The Evolution of Data Platform Architectures in Azure

Lambda, Kappa, Delta, Data Mesh

λ κ δ

Paul Andrew | Technical Architect in Azure CoE

Microsoft® MVP Most Valuable Professional

avanade

Mr Paul Andrew
Consulting Ltd

@MrPaulAndrew In/MrPaulAndrew MrPaulAndrew.com c/MrPaulAndrew

Sponsors

Microsoft

POWER BI SENTINEL™
Governance, Disaster Recovery and Auditing for Power BI

dbWatch

redgate

Coeo

ADVANCING ANALYTICS

sqlbits

pyramid

Octopus Deploy

XTEN
CLOUD | DATA | XOPS

Simpson Associates
The Data Analytics Company

Thank you!
We couldn't do it without you.

DATA RELAY

https://github.com/mrpaulandrew

CommunityEvents

Demo code, content and slides from various community events.

C++

{Event/Location}-{Month}-{Year}

What is the answer to life, the universe and everything?

Answer:
42

Answer:
It depends!

What is big data?

Answer:
It depends!

Answer:
"Any data that you cannot process
in the time that you have/want
using the technology you have."

- Buck Woody

@BuckWoodyMSFT

Volume
Velocity
Variety
Veracity
Value

What is the goal of our data solutions?

Data Collection

Data Sources

Paul's Magic Box - From the Hogwarts School of Witches & Wizardry

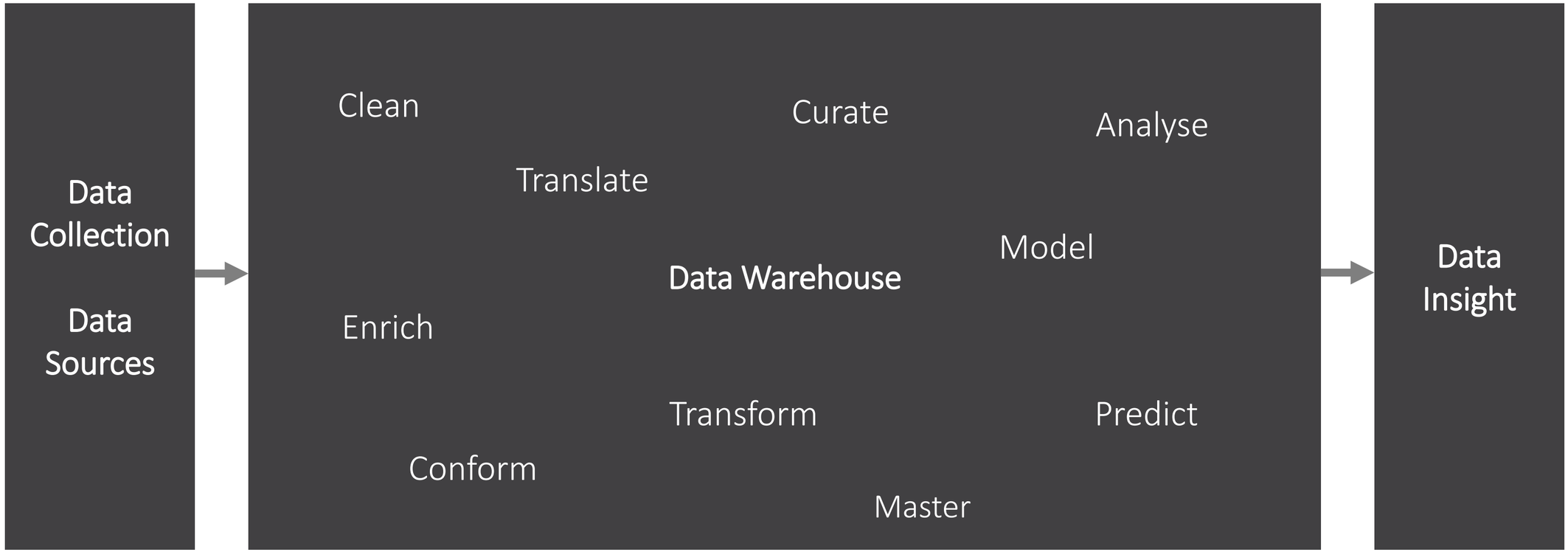
Data Insight

Data = Information = Knowledge = Power

What is the goal of our data solutions?

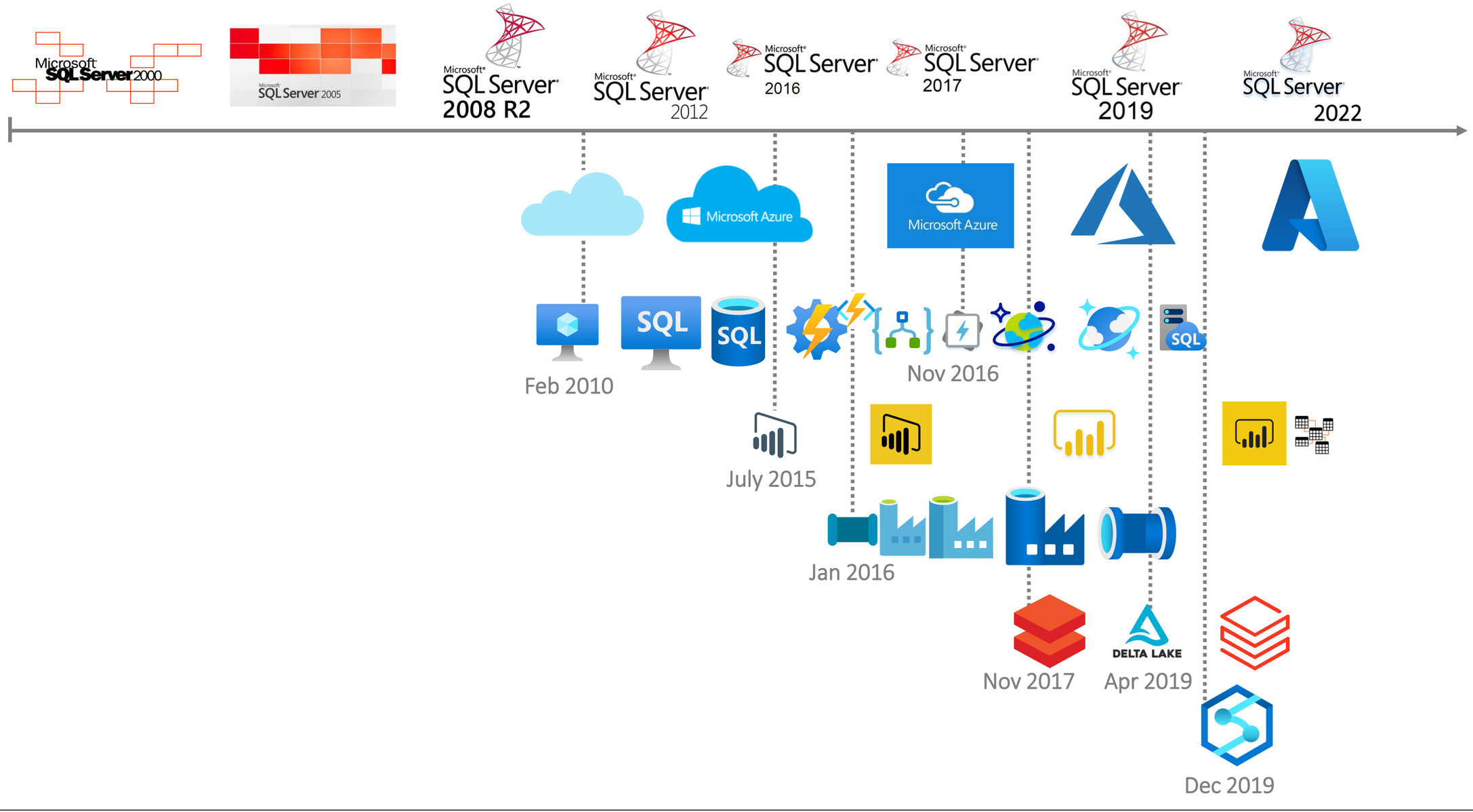
Data = Information = Knowledge = Power/Insights

What is the goal of our data solutions?

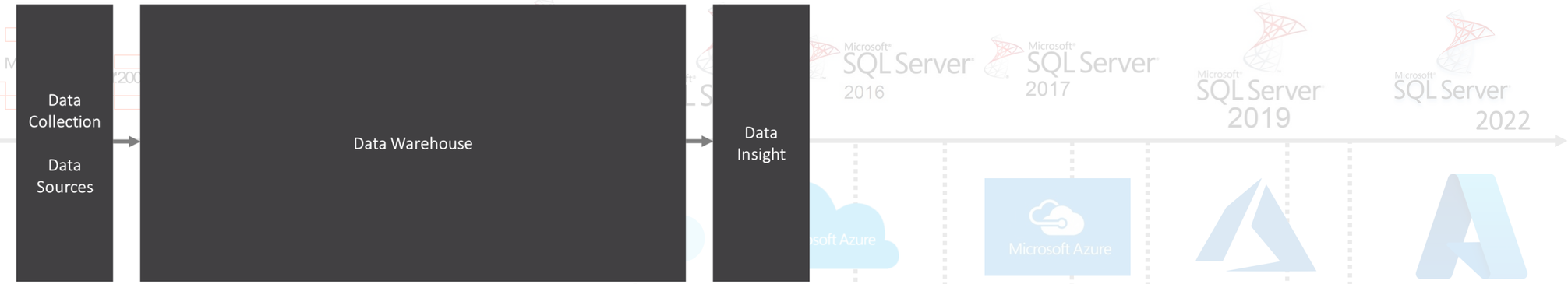


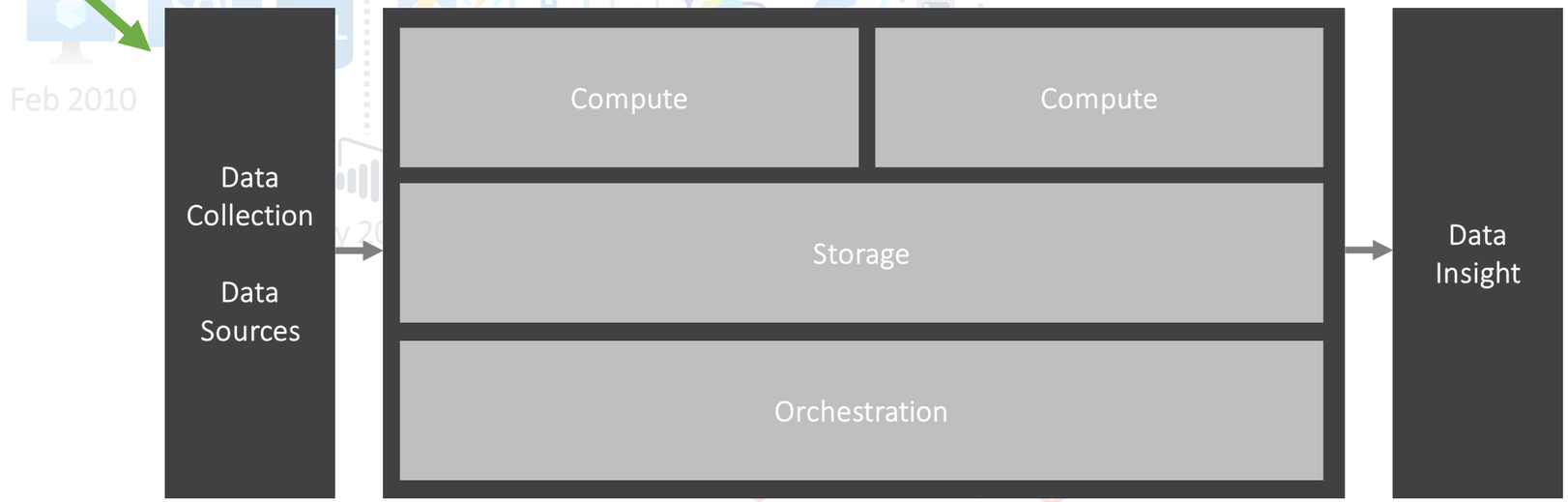
Data = Information = Knowledge = Power/Insights

An Evolution of Data Platforms

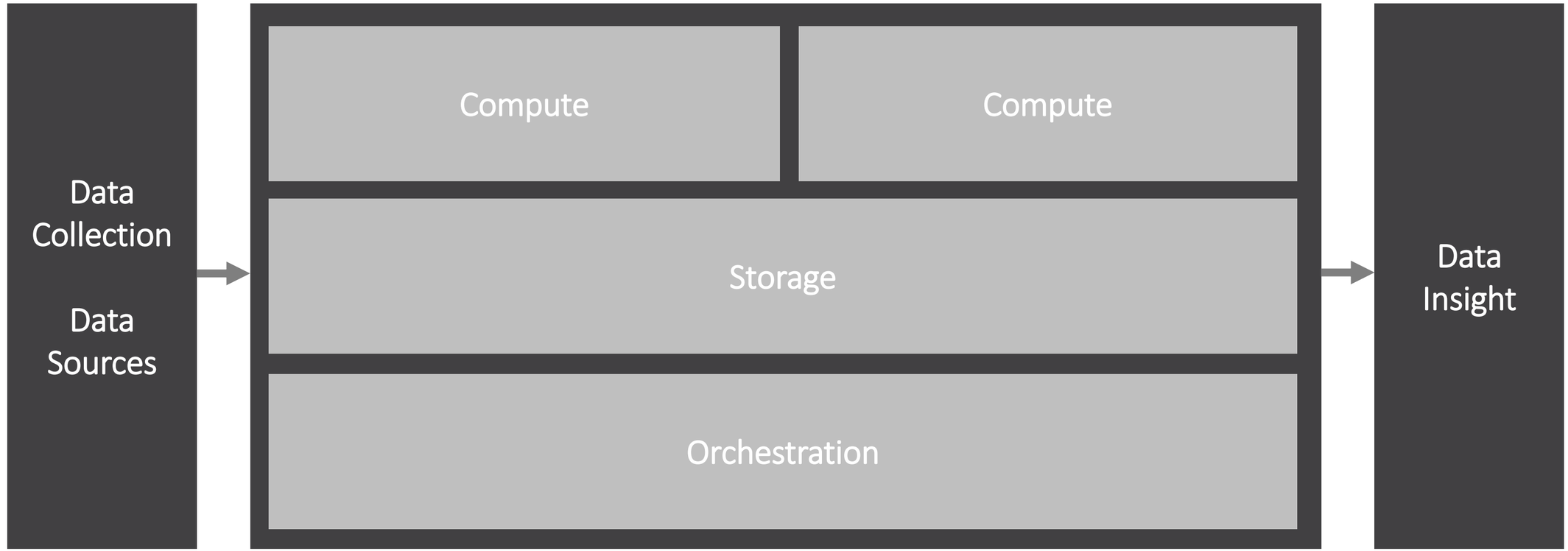


An Evolution of Data Platforms

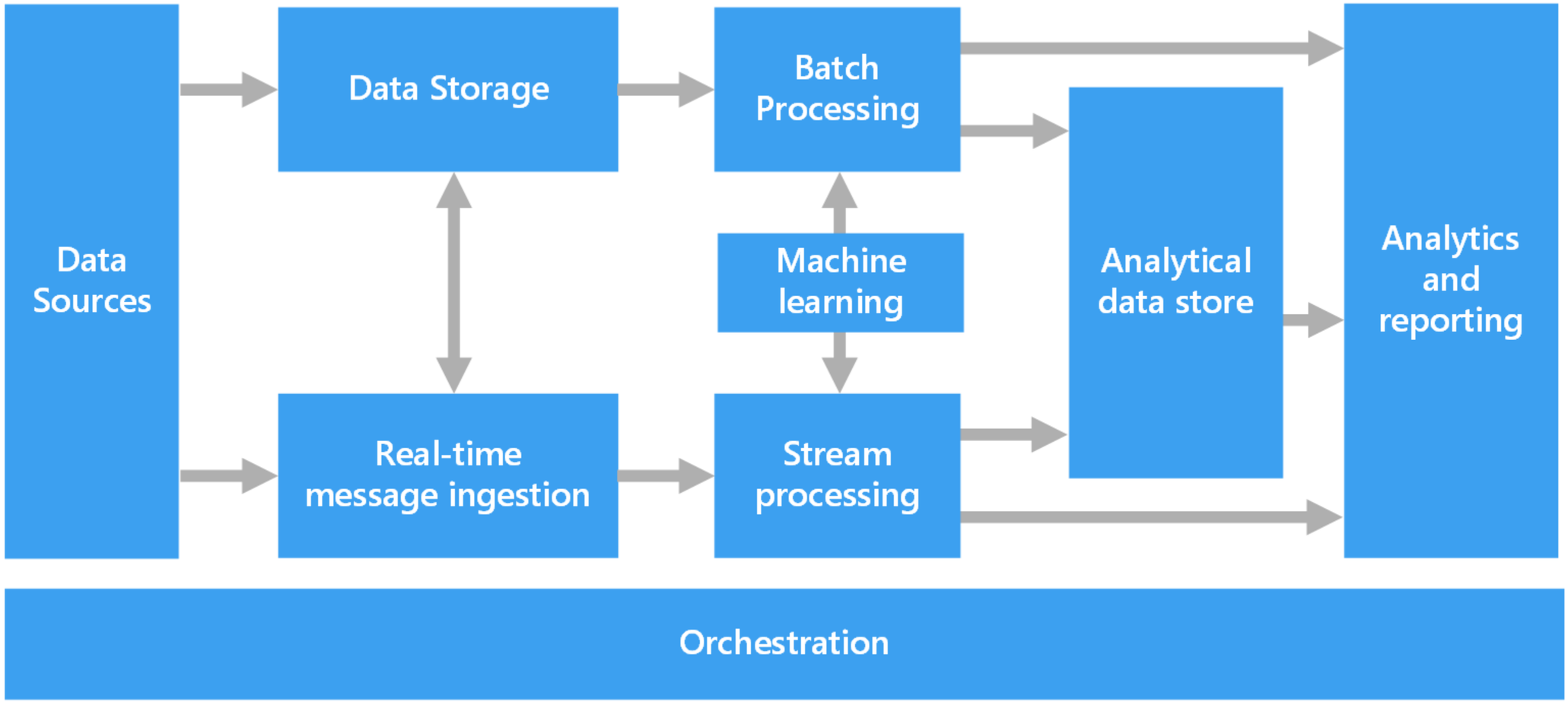




A Reference Architecture?

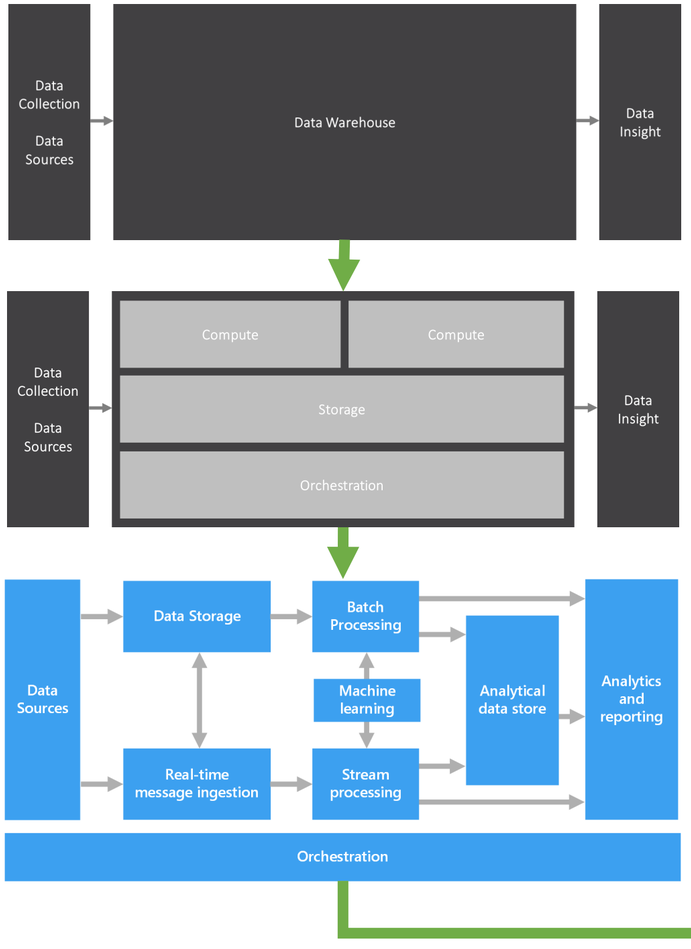


Microsoft's Components of a ~~Big~~ Data Architecture



https://docs.microsoft.com/en-us/azure/architecture/data-guide/big-data/

A Logical Data Architecture



Concepts

Resources

Layers

avanade

Sources

Ingestion

Storage & Compute

Serving

Consumption

Data Governance

Orchestration

Batch

Custom Activity

Integration Runtime

3rd Party Cloud

Data Store Transactional Push

Change Feed

Hybrid Transaction Analytics

Real-Time

Event Endpoints

Digital Twin

IoT Hub

Spark Clusters

Storage Accounts

Raw

Cleaned

Schema Aligned

Conformed

SQL Clusters

Functions

Service Fabric

Batch

Secure Redacted Store

Temporal Analysis

Stream Analytics

Spark Streaming

Analysis Services

SQL Database

Power Platform

Machine Learning

DevTest Labs

Application Gateways

Dashboards

Data Sharing

Self-Service

Mobile

Model Delivery & Real Time APIs

Time Series Insights

Authentication & Credential Management

Active Directory Services

Key Vault

AD B2C - OAuth2 Managed Identities

Source Control, Deployment, Operations, Monitoring

Log Analytics

Application Insights

Alerting Logic Apps

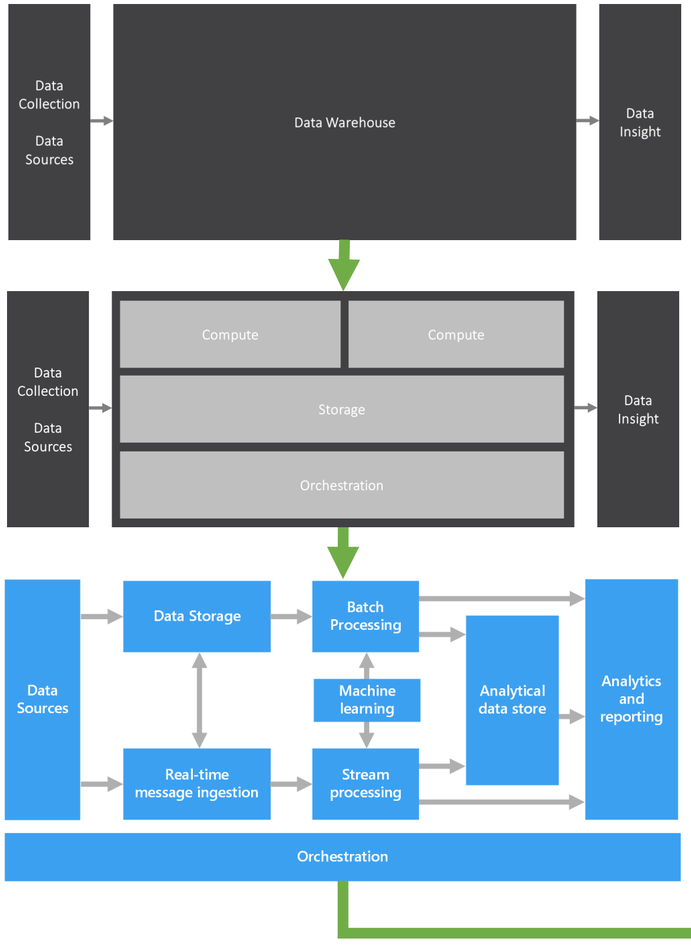
Monitoring

Cost Management

Metrics

DevOps (CI/CD)

A Logical Data Architecture



Concepts

Resources

Layers

avanade

Sources

Ingestion

Storage & Compute

Serving

Consumption

Data Governance

Batch

Hybrid Transaction Analytics

Real-Time

Orchestration

Custom Activity

Integration Runtime

3rd Party Cloud

Data Store Transactional Push

REST

Change Feed

Event Endpoints

Digital Twin

IoT Hub

Spark Clusters

Storage Accounts

Raw

Cleaned

Schema Aligned

Conformed

Secure Redacted Store

Temporal Analysis

Stream Analytics

Spark Streaming

SQL Clusters

Functions

Service Fabric

Batch

Analysis Services

SQL Database

Power Platform

Machine Learning

DevTest Labs

Application Gateways

Model Delivery & Real Time APIs

Time Series Insights

Dashboards

Data Sharing

Self-Service

Mobile

Authentication & Credential Management

Active Directory Services

Key Vault

AD B2C - OAuth2 Managed Identities

Source Control, Deployment, Operations, Monitoring

Log Analytics

Application Insights

Alerting Logic Apps

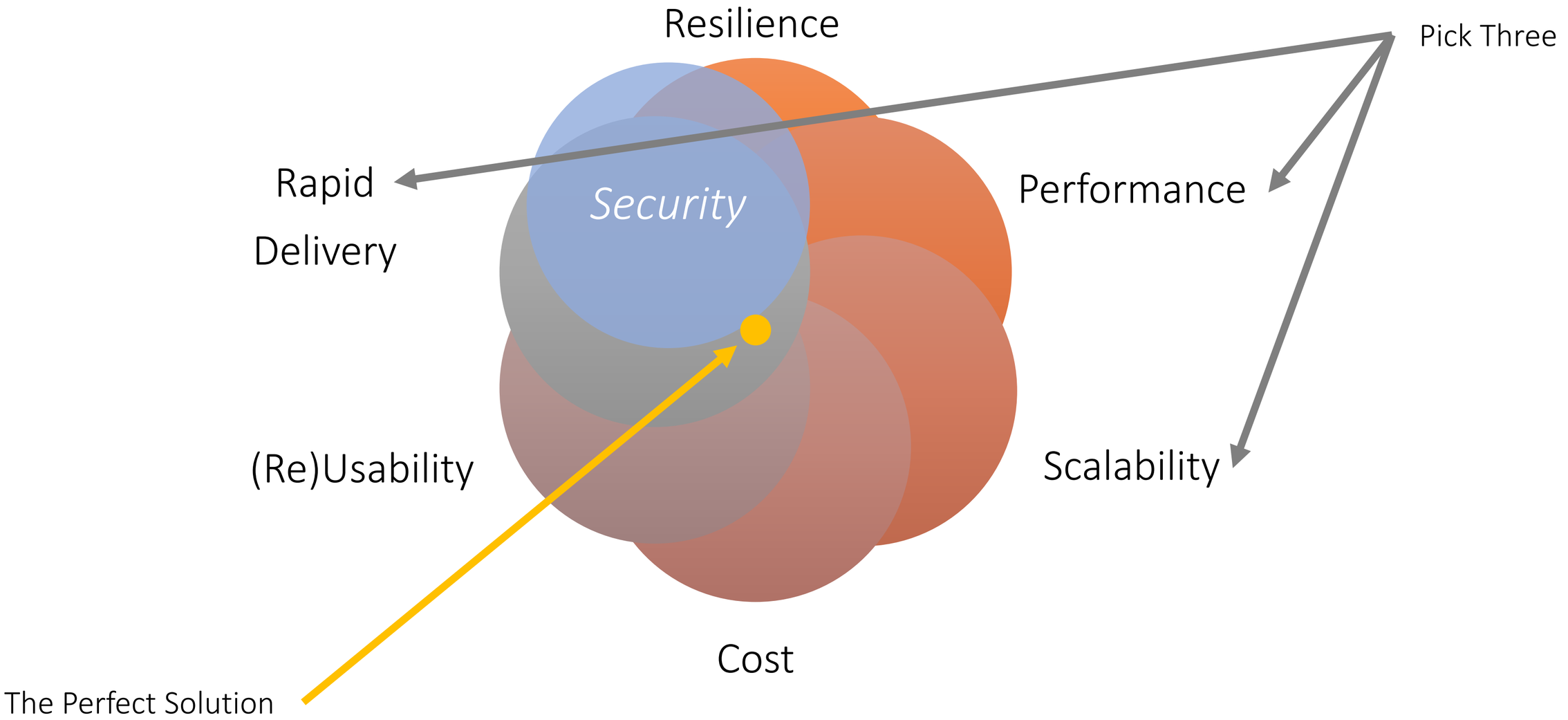
Monitoring

Cost Management

Metrics

DevOps (CI/CD)

What is our primary design focus?



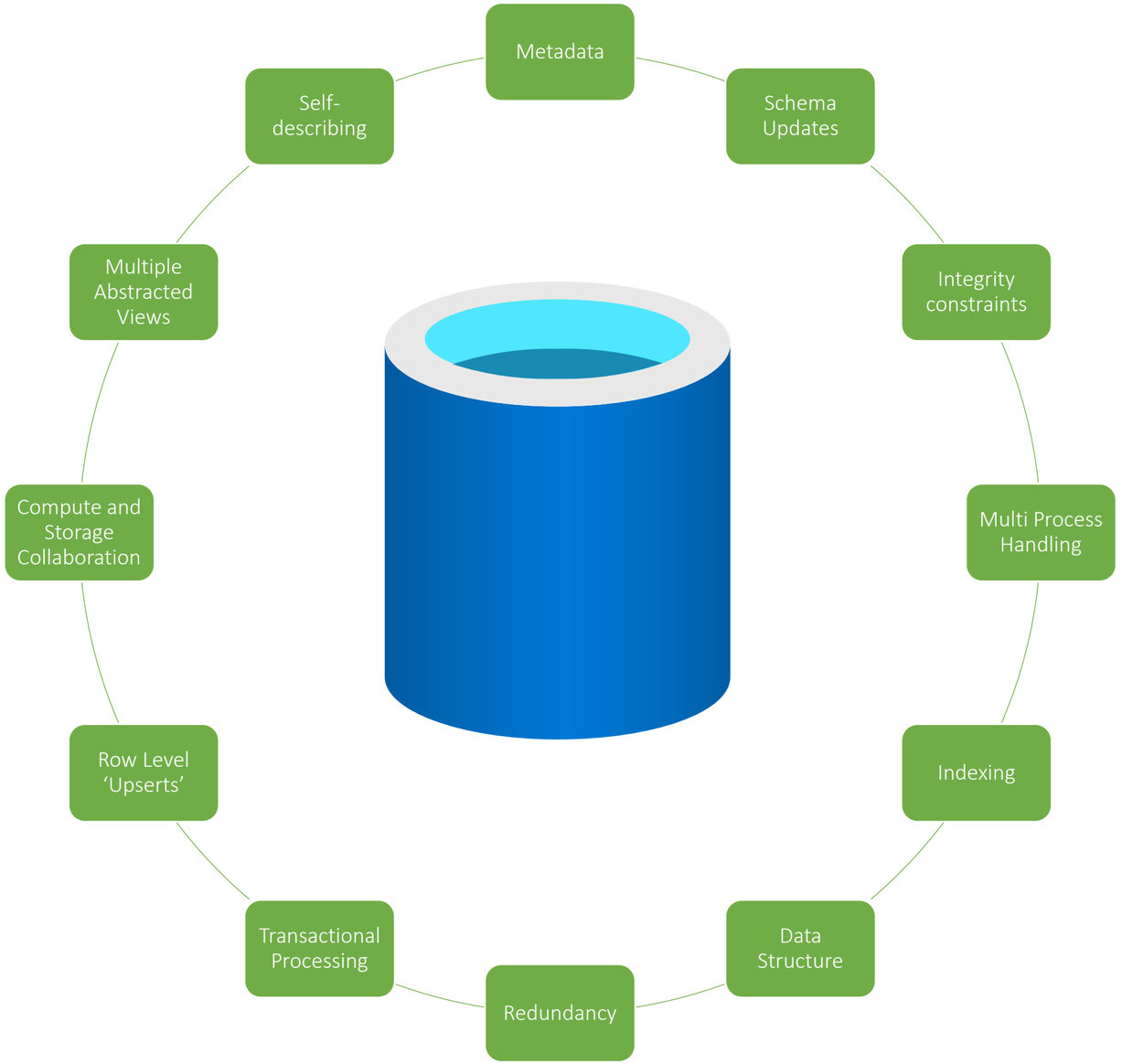
Delta* Lake

Delta.io

* We are not talking about the delta of changed records since our data processing last ran.

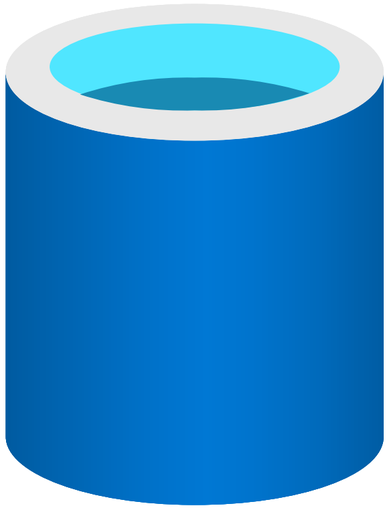
Databases

Data**B**ase
Management
System



Atomicity
Consistency
Isolation
Durability

Databases



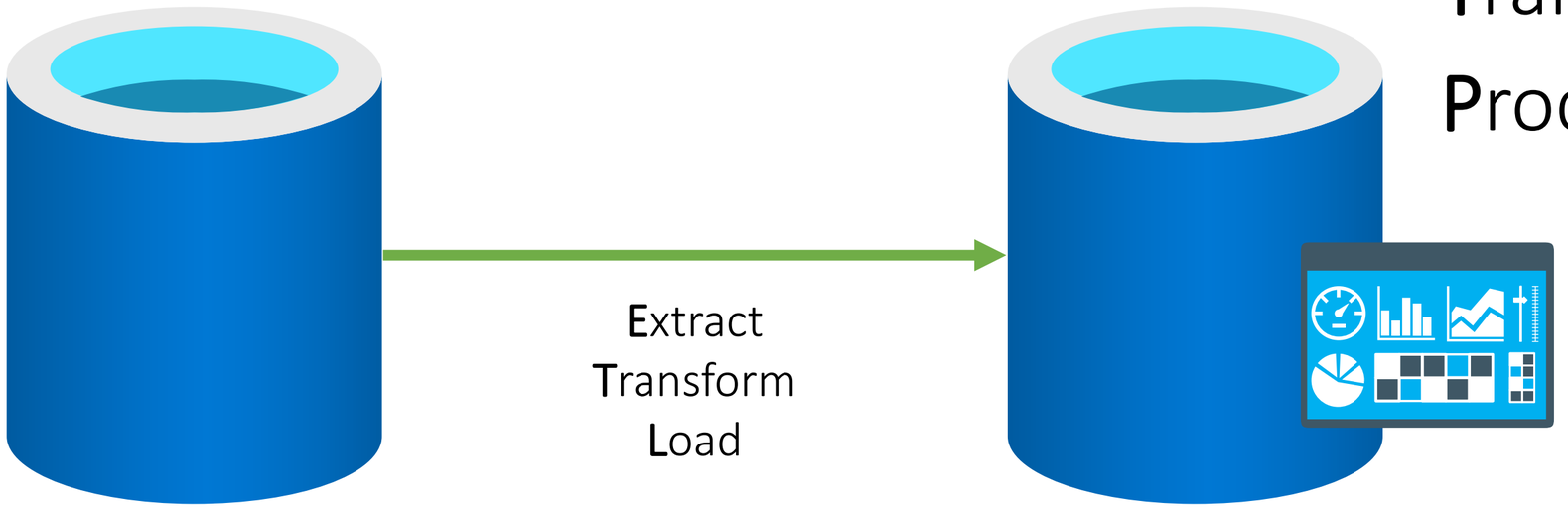
Creating a Data Warehouse

Online

Line

Transactional

Processing



Extract

Transform

Load

Offline

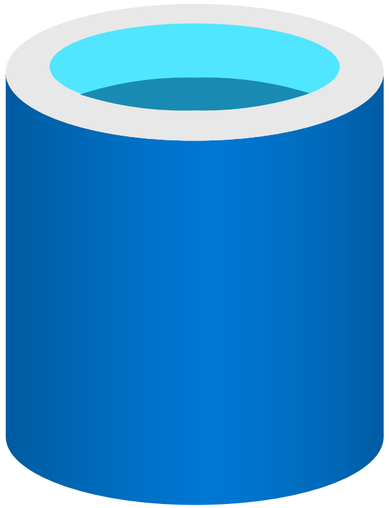
Analytical

Transactional

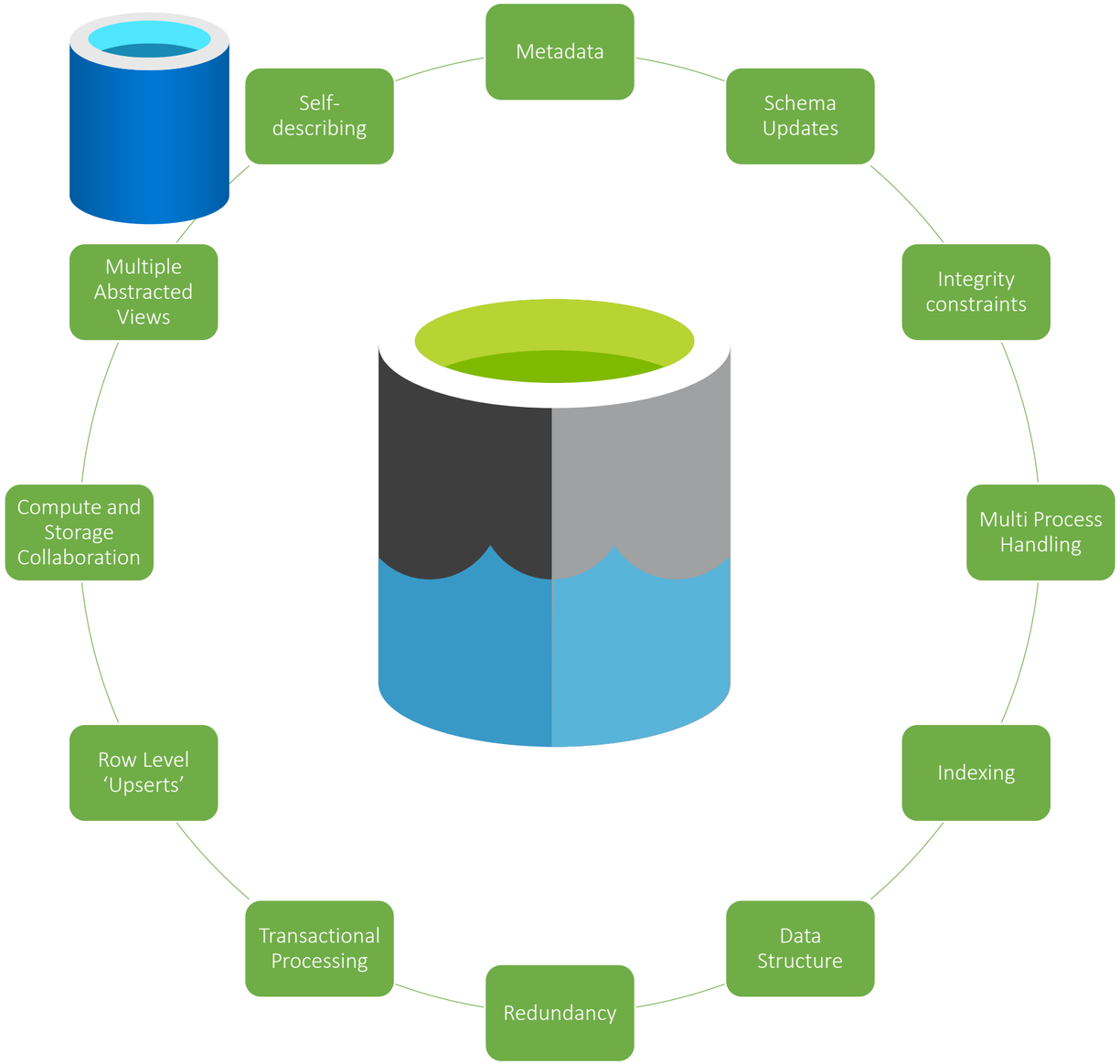
Processing

Application Data

Databases

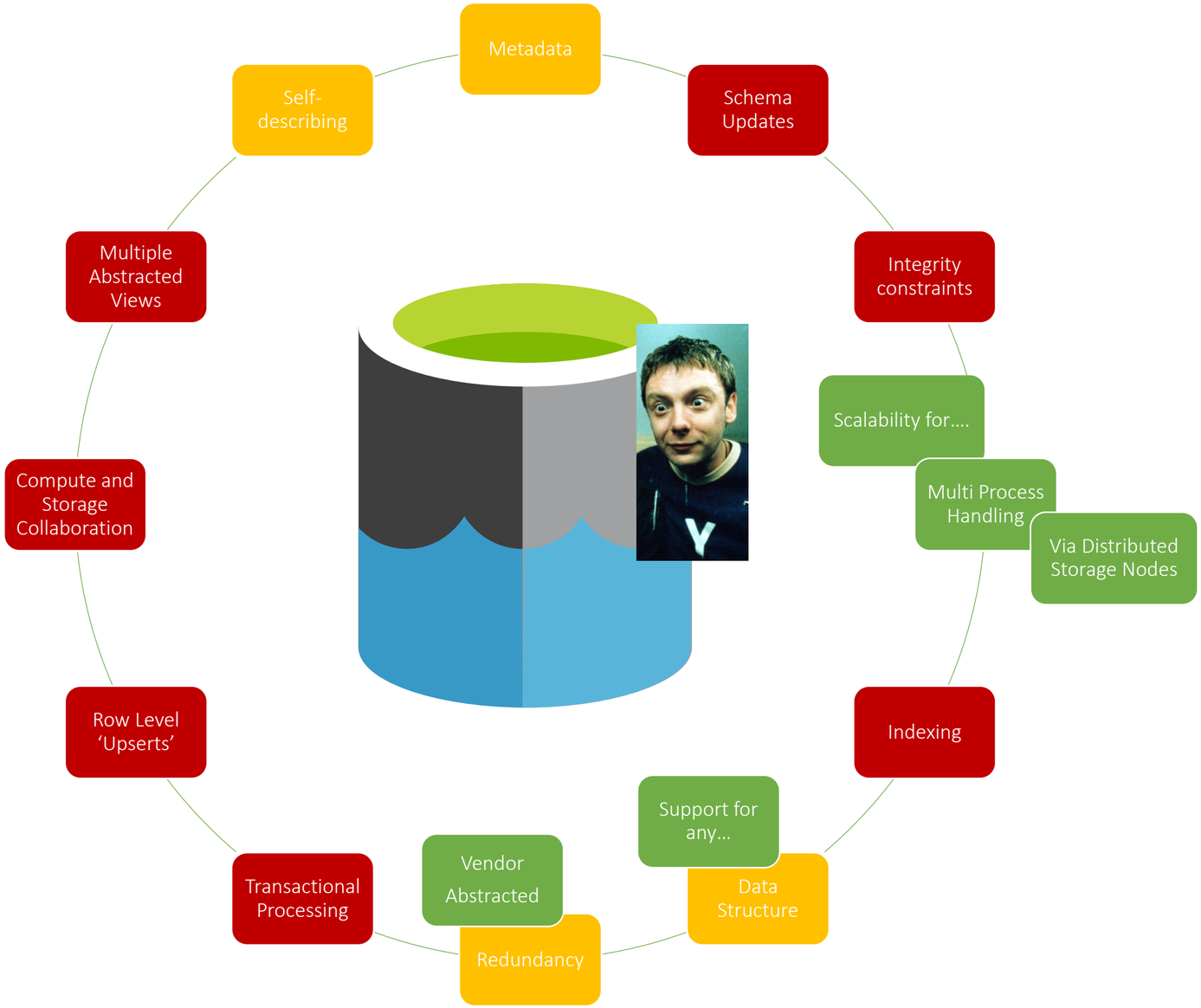


Data Lakes



Volume
Velocity
Variety
Veracity
Value

Data Lakes

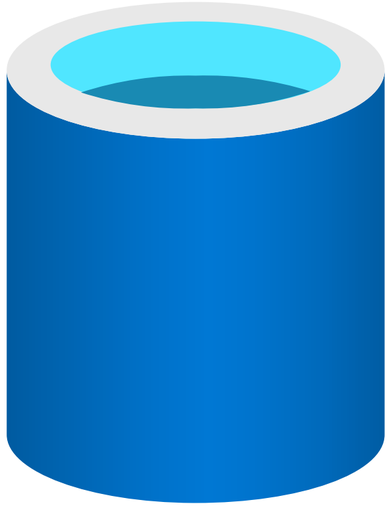


Volume
Velocity
Variety
Veracity
Value

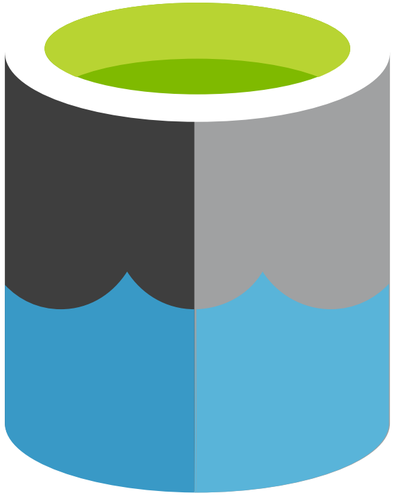
Problem Summary

Data Lakes are good, but they still lack some of the basic ACID functionality needed for data processing.

We are/were trying to use Data Lakes for everything (to replace Databases).



VS

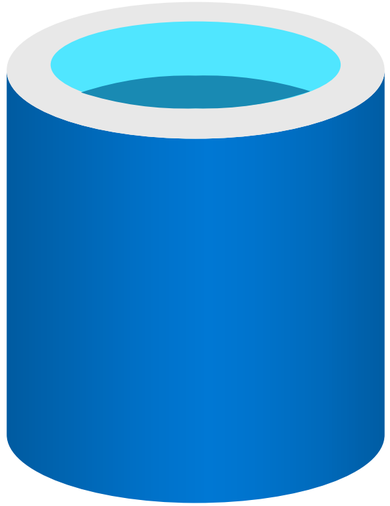


Scales Up	Scales Out
Natural Home for Structured Data	Any Data Structure
Storage Limits	No Storage Limits
Transactional Resilience	No Transactional Handling
Storage & Compute Coupled	Storage & Compute Decoupled

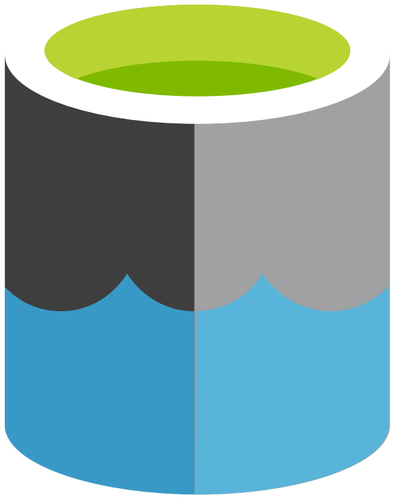
Problem Summary

Data Lakes are good, but they still lack some of the basic ACID functionality needed for data processing.

We are/were trying to use Data Lakes for everything (to replace Databases).



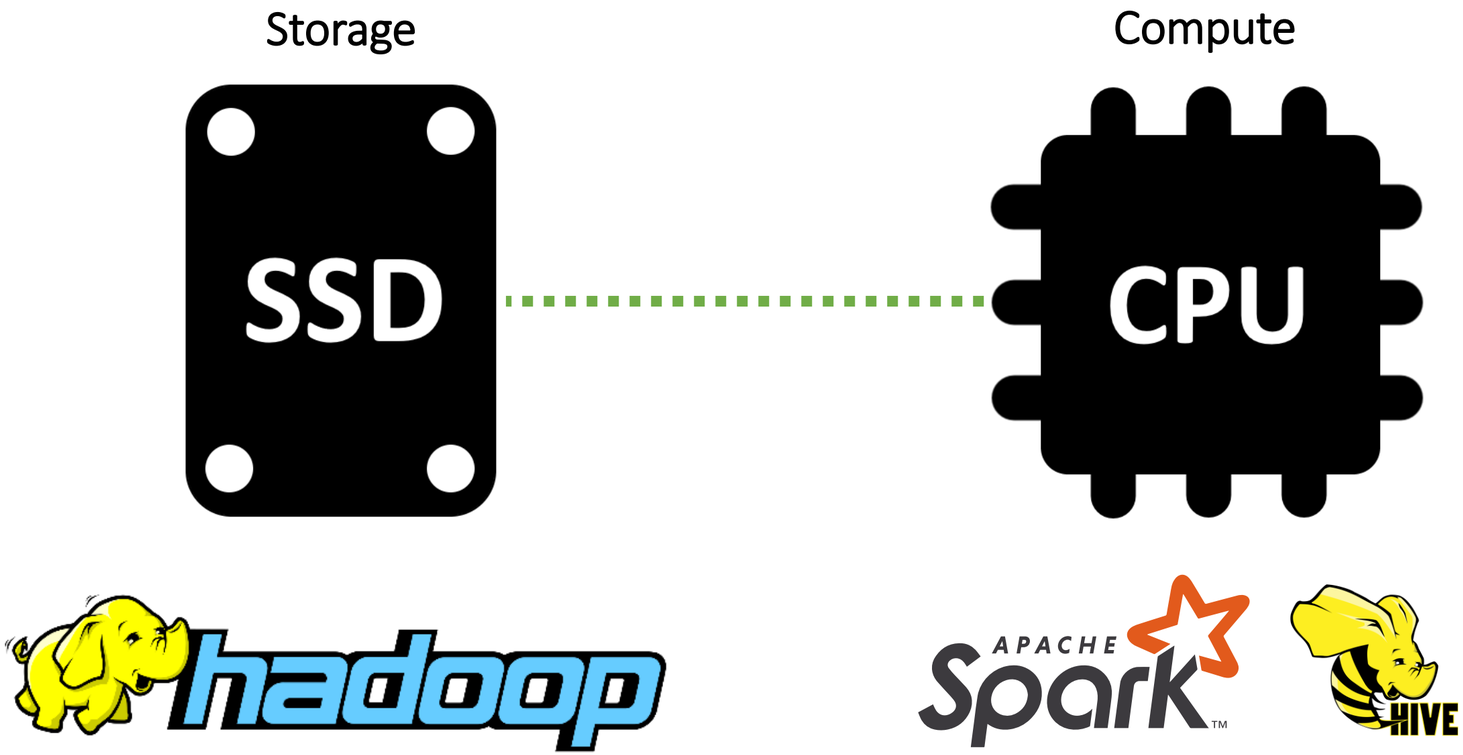
VS



Scales Up	Scales Out
Natural Home for Structured Data	Any Data Structure
Storage Limits	No Storage Limits
Transactional Resilience	No Transactional Handling
Storage & Compute Coupled	Storage & Compute Decoupled

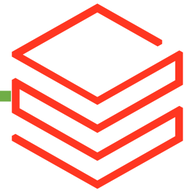
Solution

Just enable ACID transactional support for Data Lakes...



Storage & Compute ~~Decoupled~~ Working Together Again As Friends!

ACID Data Frameworks for Data Lakes



databricks®

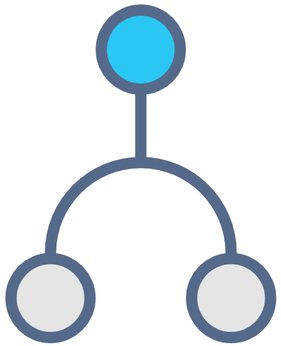
February 2019

NETFLIX

Uber

What is Delta Lake?

DELTA LAKE™ ← databricks®

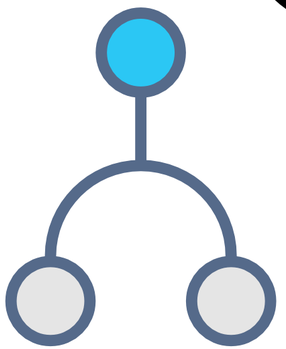


What is Delta Lake?

https://delta.io

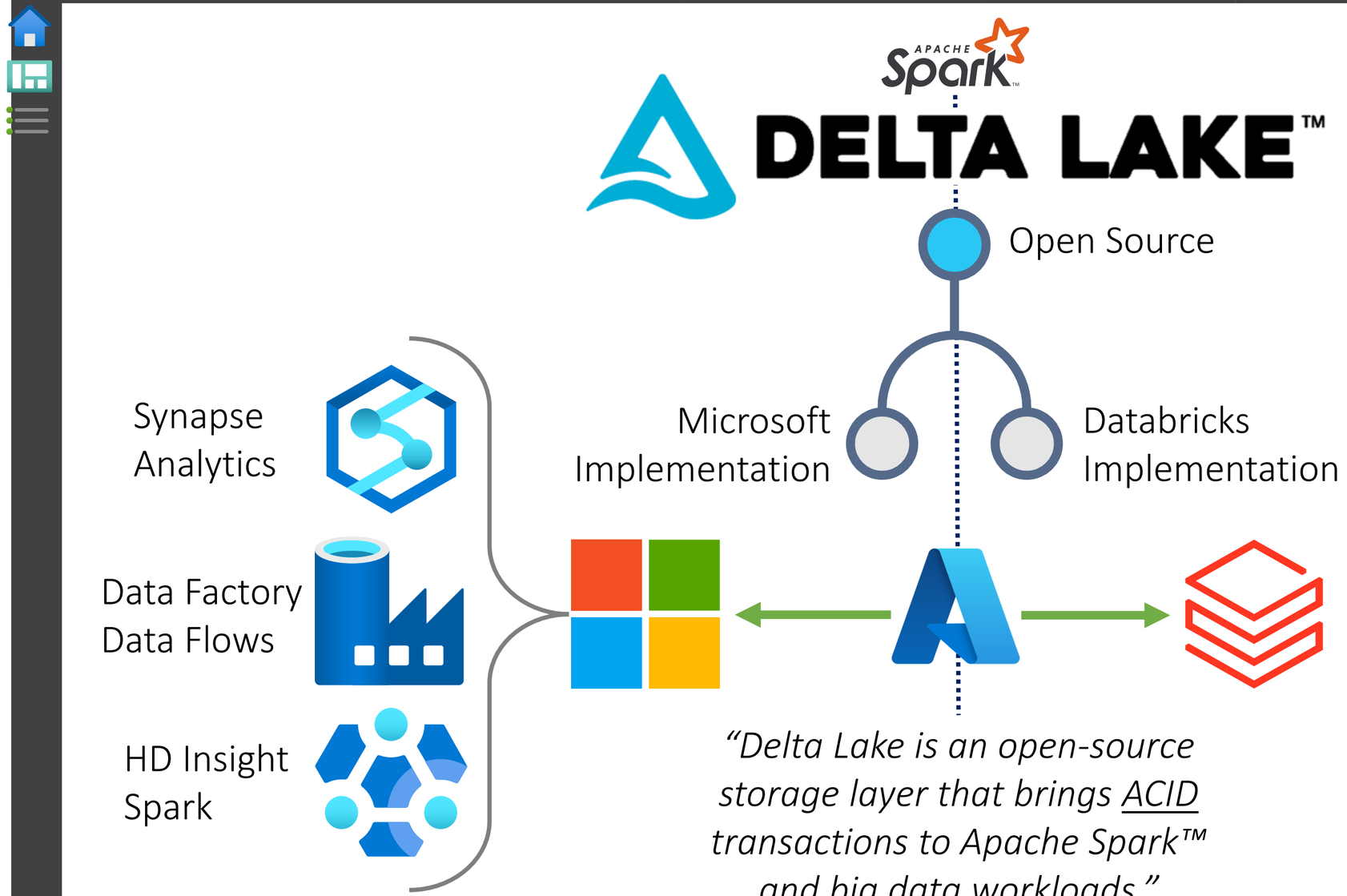
https://github.com/delta-io/delta

April 2019



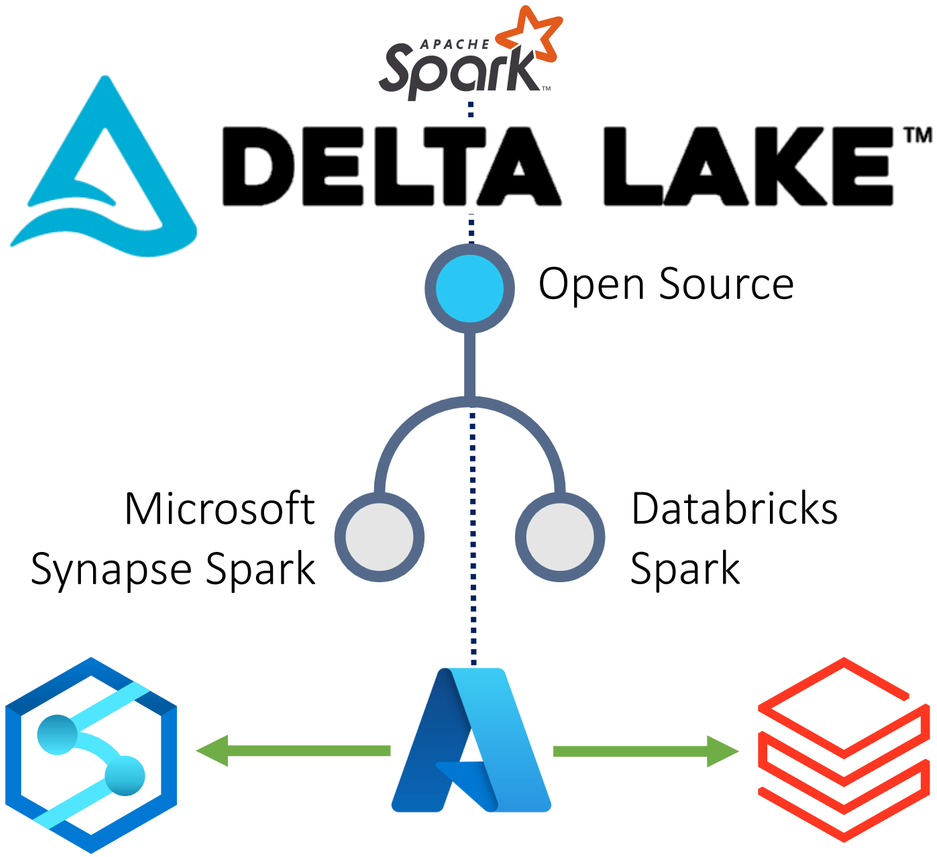
databricks®

What is Delta Lake?



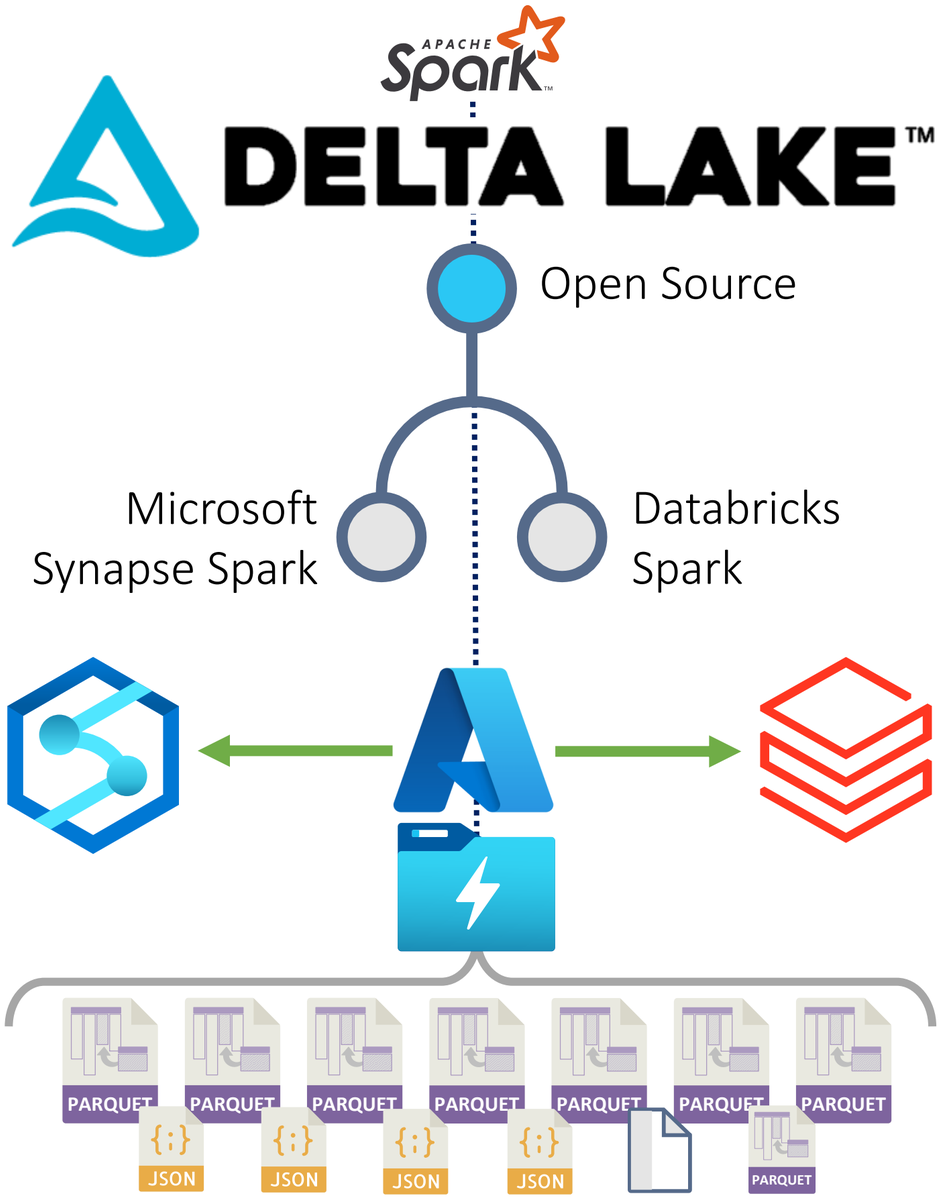
"Delta Lake is an open-source storage layer that brings ACID transactions to Apache Spark™ and big data workloads."

Which Spark Implementation is Better?



"Delta Lake is an open-source storage layer that brings ACID transactions to Apache Spark™ and big data workloads."

Which Spark Implementation is Better?



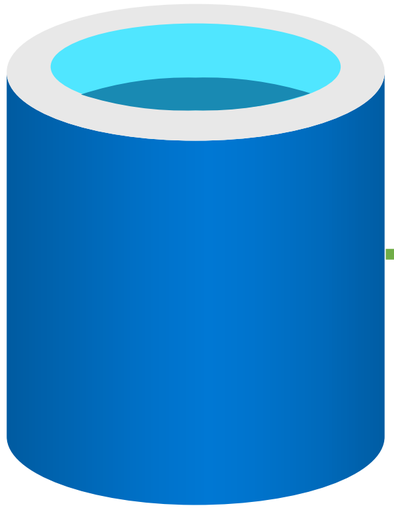
Data Warehouse

Online

Line

Transactional

Processing

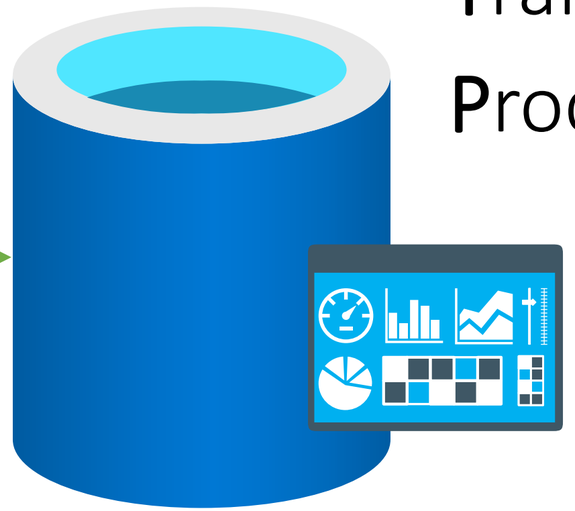


Application Data

Extract

Transform

Load



Offline

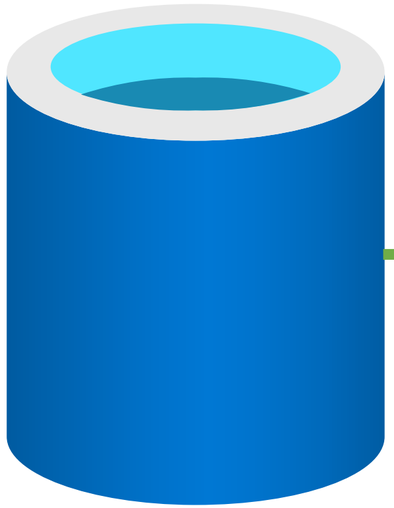
Analytical

Transactional

Processing

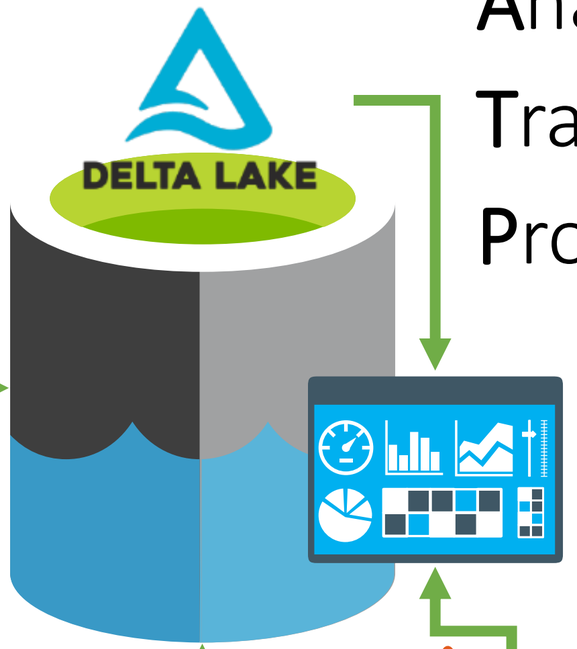
Lake House

Online
Line
Transactional
Processing

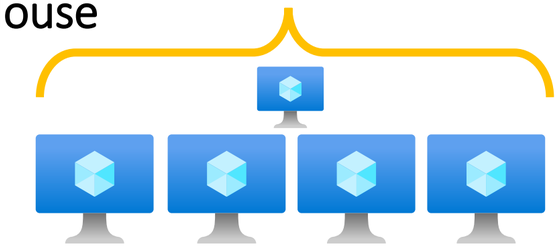


Application
Data

Extract
Transform
Load



Lake
House



Offline
Analytical
Transactional
Processing

Lake House

Not logged in Talk Contributions Create account Log in

Article Talk | Read Edit View history | Search Wikipedia

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Current events
Random article
About Wikipedia
Contact us
Donate

Contribute
Help
Learn to edit
Community portal
Recent changes
Upload file

Tools
What links here
Related changes
Special pages
Permanent link
Page information
Cite this page
Wikidata item

Print/export
Download as PDF
Printable version

Languages
العربية
Deutsch
Español
Français

The Lake House (film)

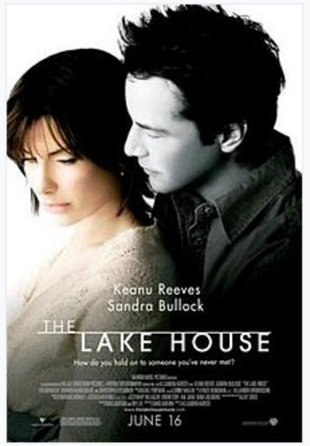
From Wikipedia, the free encyclopedia

This article includes a list of general references, but it remains largely unverified because **it lacks sufficient corresponding inline citations**. Please help to improve this article by introducing more precise citations. *(October 2017) (Learn how and when to remove this template message)*

The Lake House is a 2006 American fantasy romantic drama film directed by Alejandro Agresti, starring Keanu Reeves and Sandra Bullock (who had previously appeared together in the box office hit *Speed*). It was written by David Auburn.[2] A remake of the South Korean motion picture *Il Mare* (2000), it centers on an architect living in 2004 and a doctor living in 2006 who meet via letters left in a mailbox at the lake house where they have lived at separate points in time. They carry on correspondence over two years, remaining separated by their original difference of two years.[3]

Contents [hide]
1 Plot
2 Cast
3 Production
4 Music
5 Reception
5.1 Box office
5.2 Critical response
5.3 Home media
5.4 Awards
6 References
7 External links

The Lake House

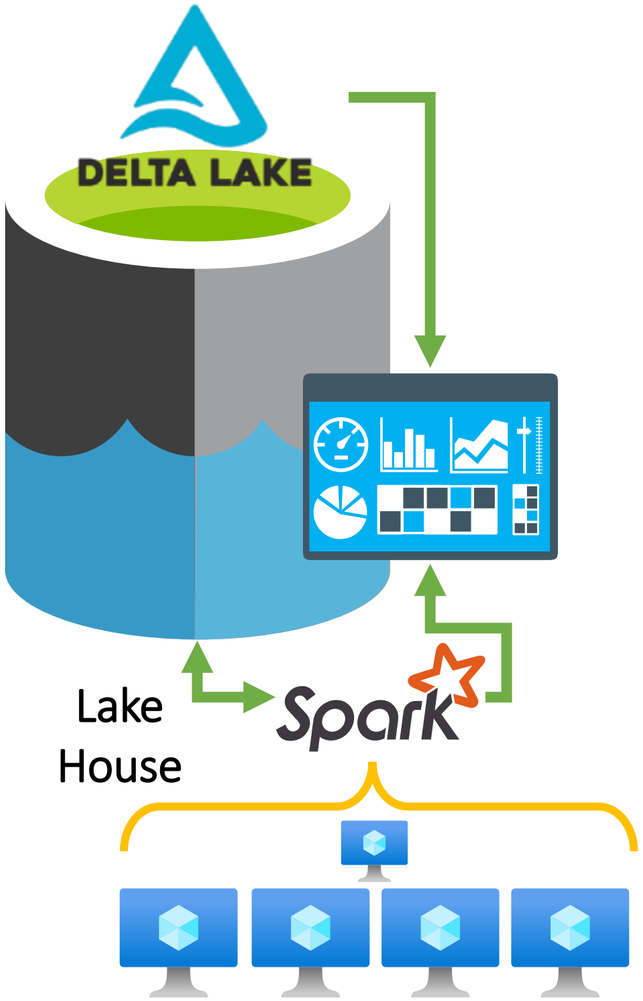


Theatrical release poster

Directed by	Alejandro Agresti
Written by	David Auburn
Based on	*Il Mare* by Kim Eun-jeong Kim Mi-yeong
Produced by	Doug Davison Roy Lee
Starring	Keanu Reeves

Plot [edit]

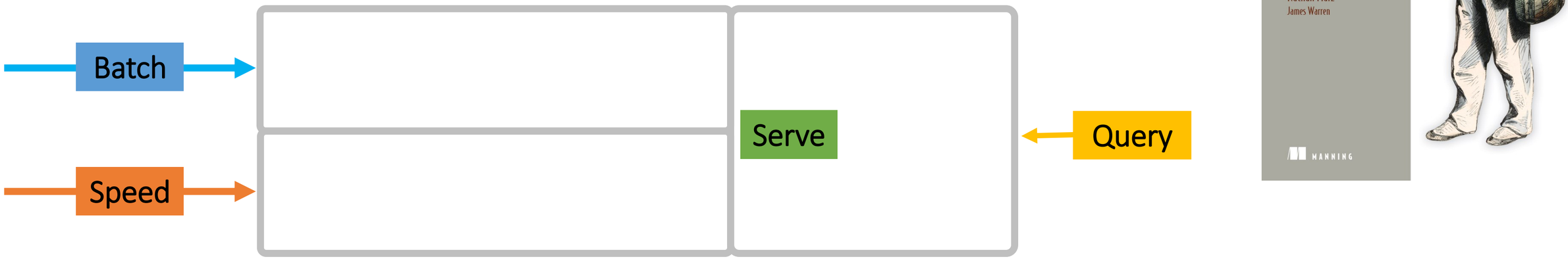
In 2006, Dr. Kate Forster (Sandra Bullock) is leaving a lake house that she has been renting in Chicago. Kate leaves a note in the mailbox for the next tenant to forward her mail, adding that the paint-embedded pawprints on the path leading to the house were already there when she arrived.

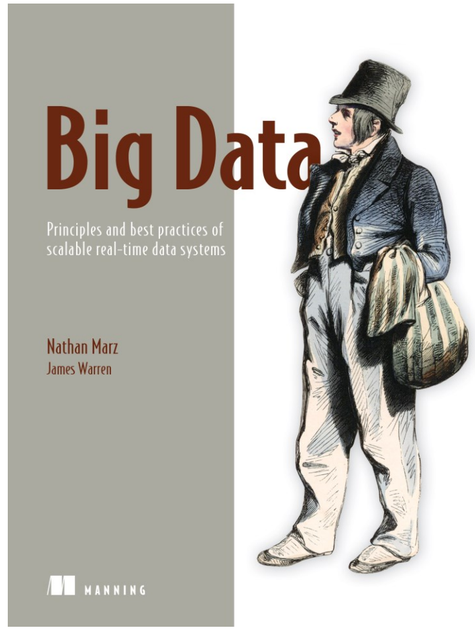


Lambda* & Kappa

* We are not talking about the computer game Half-Life.

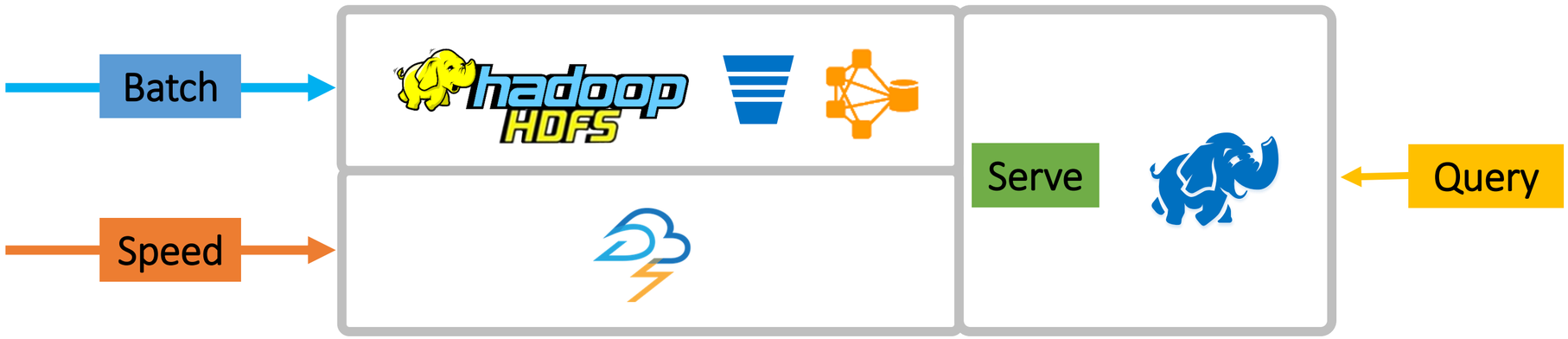
Lambda & Kappa Architectures

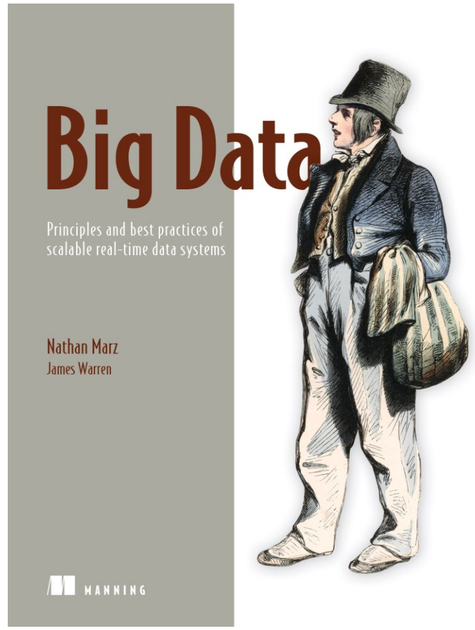




Lambda & Kappa Architectures

~2015

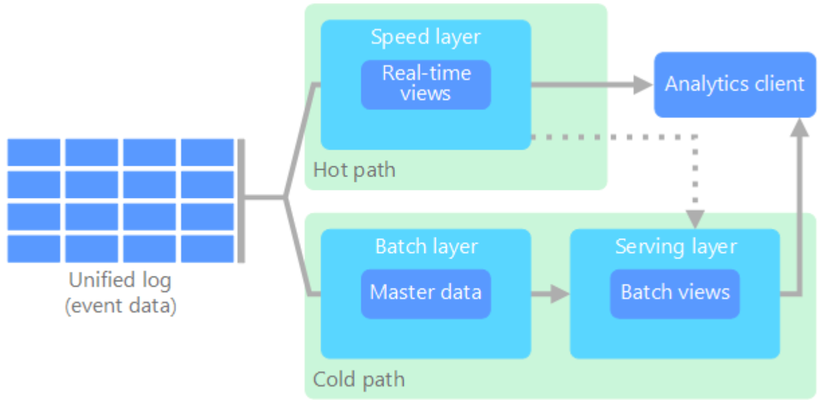




Lambda & Kappa Architectures

Lambda & Kappa Architectures

Lambda

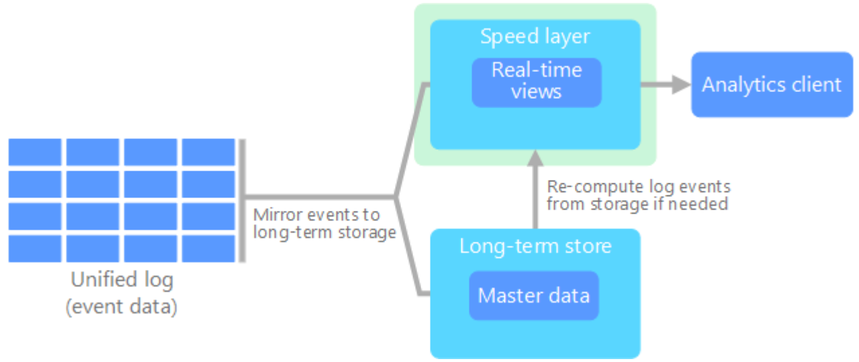


"The **lambda architecture**, first proposed by Nathan Marz, addresses this problem by creating two paths for data flow. All data coming into the system goes through these two paths:

A **batch layer** (cold path) stores all of the incoming data in its raw form and performs batch processing on the data. The result of this processing is stored as a **batch view**.

A **speed layer** (hot path) analyzes data in real time. This layer is designed for low latency, at the expense of accuracy."

Kappa

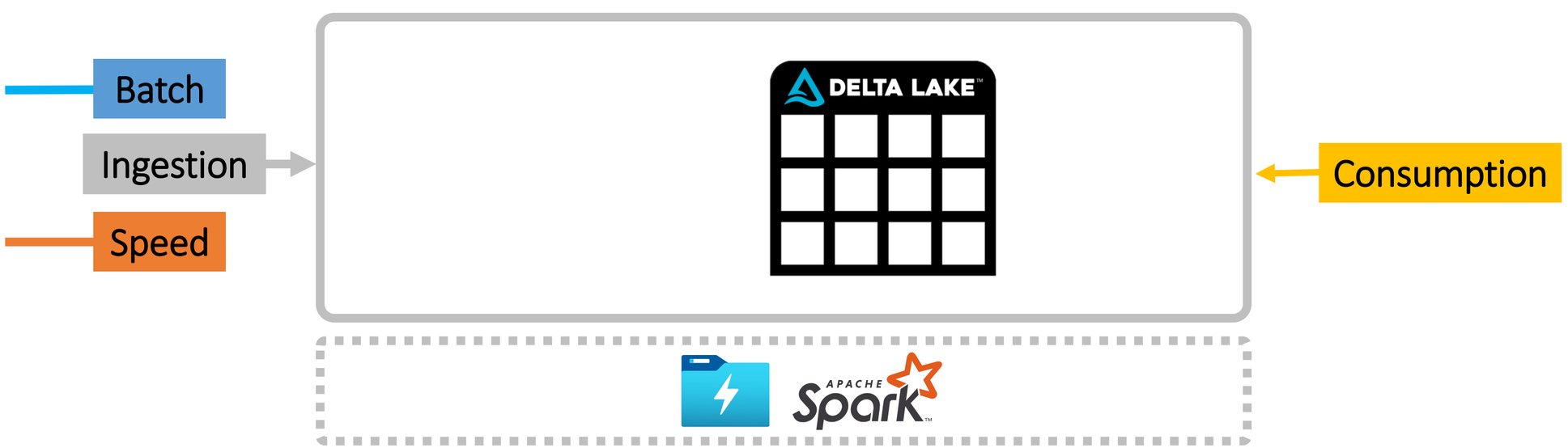


"A drawback to the lambda architecture is its complexity. Processing logic appears in two different places — the cold and hot paths — using different frameworks. This leads to duplicate computation logic and the complexity of managing the architecture for both paths.

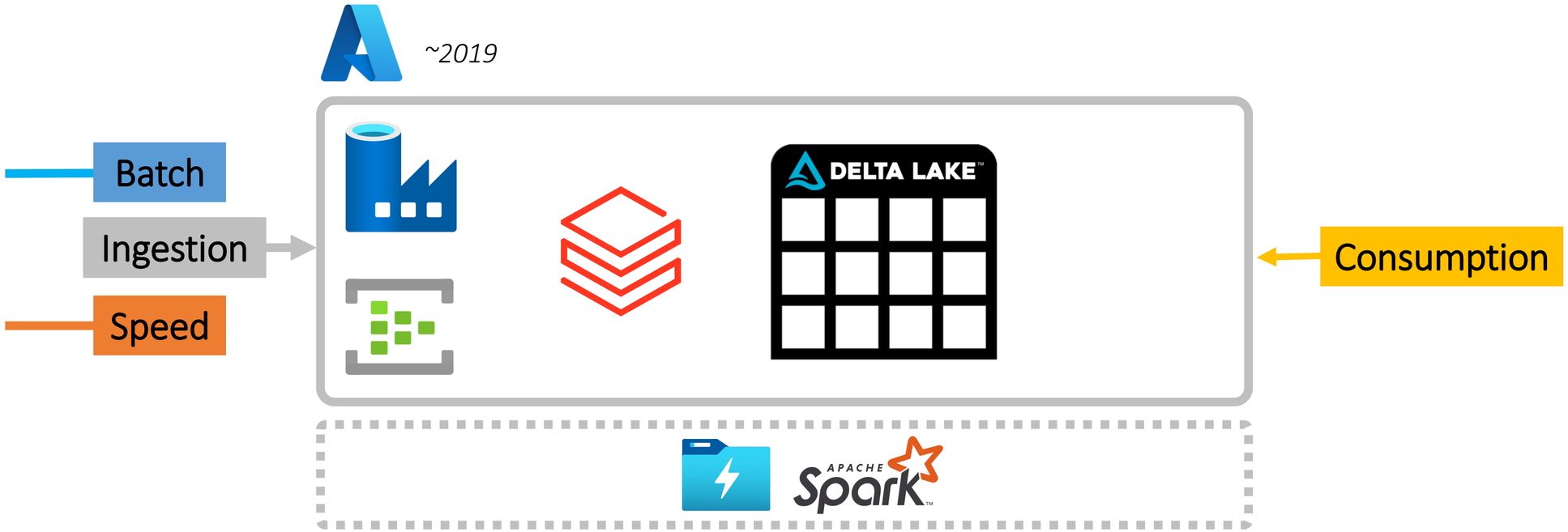
The **kappa architecture** was proposed by Jay Kreps as an alternative to the lambda architecture. It has the same basic goals as the lambda architecture, but with an important distinction: All data flows through a single path, using a stream processing system."

https://docs.microsoft.com/en-us/azure/architecture/data-guide/big-data/

Lambda & Kappa Architectures



Lambda & Kappa Architectures



Lambda & Kappa Architectures

~2021

Batch

Ingestion

Speed

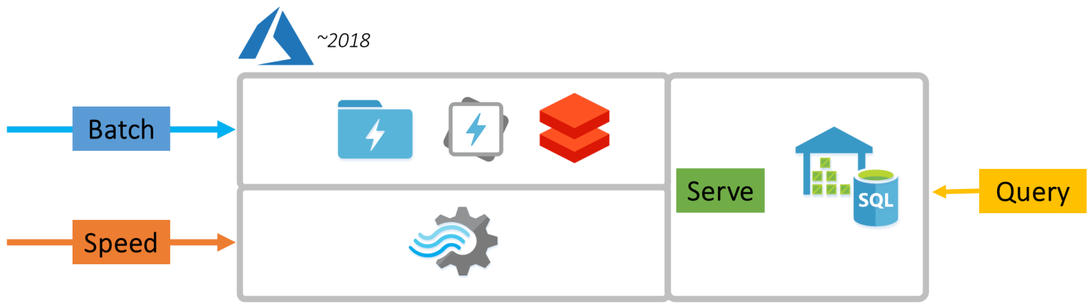
DELTA LAKE

Consumption

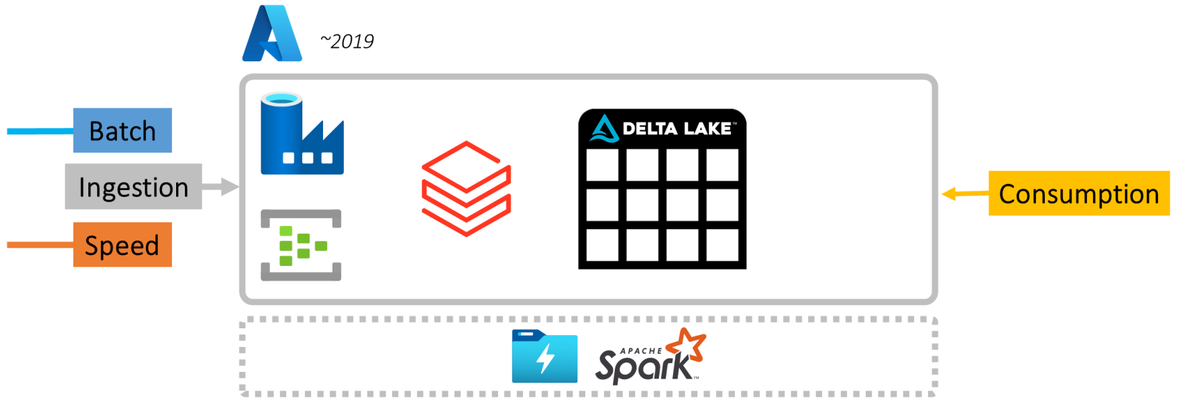
APACHE Spark

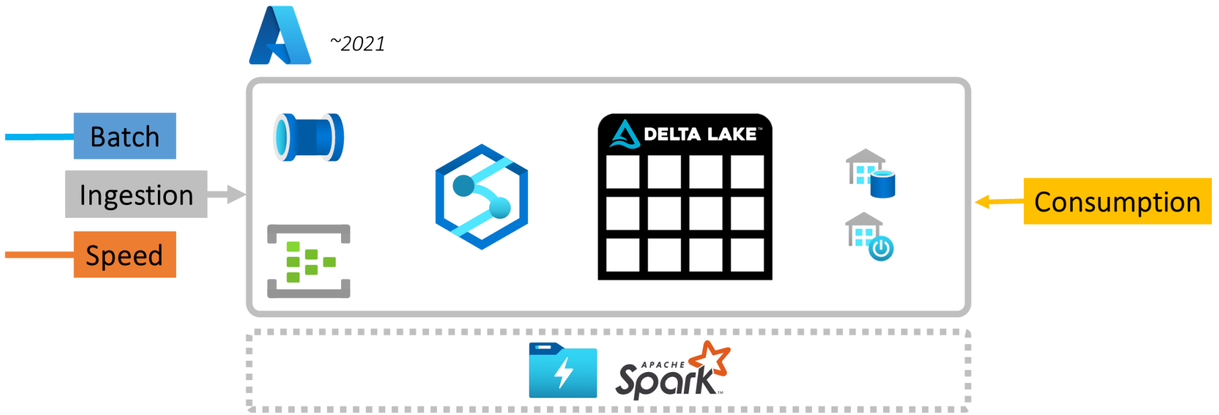
Delta Lake in the Context of Lambda & Kappa

Lambda



Kappa

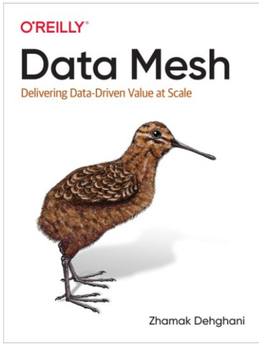


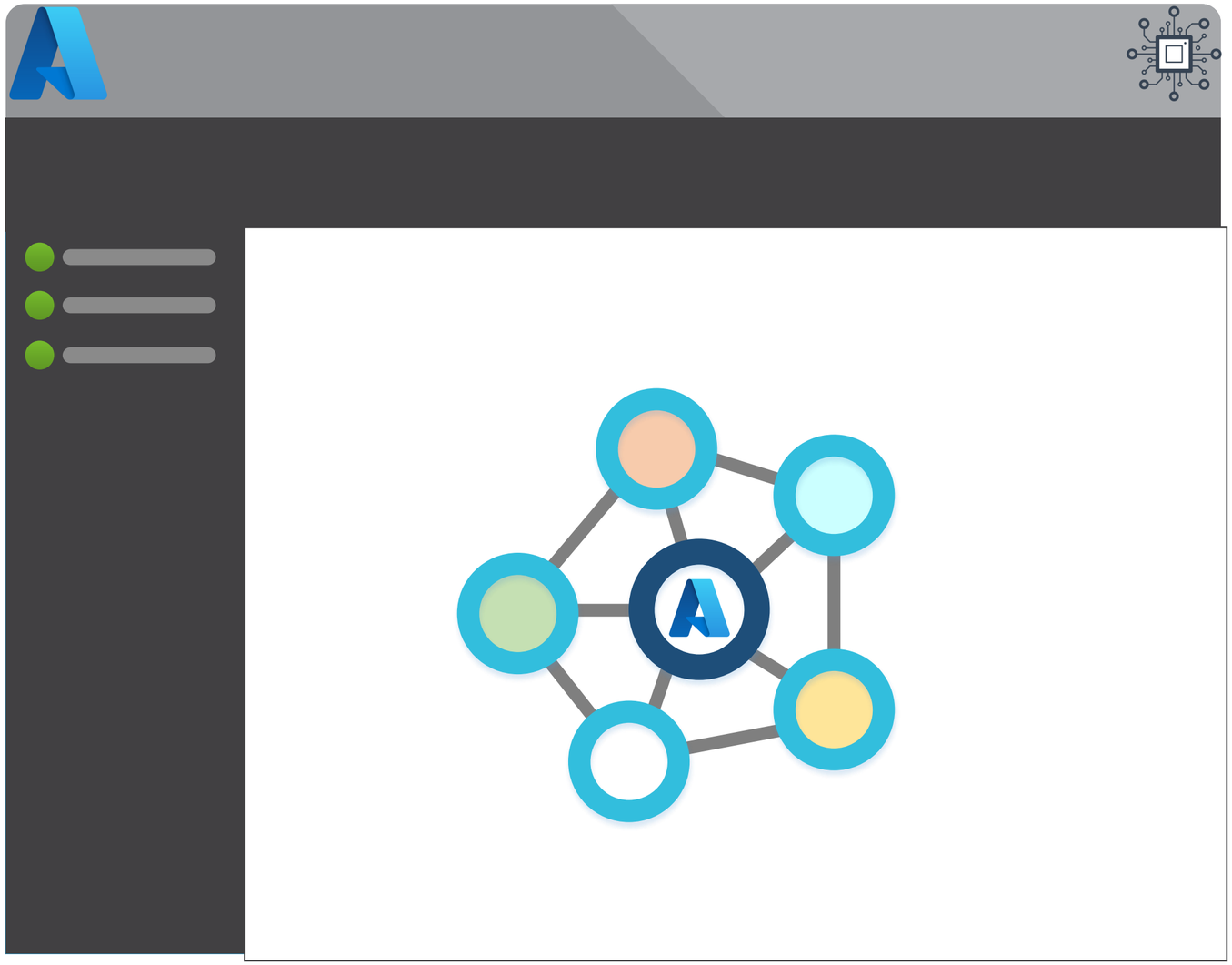


Data Mesh

- Zhamak Dehghani

@zhamakd

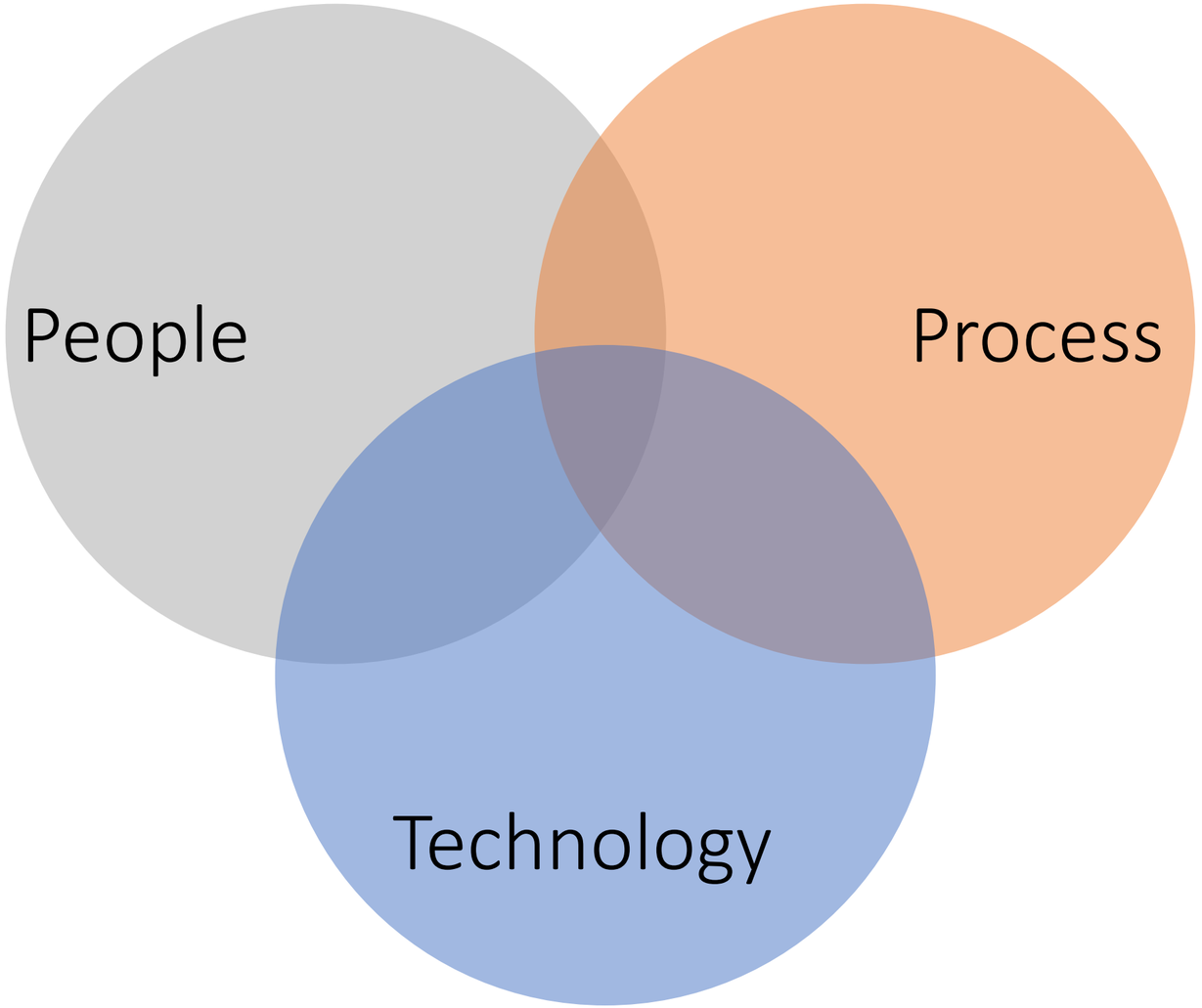




1. Domain-oriented decentralised data ownership and architecture.
2. Data as a product.
3. Self-serve data infrastructure as a platform.
4. Federated computational governance.

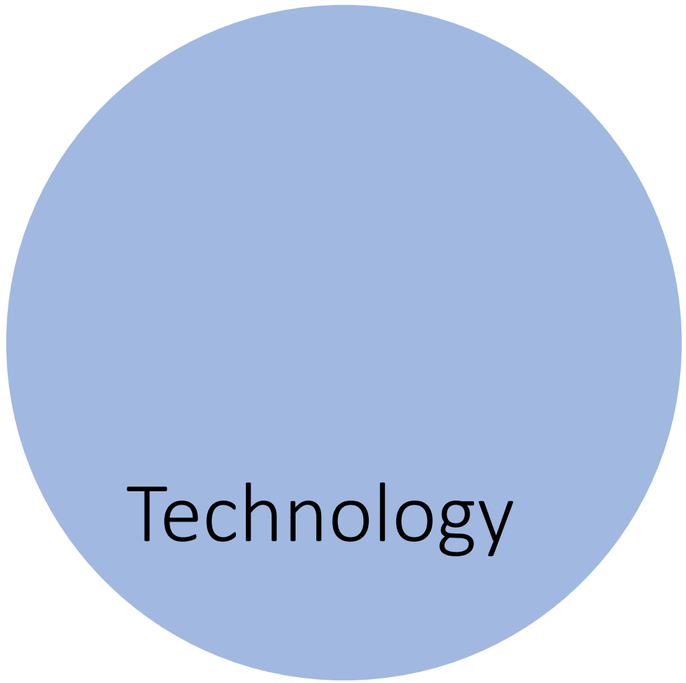
Introducing the Data Mesh

- Culture
- Personas
- Structure
- Owners
- Teams
- Skills
- Operations
- Guilds

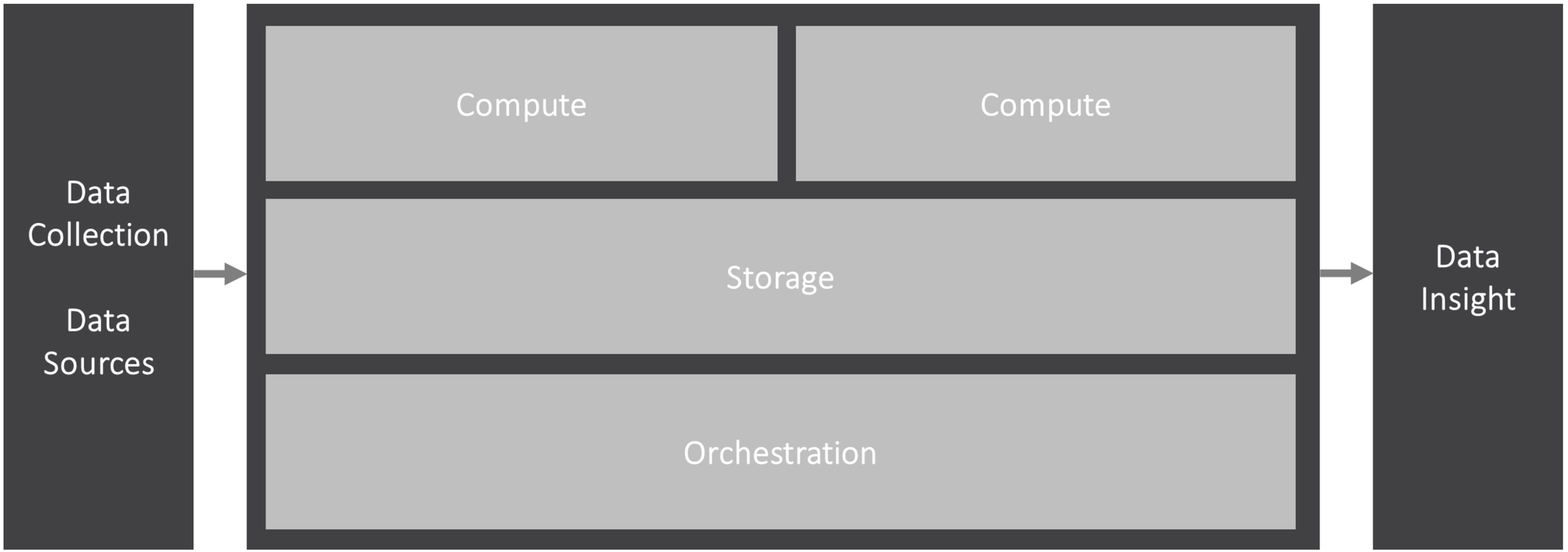


- Works of Working
- Agility
- Governance
- Workloads
- Requirements Gathering
- Maturity
- Frameworks
- Time Zones

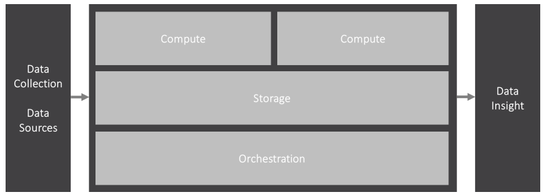
Introducing the Data Mesh

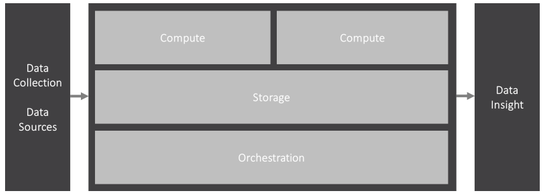


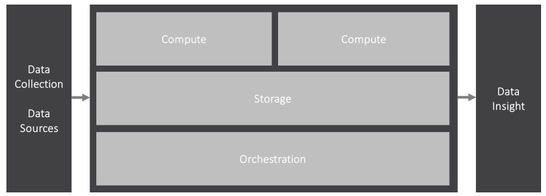
Data Products



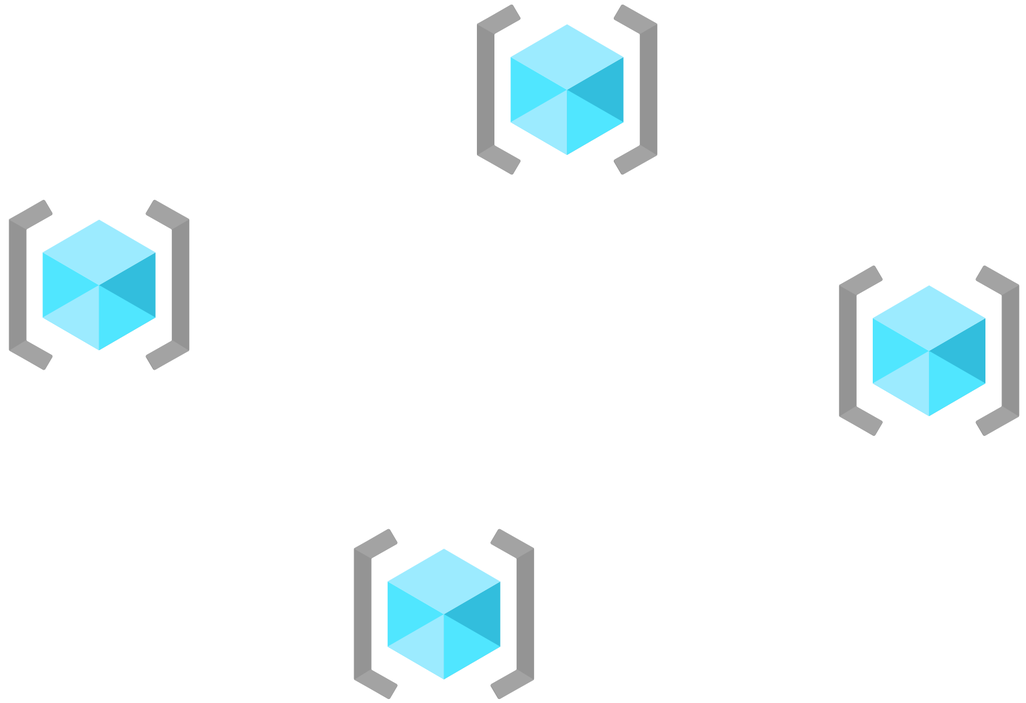
Data Products in Azure



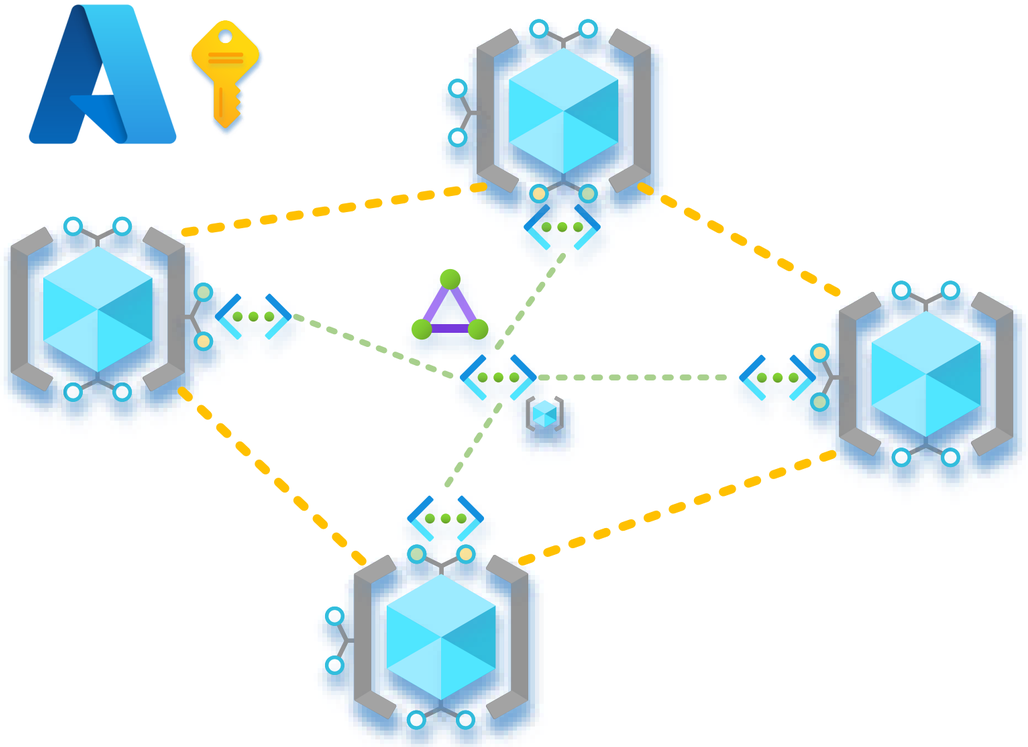




Data Products in Azure



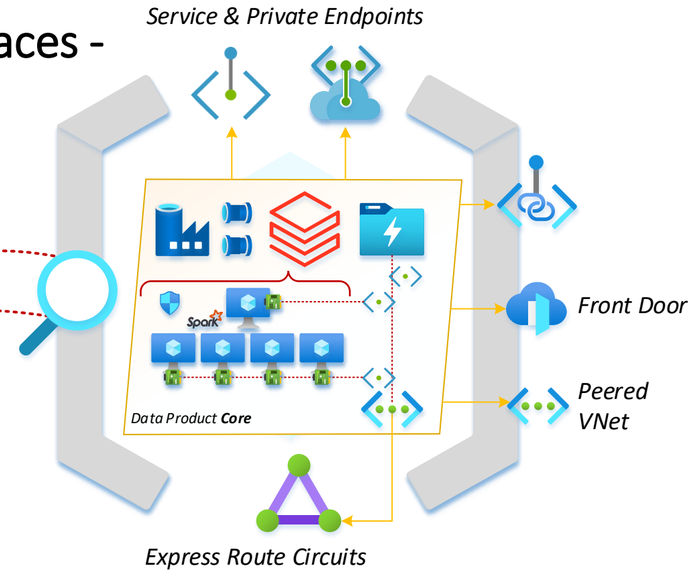
Data Products in Azure with Interfaces

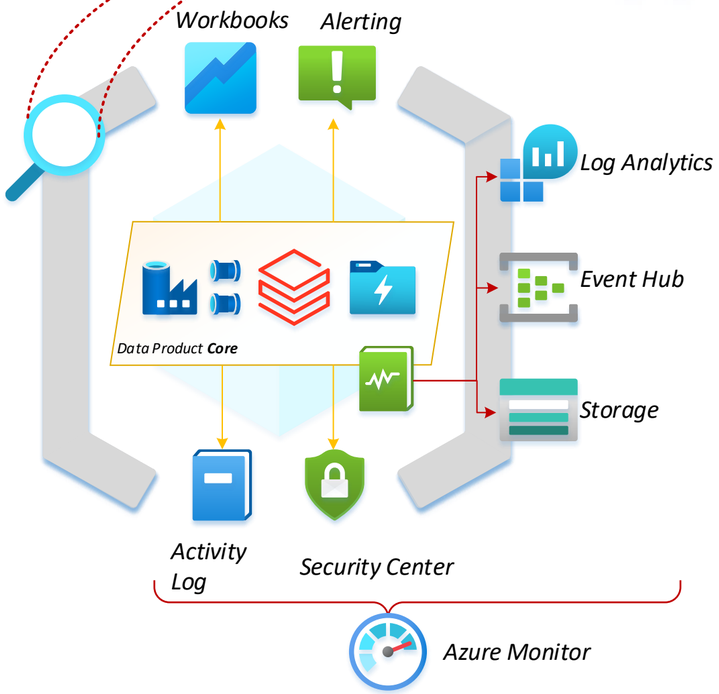


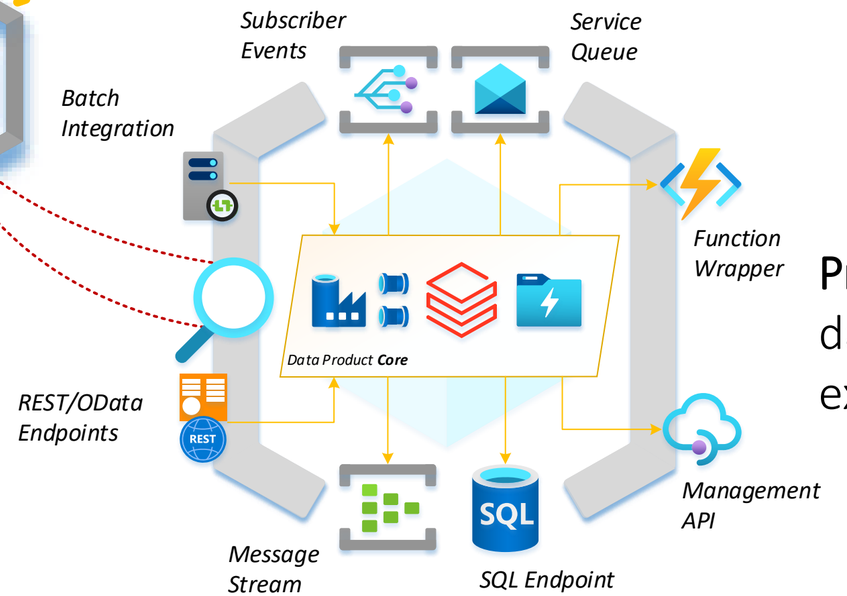
Data Products in Azure with Interfaces

Secondary Interfaces - operational reporting and logging.

Tertiary Interfaces - resource connectivity.

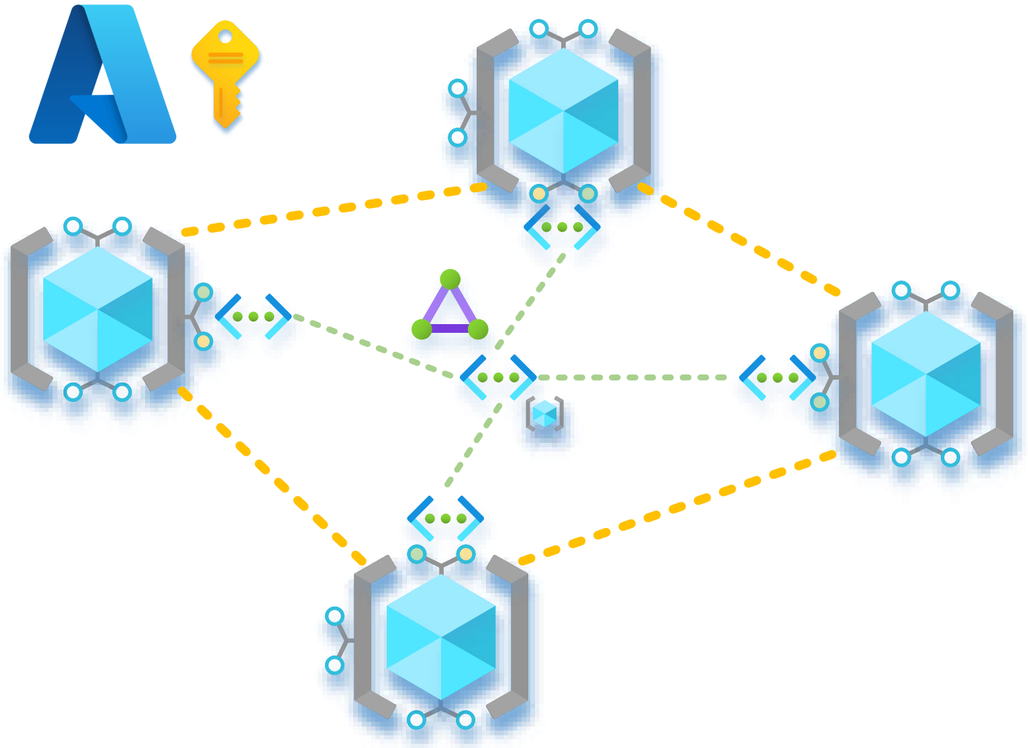




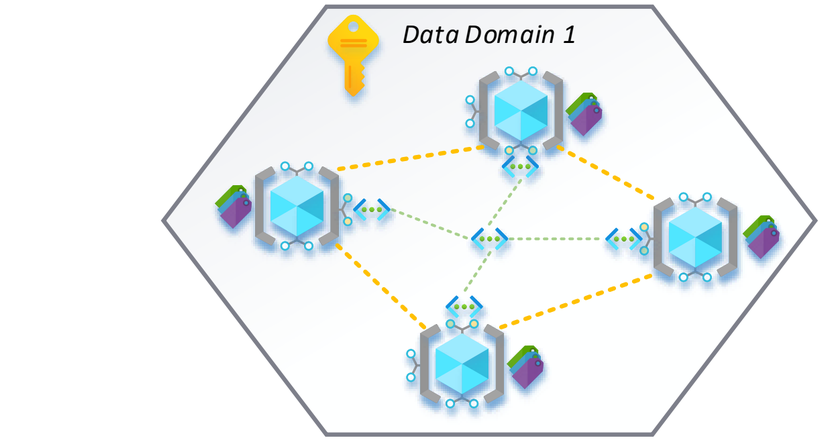


Primary Interfaces – data integration and exchange.

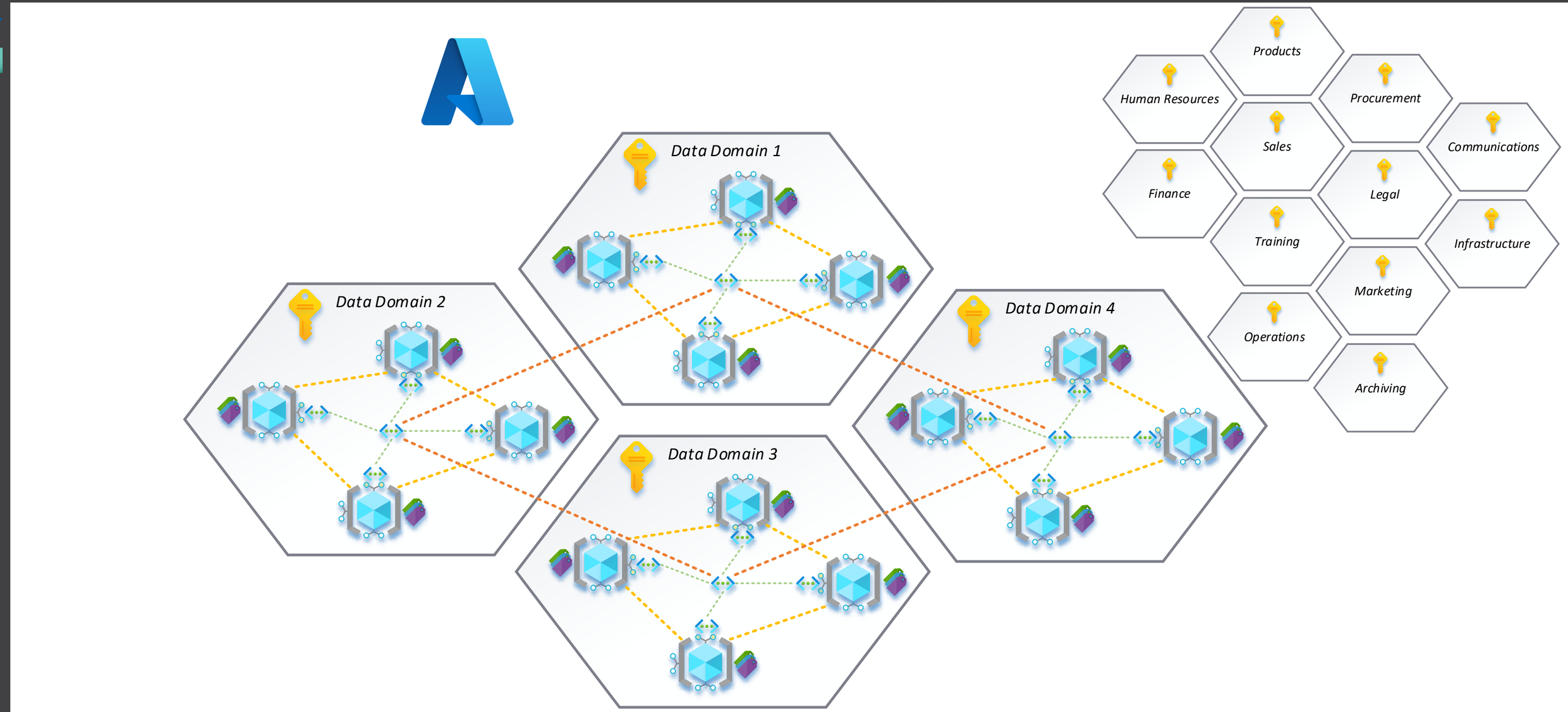
Data Domains in Azure



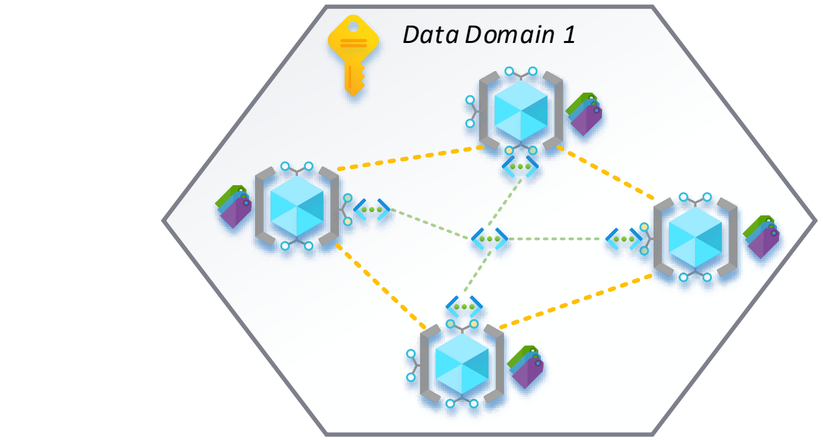
Data Domains in Azure

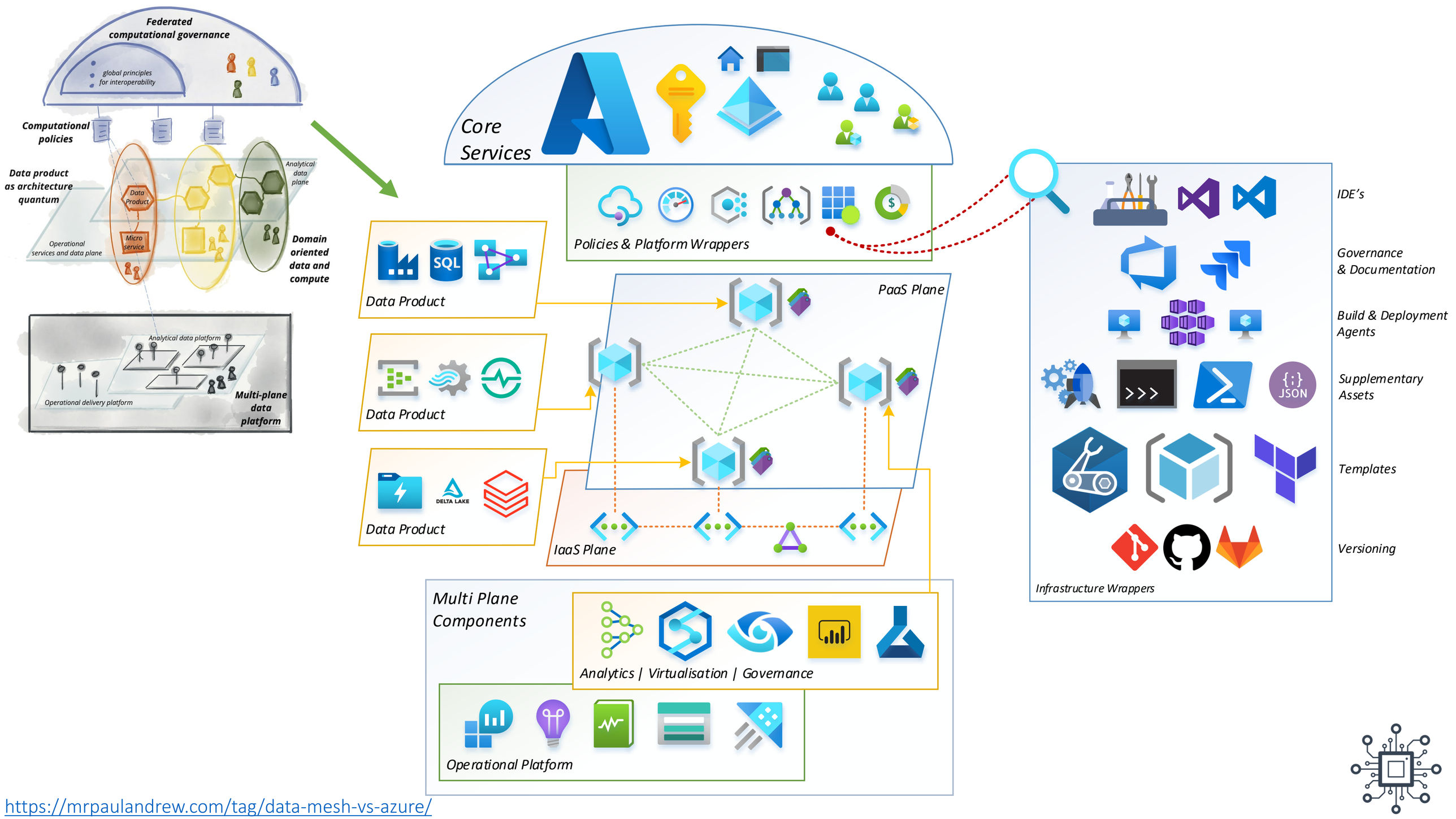


Data Domains in Azure



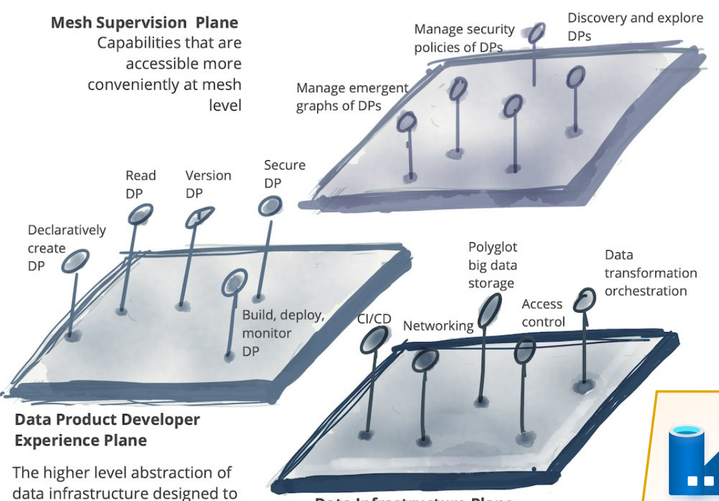
Data Domains in Azure





https://mrpaulandrew.com/tag/data-mesh-vs-azure/

Mesh Supervision Plane
Capabilities that are accessible more conveniently at mesh level



Data Product Developer Experience Plane

The higher level abstraction of data infrastructure designed to support the common data product developer journey

Data Infrastructure Plane
Providing the underlying infrastructure required to build, run, monitor data products

Core Services

Policies & Platform Wrappers

Data Product

Data Product

Data Product

PaaS Plane

IaaS Plane

SaaS Plane

PaaS Plane

IaaS Plane

Multi Plane Components

Analytics | Virtualisation | Governance

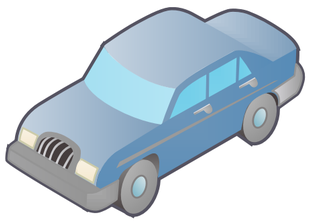
Operational Platform

The Evolution of Data Platform Architectures in Azure

Lambda, Kappa, Delta, Data Mesh

… should we be considering a solution that offers all these principals in a nirvana of data insight perfection?

Yes!



1. Domain-oriented decentralised data ownership and architecture.
2. Data as a product.
3. Self-serve data infrastructure as a platform.
4. Federated computational governance.

1. Don’t centralise everything.
2. Don’t create silo’s.
3. Abstract specialised skills.
4. Don’t create monolithic systems.

Thank you for listening…

Paul Andrew

Microsoft® MVP Most Valuable Professional

avanade

Mr Paul Andrew Consulting Ltd

Blog: mrpaulandrew.com
YouTube: c/mrpaulandrew
Email: paul@mrpaulandrew.com

Twitter: @mrpaulandrew
LinkedIn: In/mrpaulandrew

GitHub: github.com/mrpaulandrew

Contact Details