
- Includes:
 - Jupyter notebook document format (.ipynb)
 - [Jupyter Interactive web-based Notebook](#)
 - [Jupyter Qt console](#)
 - [Jupyter Terminal console](#)
 - [Notebook viewer \(nbviewer\)](#)

Supported languages (Kernels)



See [full list here](#)

Integration with BI Reporting Tools

HDInsight Spark integrates with these BI tools to report on Spark data



Ingestion



Processing



Staging



Serving

ADLS
Azure DW
Azure SQL DB
Hbase
Cassandra
Azure Storage
Power BI

Enrichment and Curation

Azure Data Factory

Azure ML

Modern Data Lifecycle

Dashboards



Interactive

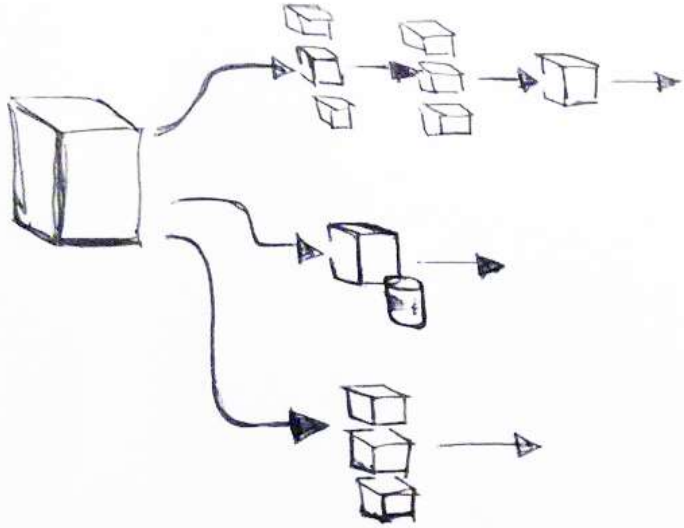


Exploration





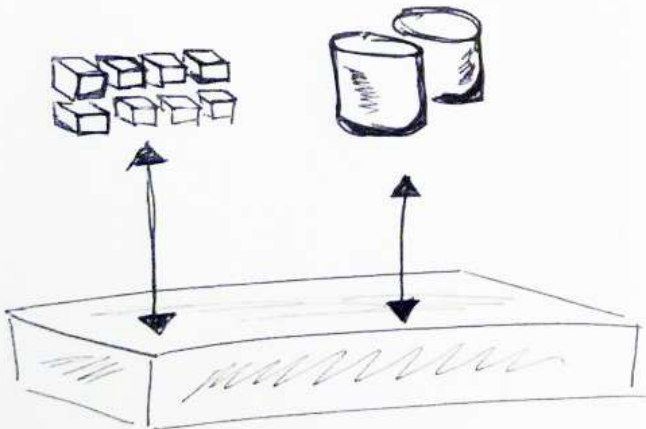
Serving



Typically serving here is constrained:

- *Constrained on particular access patterns*
- *Constrained on dashboard scenarios*

Serving here optimized for reducing “observation latency”



Typically serving here can be unconstrained

- *Still used for dashboards*
- *Used for data exploration and ML*
- *Used for interactive BI*
- *Often used for broad sharing*

Azure DW and Analytical Workloads

Store large volumes of data.

Consolidate disparate data into a single location.

Shape, model, transform and aggregate data.

Perform query analysis across large datasets.

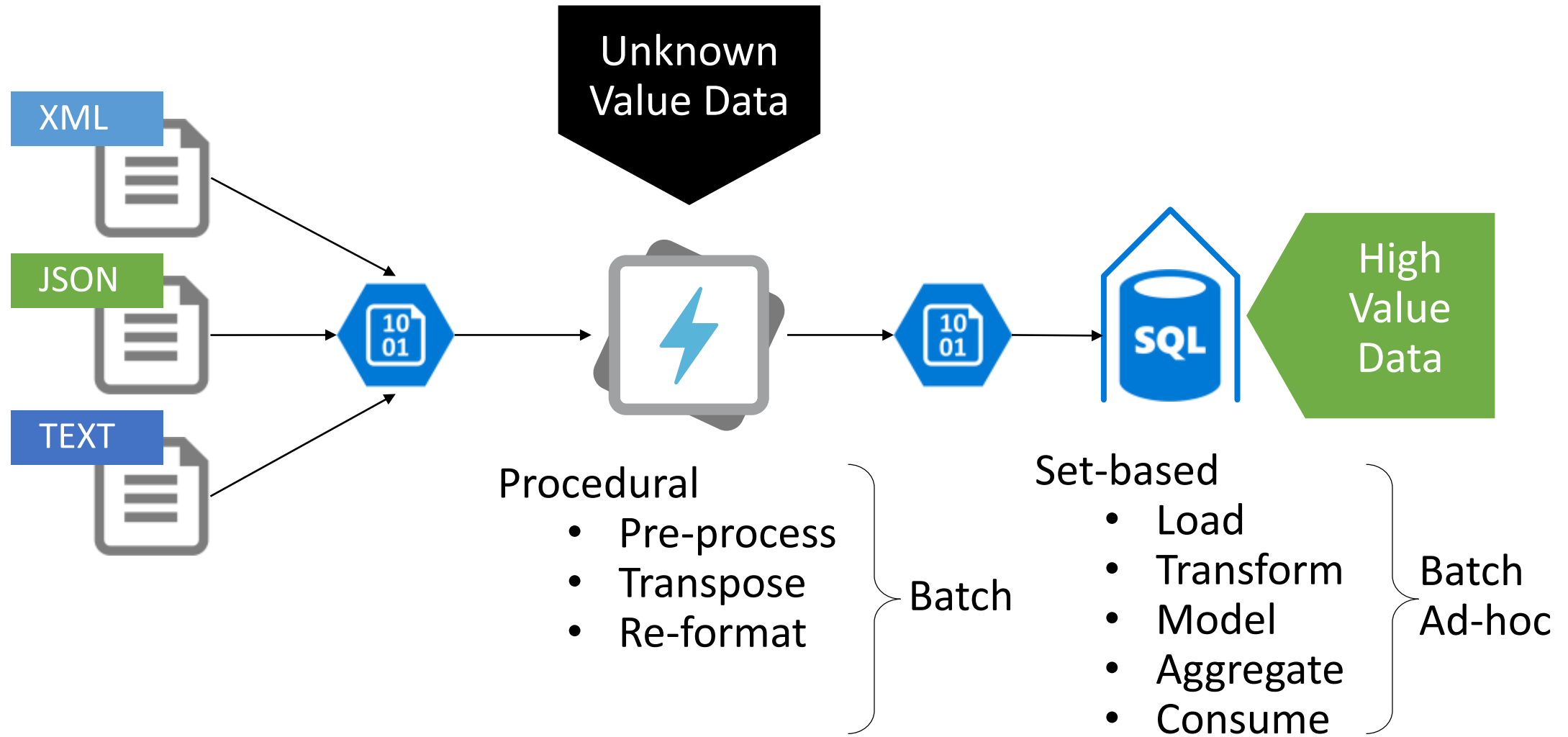
Ad-hoc reporting across large data volumes.

All using simple SQL constructs.

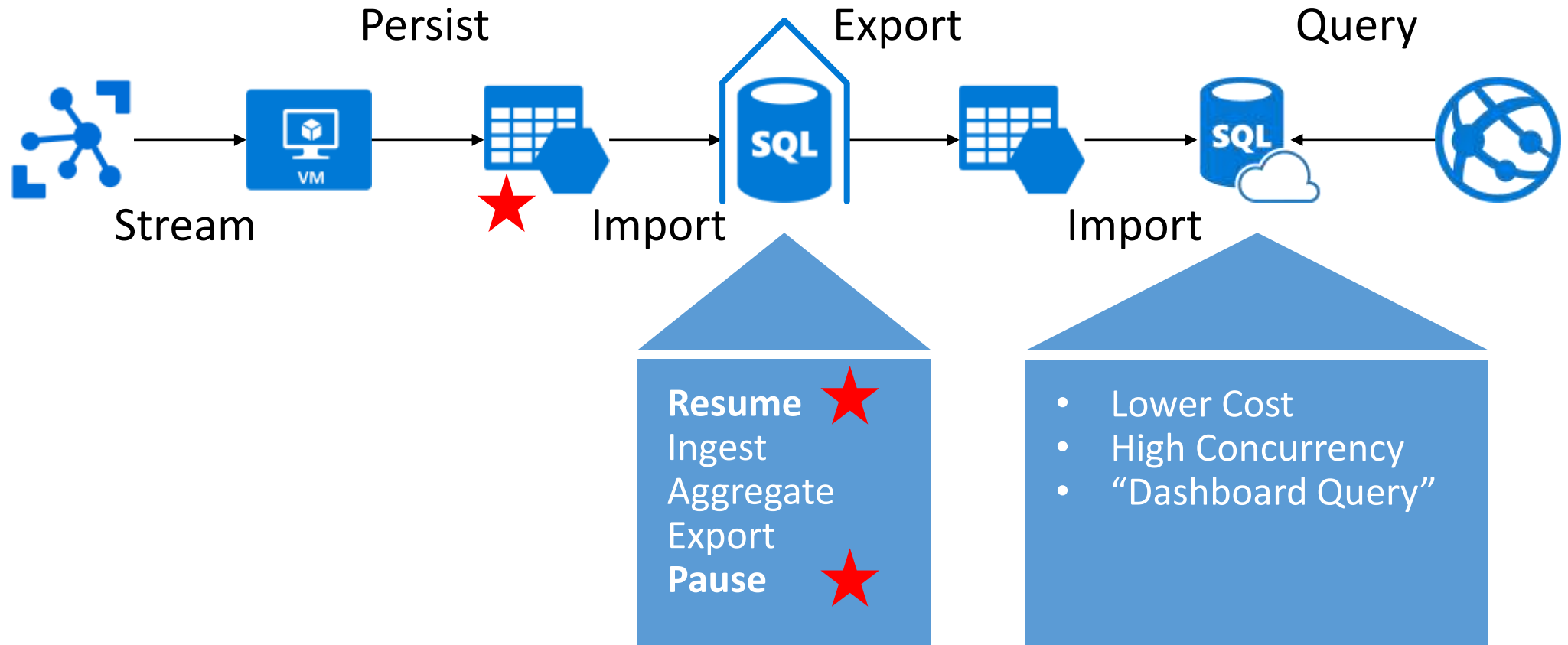


“SQL on SQL”

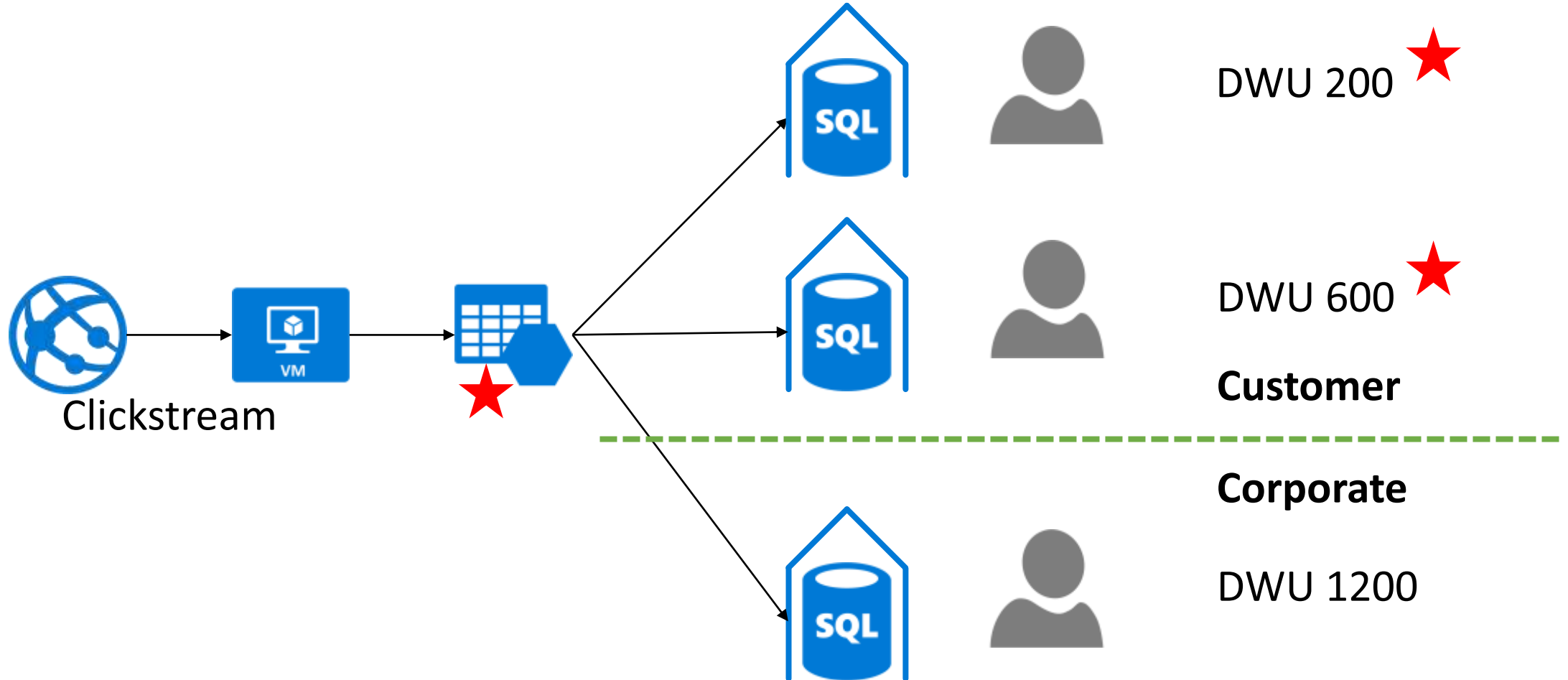
ADL and SQLDW



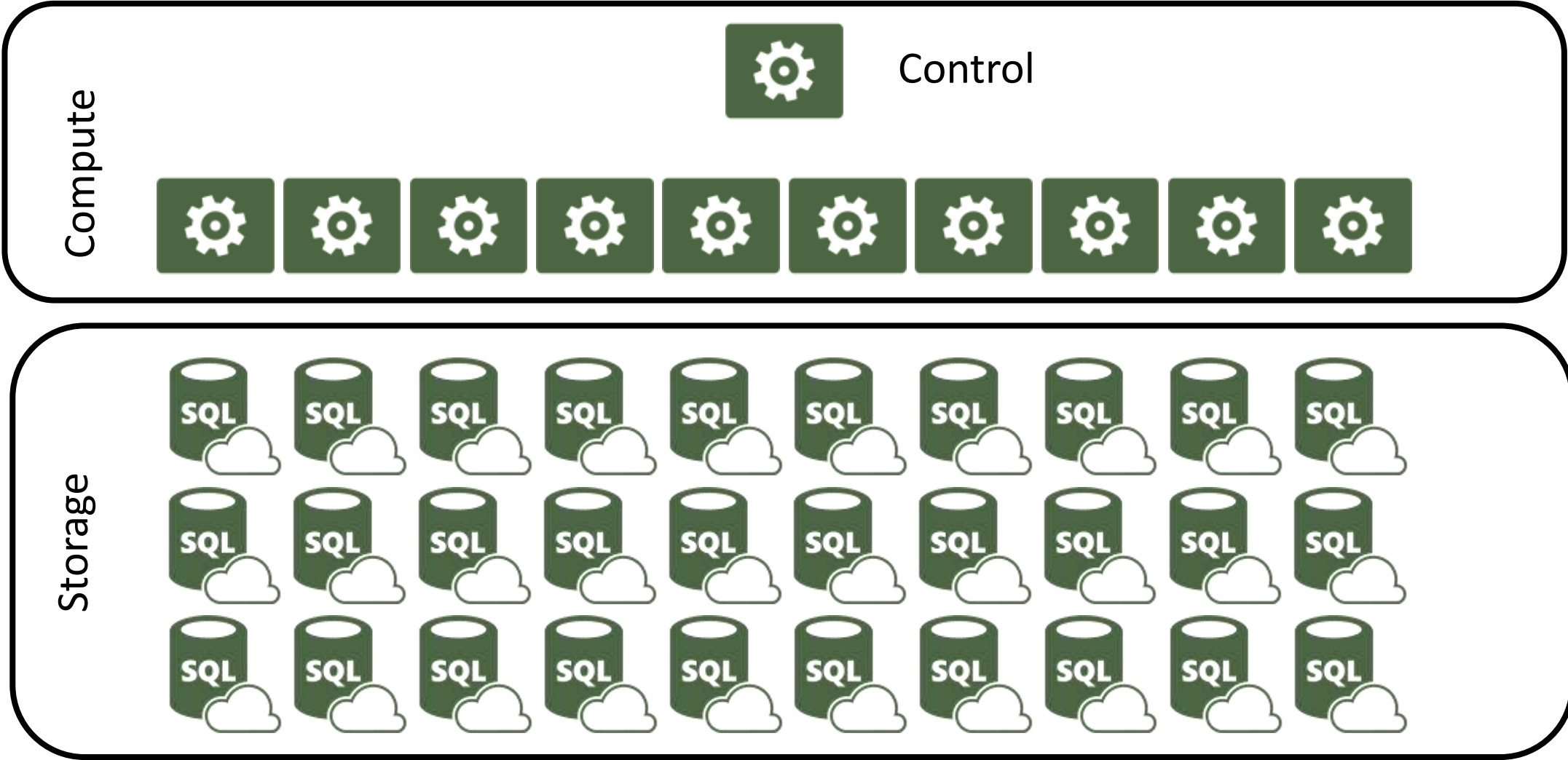
Pattern: Compute consumption



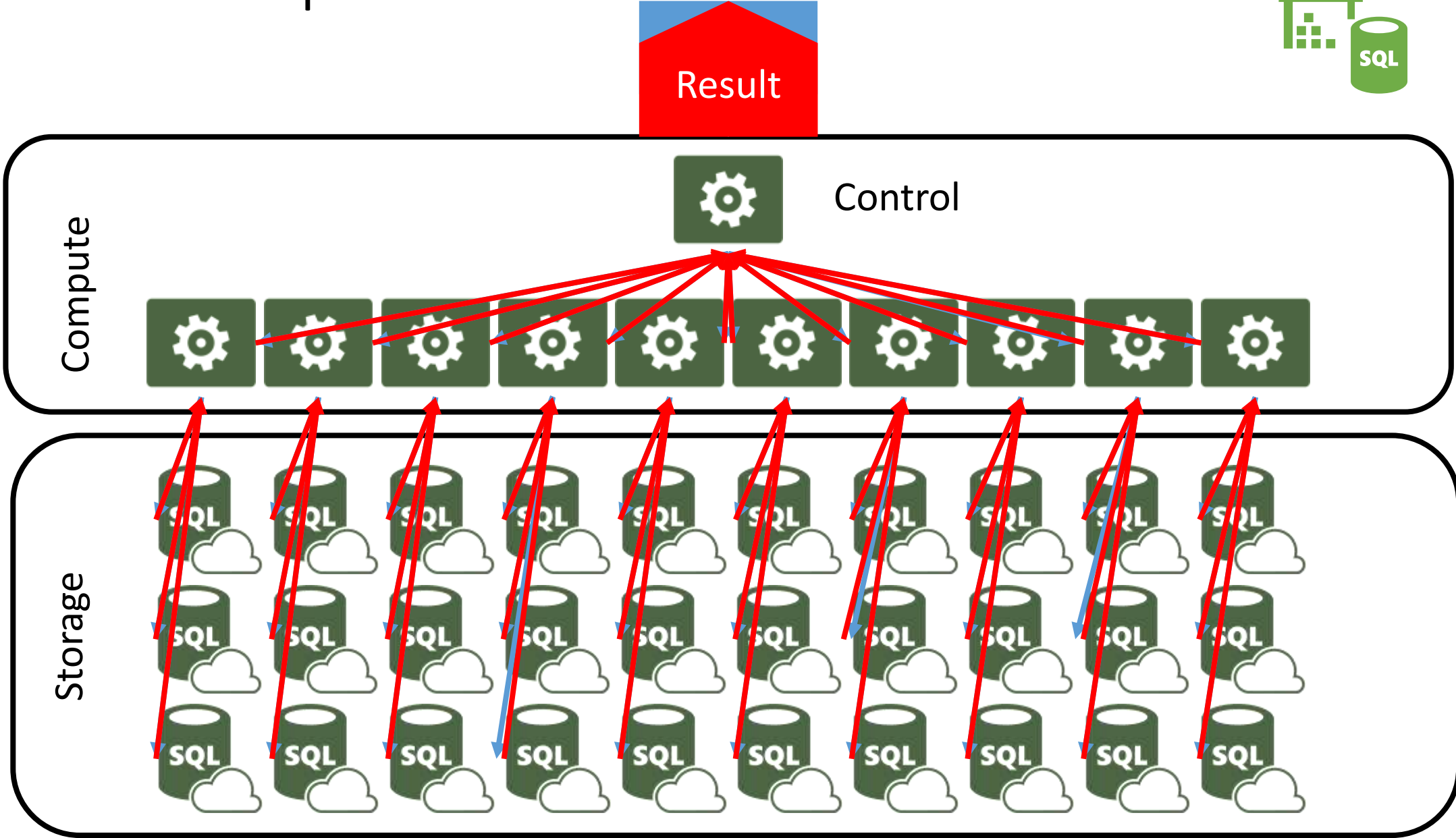
Pattern: SaaS customer isolation



Logical overview



Distributed queries



Azure Data Warehouse

where the smarts happen



Interactive and Exploration

Microsoft Azure Machine Learning | Home Studio Gallery PREVIEW

Browse ▾

Search for entities by name, algorithms or tags

Discover. Learn. Share.

Learn more ▾



EXPERIMENT

Experiment for the Guide to R in Azure

This experiment is developed in Azure Machine Learning.

forecasting predictive analytics

graphics

845 224 7 months ago



Stephen Elston

Machine Learning APIs

MACHINE LEARNING API

Text Analytics



MACHINE LEARNING API

Speech APIs



MACHINE LEARNING API

Recommendations



MACHINE LEARNING API

Customer



Refine by

Categories

- ☒ Experiment
- ☐ Machine Learning API
- ☐ Tutorial

Show

- ☐ Microsoft content only

Tags

- ☐ R
- ☐ Classification
- ☐ classification
- ☐ Apply Transformation
- ☐ Template

Show all

Algorithms used

- ☐ Two-Class Boosted Decision Tree
- ☐ Two-Class Logistic Regression
- ☐ Two-Class Support Vector Machine
- ☐ Linear Regression

Results

You've selected: Experiment x Clear all

EXPERIMENT
Sample 1: Download dataset from UCI: Adult...



This sample demonstrates how to download a dataset from a http location, add column names to the...

reader http reader input
42420 4982 25 days ago

Microsoft

EXPERIMENT
Binary Classification; Twitter sentiment analy...



This experiment demonstrates the use of the Execute R Script, Feature Selection, Feature Hashing module...

Two-Class Support Vector Machine
Classification Text Mining
8445 3061 25 days ago

Microsoft

EXPERIMENT
Regression: Demand estimation



This experiment demonstrates demand estimation using regression with UCI bike rental data.

Boosted Decision Tree Regression
demand estimation
5673 1686 25 days ago

Microsoft

MACHINE LEARNING API

Customer Churn Prediction

by Microsoft April 22, 2015

Description

What if you could know which of your customers are more likely to stop doing business with you next month?

Customer Churn Prediction is a service built with Azure Machine Learning. It's designed to predict when a customer (player, subscriber, user, etc.) is likely to end his or her relationship with a company or service. Being able to predict which customers have a high risk of leaving the relationship with a company provides the company with a window of opportunity to approach them and reduce the likelihood of their leaving.

How does this service work?

By providing historical data to the service, you enable it to learn and train a model that fits your business. After the model has learned this patterns, you will be able to provide recent history for a set of customers and will receive a prediction identifying the subset of customers that have high risk of leaving your business.

No coding required

Get started doing churn analysis today without having to write a single line of code by visiting the [Customer churn](#)



SIGN UP

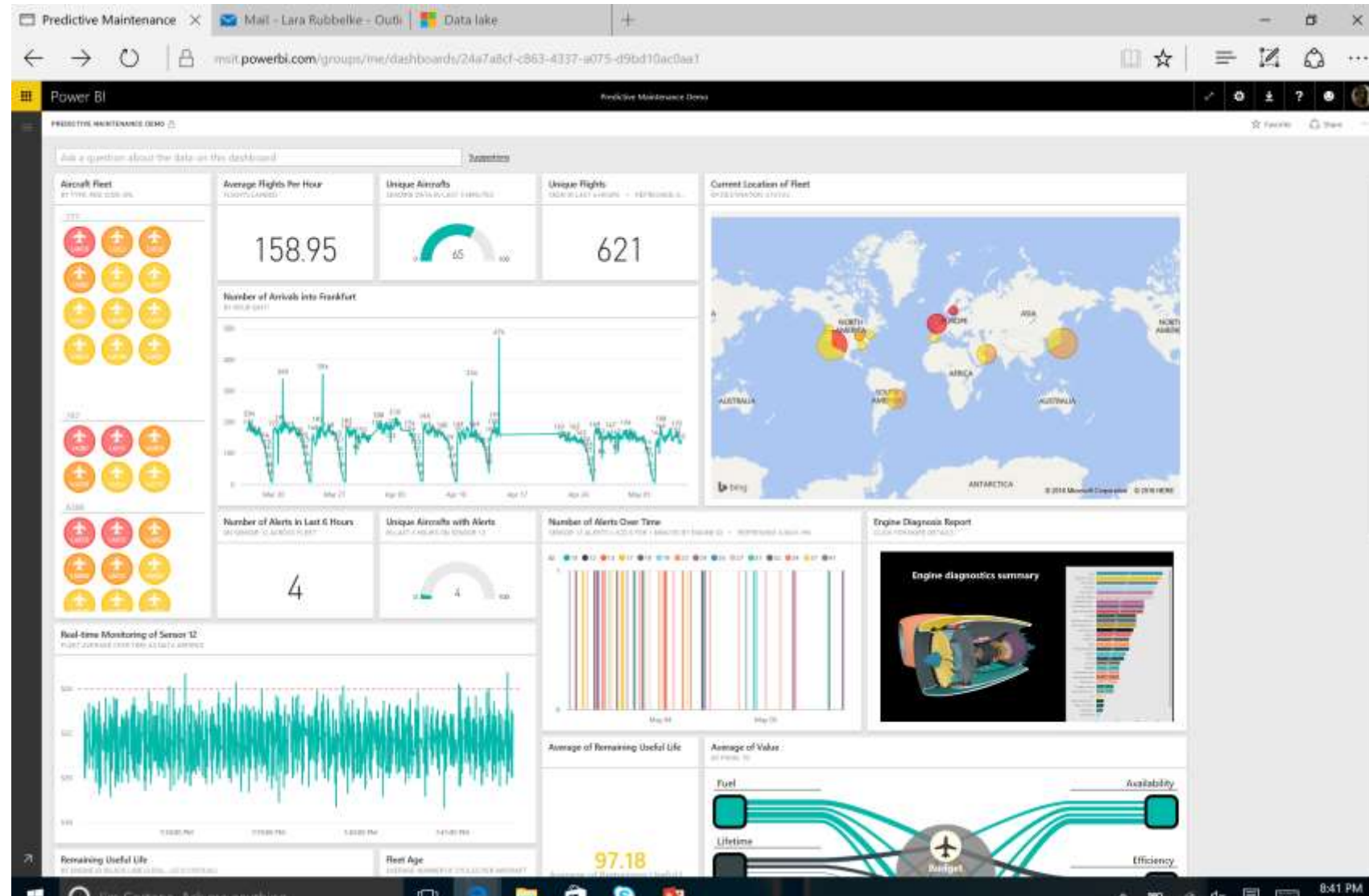
TRY IT NOW

2721 views

Tweet LinkedIn




Interactive and Dashboards



Ingestion

Event Hubs
IoT Hubs
Service Bus
Kafka




Processing

HDInsight
ADLA
Storm
Spark
Stream Analytics



Staging

ADLS
Azure Storage
Azure SQL DB



Serving

ADLS
Azure DW
Azure SQL DB
Hbase
Cassandra
Azure Storage
Power BI



Enrichment and Curation

Azure Data Factory

Azure ML

Modern Data Lifecycle

