

The Evolution of Data Platform Architectures in Azure

Lambda, Kappa, Delta, Data Mesh

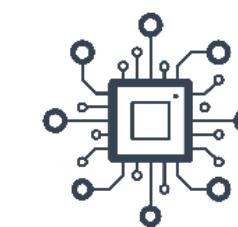
λ

κ

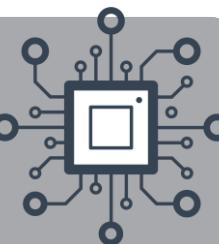
δ



Paul Andrew | Technical Architect in Azure CoE



Mr Paul Andrew
Consulting Ltd



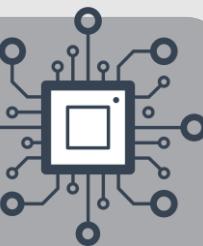
<https://github.com/mrpaulandrew>

CommunityEvents

Demo code, content and slides from various community events.

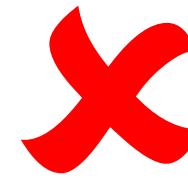


[{Event/Location}-{Month}-{Year}](#)



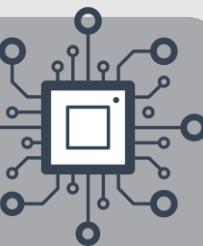
What is the answer to life, the universe and everything?

Answer:
42



Answer:
It depends!





What is big data?

Answer:

It depends!



Answer:

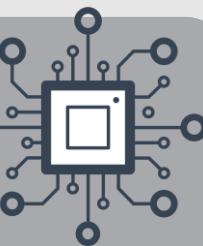
“Any data that you cannot process
in the time that you have/want
using the technology you have.”



- Buck Woody

@BuckWoodyMSFT

Volume
Velocity
Variety
Veracity
Value



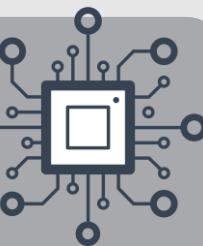
What is the goal of our data solutions?

Data
Collection

Data
Sources

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

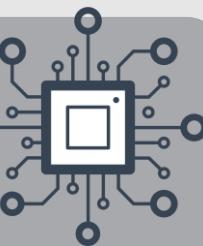
Data
Insight



What is the goal of our data solutions?



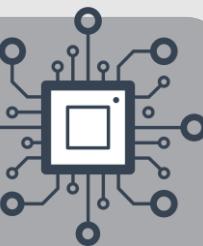
Data = Information = Knowledge = Power/Insights



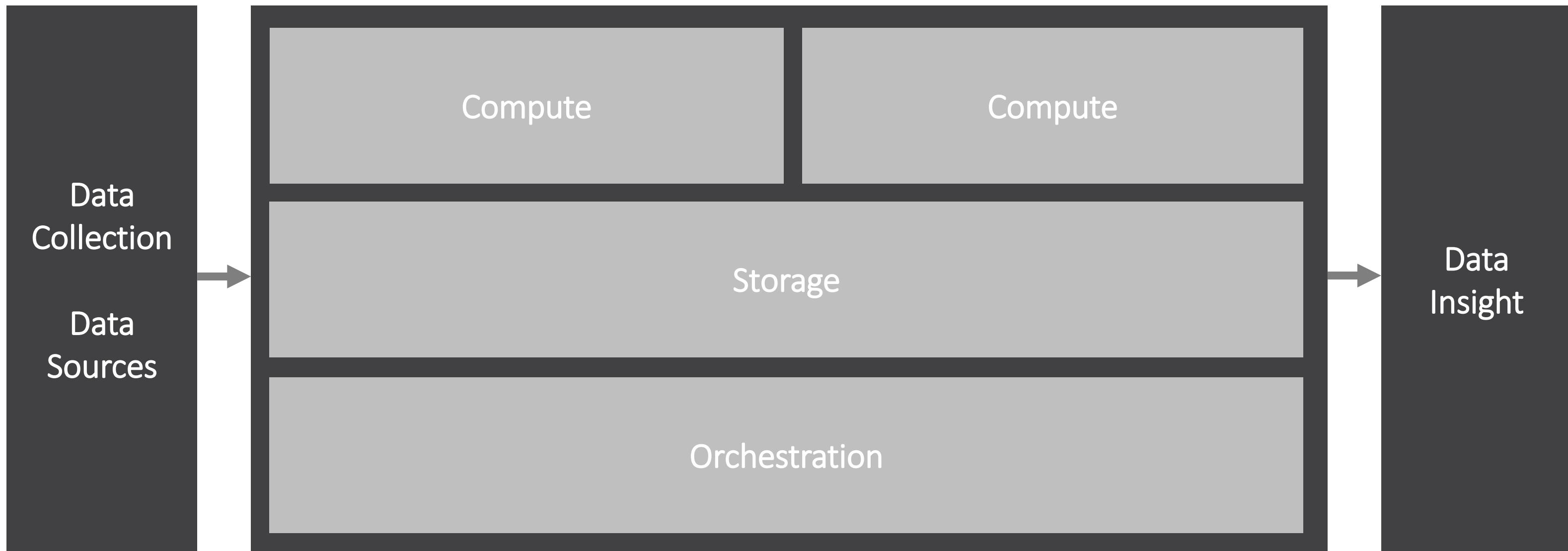
What is the goal of our data solutions?



Data = Information = Knowledge = Power/Insights



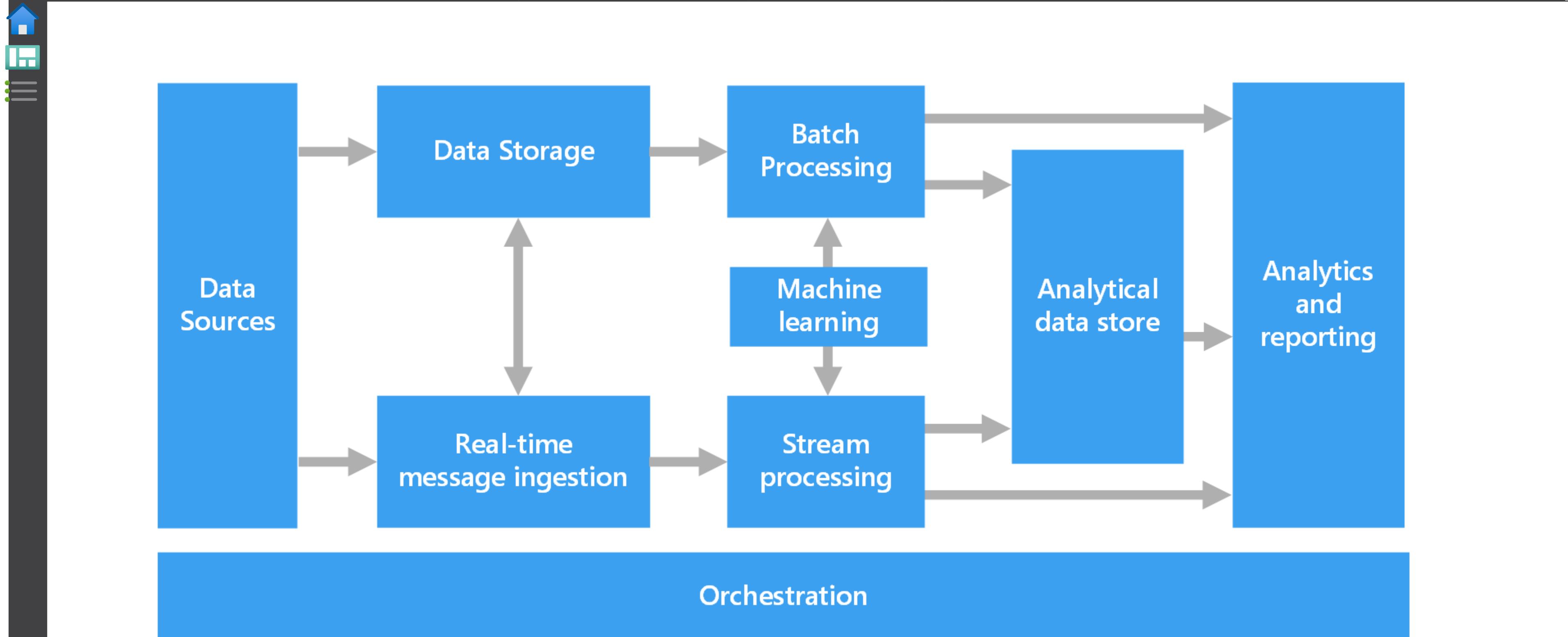
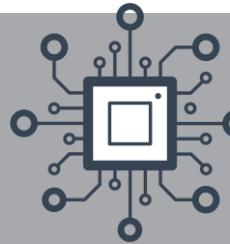
What is the goal of our data solutions?



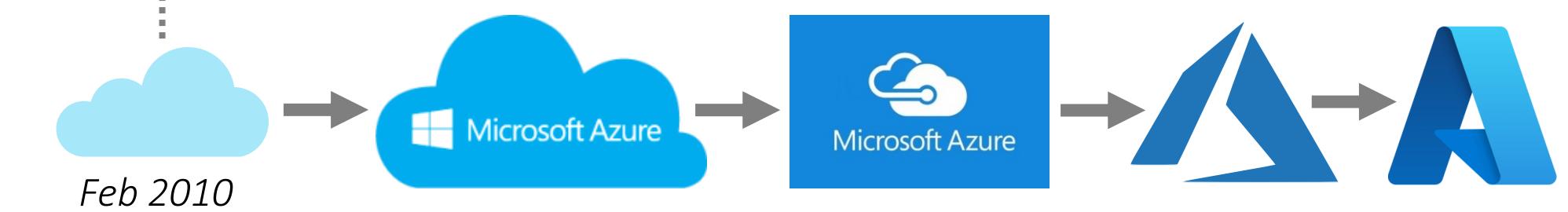
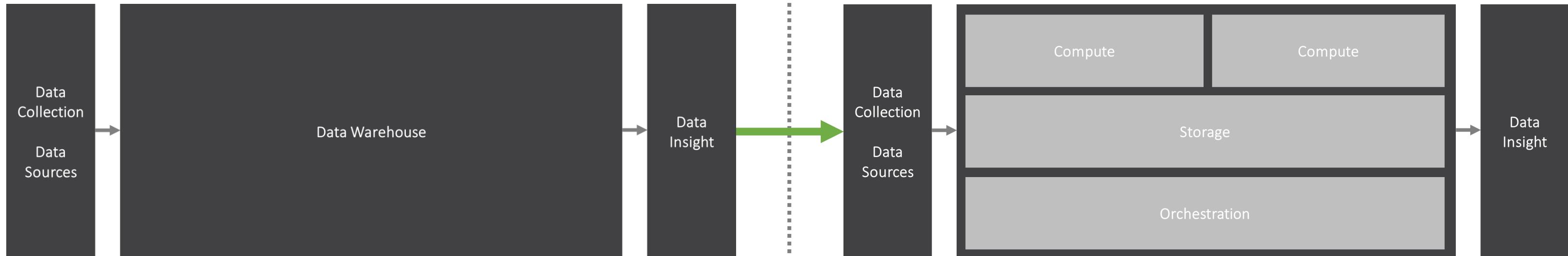
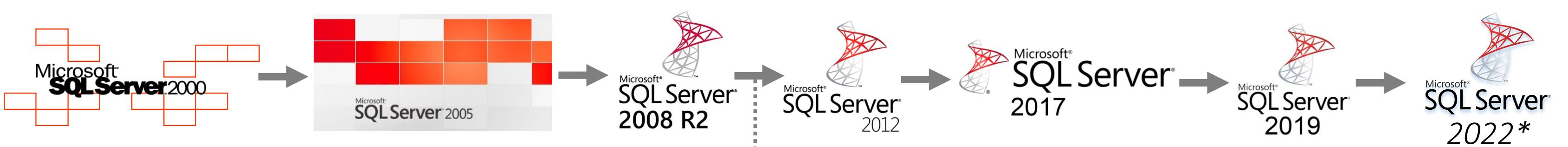
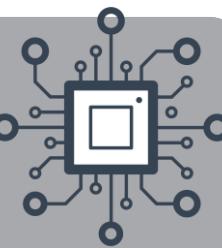
Data = Information = Knowledge = Power/Insights



Microsoft's Components of a Big Data Architecture

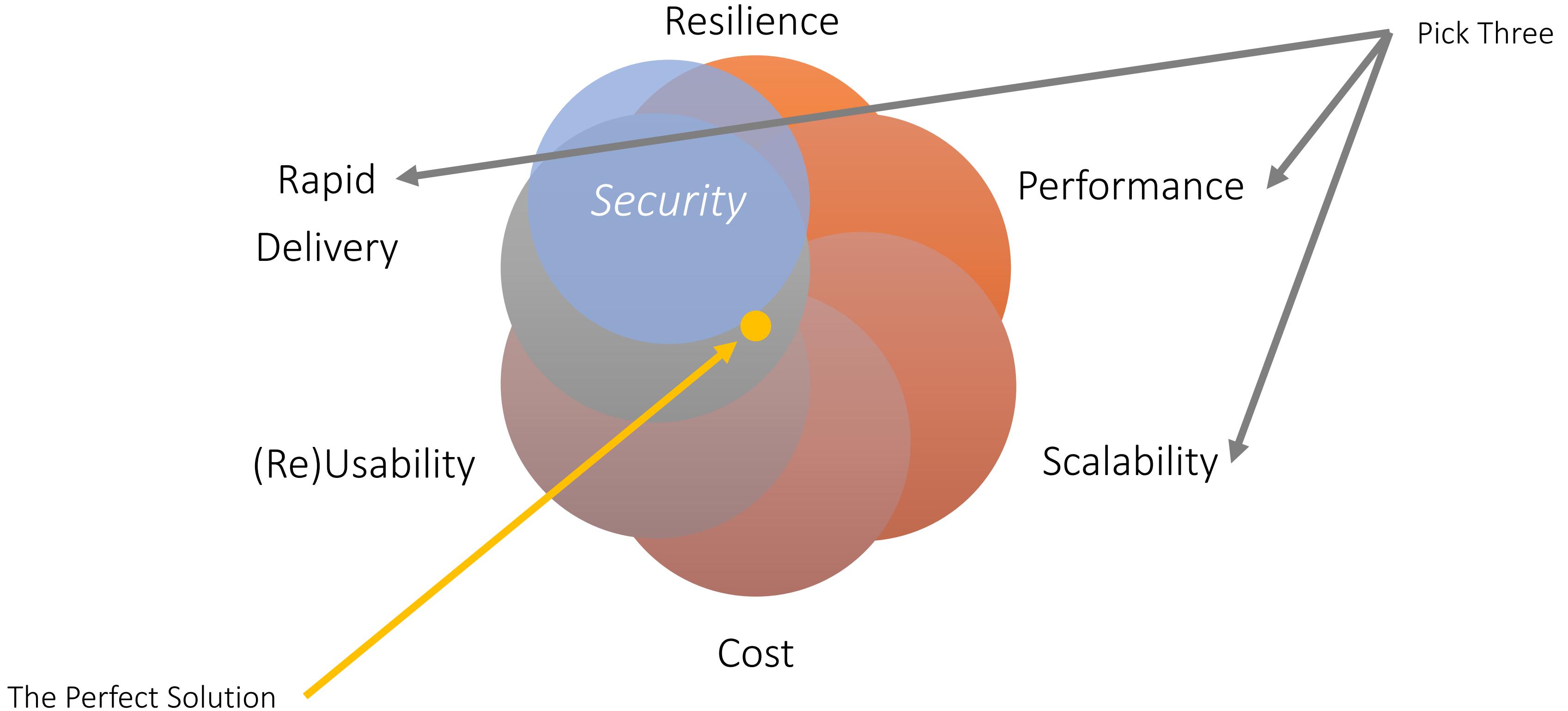
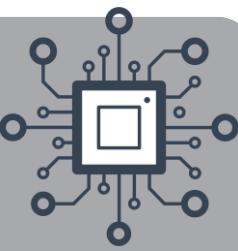


Data Platform Evolution



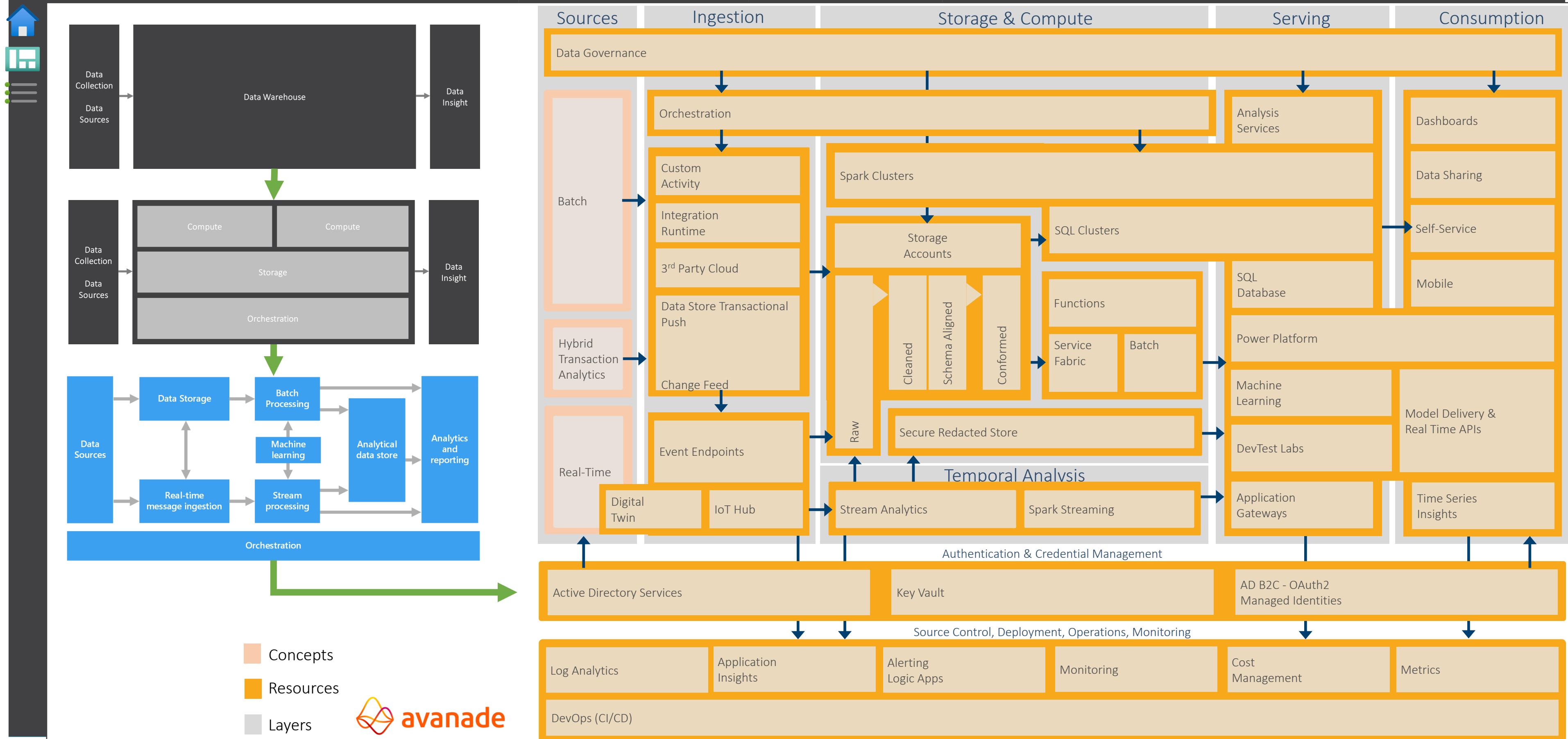
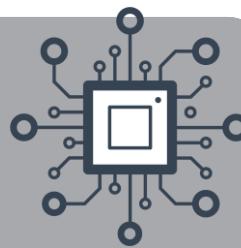


What is our primary design focus?



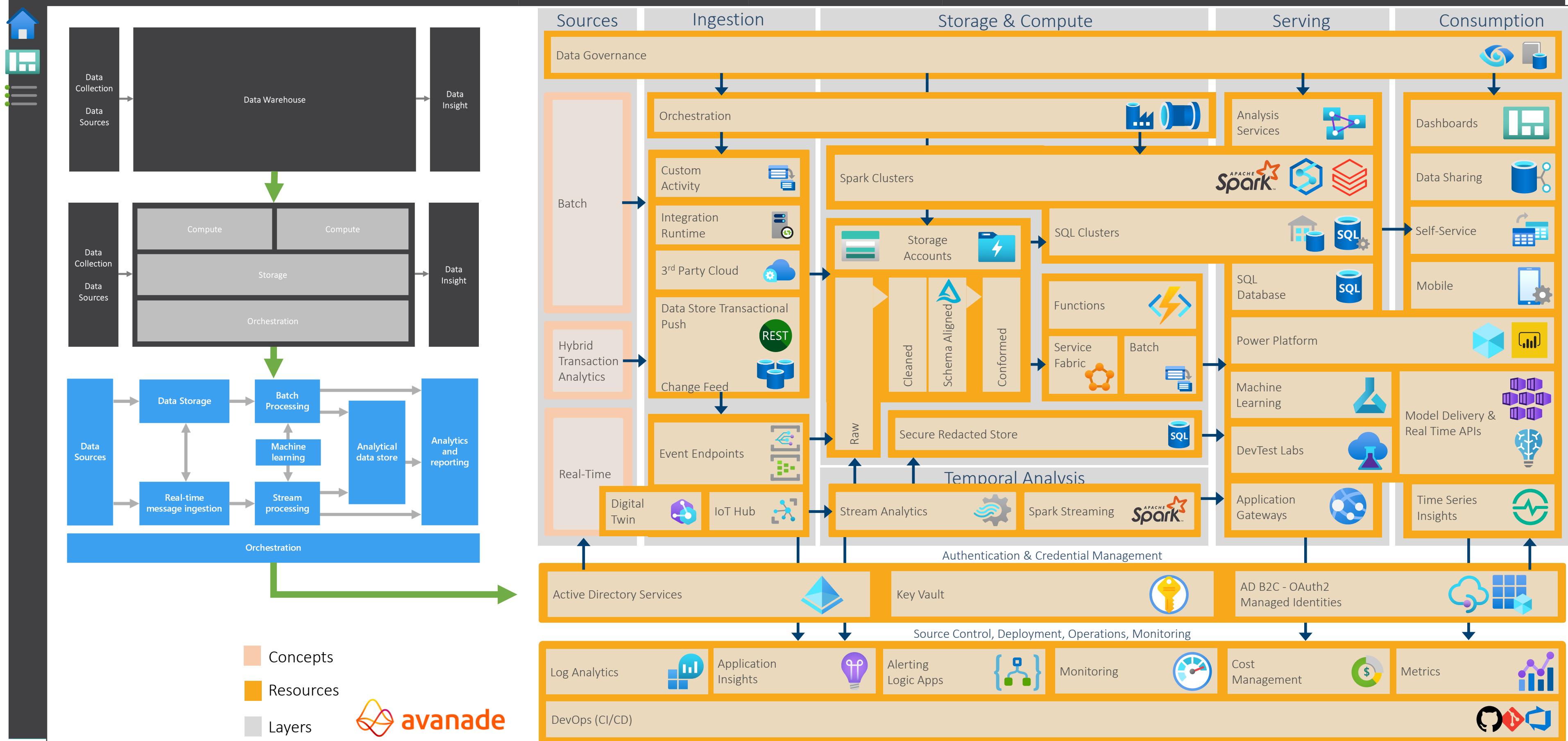
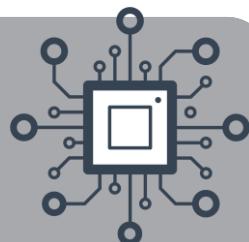


A Logical Data Architecture





A Logical Data Architecture



Delta* Lake

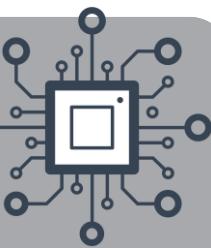
Delta.io 



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

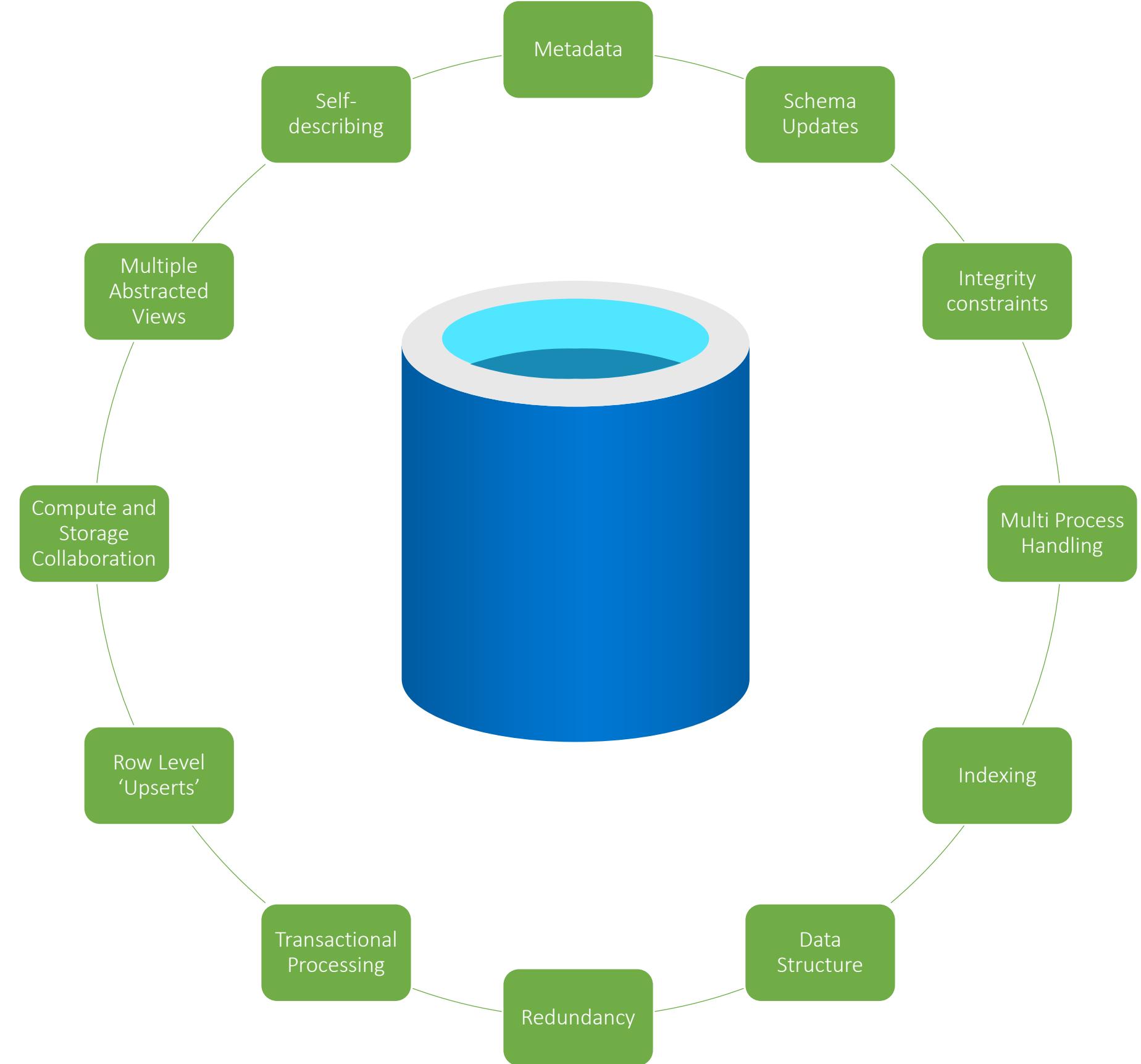


Databases



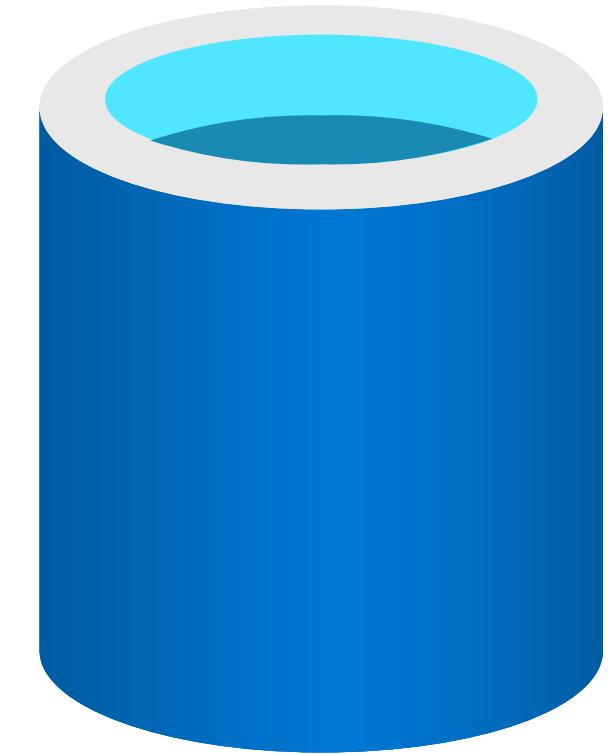
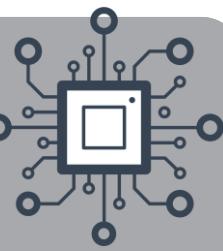
DataBase Management System

Atomicity
Consistency
Isolation
Durability



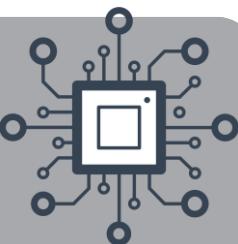


Databases

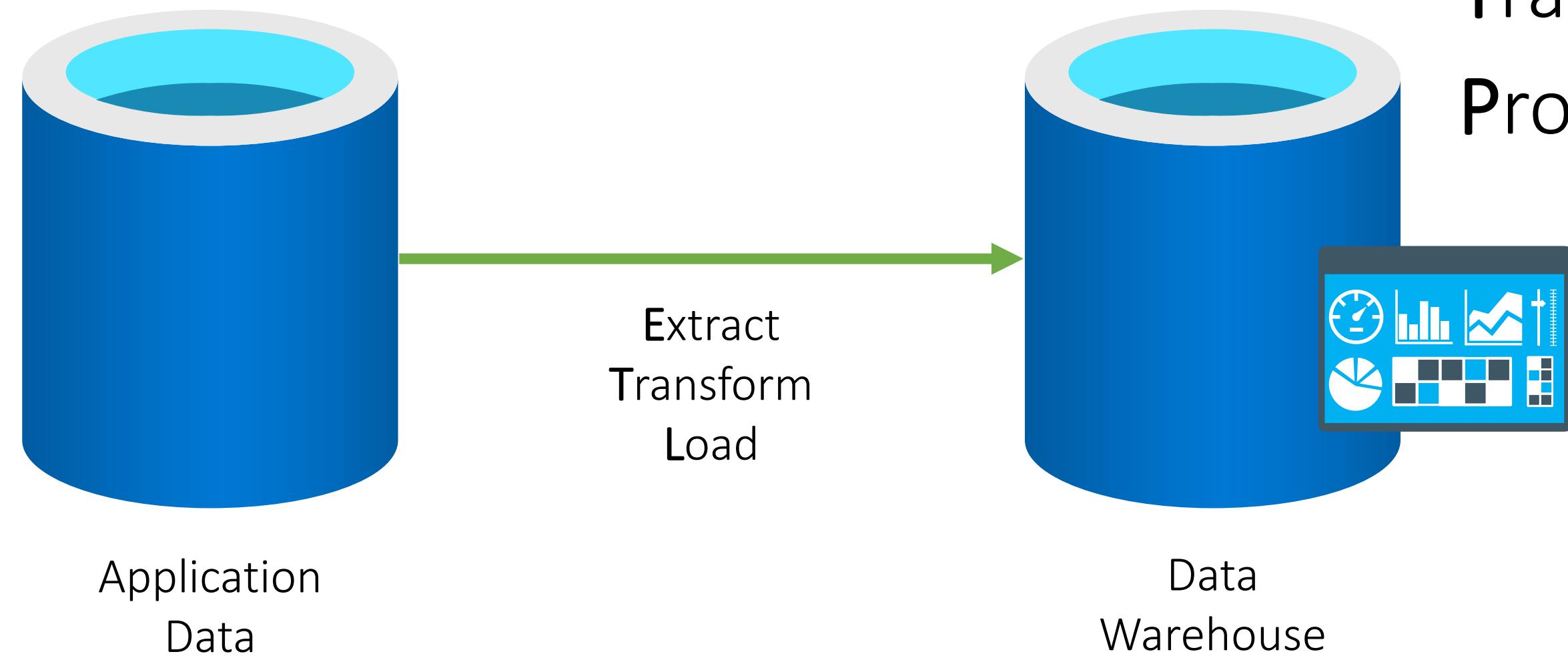




Creating a Data Warehouse



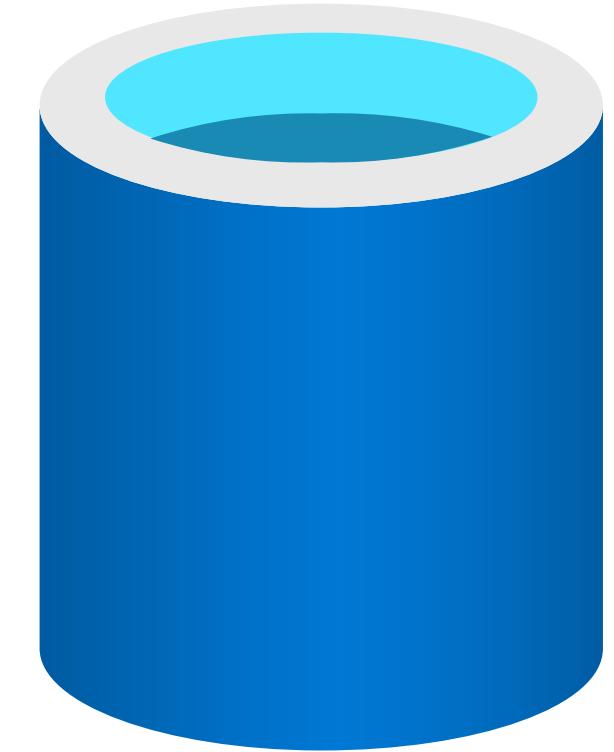
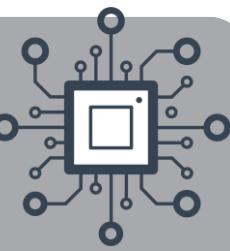
Online
Line
Transactional
Processing



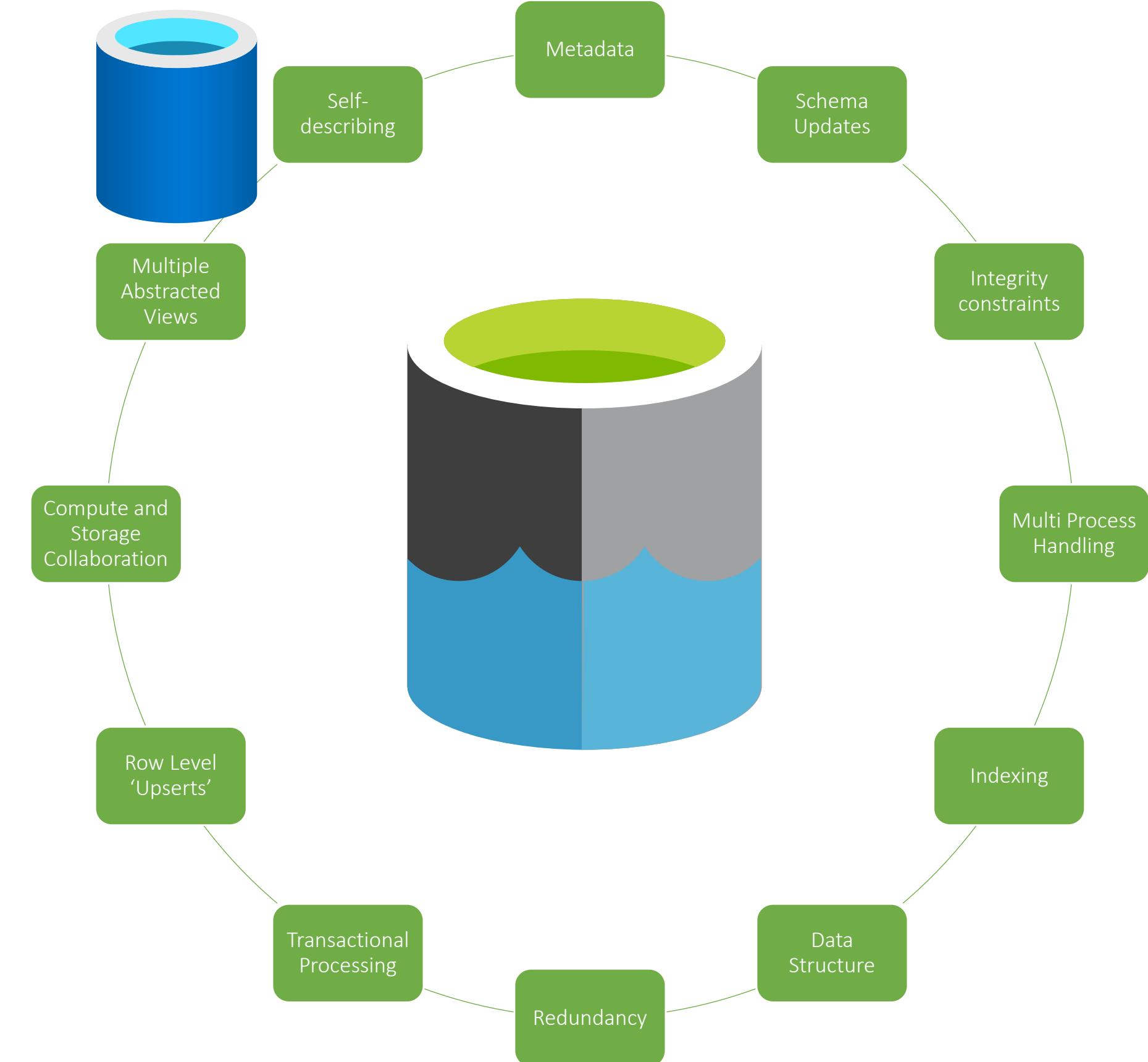
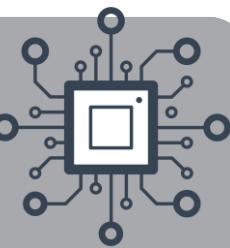
Offline
Analytical
Transactional
Processing



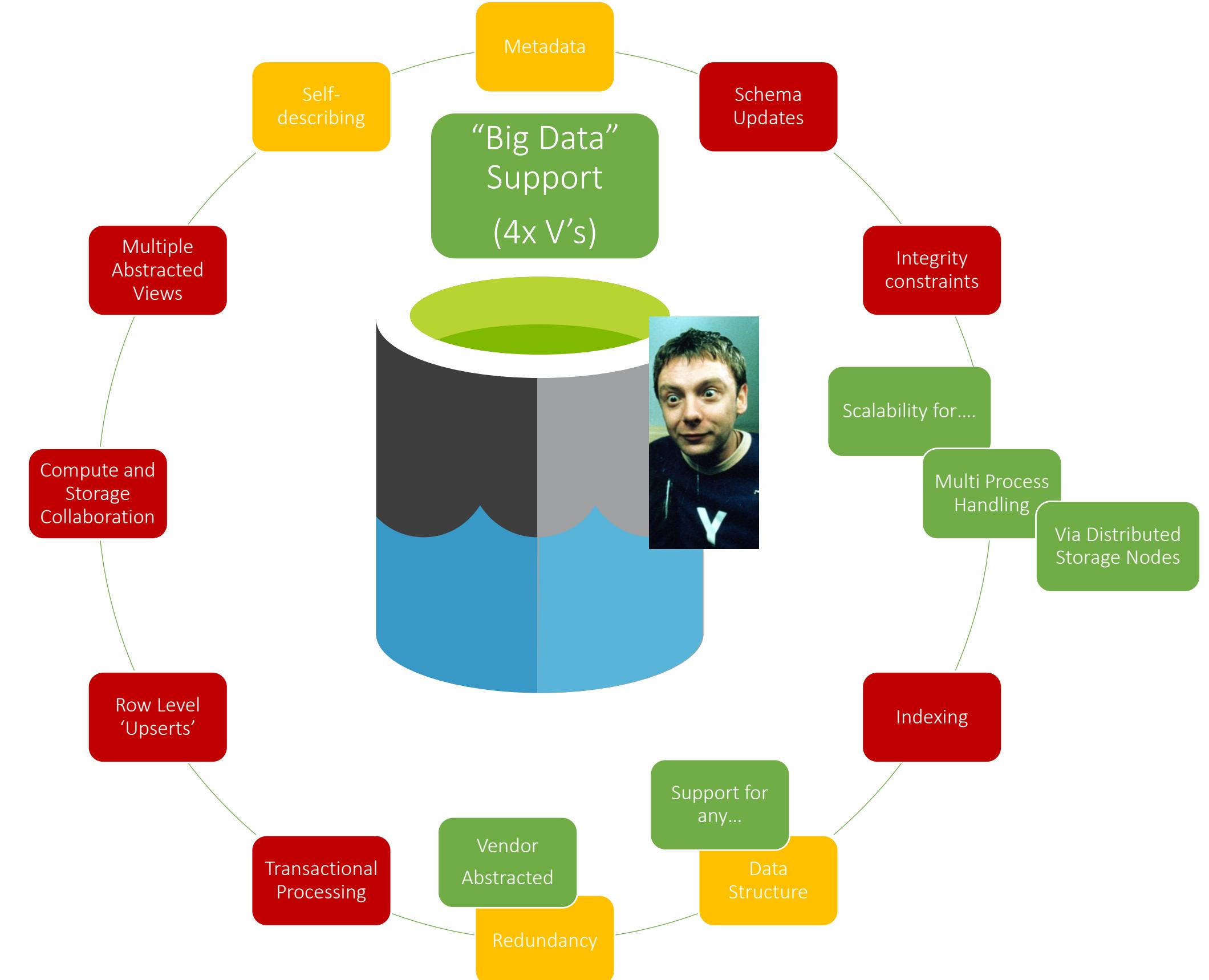
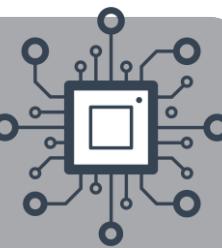
Databases



Data Lakes

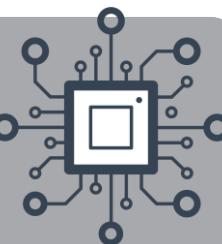


Data Lakes





Problem Summary

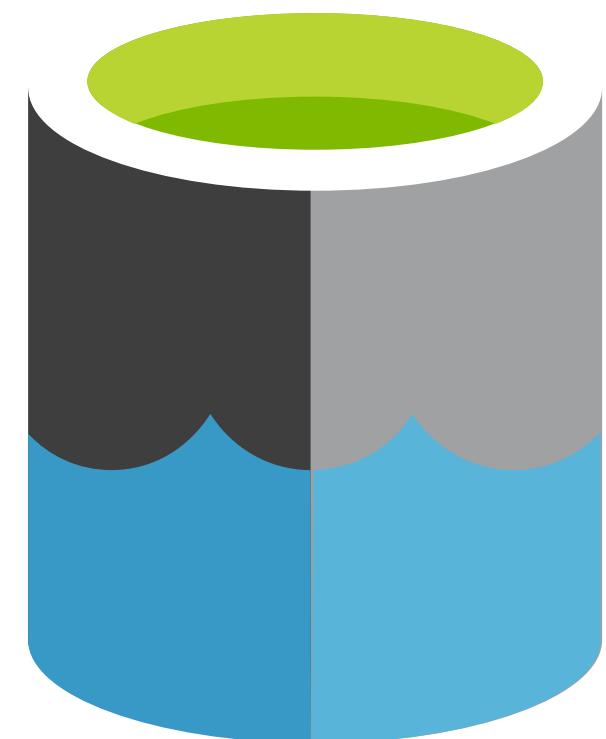


Data Lakes are good, but they still lack some of the basic ACID functionality needed for data processing.

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

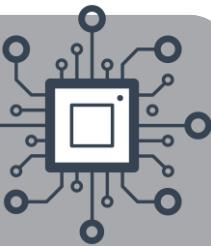


VS



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

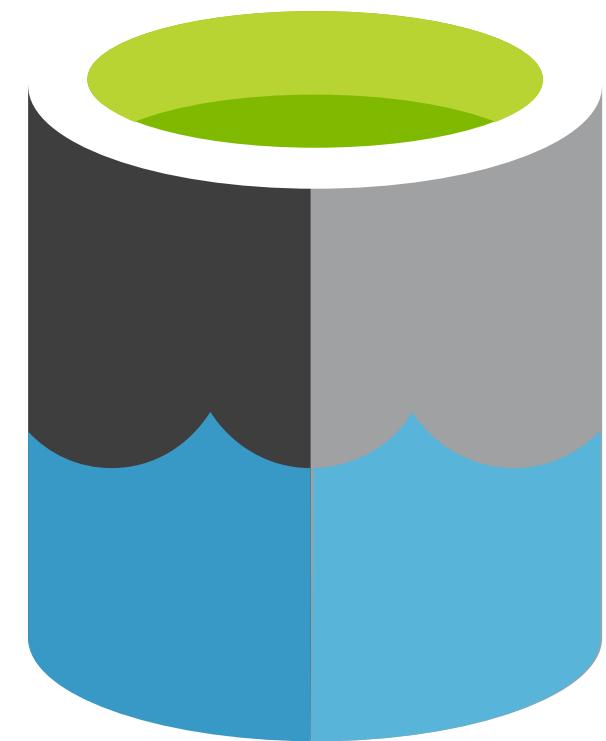


Data Lakes are good, but they still lack some of the basic ACID functionality needed for data processing.

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



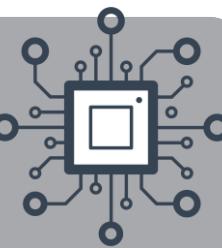
VS



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

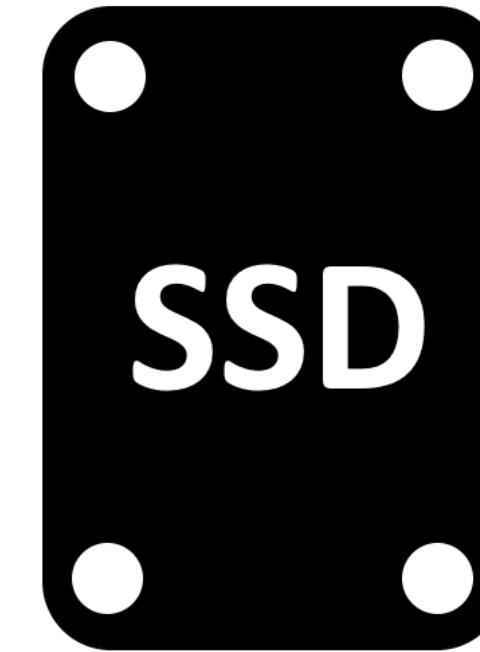


Solution

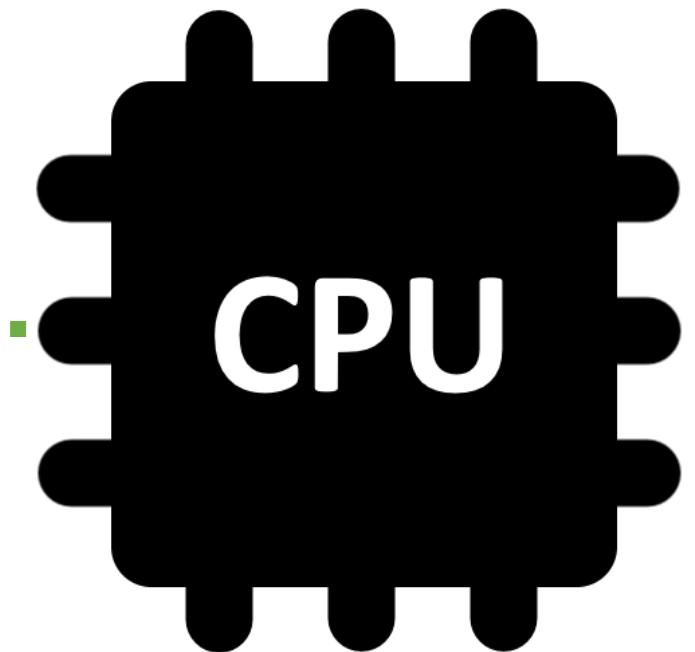


Enable ACID transactional support for Data Lakes...

Storage



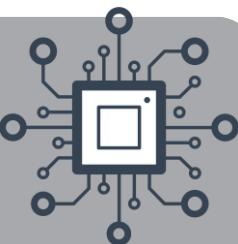
Compute



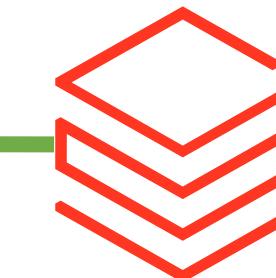
Storage & Compute ~~Decoupled~~ Working Together Again As Friends!



ACID Data Frameworks for Data Lakes



DELTA LAKE™



databricks®

February 2019

ICEBERG 

NETFLIX



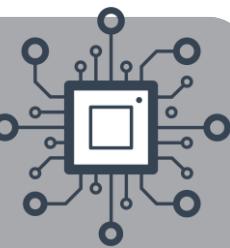
Apache

hoodie

Uber



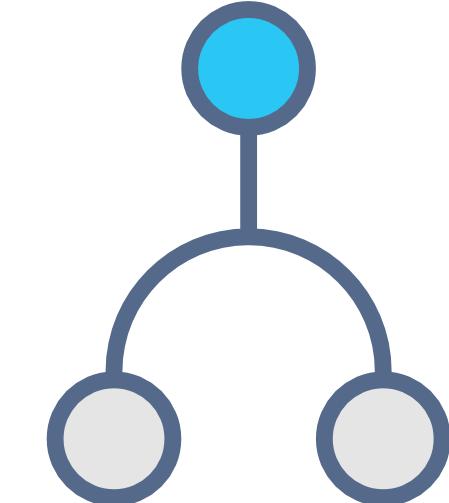
What is Delta Lake?



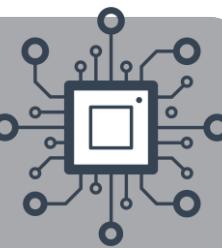
DELTA LAKE™



databricks®

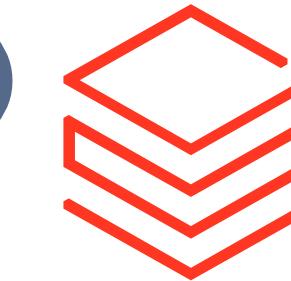
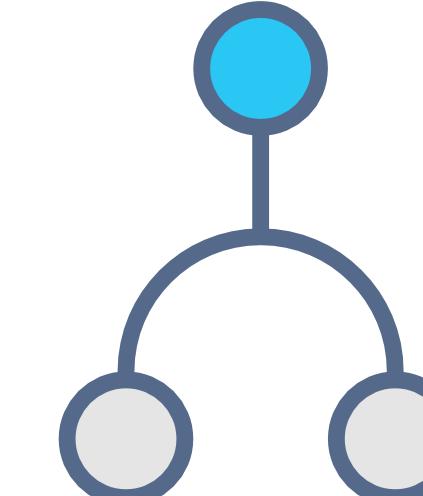
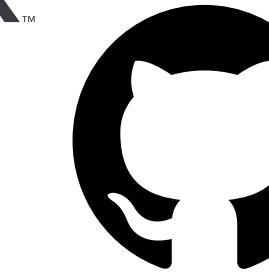


What is Delta Lake?



DELTA LAKE™

APACHE
Spark



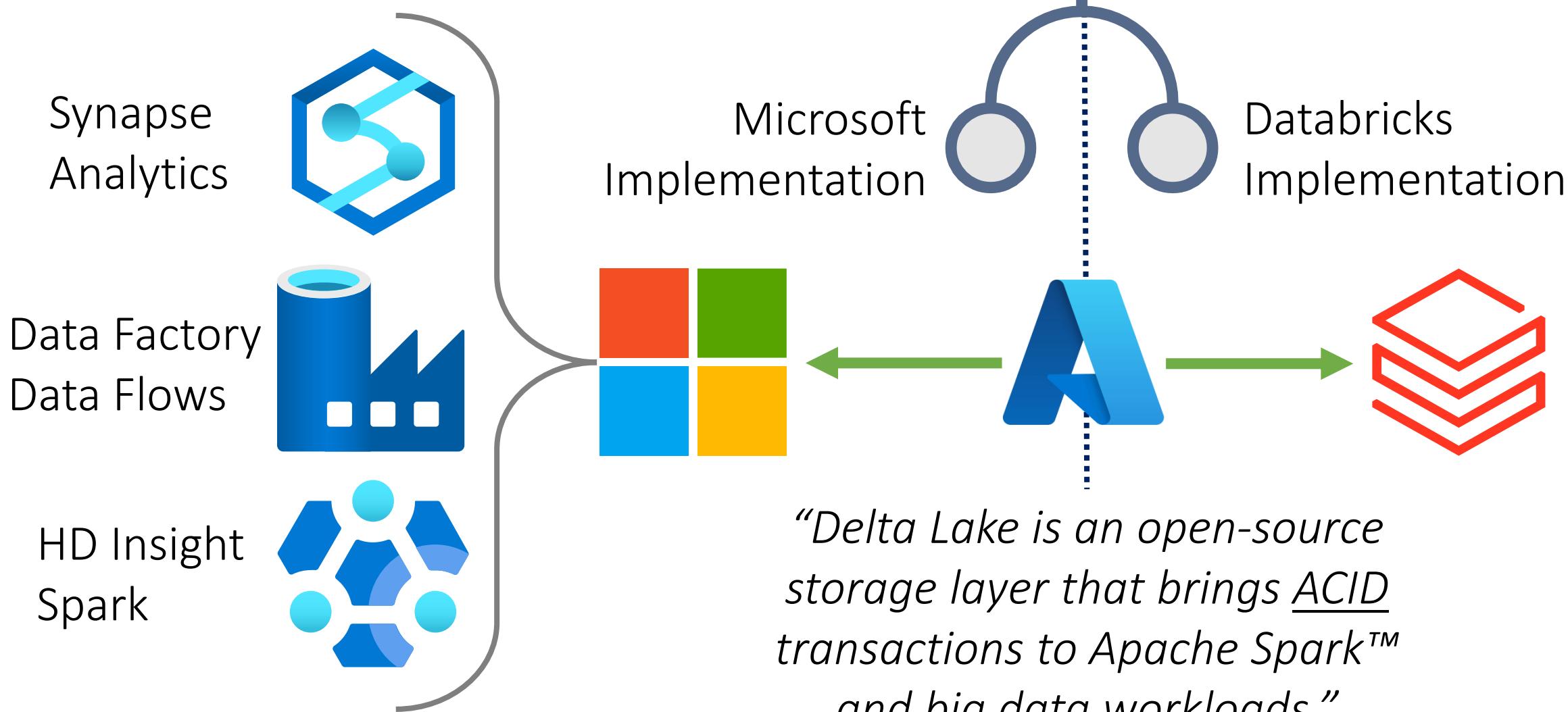
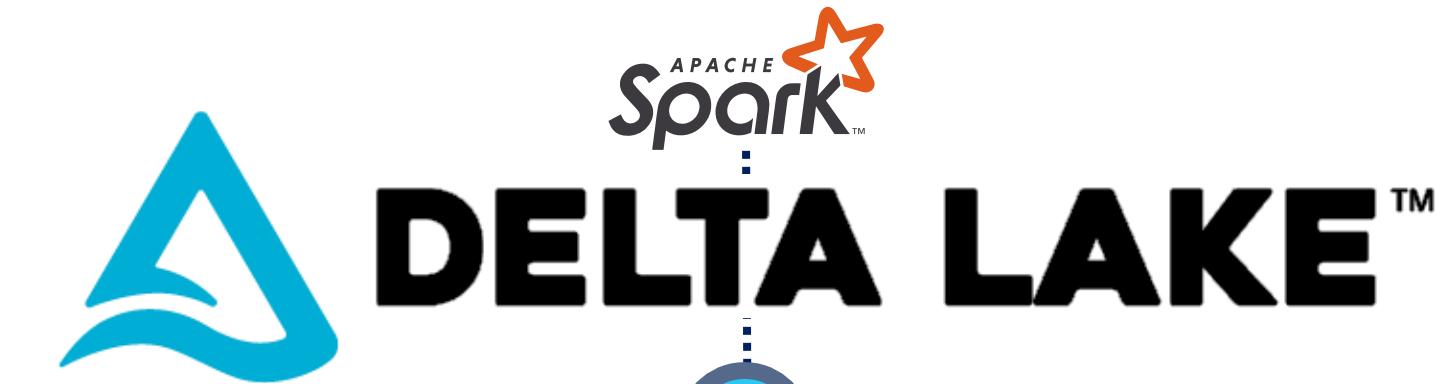
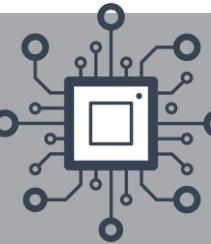
databricks®

<https://delta.io>

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

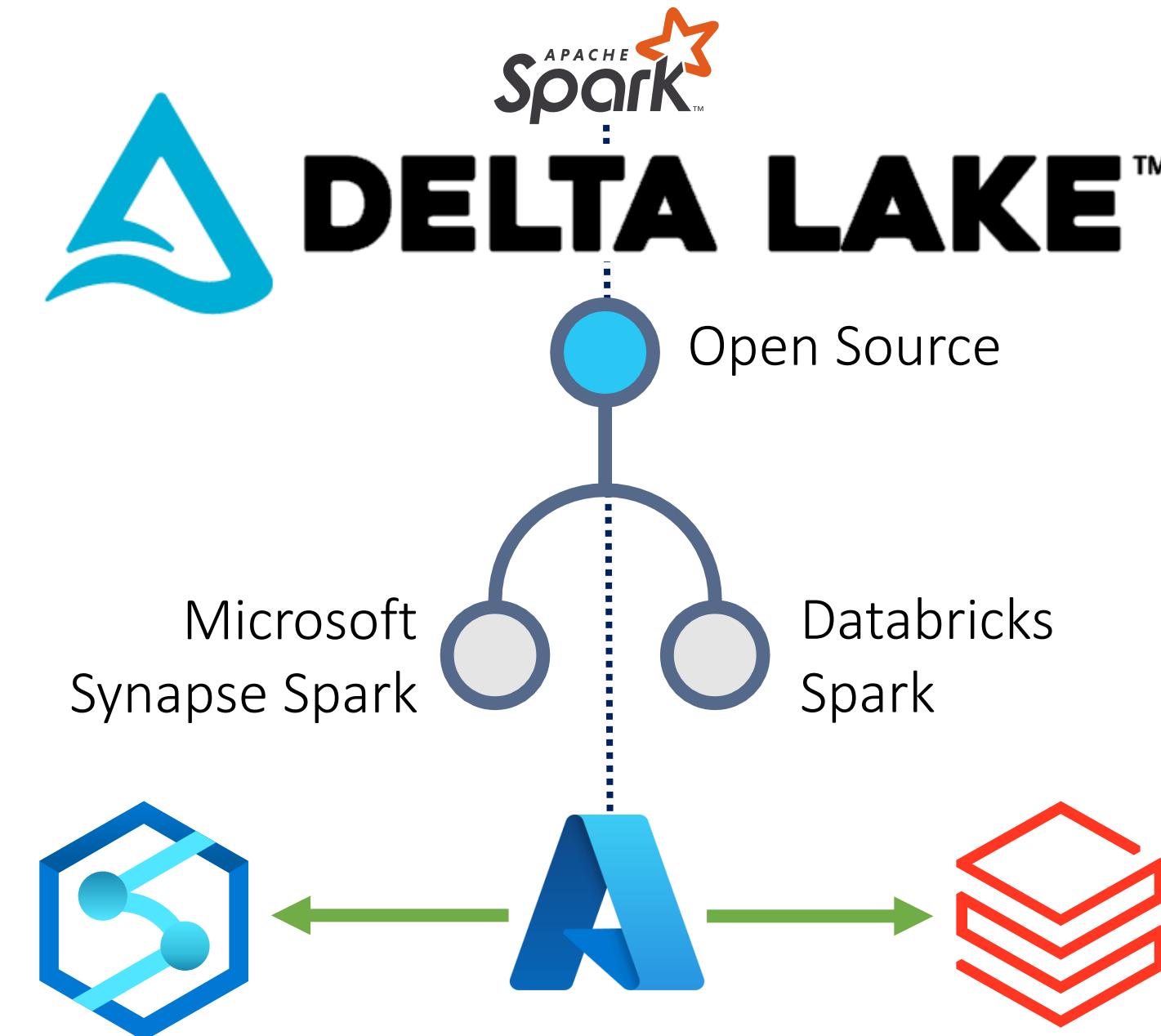
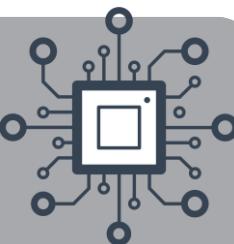
April 2019

What is Delta Lake?





Which Spark Implementation is Better?

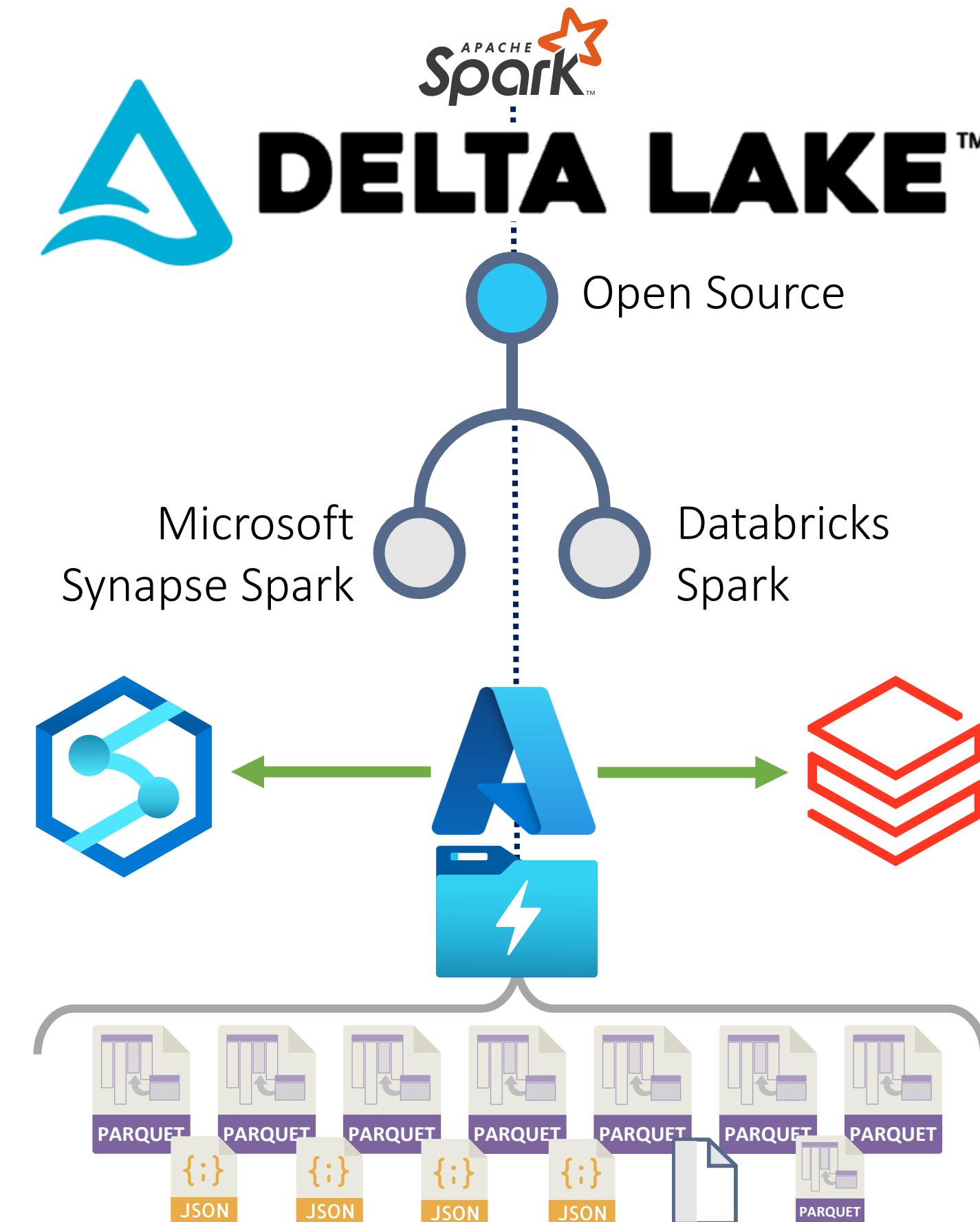
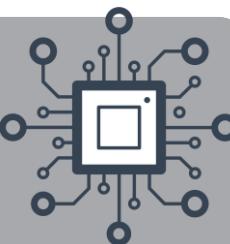


“Delta Lake is an open-source storage layer that brings ACID transactions to Apache Spark™ and big data workloads.”



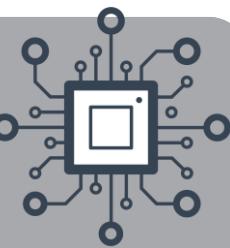


Which Spark Implementation is Better?





Lake House

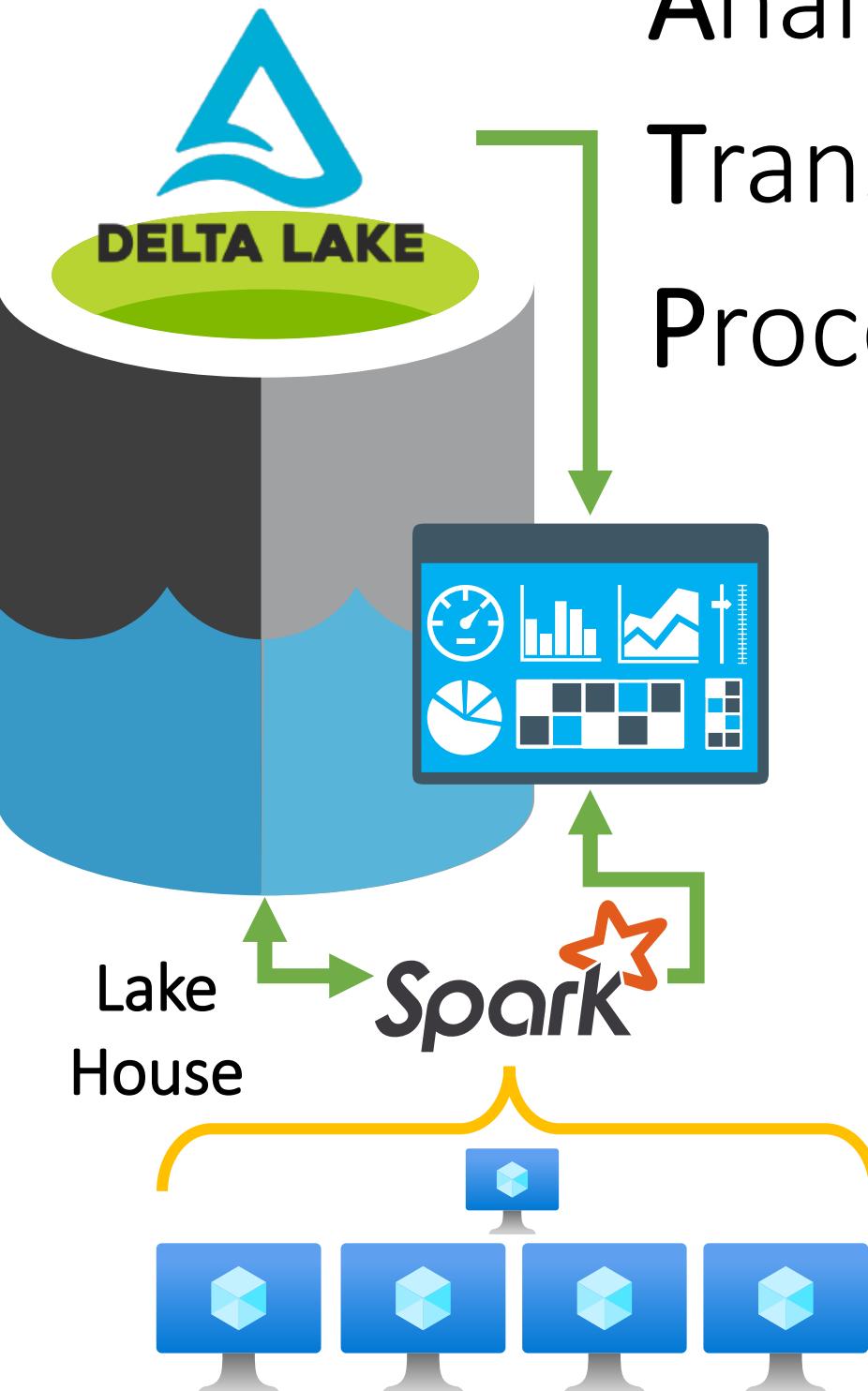


Online
Line
Transactional
Processing



Application
Data

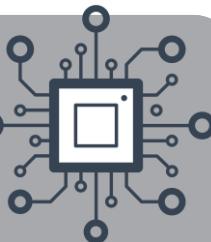
Extract
Transform
Load



Offline
Analytical
Transactional
Processing



Lake House



WIKIPEDIA
The Free Encyclopedia

Article Talk Read Edit View history Search Wikipedia

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 *II 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

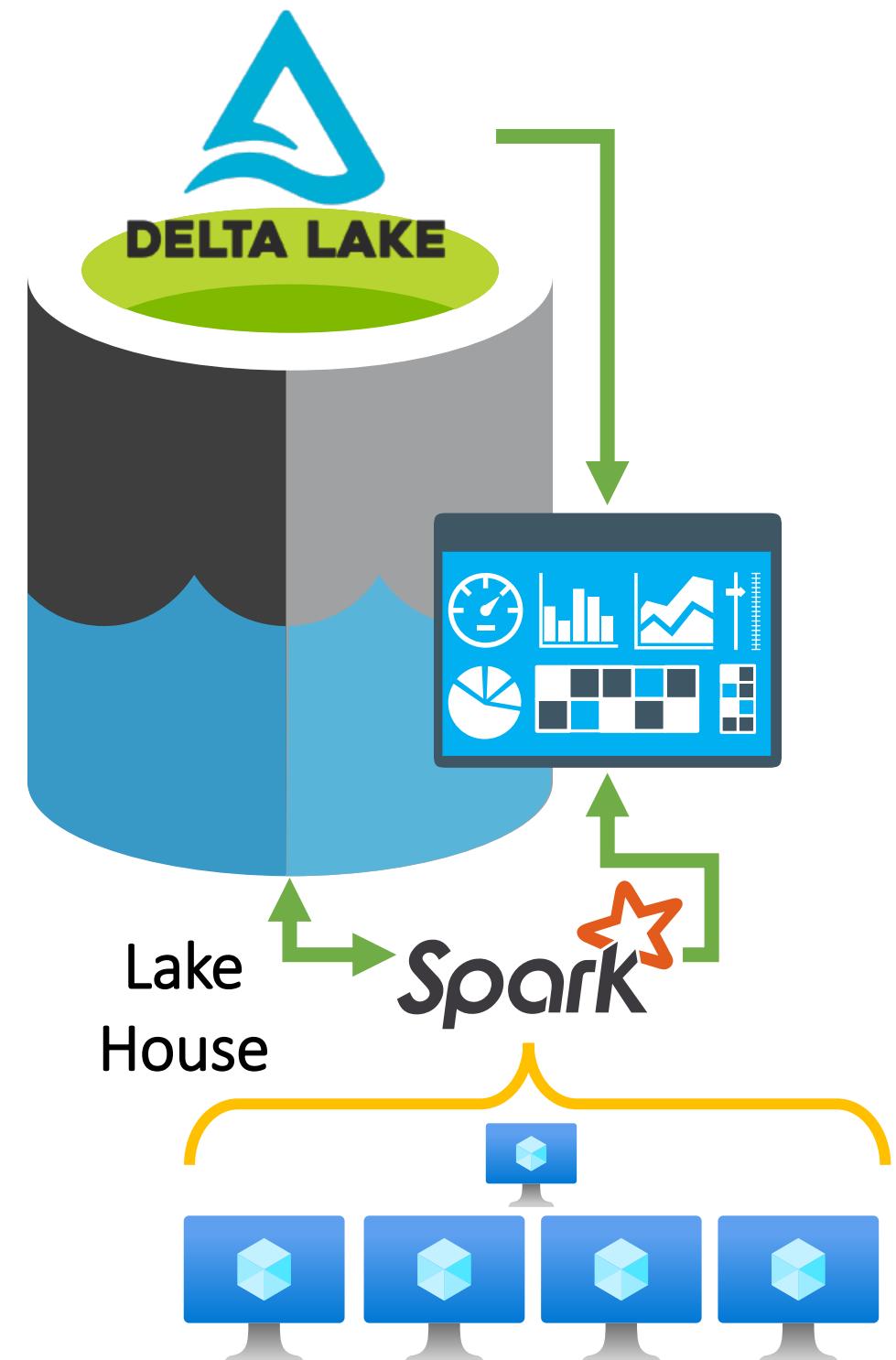
Plot [edit]

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.



Theatrical release poster

Directed by	Alejandro Agresti
Written by	David Auburn
Based on	<i>II Mare</i> by Kim Eun-jeong Kim Mi-yeong
Produced by	Doug Davison Roy Lee
Starring	Keanu Reeves



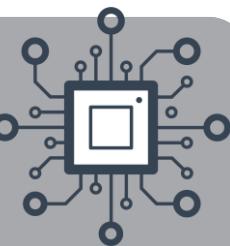
Lambda* & Kappa



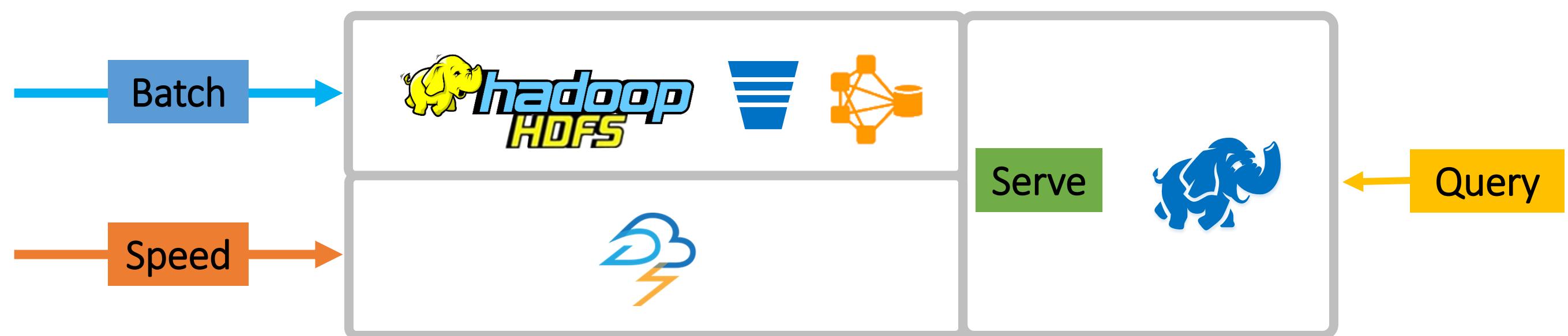
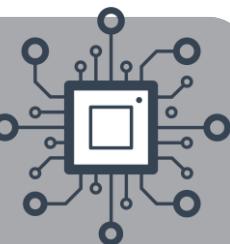
* We are not talking about the computer game Half-Life.



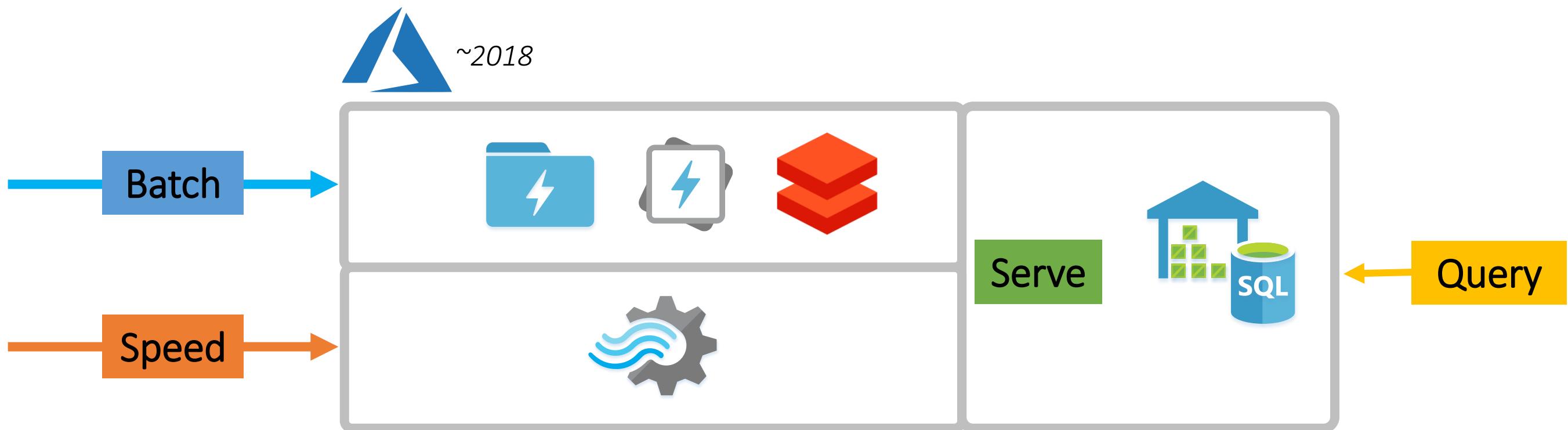
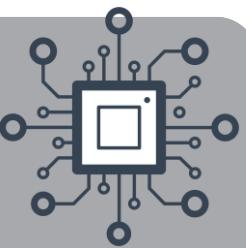
A Lambda & Kappa Architectures



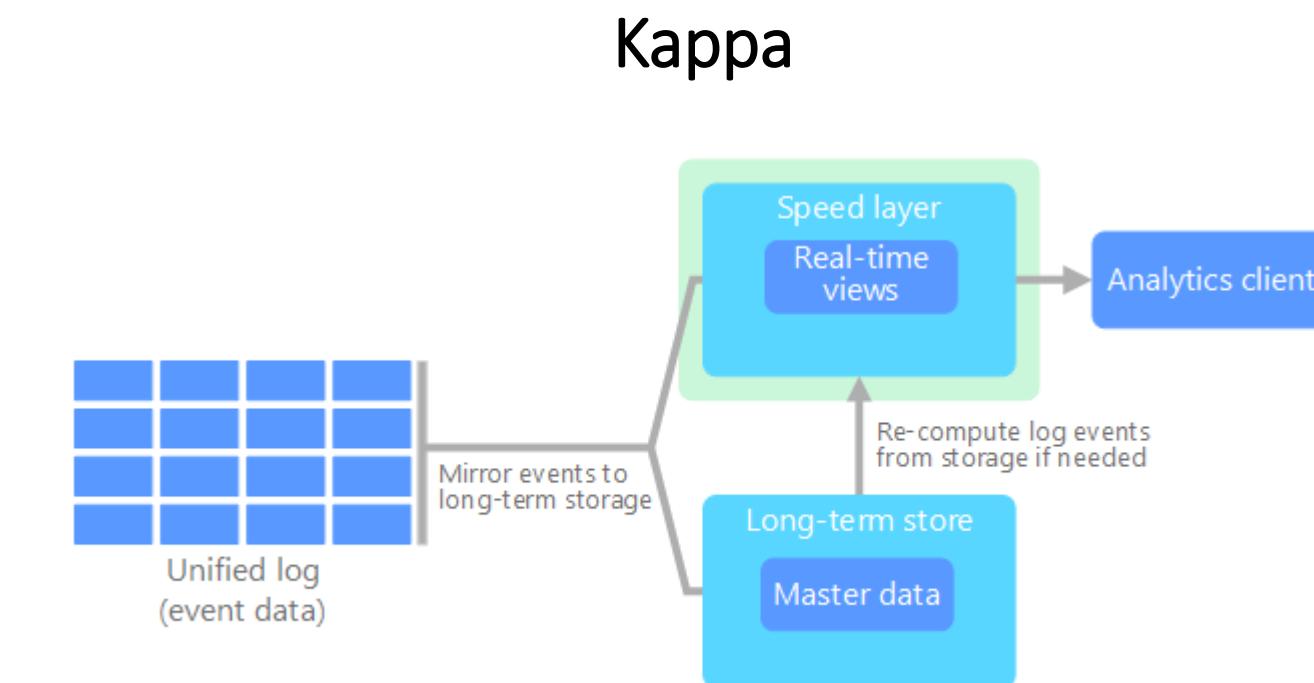
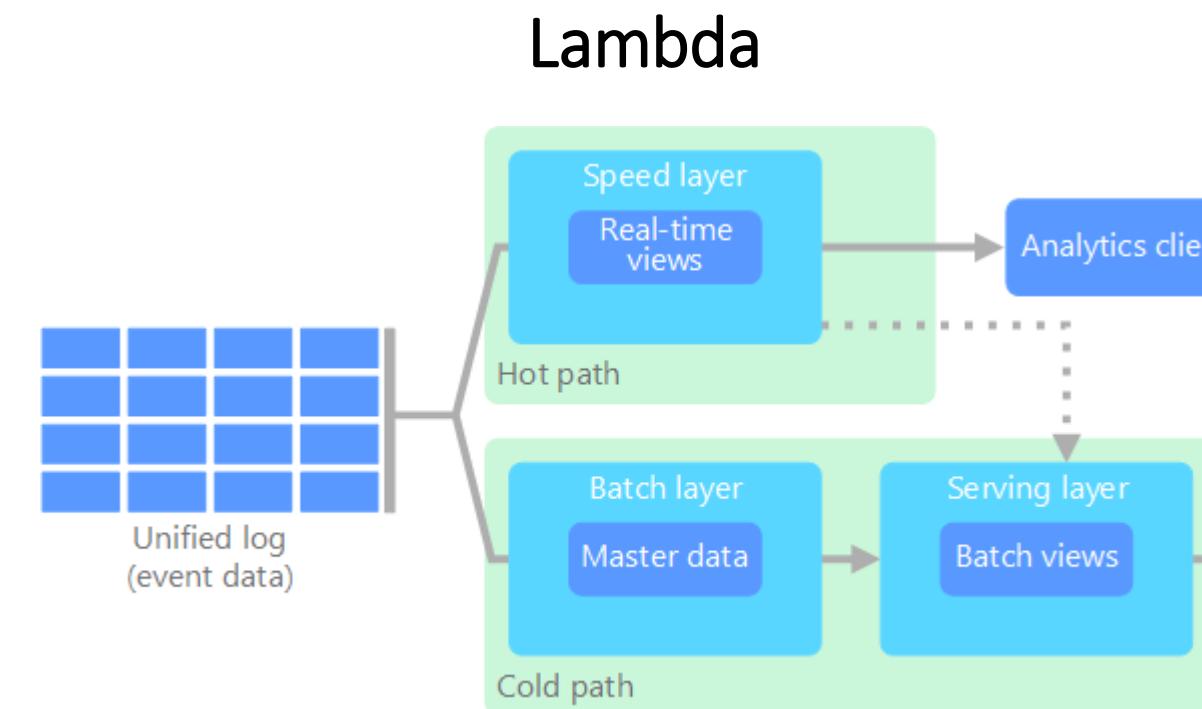
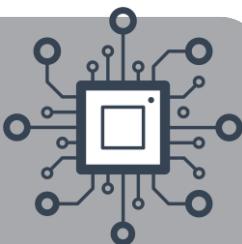
A Lambda & Kappa Architectures



A Lambda & Kappa Architectures



Lambda & Kappa Architectures



"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:

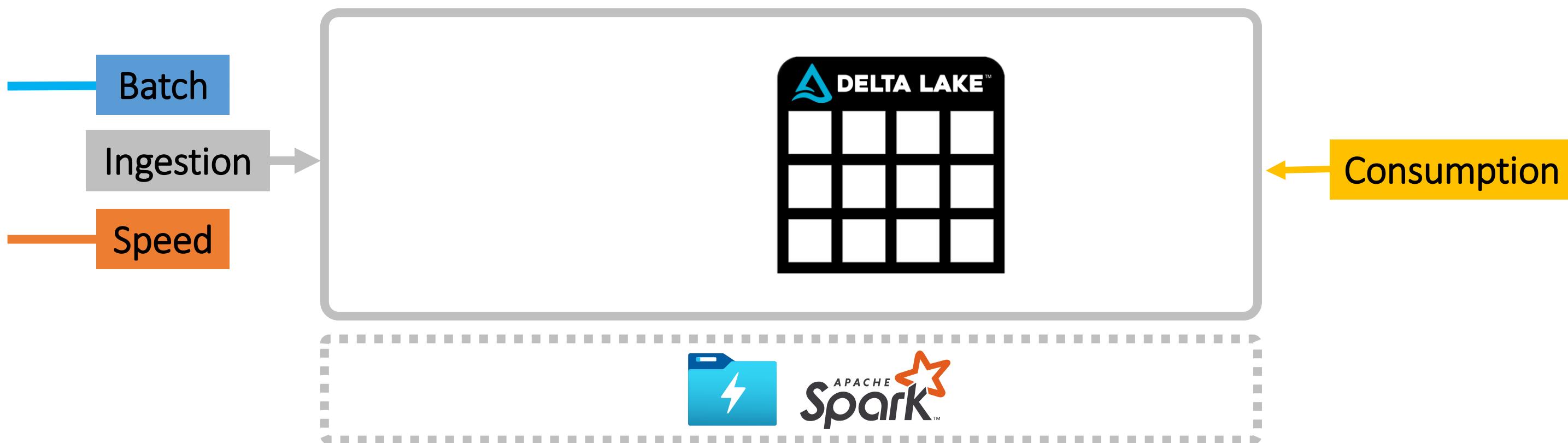
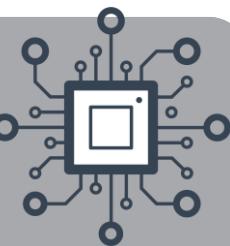
A **batch layer** (cold path) stores all of the incoming data in its raw form and performs batch processing on the data. The result of this processing is stored as a **batch view**.

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

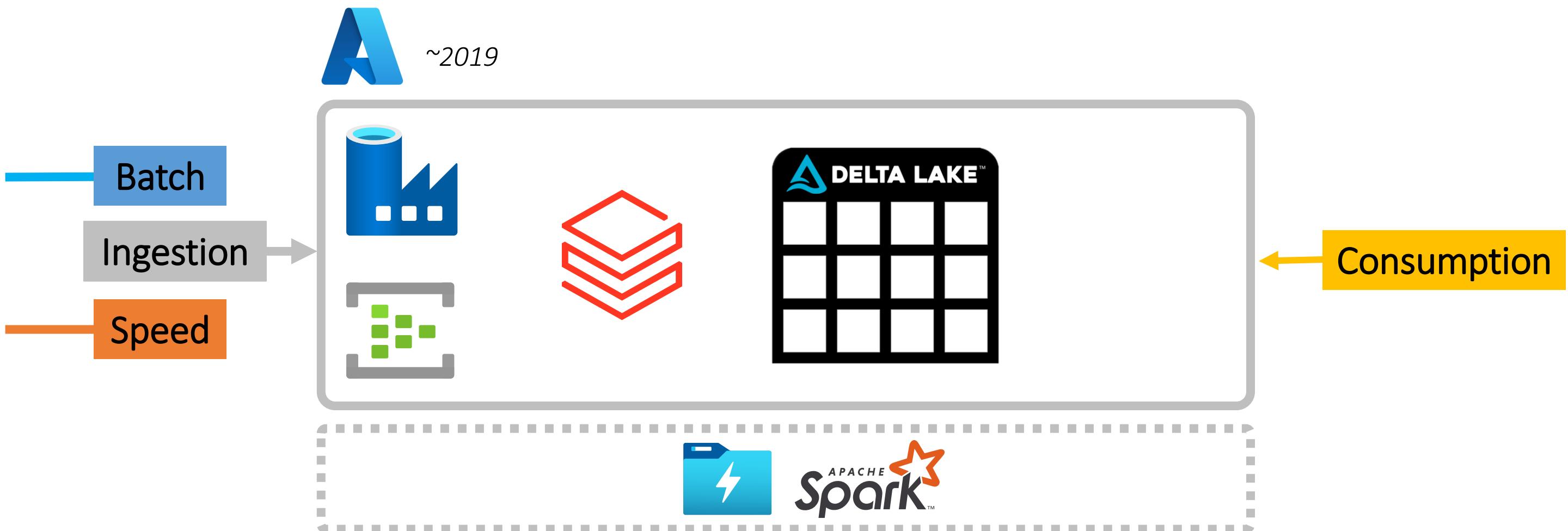
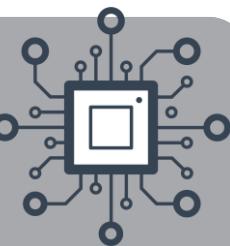
"A drawback to the lambda architecture is its complexity. Processing logic appears in two different places — the cold and hot paths — using different frameworks. This leads to duplicate computation logic and the complexity of managing the architecture for both paths.

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."

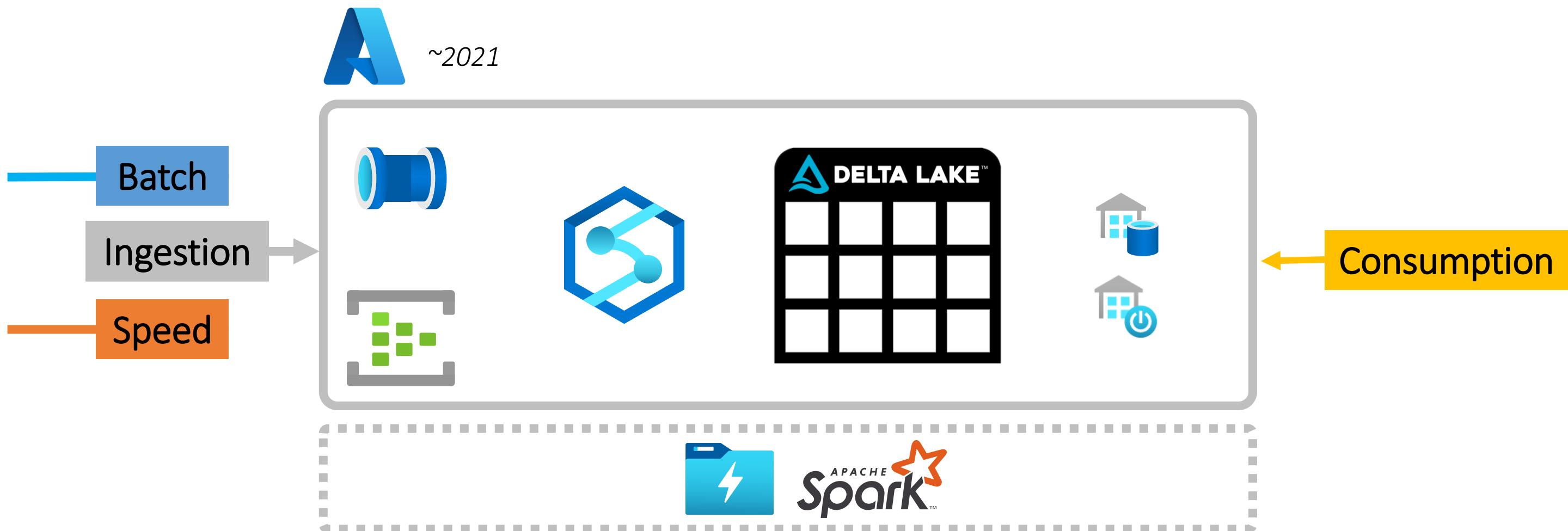
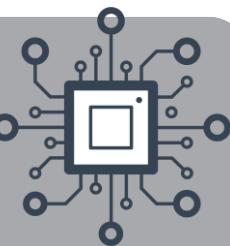
A Lambda & Kappa Architectures



A Lambda & Kappa Architectures

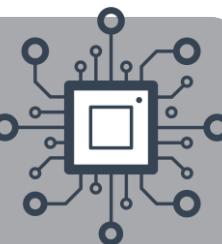


A Lambda & Kappa Architectures





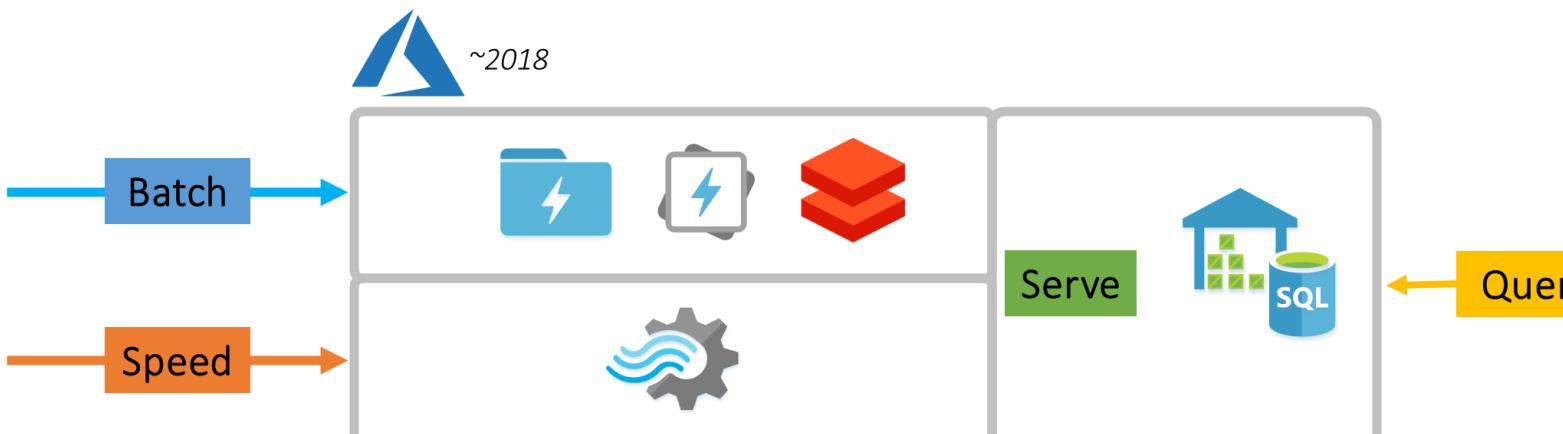
Delta Lake in the Context of Lambda & Kappa



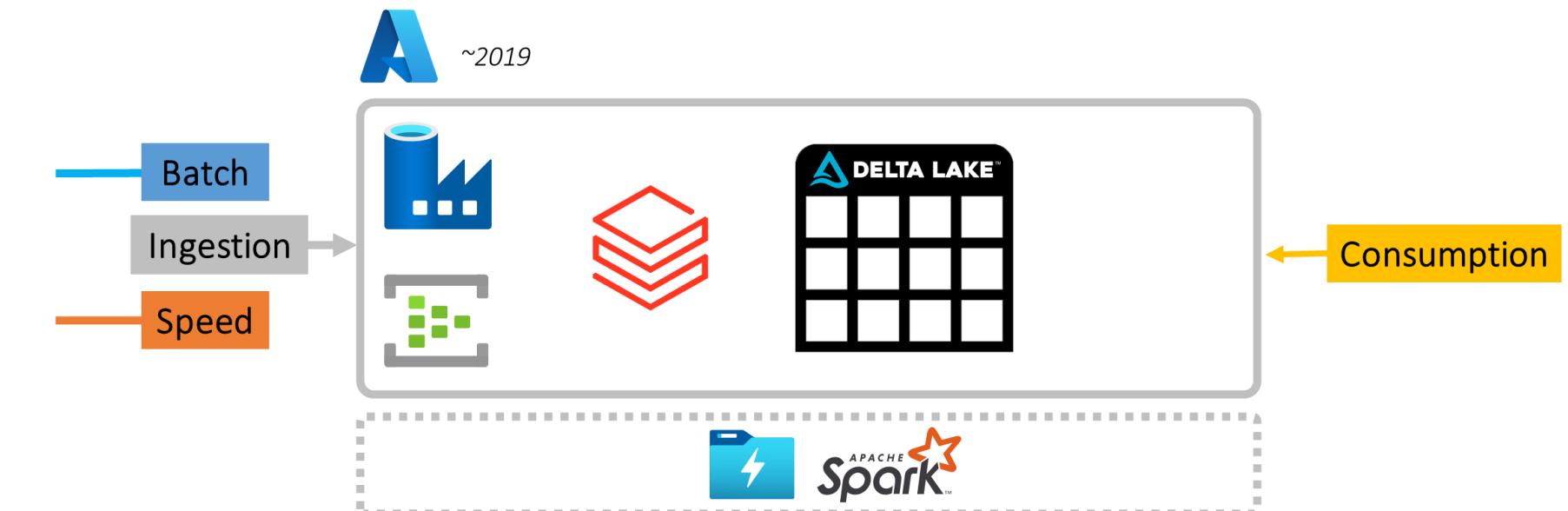
Lambda



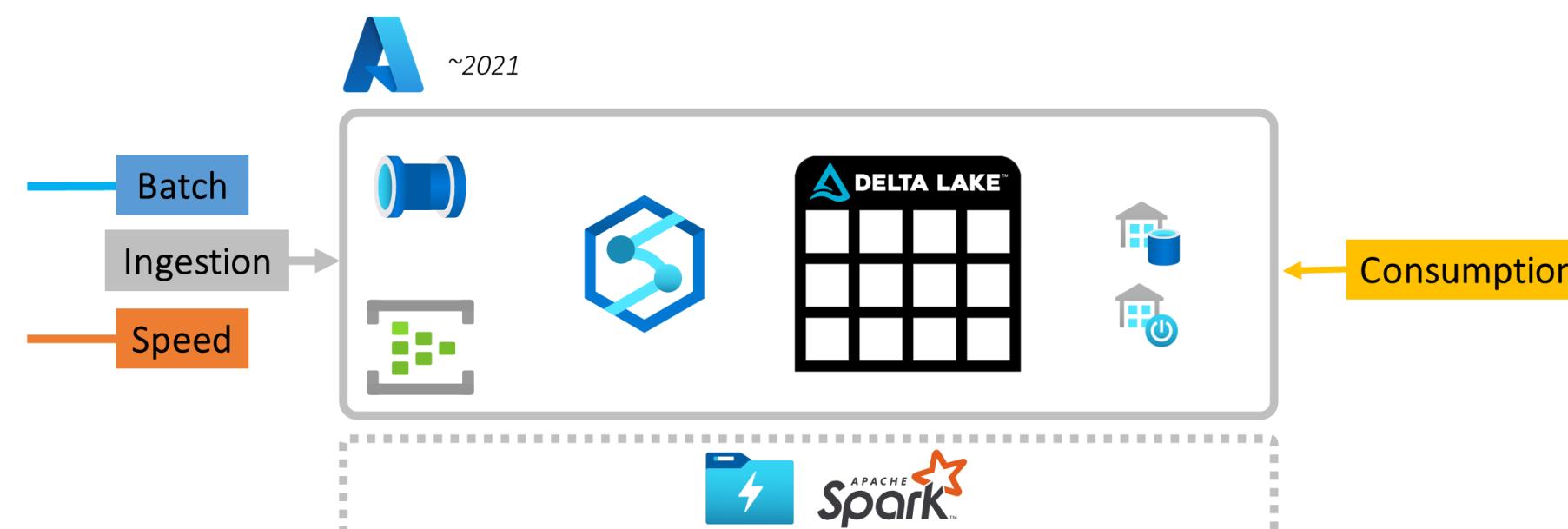
~2018



Kappa



~2019



~2021

Data Mesh

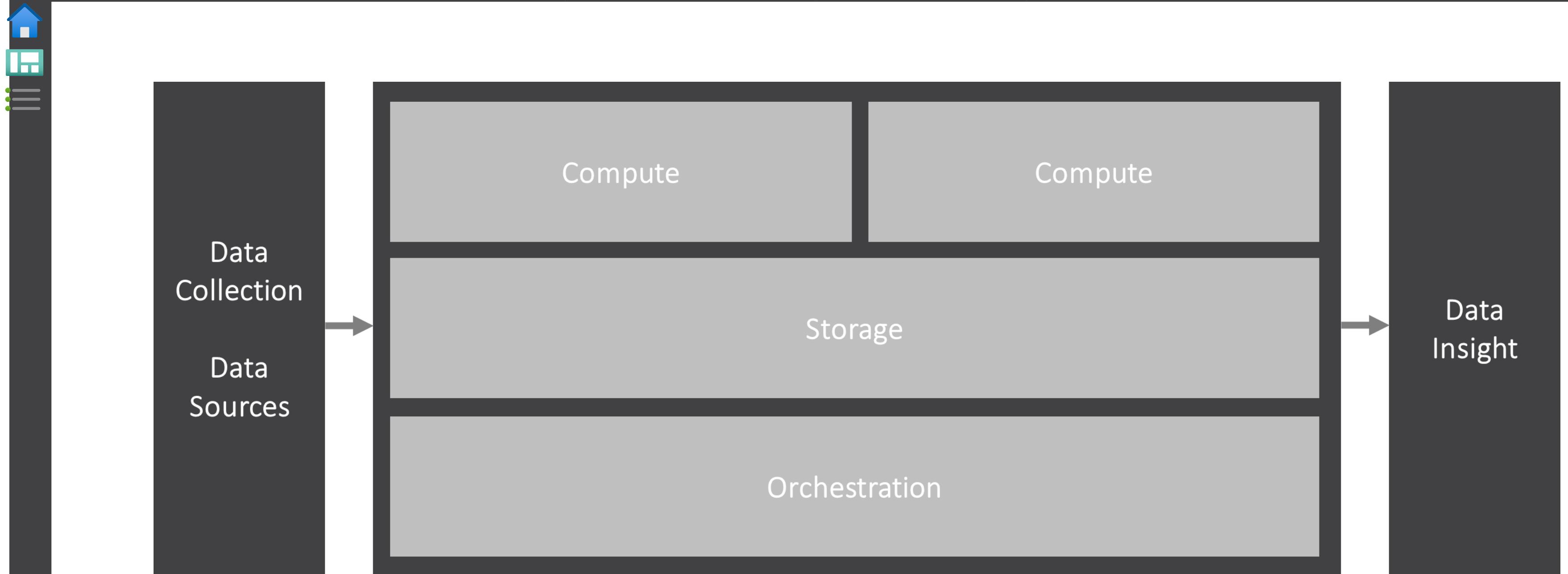
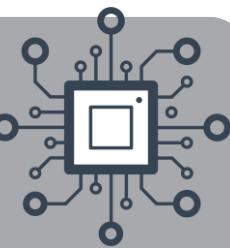
- Zhamak Dehghani

@zhamakd

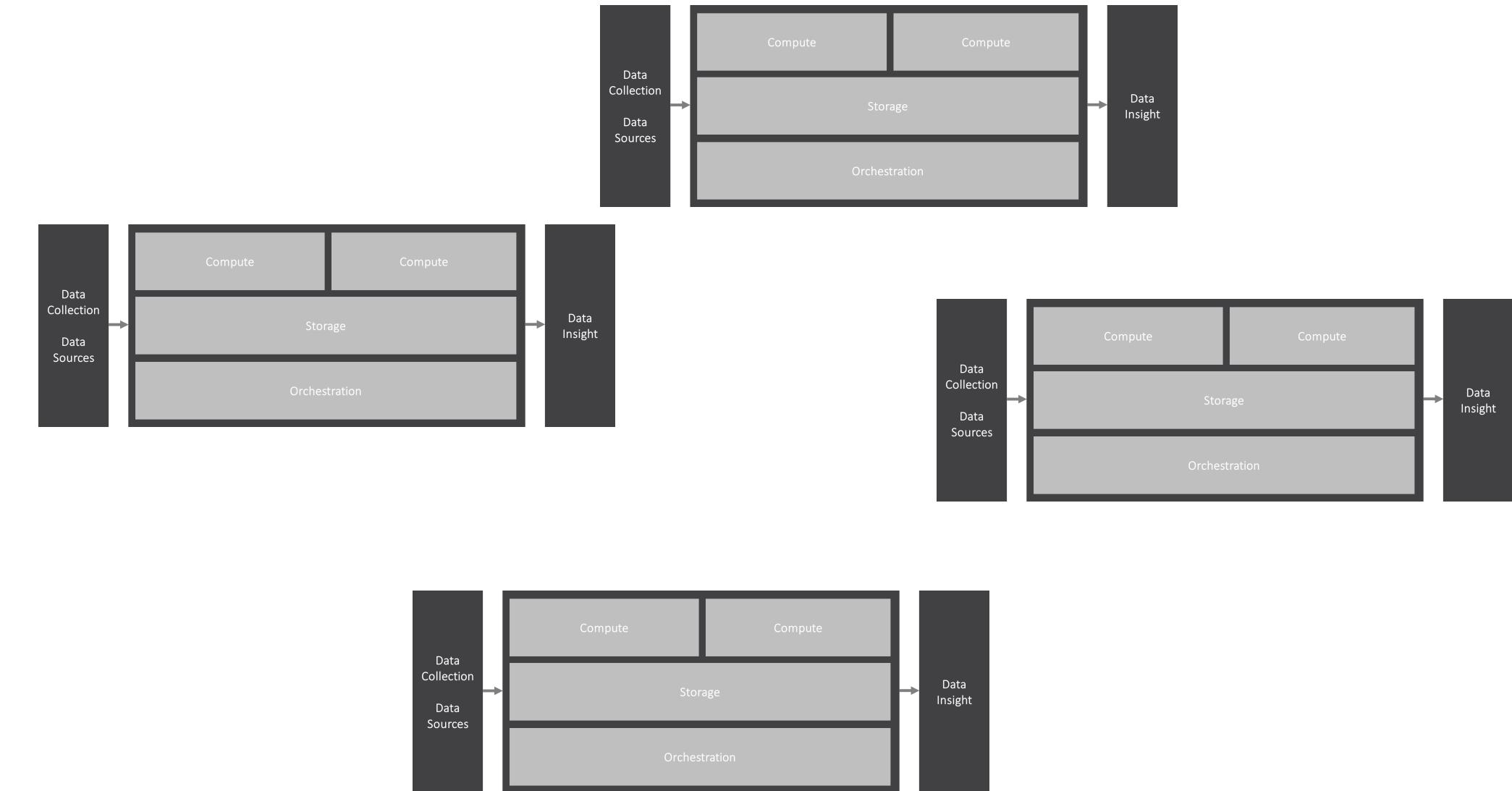
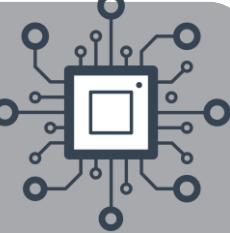
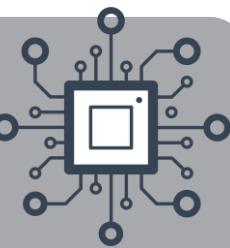


1. Domain-oriented decentralised data ownership and architecture.
2. Data as a product.
3. Self-serve data infrastructure as a platform.
4. Federated computational governance.

Data Products

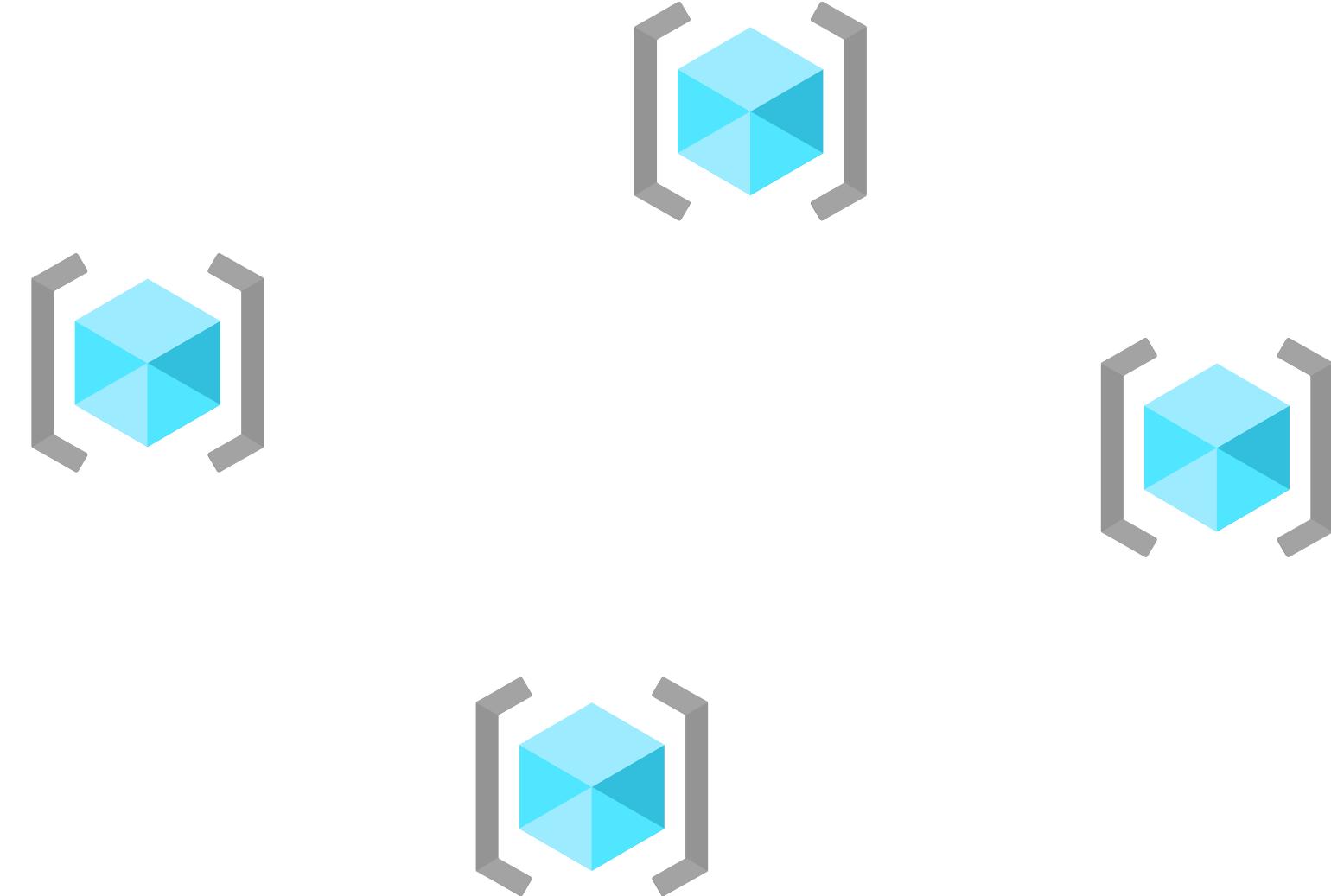
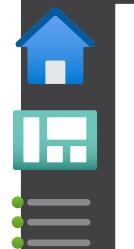
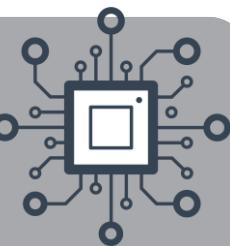


Data Products in Azure



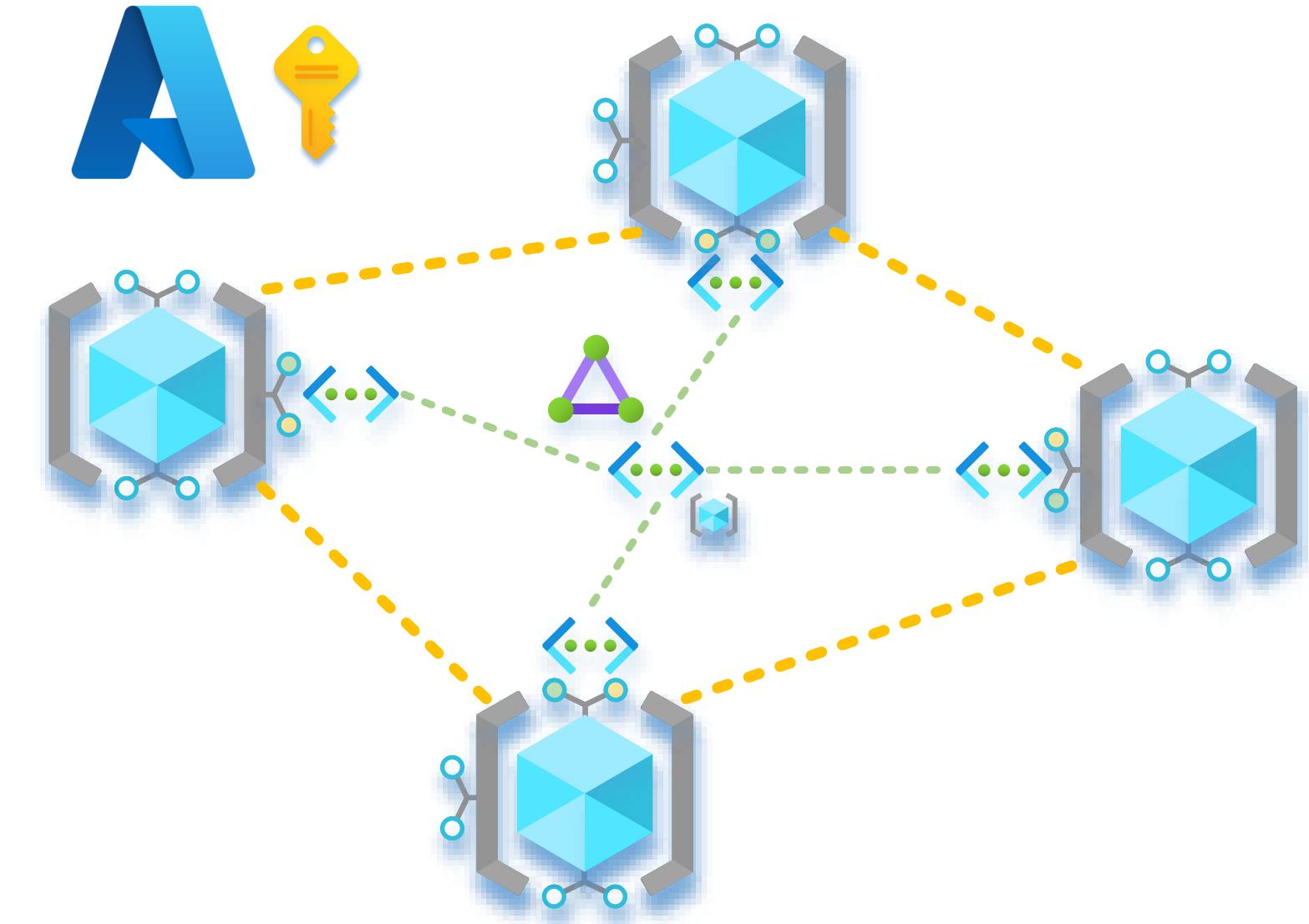
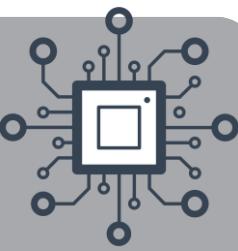


Data Products in Azure

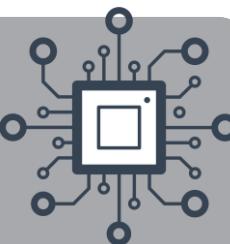




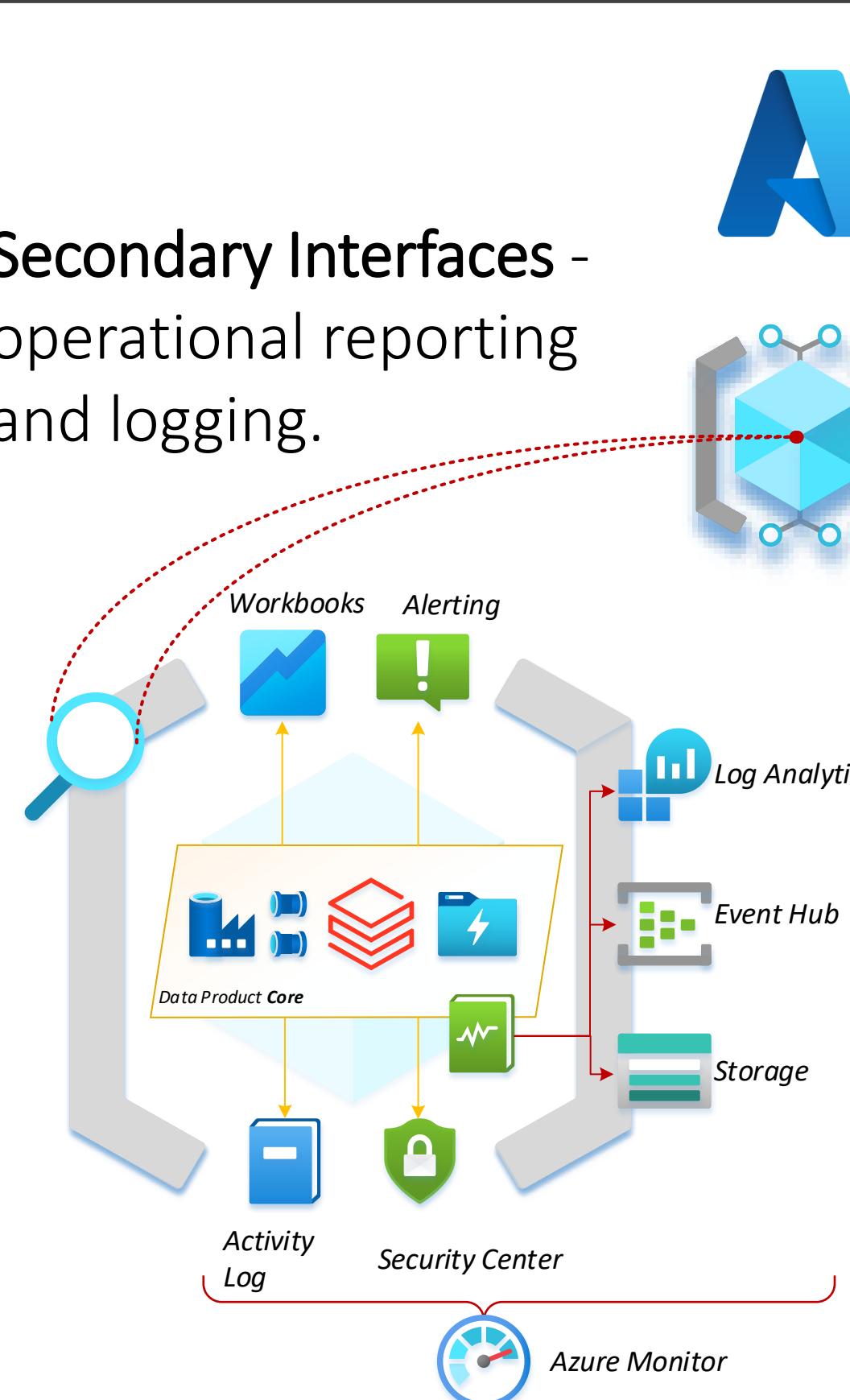
Data Products in Azure with Interfaces



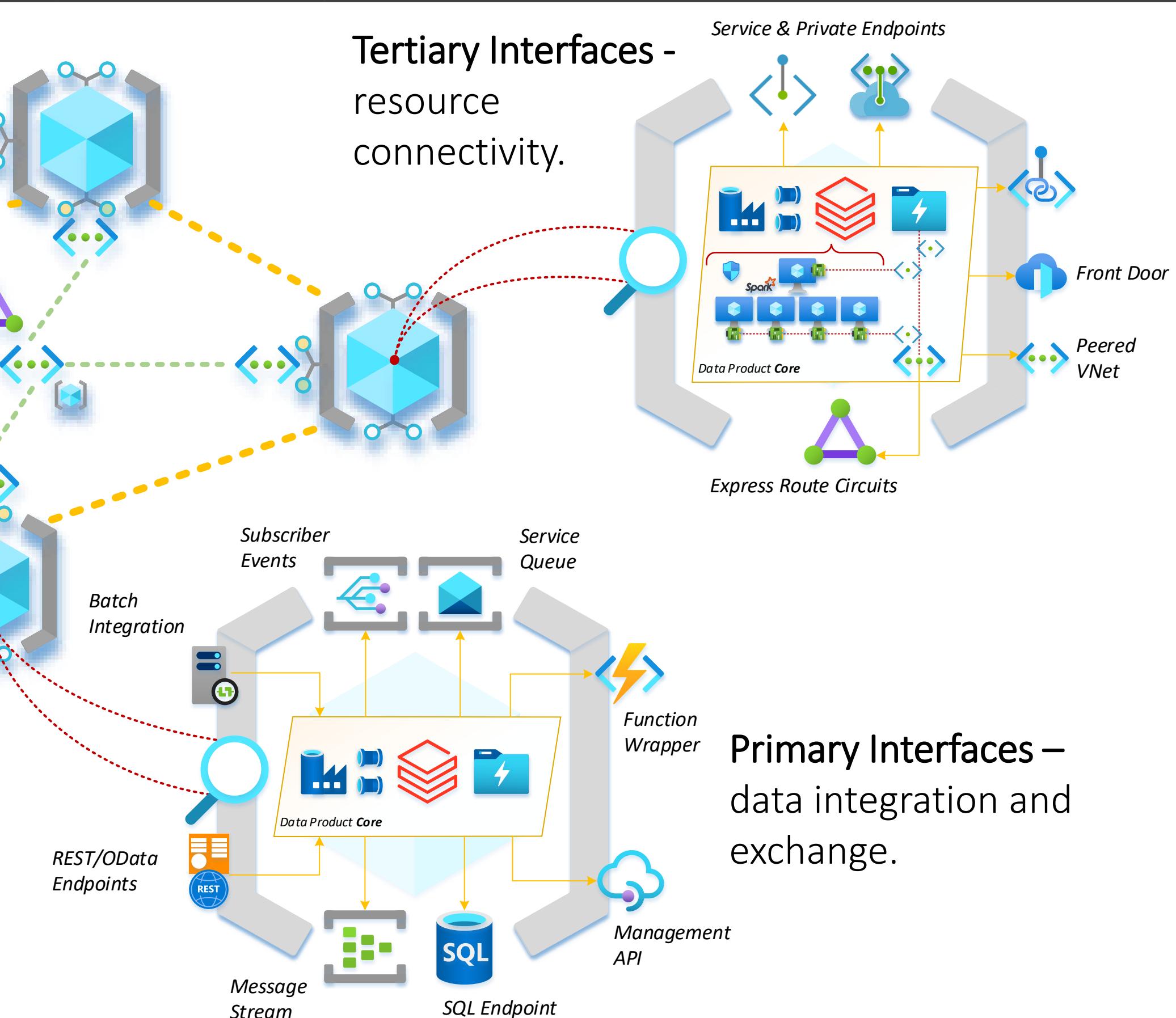
Data Products in Azure with Interfaces



Secondary Interfaces –
operational reporting
and logging.



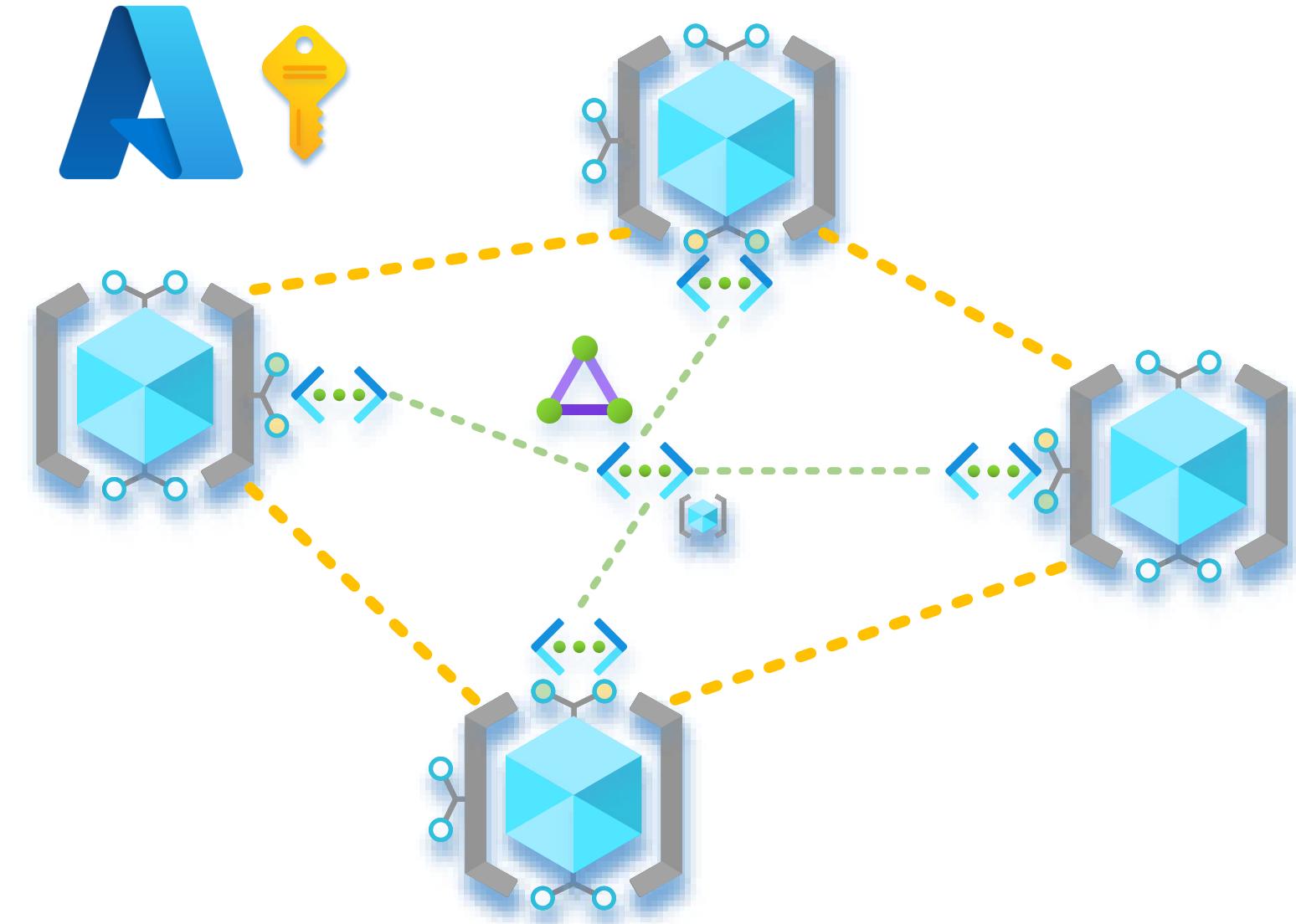
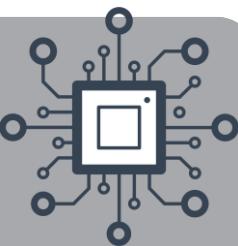
Tertiary Interfaces –
resource
connectivity.



Primary Interfaces –
data integration and
exchange.

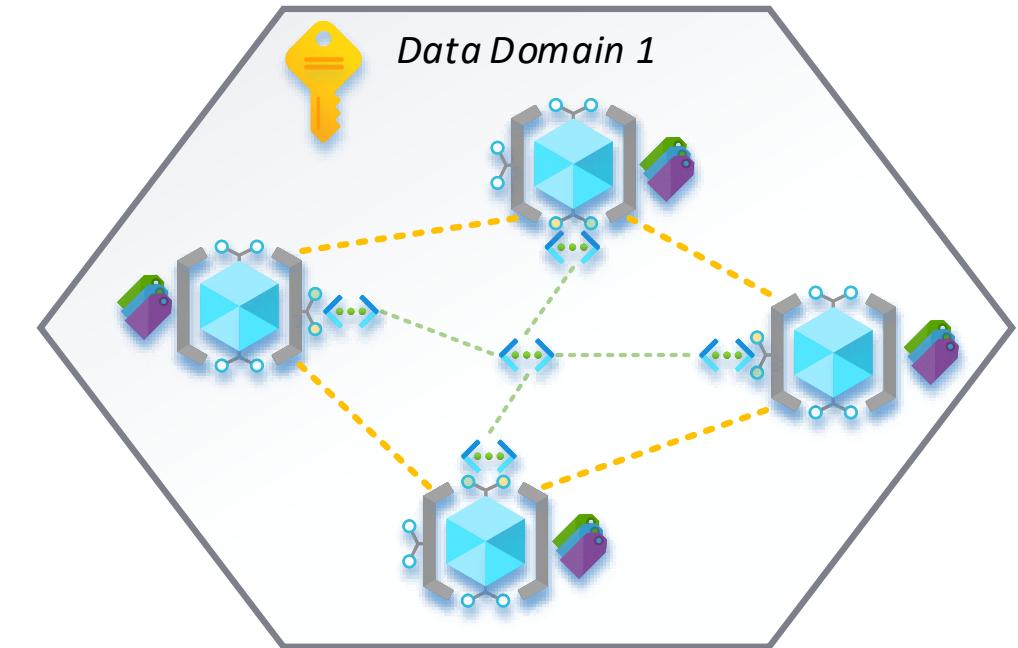
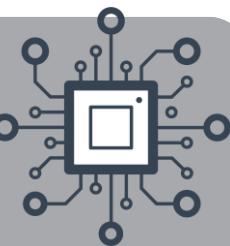


Data Domains in Azure



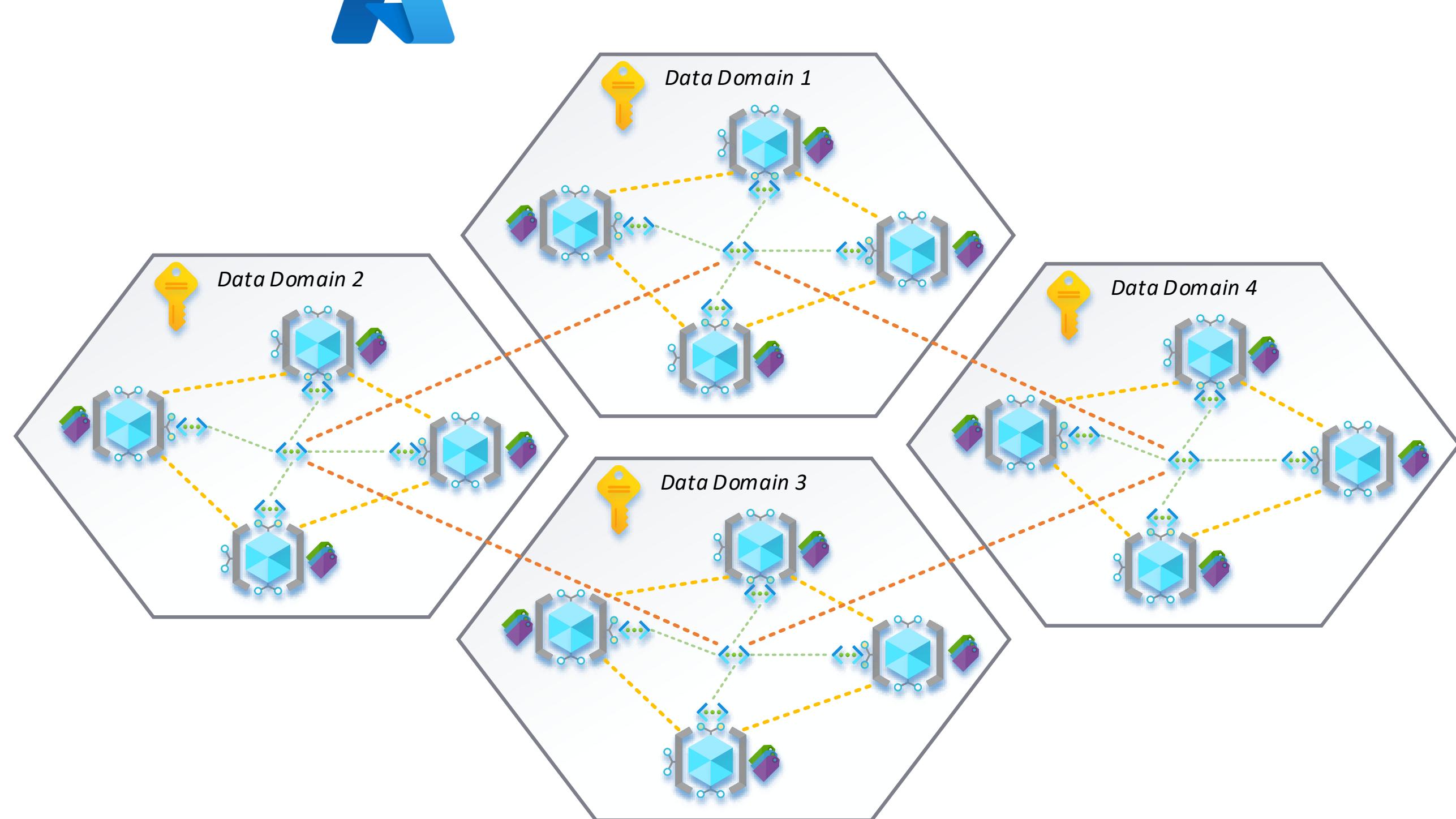
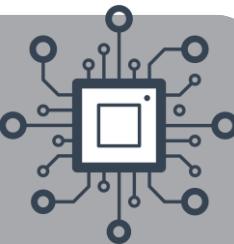


Data Domains in Azure



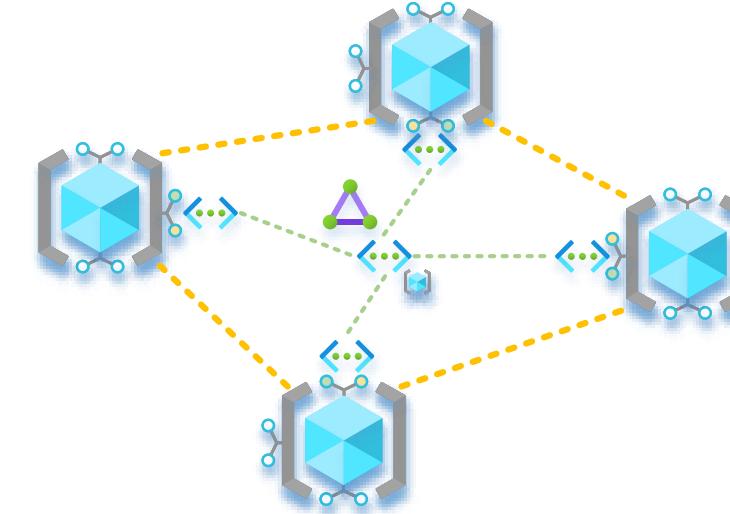
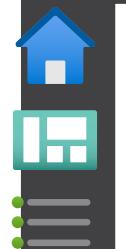
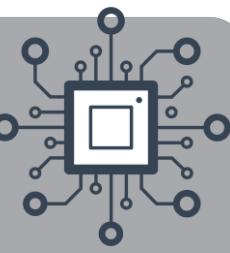


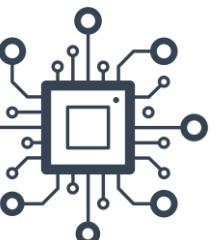
Data Domains in Azure



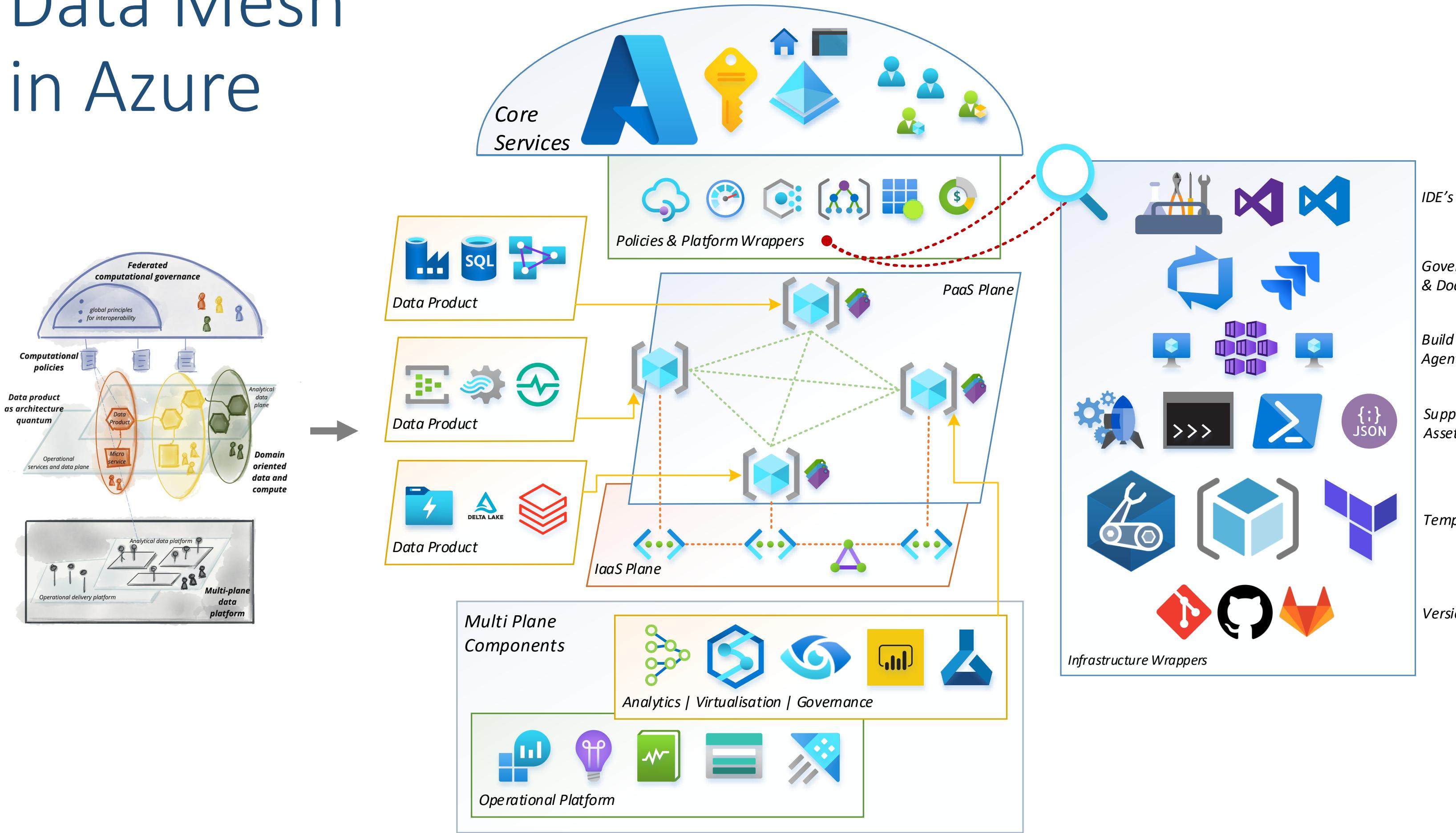


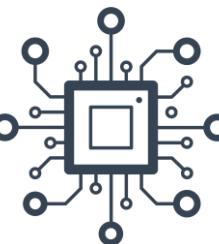
Data Domains in Azure





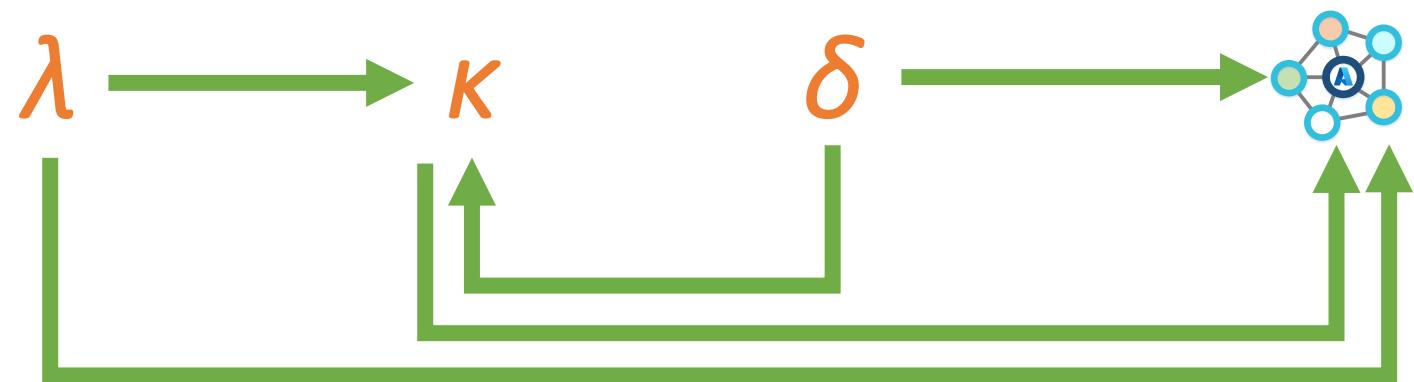
Data Mesh in Azure





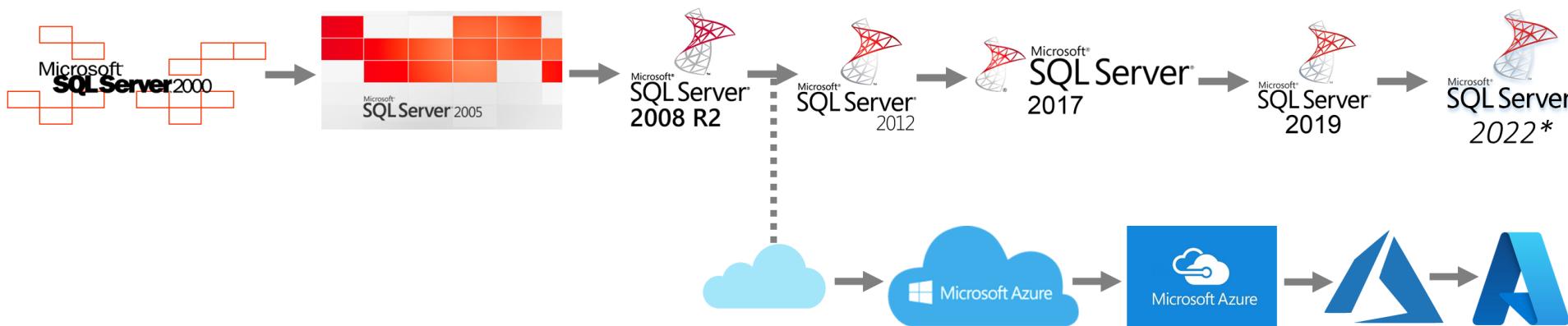
The Evolution of Data Platform Architectures in Azure

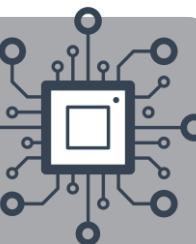
Lambda, Kappa, Delta, Data Mesh



... should we be considering a solution that offers all these principals in a nirvana of data insight perfection? **Yes!**

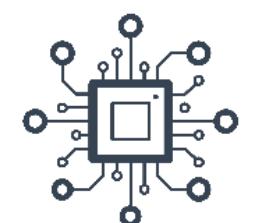
1. Domain-oriented decentralised data ownership and architecture.
2. Data as a product.
3. Self-serve data infrastructure as a platform.
4. Federated computational governance.





Thank you for listening...

Paul Andrew



Mr Paul Andrew
Consulting Ltd

Blog: mrpaulandrew.com

YouTube: [c/mrpaulandrew](https://www.youtube.com/c/mrpaulandrew)

Email: paul@mrpaulandrew.com

Twitter: [@mrpaulandrew](https://twitter.com/mrpaulandrew)

LinkedIn: [In/mrpaulandrew](https://www.linkedin.com/in/mrpaulandrew)

Github: github.com/mrpaulandrew

Contact Details



sqlbits
Session Feedback

