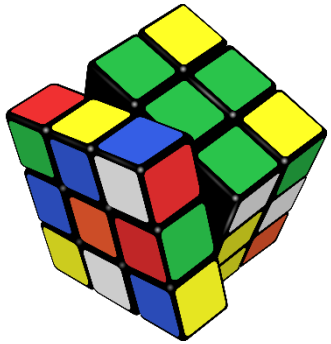


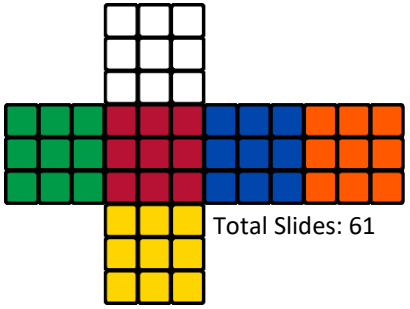
Big Data Engineering For Analytics

Big Data Architecture

Suria R Asai

(suria@nus.edu.sg)

NUS-ISS

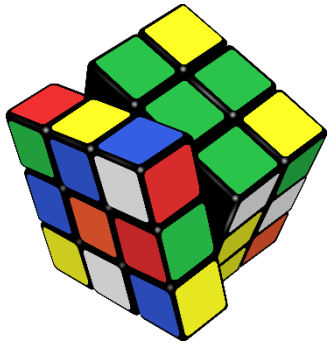


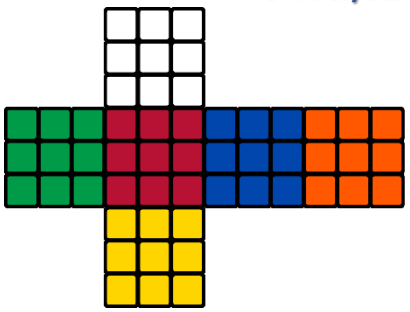
Learning Objectives

- Understand ***data architecture*** by examining how data is managed from *collection* through to *transformation*, *distribution* and *consumption* in an organization.
- Learn about the ***data engineering lifecycle***, a framework describing "*cradle to grave*" data engineering operating in terms of *principles*.
- Understand the various ***deployment options*** such as on-premise, collocated and public cloud processing platforms.

Agenda

- The (Big) Data Architecture
- (Big) Data Engineering Life Cycle
- Hadoop Ecosystem
- Kubernetes Orchestration
- Commercial Cloud Products
- Summary



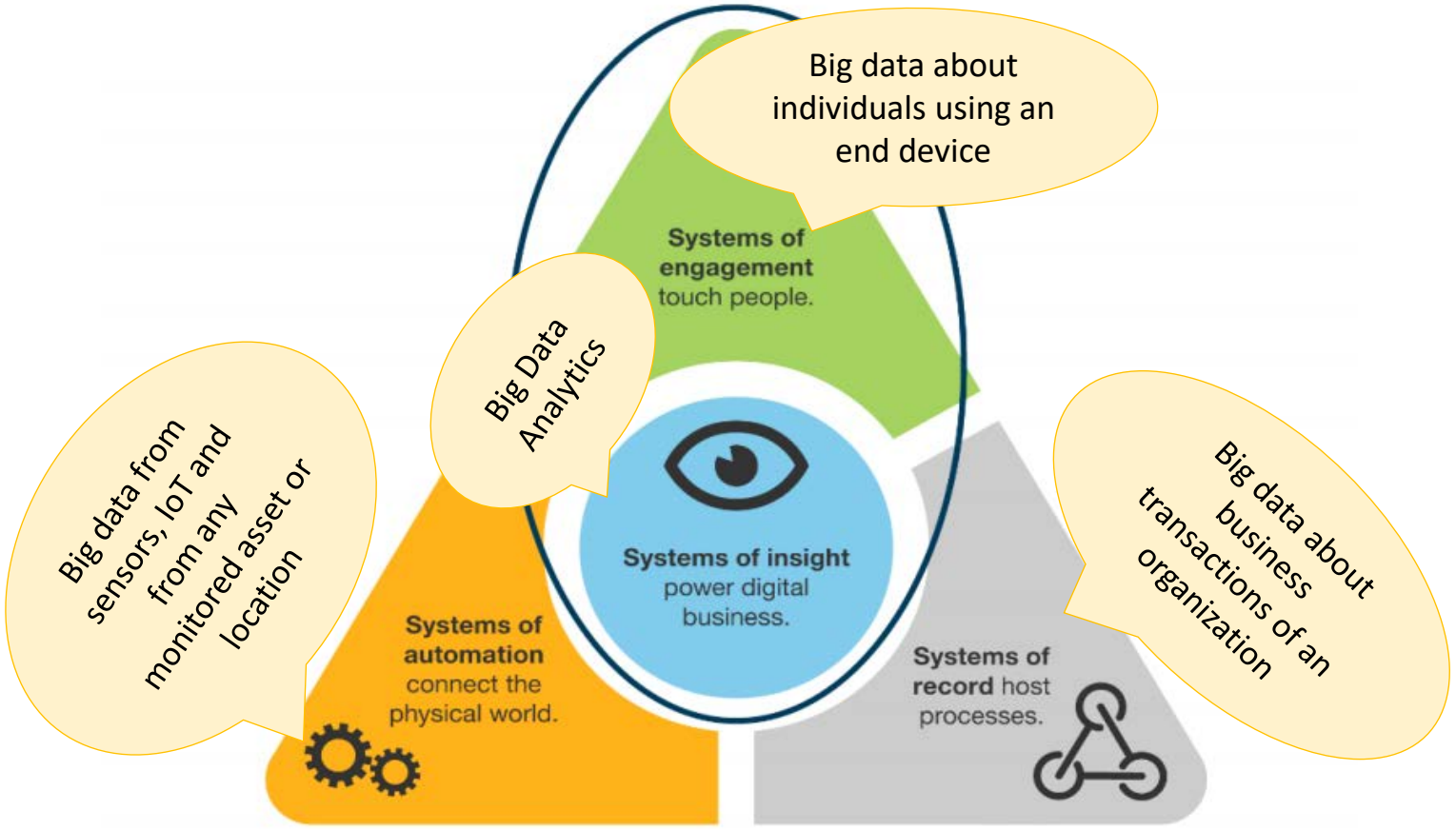


The (Big) Data Architecture

I propose to consider this question "Can machines think?"

Alan Turning

Different Systems . . .

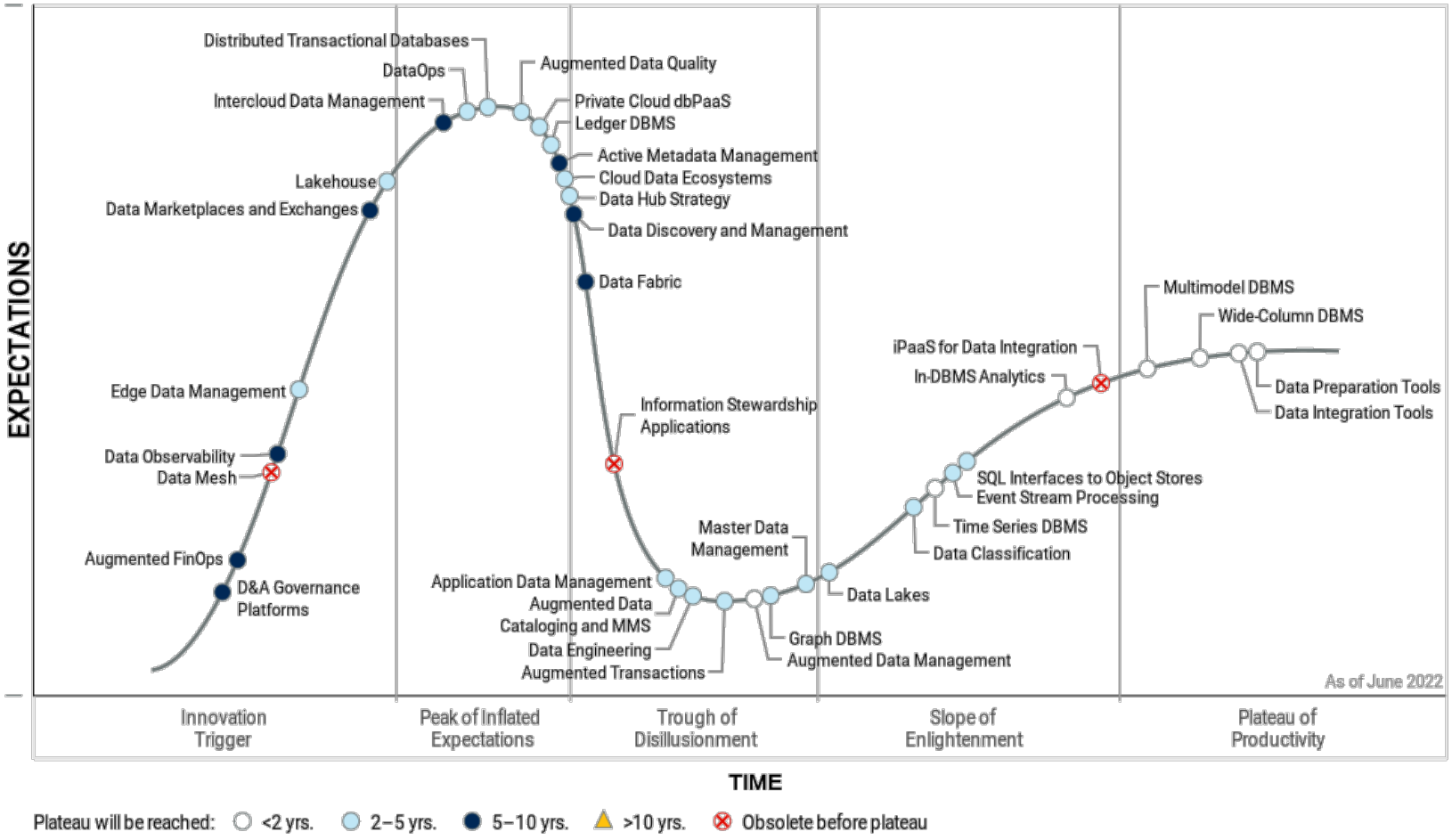


Source: "Digital Insights Are The New Currency Of Business" Forrester report

125542 Source: Forrester Research, Inc. Unauthorized reproduction, citation, or distribution prohibited.

Gartner Hype Cycle

Hype Cycle for Data Management, 2022



Gartner

Big Data for Server Side

CLOUD OFFERINGS

Example: Google Cloud Dataproc, Amazon EMR, Azure HDInsight and IBM Analytics.

Data Processing Pipeline

Processing Types:

1. Batch
2. Interactive
3. Near Real Time
4. Real Time
5. Time Frame

Eg: Apache Hadoop YARN SPARK

DATA LAKE

INTELLIGENT INSIGHTS

Insight Discovery and Action

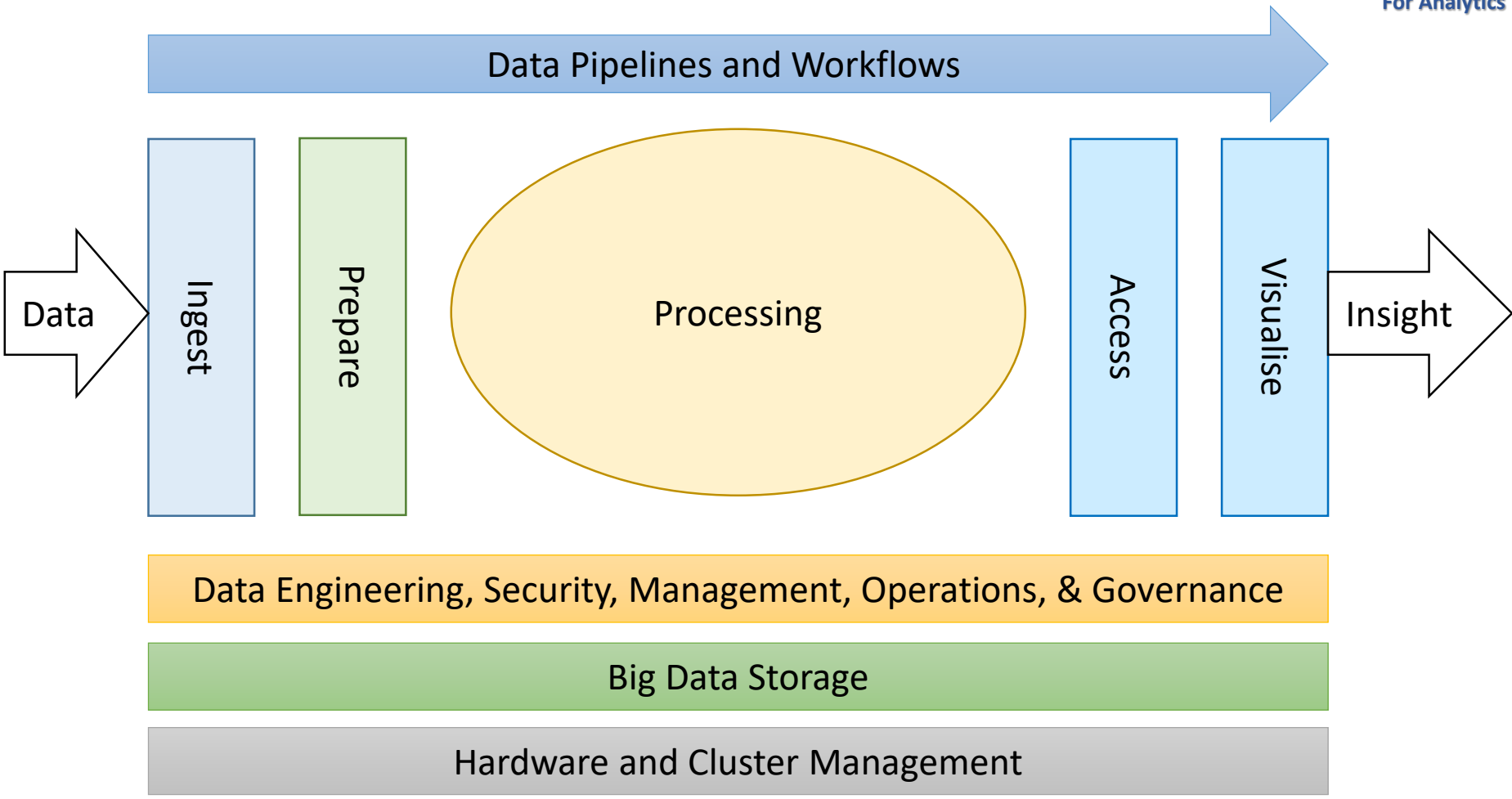
Big Data for End Device Computing

Intelligent Decisions

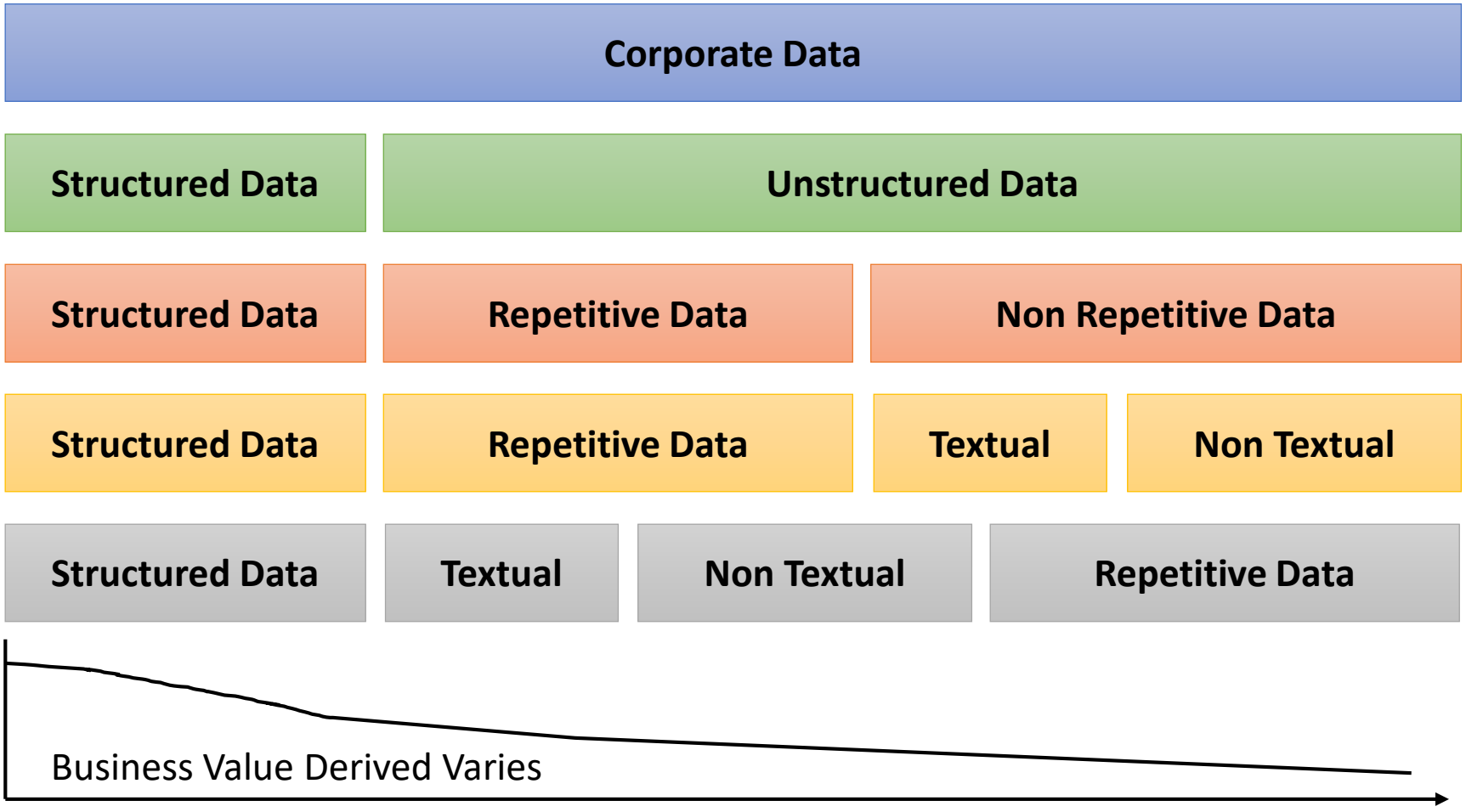
- Slow Precise
- Quick & Dirty

1. Constraints
2. Provenance
3. Mobility
4. Peer-to-Peer
5. Scheduling or Event Based

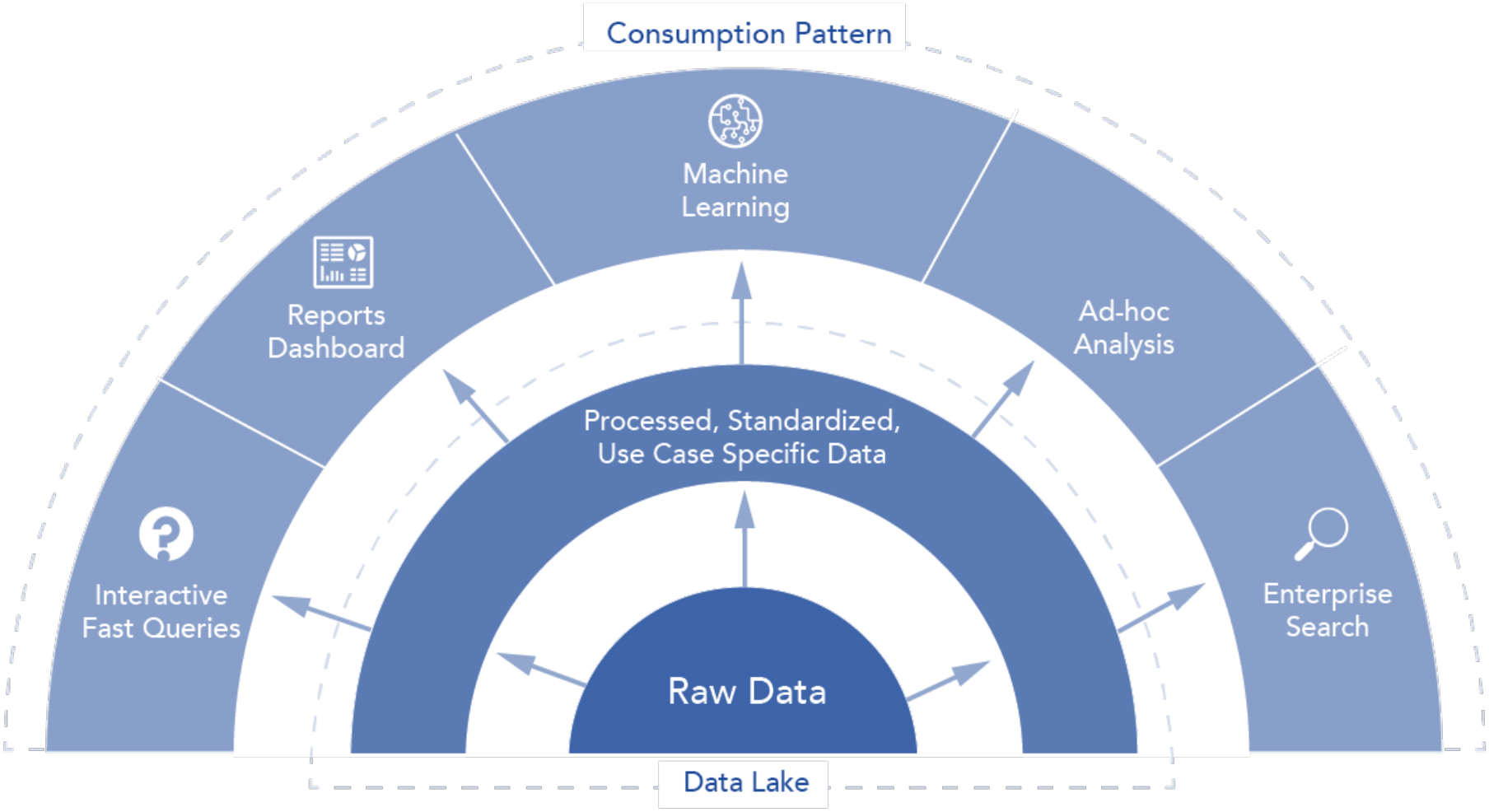
Components of Reference Architecture



The totality of corporate data.

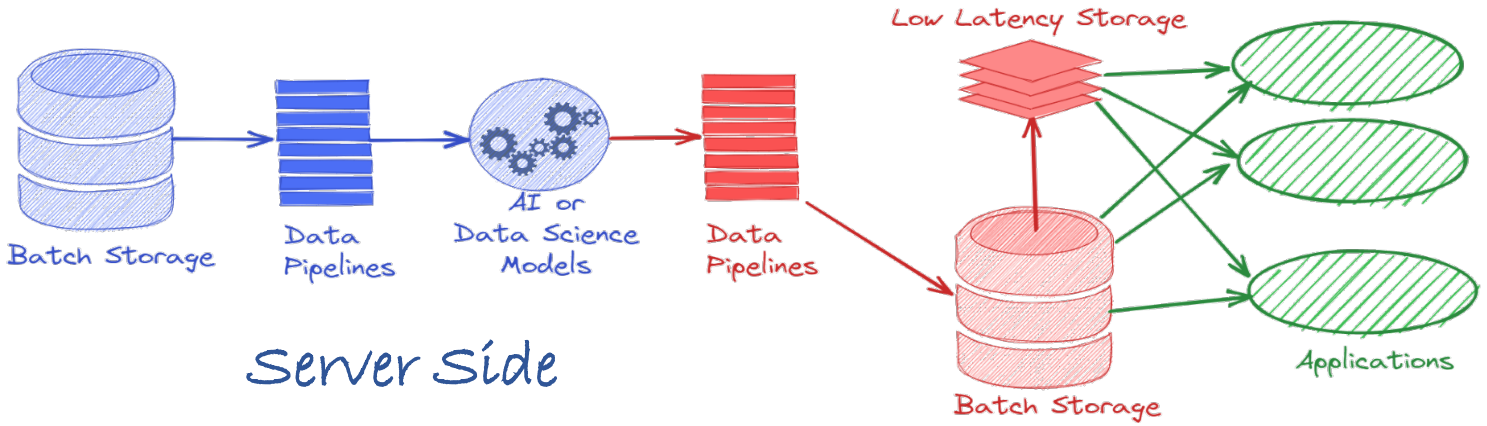


Consumption Patterns

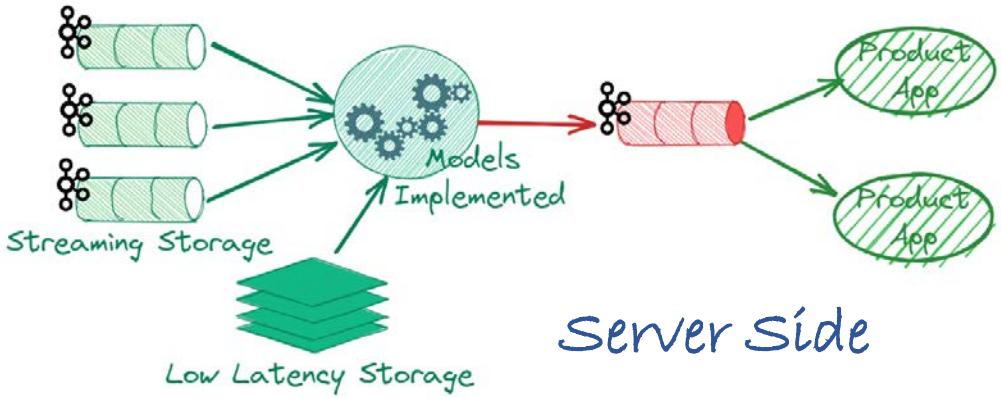


Types of Processing - 1

Batch Processing

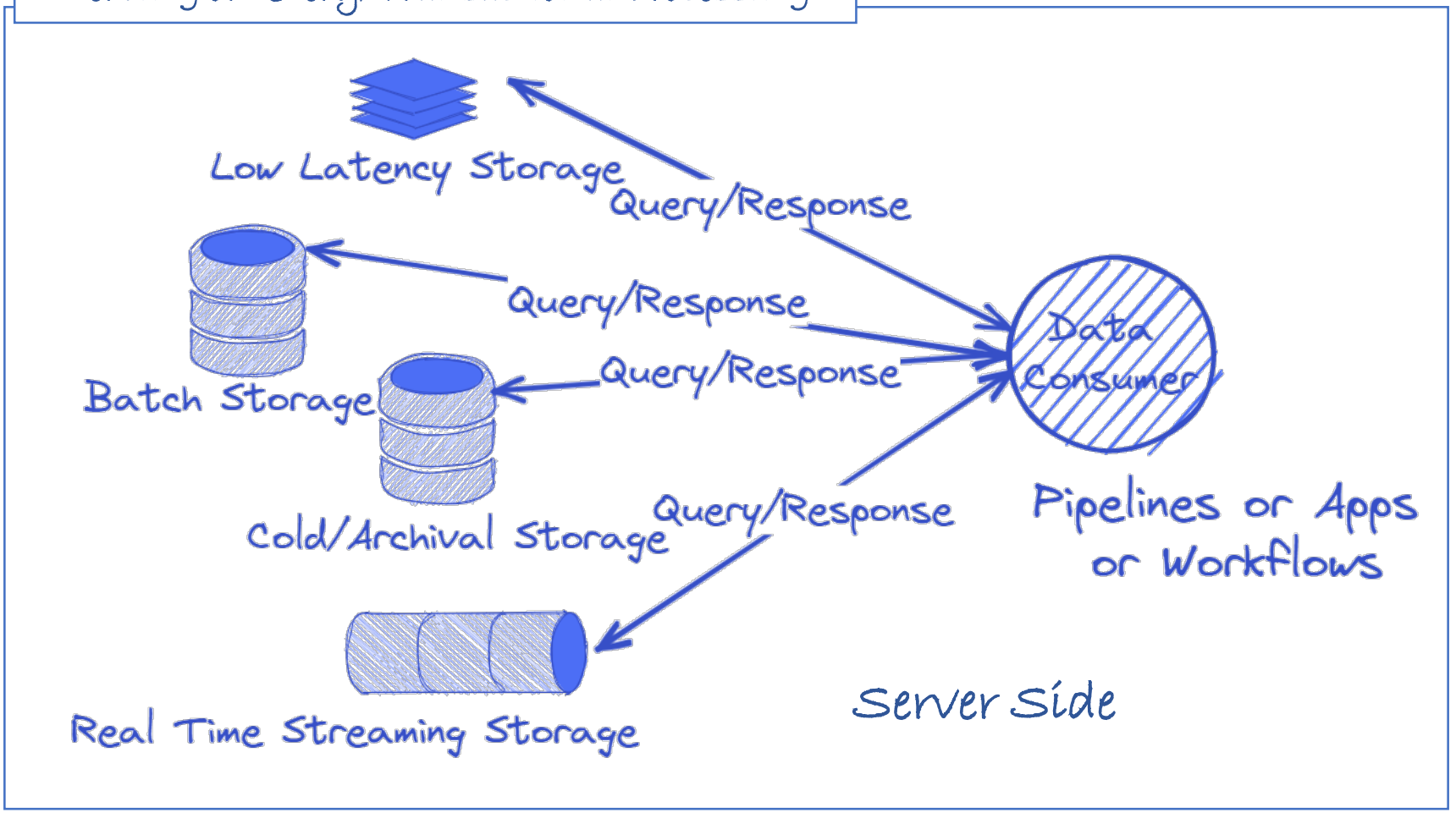


Real Time Stream Processing



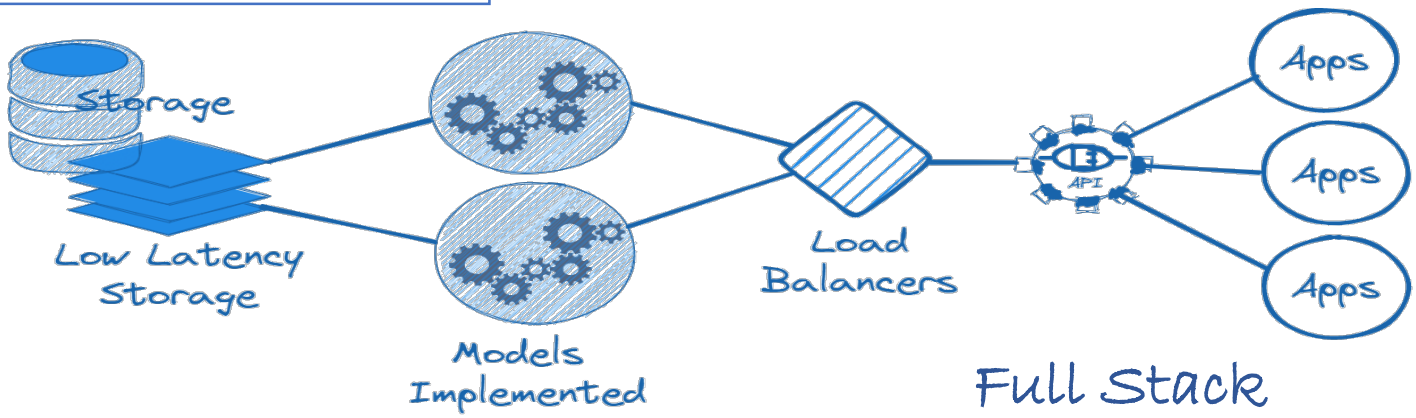
Types of Processing - 2

Serving or Query/Transactional Processing

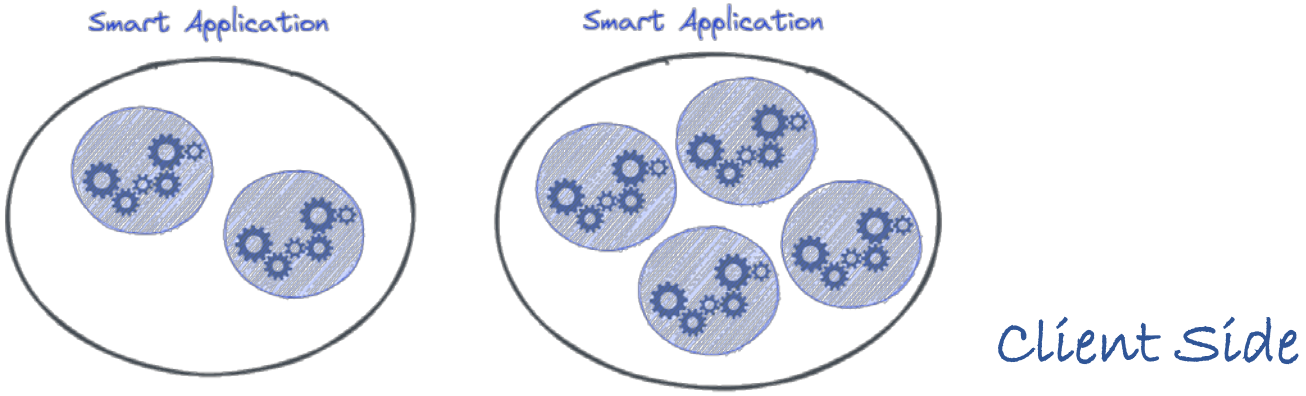


Types of Processing – 3.

Request Response Processing



Edge

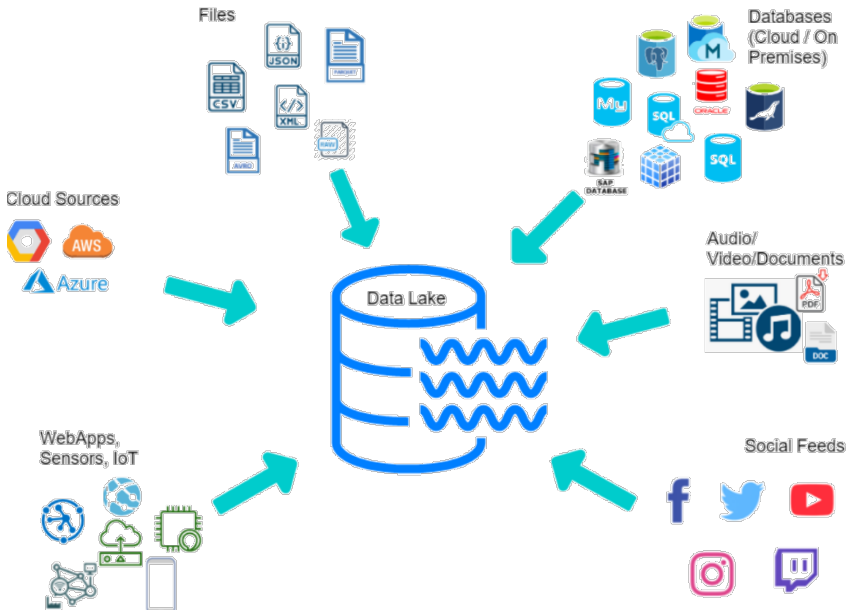


Working of Layers

Batch Layer	•Stored immutable data •Constantly growing in size •Recomputed views all the time
Speed Layer	•Constant stream of data •Stores mutable data •Less in size/ volume •Views live for a specified period and discarded at intervals
Serving Layer	•Responsible for indexing and making sure exposed batch views perform well •Exposes real-time views created incrementally by the speed layer •Merges result from both batch and speed views in a consistent fashion

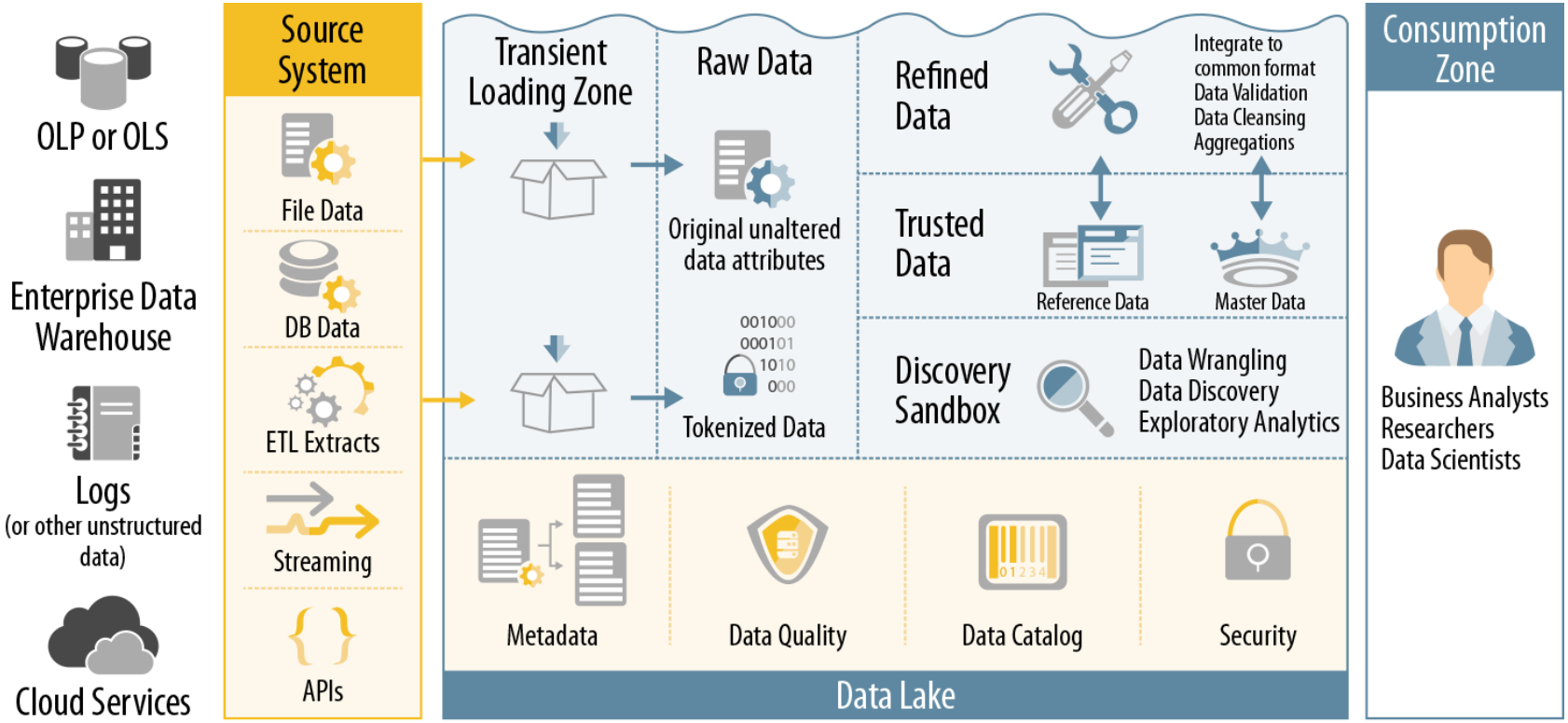
The idea of Data Lake

A data lake is a collection of data storage instances combined with one or more processing capabilities. Most data assets are copied from diverse enterprise sources and are stored in their **raw** arrival state, so they can be refined and repurposed repeatedly for **multiple use cases**. Ideally, a data lake will **store** and **process** data of any structure, latency or container *(files, documents, result sets, tables, formats, BLOBs, messages, etc.)*.



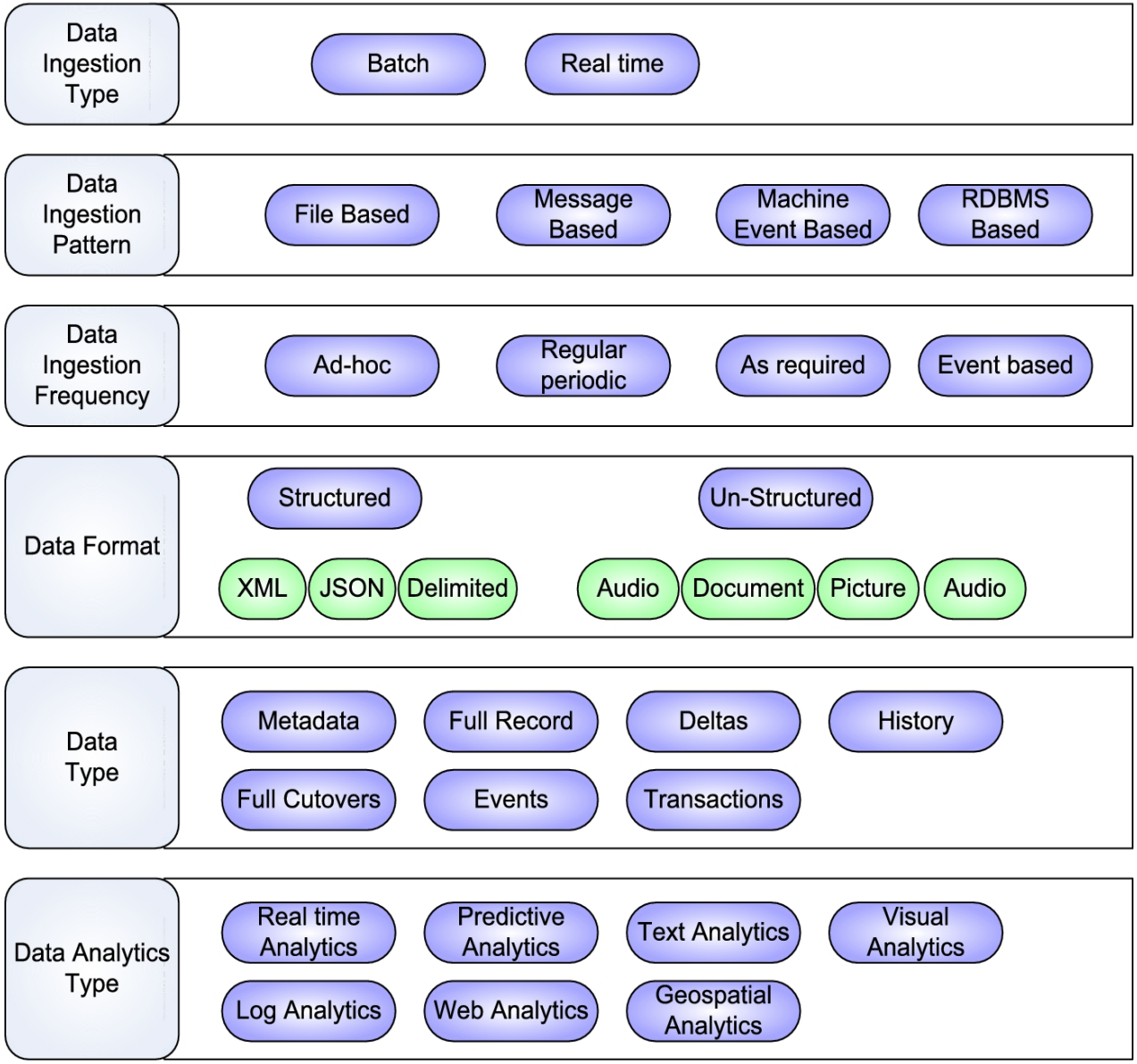
© https://vitalflux.com/data-lake-design-principles-best-practices/

Data Lake Architecture



© Architecting Data Lakes Report by Alice La Plante and Ben Sharma

Big Data Processing Framework



Big Data Engineering For Analytics

- **Data Ingestion Type**

Is the data being ingested in a batch process or in real time?

- **Data Ingestion Pattern**

How is the data being ingested? Does it involve transferring files, using messages, connecting to RDBMS or via machine events?

- **Data Ingestion Frequency**

How frequently is the data being ingested?

- **Data Format**

What data formats are involved?

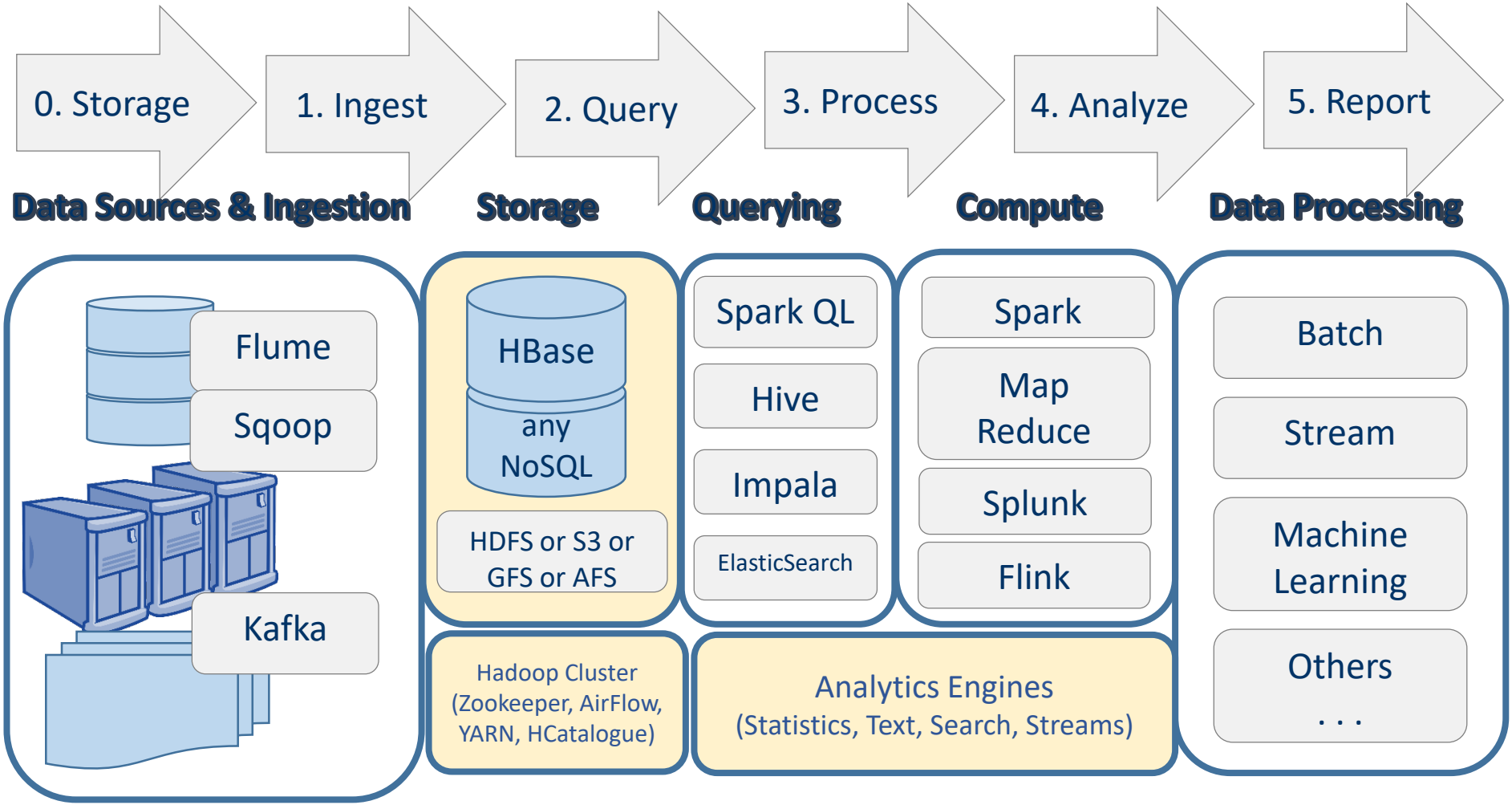
- **Data Analytics Type**

What type of analytics is required on the ingested data? Do we require text, visual, sentimental, or predictive analytics?

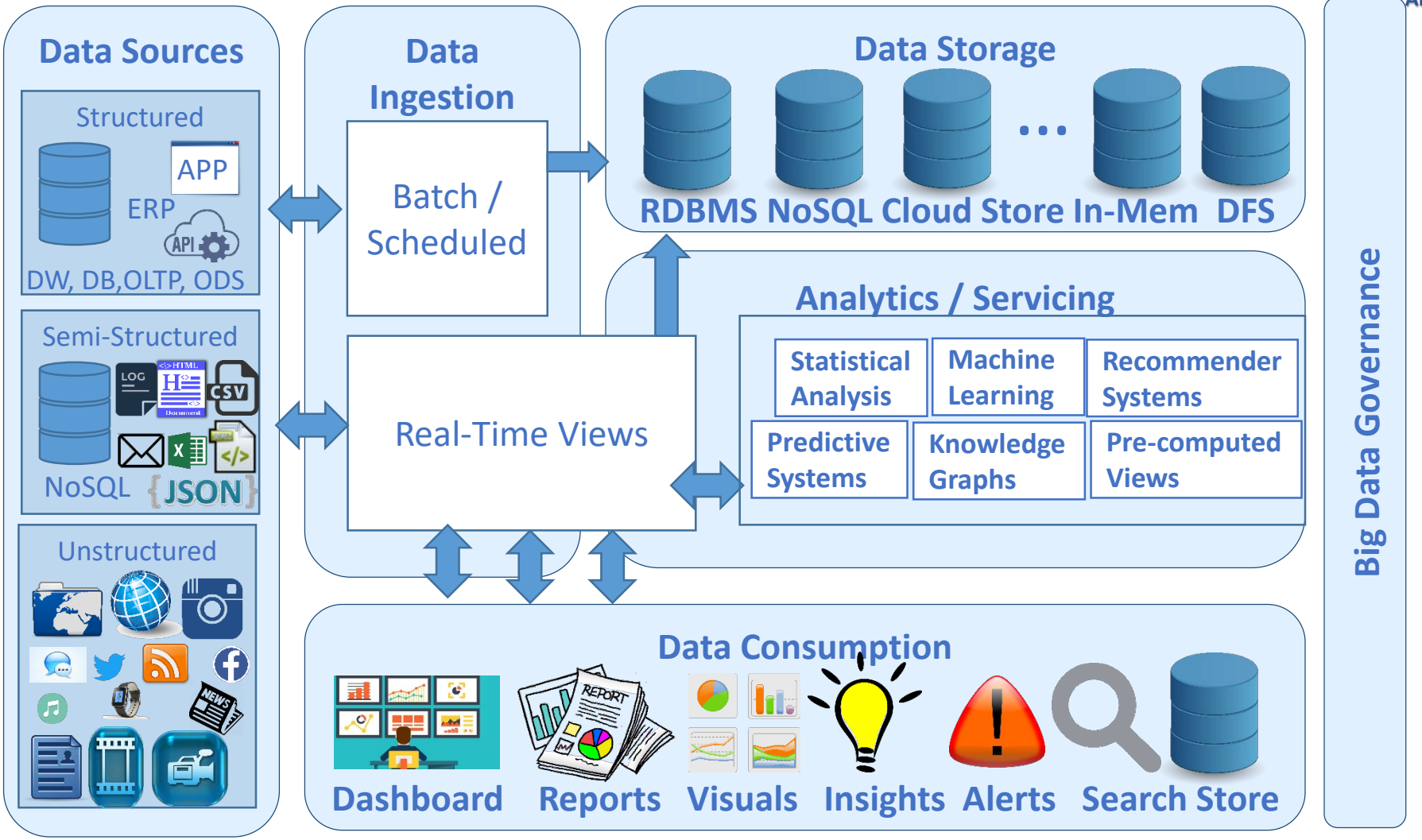
- **Data Types**

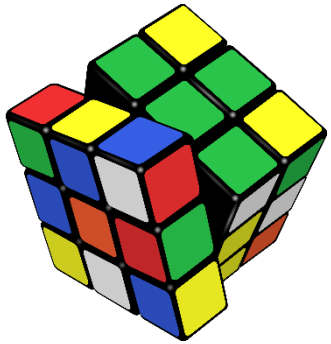
What type of data is involved?

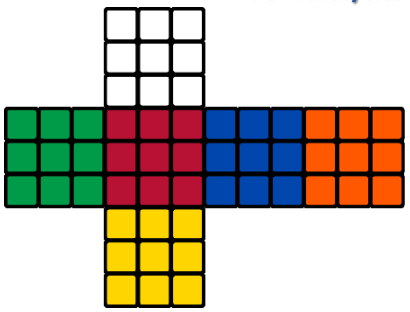
Recall The Processing layers . . .



A Generic Big Data Architecture







(Big) Data Engineering Life Cycle

Stories are just data with a soul. . .

Dr. Brené Brown

Processing Spectrum

Infrastructure

- Storage
  - RDBMS
  - File Store
  - Object Store
  - NoSQL
    - Column Family
    - Document
    - Key Value
    - In Memory
  - NewSQL

Analytics Algorithms

- Cognitive
- Prescriptive
- Predictive
- Diagnostic
- Descriptive

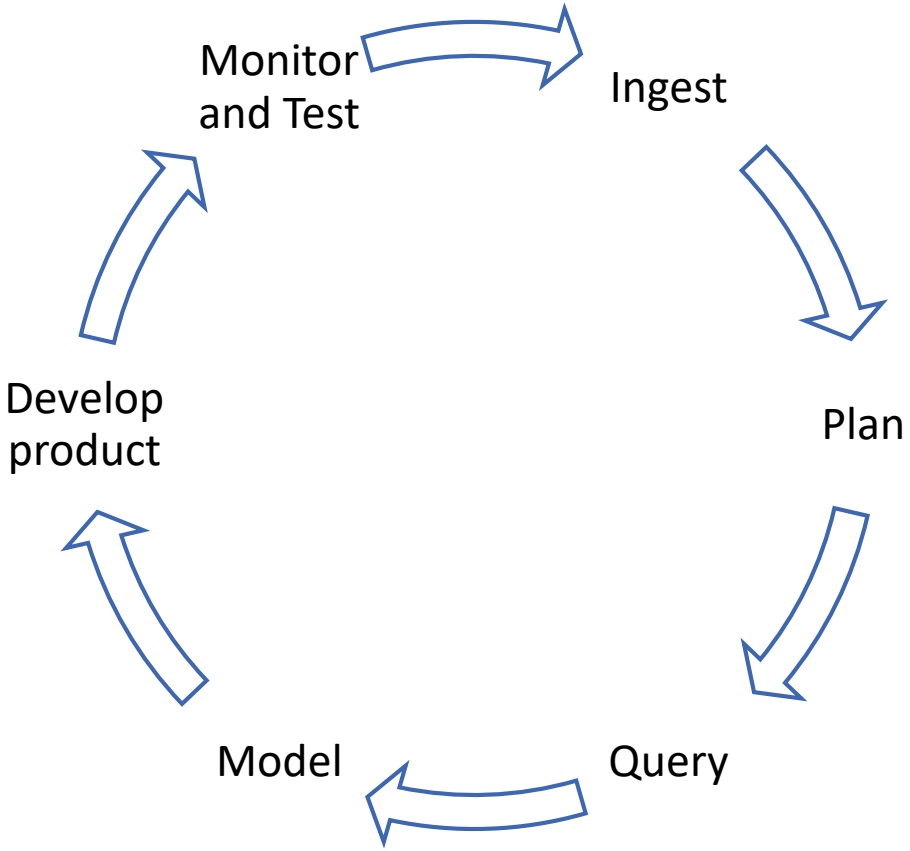
AI and Machine Learning Models

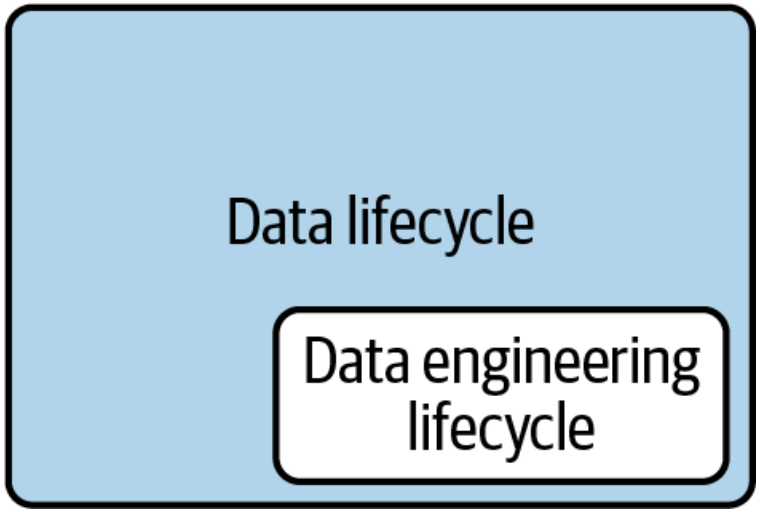
- Machine Learning
  - Ensemble
  - Recommender
  - Predictors
  - Reinforced
- Deep learning
  - Vision
  - Speech
  - Cognitive
  - NLP
- Quantum
- Synthetic Media
- Horizontal AI
- Smart Systems

Applications Spectrum

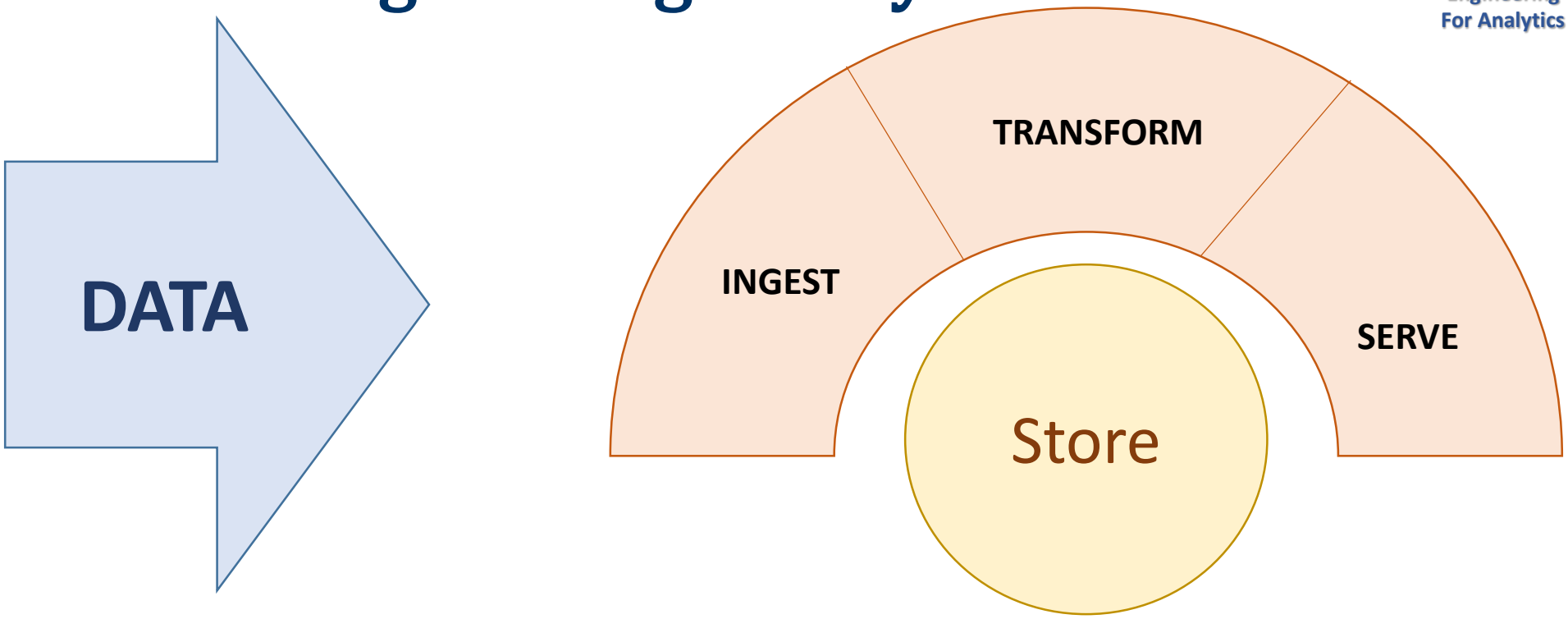
- Domain
  - Finance
  - Healthcare
  - Insurance
  - Commerce
  - Logistics
  - Marketing
  - Sales
  - CRM
  - Education
  - Human Capital
  - Transportation
  - Agriculture
  - Life Sciences
- B2B B2C
- Platforms

Data Engineering Life Cycle





Data Engineering Life Cycle



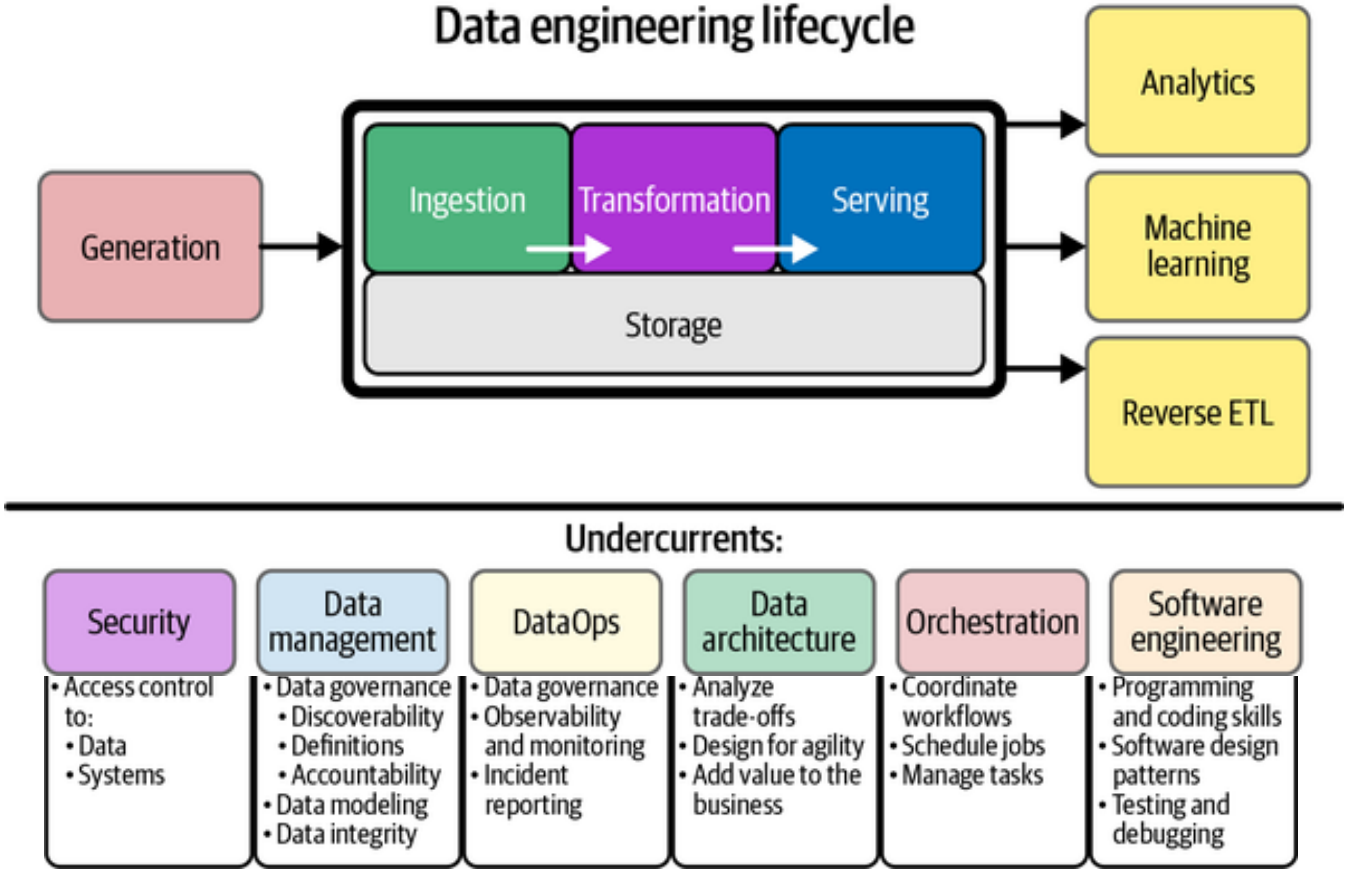
Data Management

Data Orchestration

Data Engineering and DataOps

Software Engineering and DevOps

Components and undercurrents of the data engineering lifecycle



(c) Fundamentals of Data Engineering by Joe Reis, Matt Housley Published by O'Reilly Media, Inc.

Principles of Data Engineering

- Choose common components wisely.
- Plan for failure.
- Architect for scalability.
- Architecture is leadership.
- Always be architecting.
- Build loosely coupled systems.
- Make reversible decisions.
- Prioritize security.
- Embrace Automation - DevOps, DataOps, MLOps, FinOps.

On Premise or Colocation Services or Cloud Services?

Distributed Computing Infrastructure

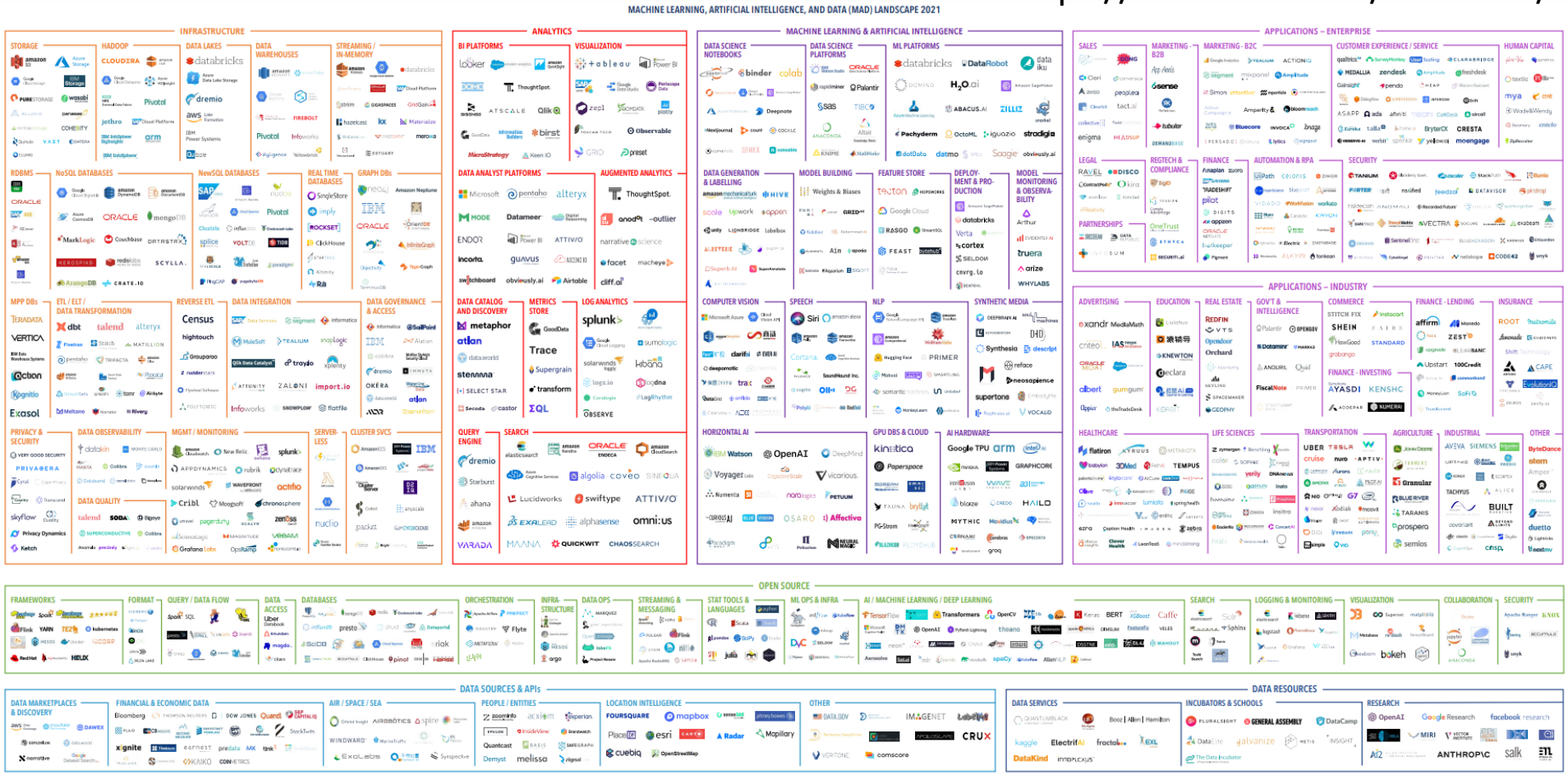
Who owns the machines?

Who deploys the software?

How does scaling happen?

Spoiled for choices

https://mattturck.com/data2021/



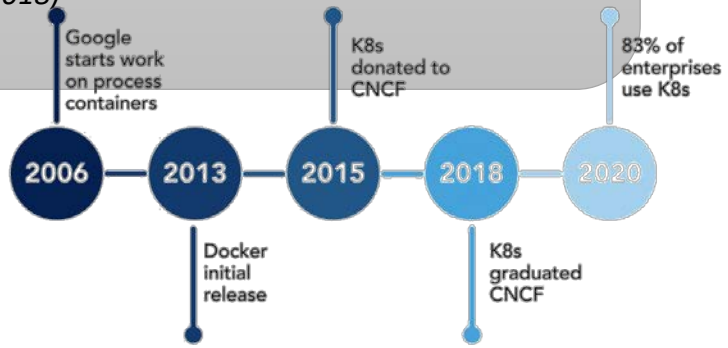
Generations of Data Engineering

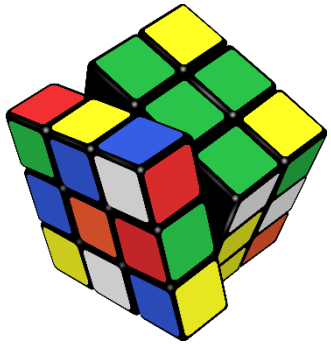
Hadoop Era

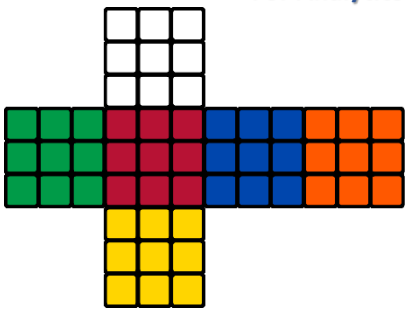
Till 2018

Kubernetes Era

Since k8s graduated from CNCF (2018)







Hadoop Eco System

It seems probable that once the machine thinking method had started, it would not take long to outstrip our feeble powers… They would be able to converse with each other to sharpen their wits. At some stage therefore, we should have to expect the machines to take control.

Alan Turing

The Basic Hadoop Blocks

HDFS

- A distributed file system designed to run on commodity hardware.

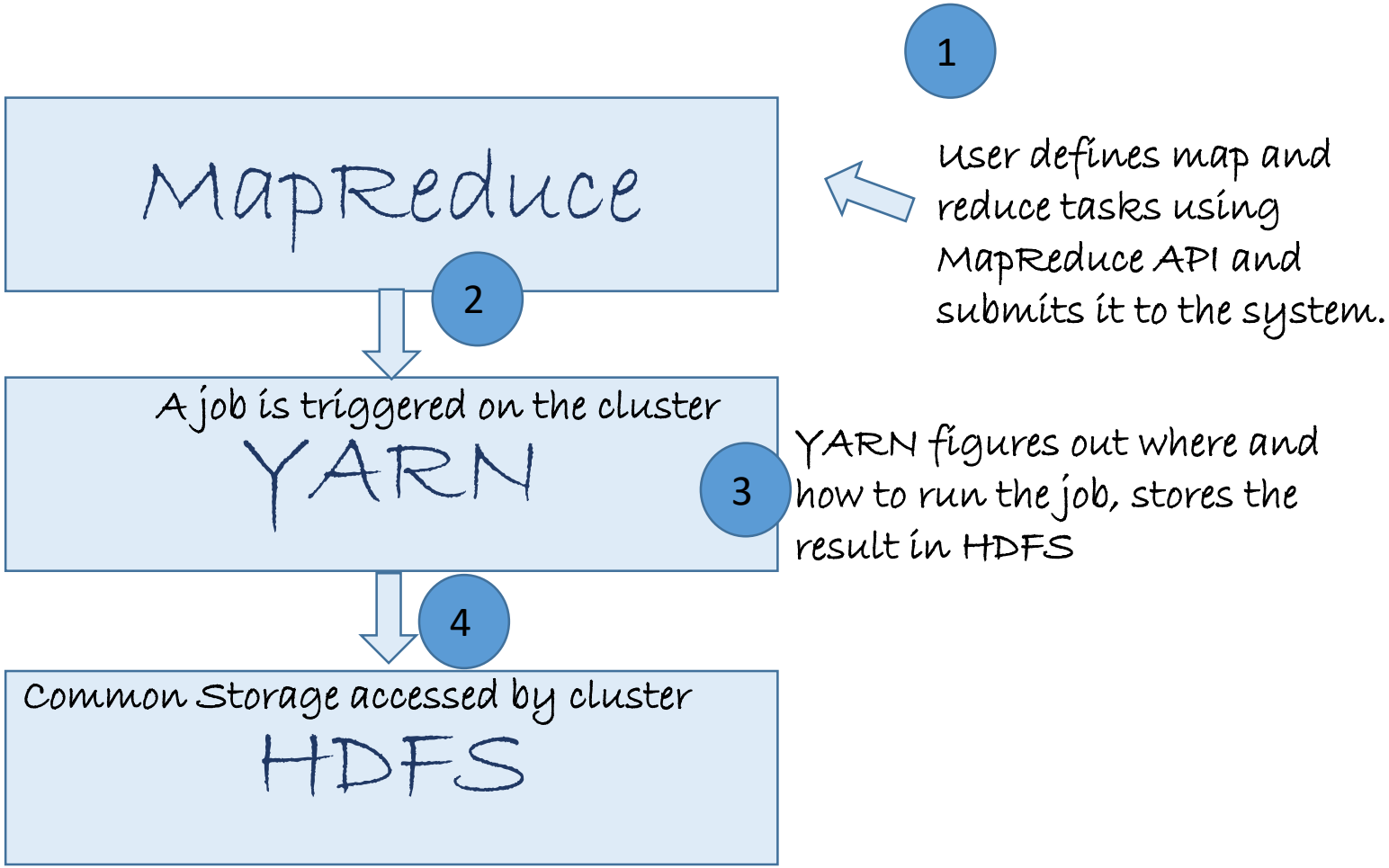
Map Reduce

- A framework to define a data processing task

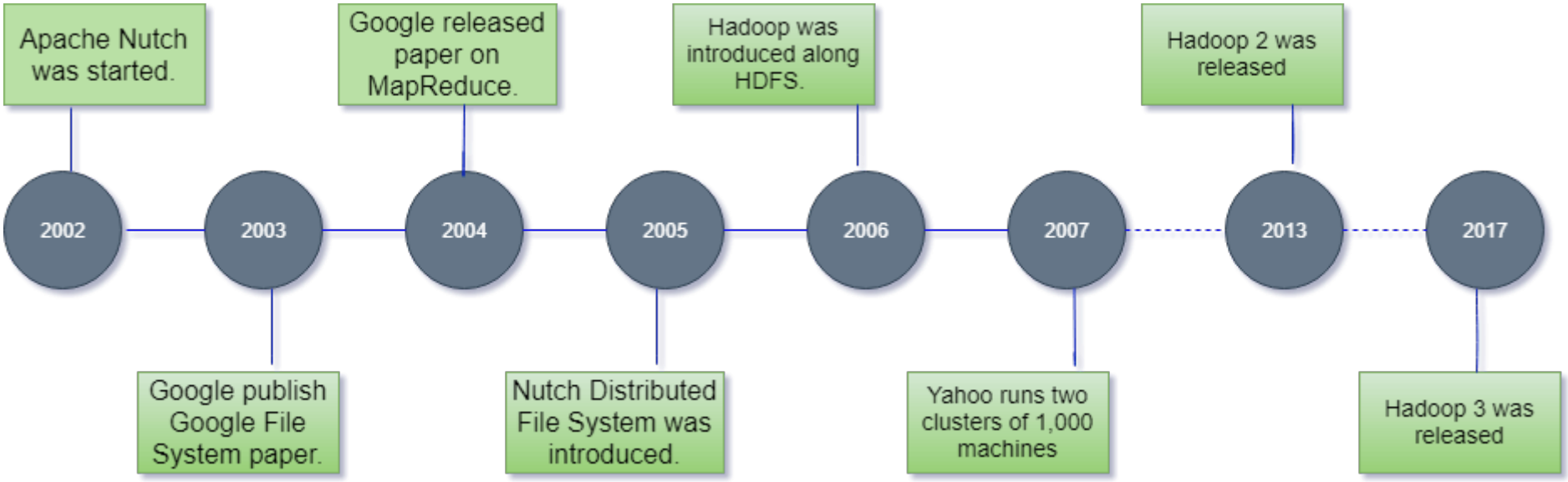
YARN

- Yet Another Resource Negotiator – A framework to run the data processing task

Coordination between Hadoop Blocks



Apache Hadoop History

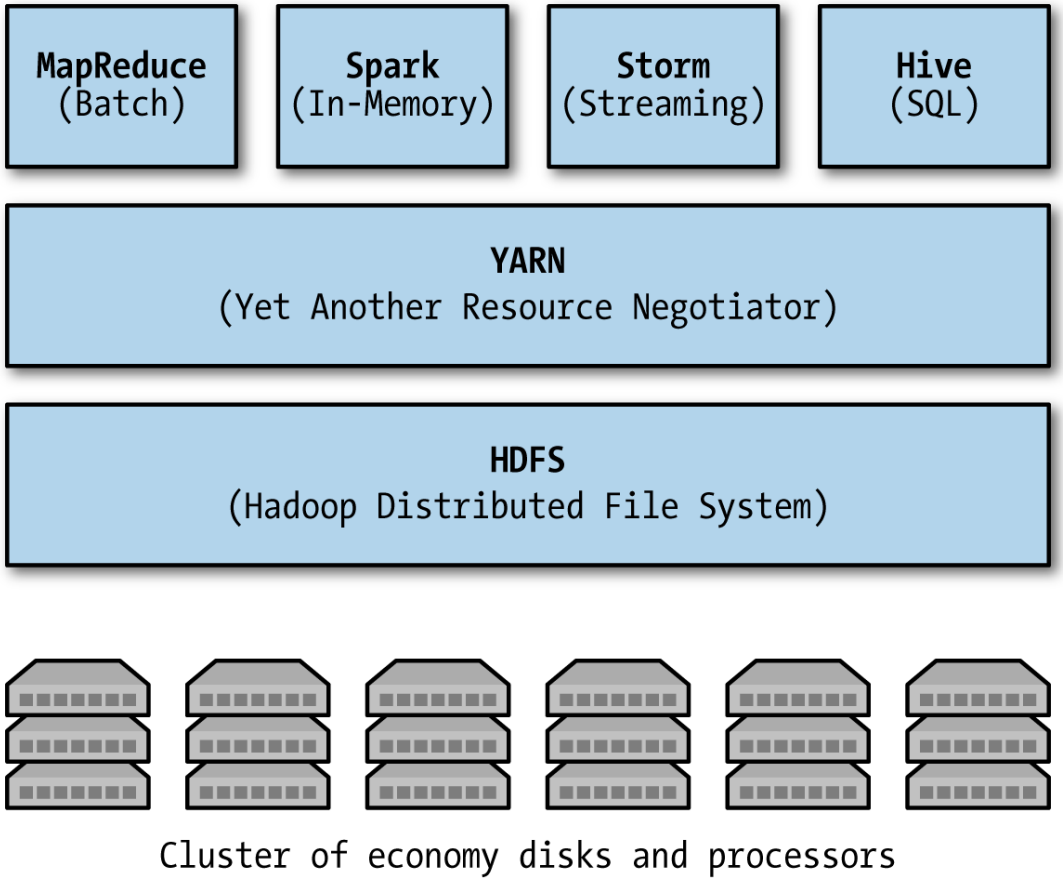


Advantages of Apache Hadoop

- **Low cost—Runs on commodity hardware:**
  - Can run on average performing commodity hardware
  - helps in controlling cost at the same time achieve scalability and performance
  - Adding or removing nodes from the cluster is simple
- **Storage flexibility**:
  - Can store data in raw format, process unstructured and semi-structured data better
- **Open source community:**
  - Supported by many contributors with a growing network of developers worldwide
  - Organizations such as Yahoo, Facebook, Cloudera, Horton works etc have contributed.
- **Fault tolerant**:
  - Massively scalable, reliable in terms of data availability and fault tolerant
  - Architecture assumes that nodes can go down and focusses on process of data
- **Complex data analytics**:
  - Data science has also grown leaps and bounds
  - Complex and heavy computation intensive algorithms need scalable algorithms for a very large-scale data for speed

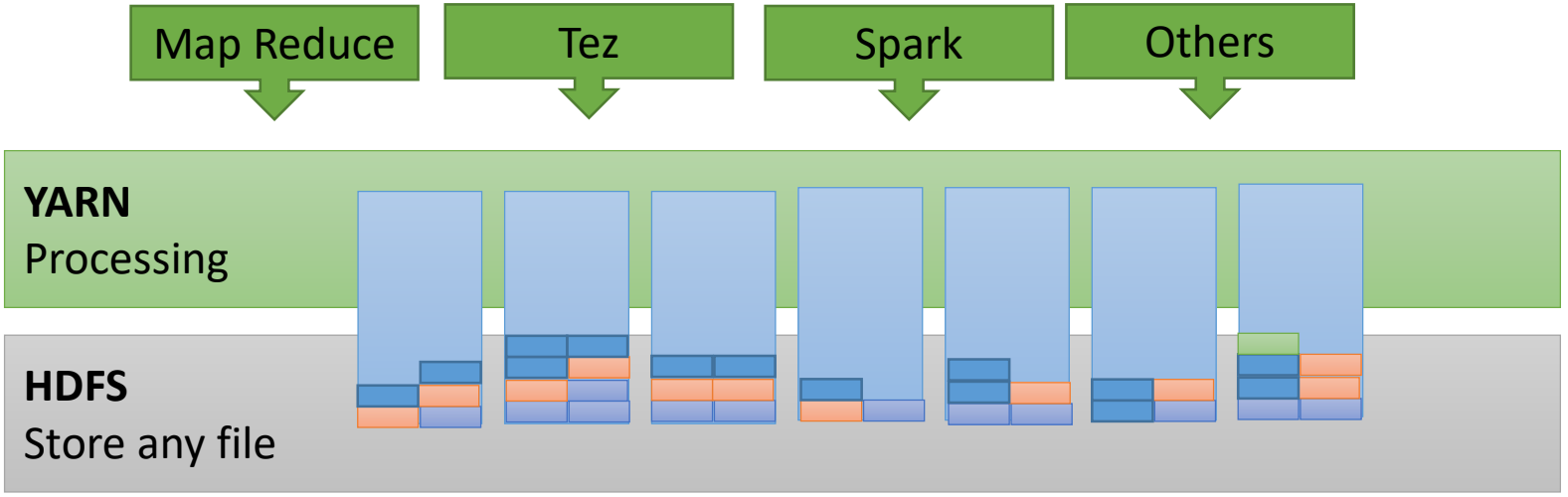
Big Data Processing with Hadoop

- Hadoop is composed of two primary components: HDFS and YARN.
  - HDFS is the Hadoop Distributed File System, responsible for managing data stored on disks across the cluster.
  - YARN acts as a cluster resource manager, allocating computational assets to applications that wish to perform a distributed computation.



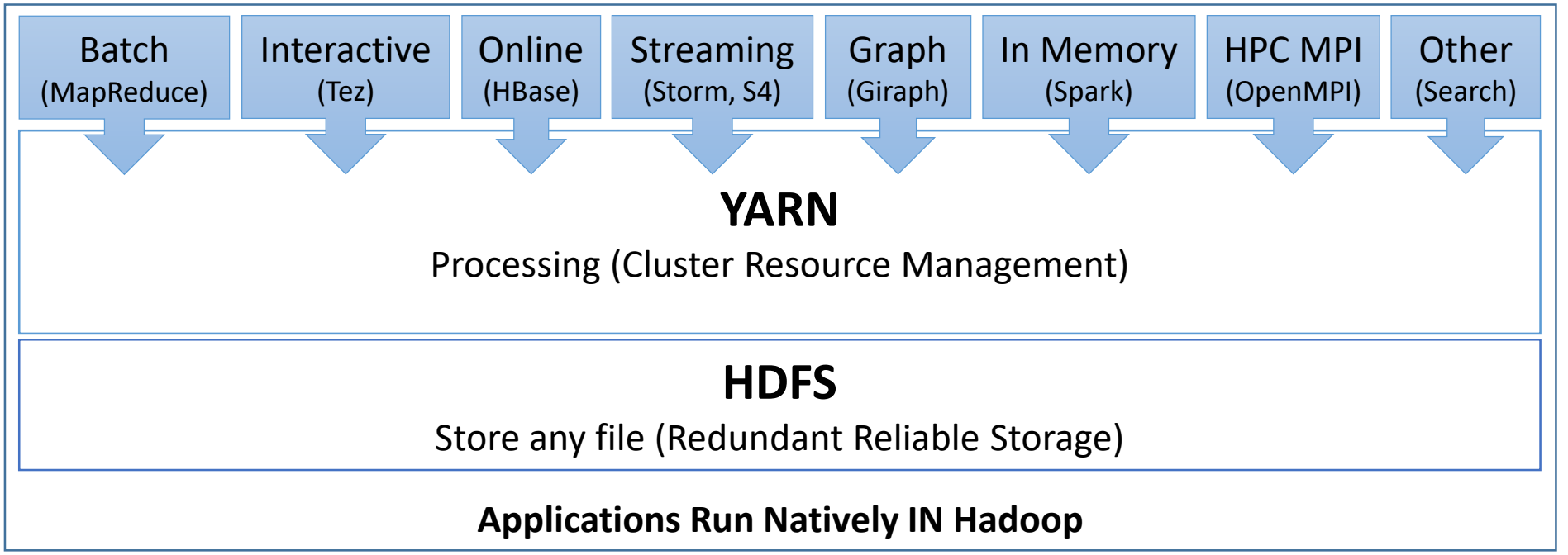
Hadoop Is. . .

- A generic software framework
  - For distributed Storage of Large Data Sets
  - And distributed processing of many concurrent tasks
  - On clusters built from commodity hardware
  - Designed to handle failures automatically

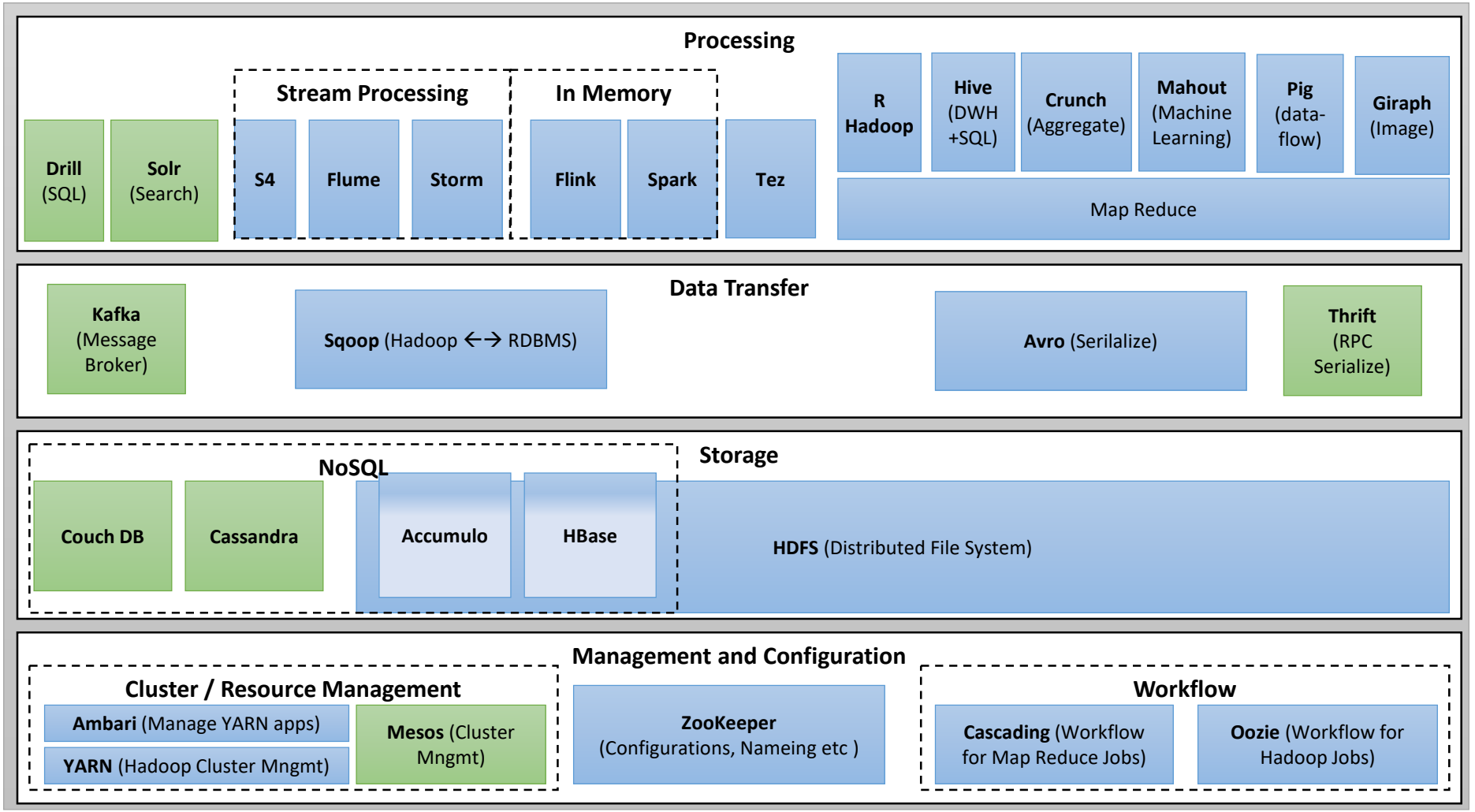


Core Hadoop Platform

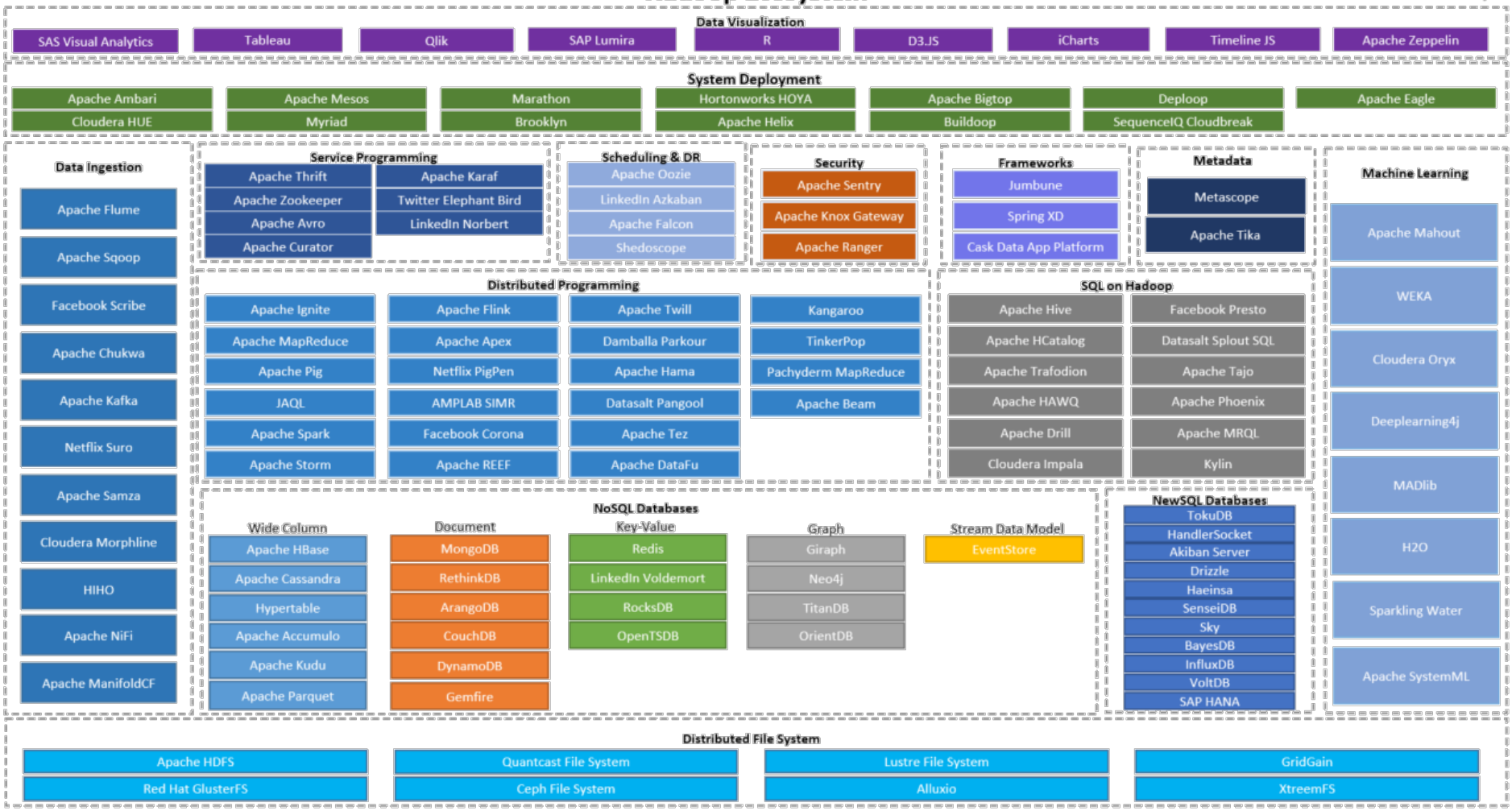
- **Hadoop Distributed File System (HDFS)**: A distributed file system that provides high-throughput access to application data
- **Hadoop YARN**: A framework for job scheduling and cluster resource management
- **Hadoop MapReduce**: A YARN-based system for parallel processing of large datasets



Hadoop Eco System

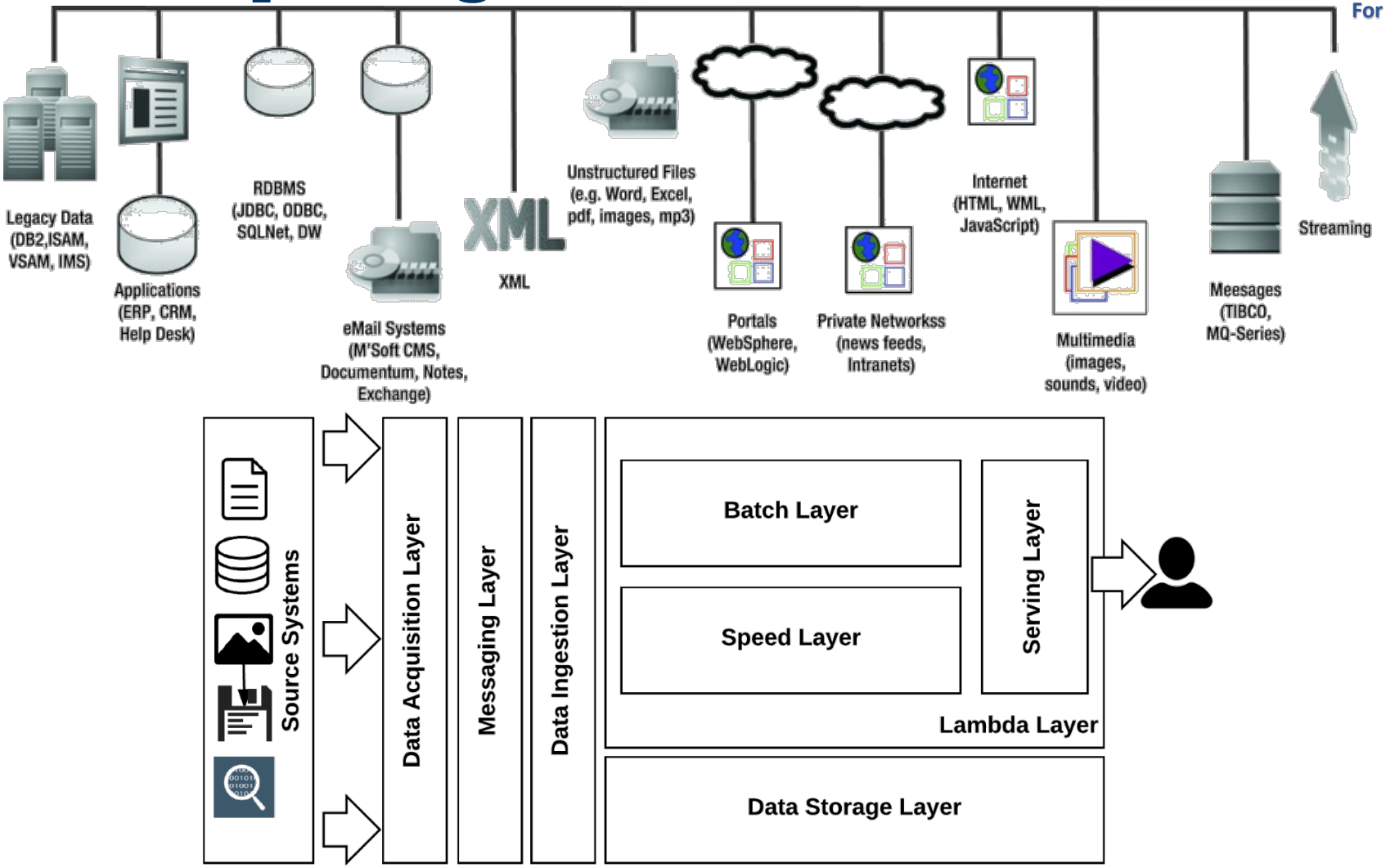


Eco System



Source: https://mydataexperiments.com/2017/04/11/hadoop-ecosystem-a-quick-glance/ http://hadoopecosystemtable.github.io/

Hadoop Design Considerations



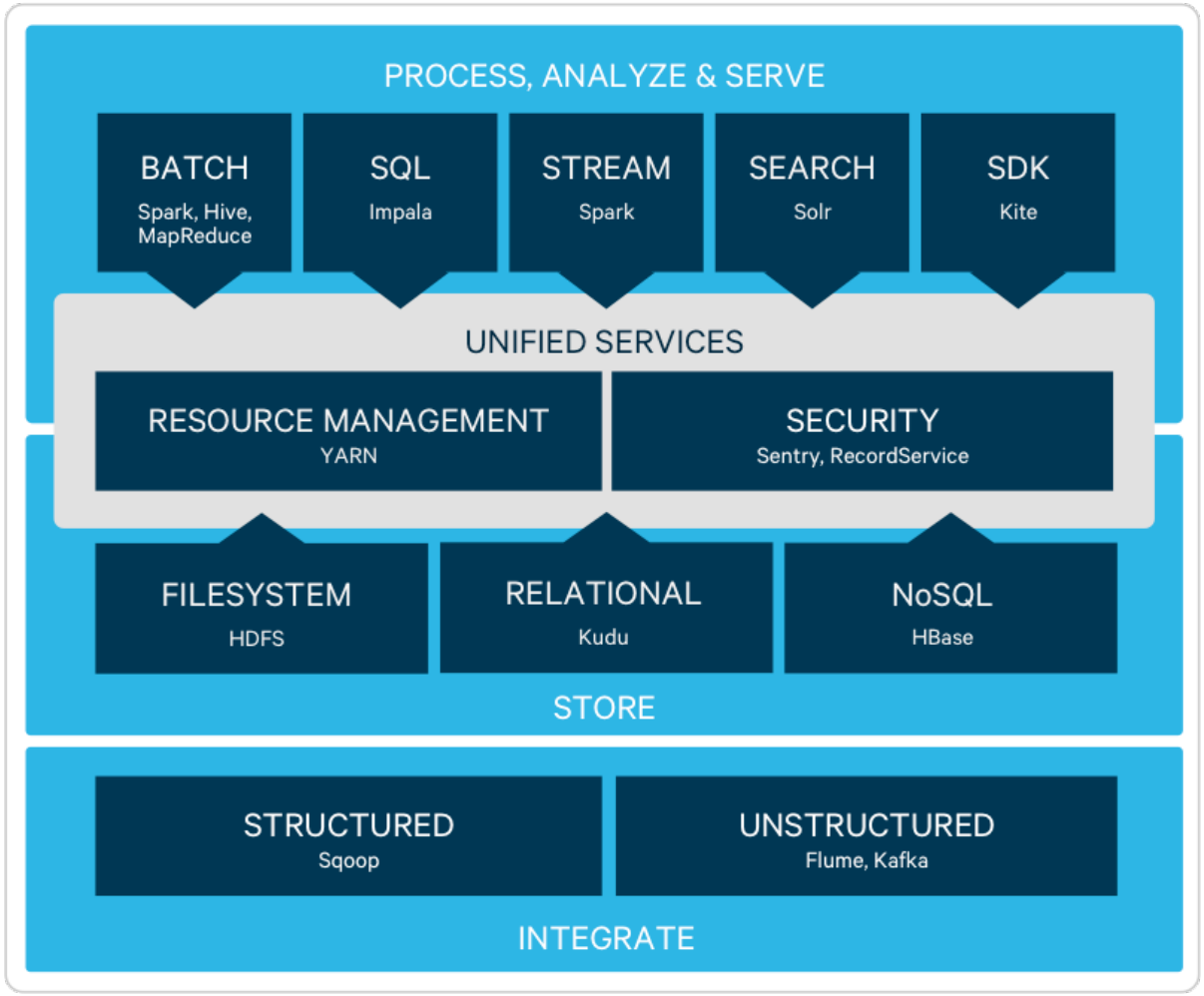
Hadoop Distributions Available

Big Data Engineering For Analytics

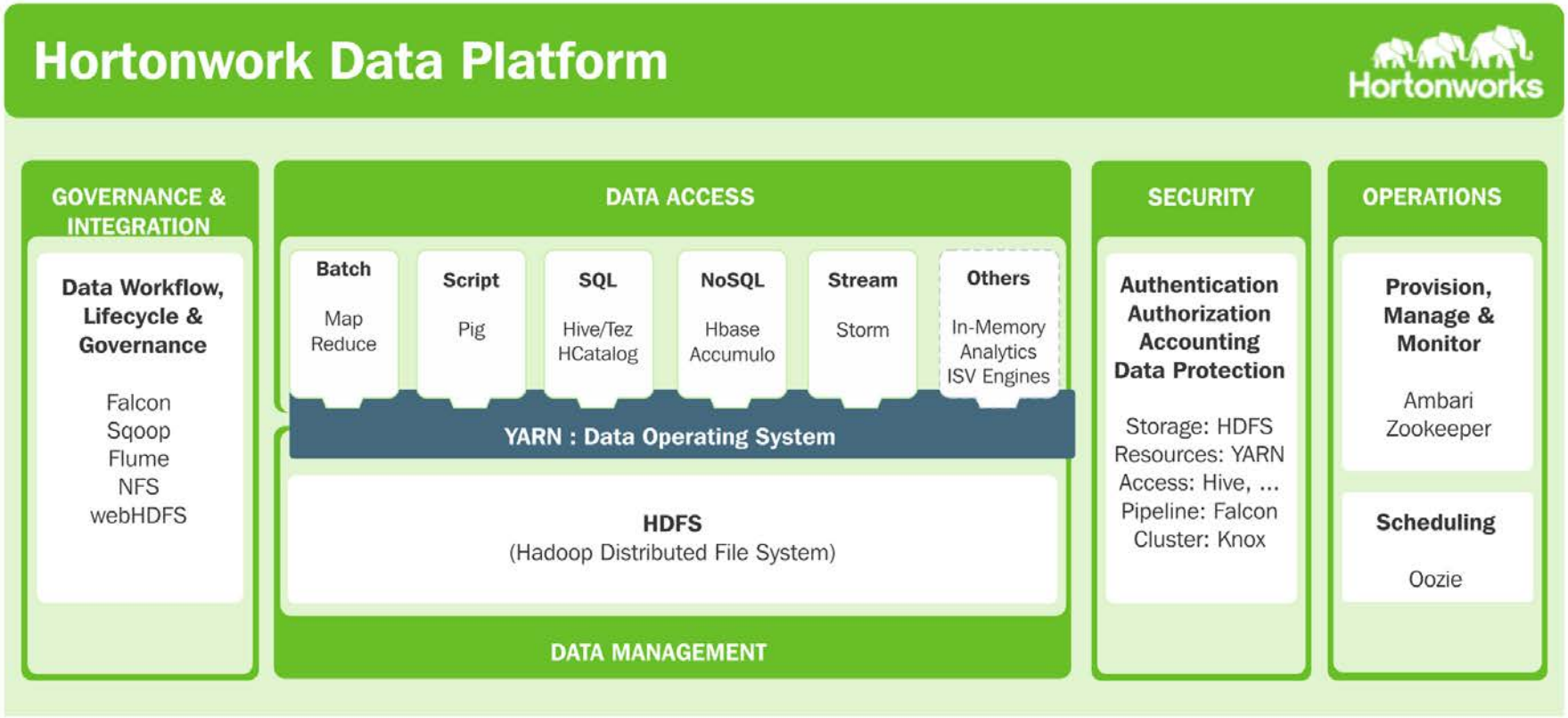
- **Cloudera**:
  - The most widely used Hadoop distribution with the biggest customer base as it provides good support and has some good utility components such as Cloudera Manager, which can create, manage, and maintain a cluster, and manage job processing, and Impala is developed and contributed by Cloudera which has real-time processing capability.
- **Hortonworks**:
  - Hortonworks uses an open source Hadoop. Ambari was developed and contributed to Apache by Hortonworks. Hortonworks offers a very good, easy-to-use sandbox for getting started. Hortonworks contributed changes that made Apache Hadoop run natively on the Microsoft Windows platforms including Windows Server and Microsoft Azure.
- **DataBricks:**
  - Databricks grew out of the AMPLab project at University of California, Berkeley that was involved in making Apache Spark, a distributed computing framework built atop Scala.
- **MapR**:
  - MapR distribution of Hadoop uses different concepts than plain open source Hadoop and its competitors, especially support for a network file system (NFS) instead of HDFS for better performance and ease of use. In NFS, Native Unix commands can be used instead of Hadoop commands. MapR have high availability features such as snapshots, mirroring, or stateful failover.
- **Amazon Elastic MapReduce (EMR)**:
  - AWS's Elastic MapReduce (EMR) leverages its comprehensive cloud services, such as Amazon EC2 for compute, Amazon S3 for storage, and other services, to offer a very strong Hadoop solution for customers who wish to implement Hadoop in the cloud. EMR is much advisable to be used for infrequent Big Data processing. It might save you a lot of money.

https://en.wikipedia.org/wiki/Category:Big_data_companies

Cloudera Distribution

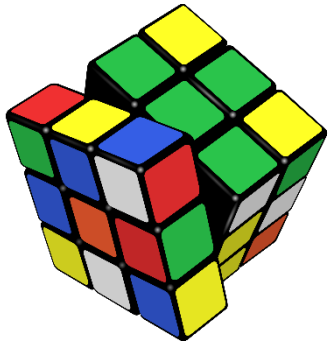


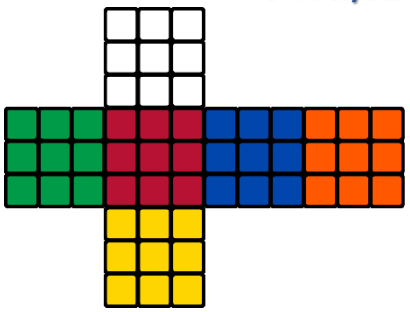
Hortonworks Data Platform



Hadoop Solution Examples

- Extract/Transform/Load (ETL)
- Text mining
- Index building
- Graph creation and analysis
- Pattern recognition
- Collaborative filtering
- Prediction models
- Sentiment analysis
- Risk assessment
- Log processing
- Recommendation systems
- Business intelligence/data warehousing
- Video and image analysis
- Archiving



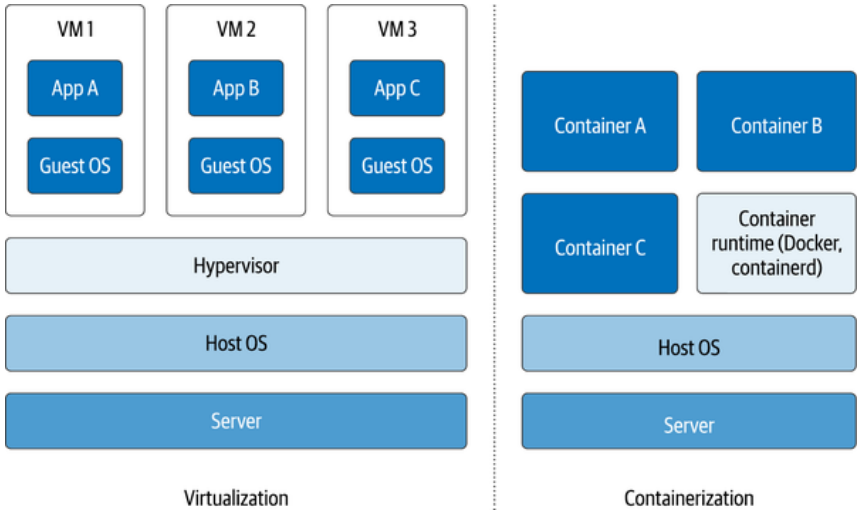


Kubernetes Orchestration

One time I tried to explain Kubernetes to someone. Then we both didn't understand it!!

~Tweet by @SwiftOnSecurity

Comparing containerization to virtualization



Virtual Images run a guest operating system per VM, with a hypervisor layer to implement system calls onto the underlying host operating system.

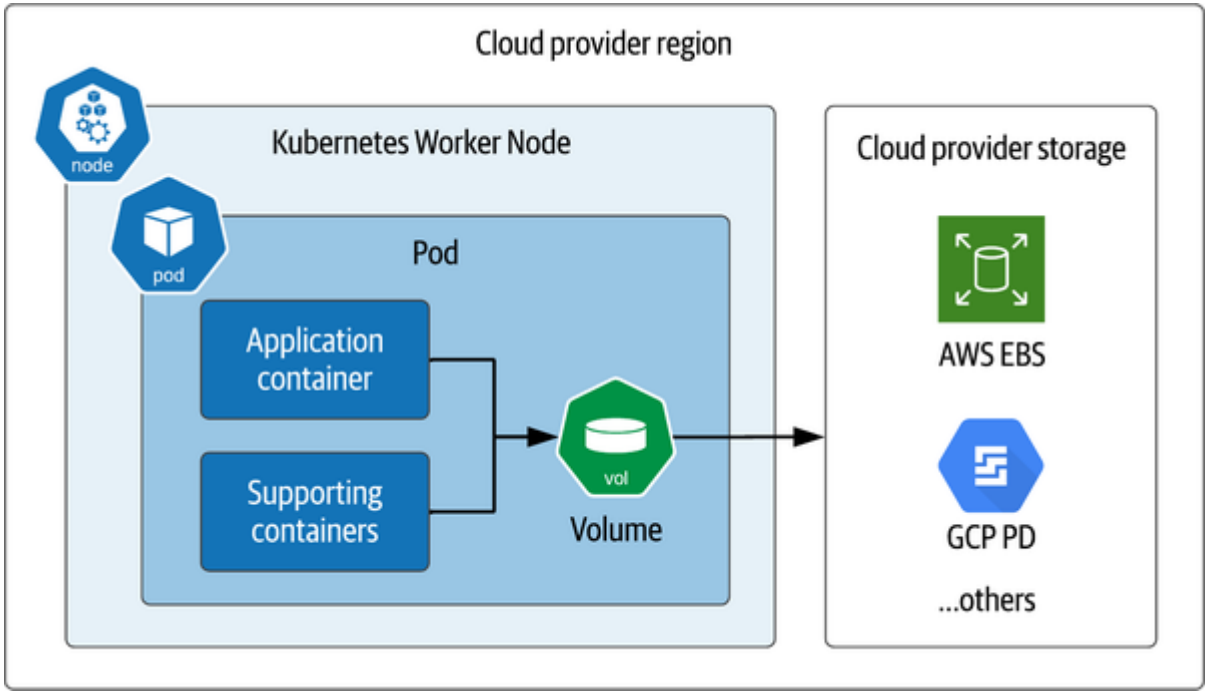
Containers are designed to be disposable and replaceable, so they need to start quickly and use as few resources for overhead processing as possible. For this reason, most container images are built from base images containing streamlined, Linux-based, open-source operating systems.

Benefits of Big Data on Kubernetes

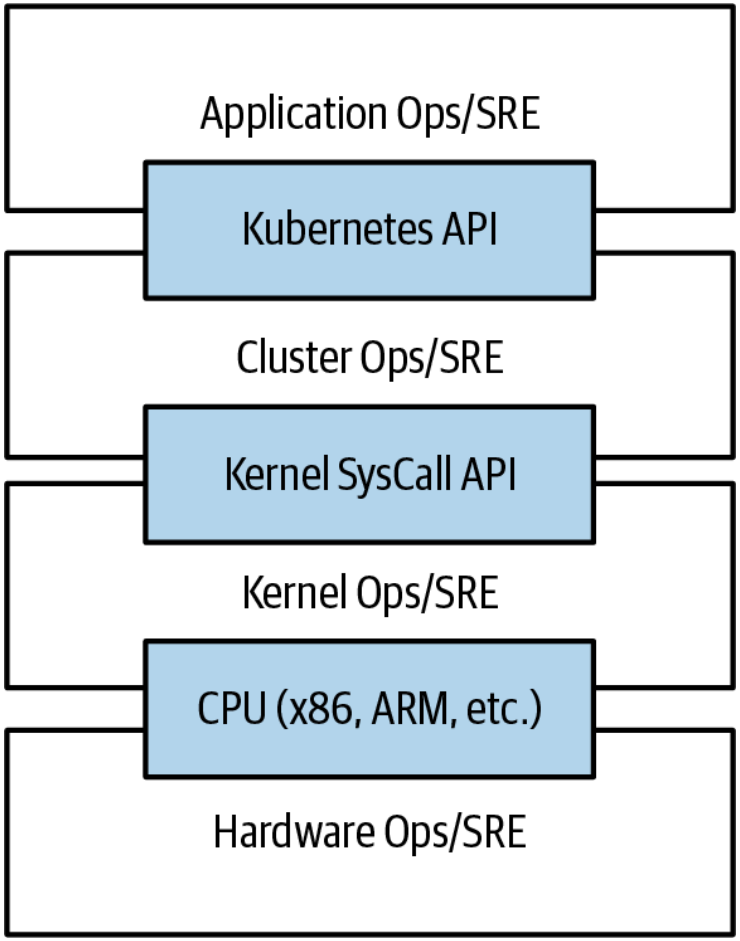
- Support multiple standby NameNodes.
- Supports multiple NameNodes for multiple namespaces.
- Storage overhead reduced.
- Support GPUs.
- Intra-node disk balancing.
- Support for Opportunistic Containers and Distributed Scheduling.
- Support for Data Lake and Object Storage System via file-system connectors.

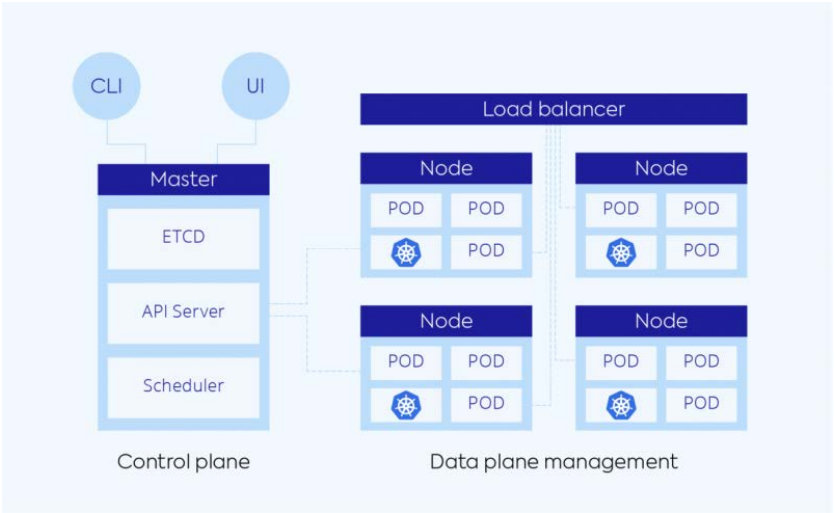
Kubernetes Resources for Data Storage

- Kubernetes provides to help manage the three commodities of cloud computing: ***compute***, ***network***, and ***storage***.
- Kubernetes resources that can be used to manage stateful workloads, including *Volumes, PersistentVolumes, PersistentVolumeClaims, and StorageClasses*.
- Kubernetes also helps in managing persistence in containerized applications in general.



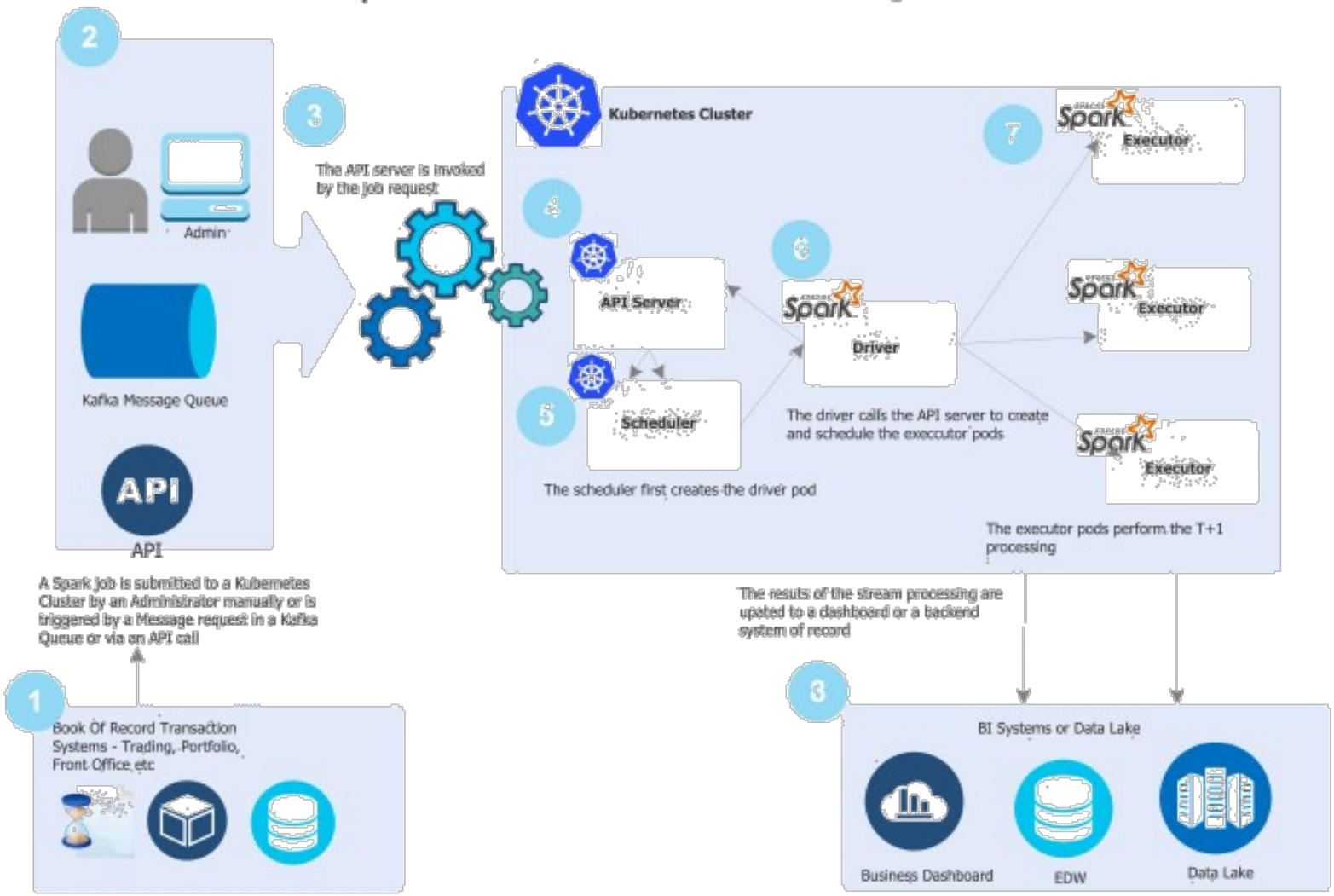
Kubernetes Separation of Concern



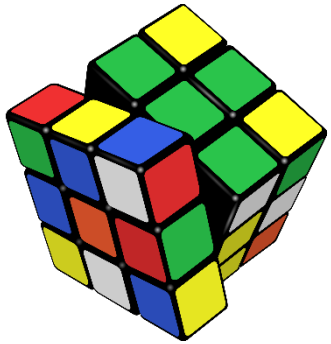


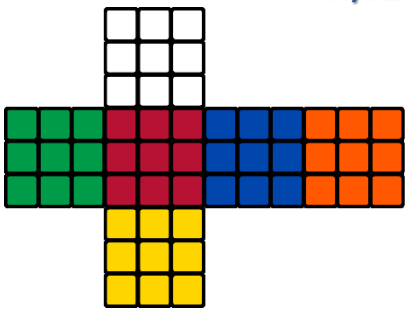
Kubernetes offers users a way to automate many of the manual tasks involved with operating containers such as autoscaling, resiliency management, metrics monitoring and more.

Spark and Kubernetes Integration



https://www.vamsitalkstech.com/architecture/big-data-kubernetes-a-reference-architecture-for-spark-with-kubernetes-2-4/





Commercial Cloud Products

Mathematical reasoning may be regarded rather schematically as the exercise of a combination of two facilities, which we may call intuition and ingenuity.

Alan Turing

Amazon Data Lake and Analytics Products

Analytics	Interactive analytics	Amazon Athena
	Big data processing	Amazon EMR
	Data warehousing	Amazon Redshift
	Real-time analytics	Amazon Kinesis Data Analytics
	Operational analytics	Amazon Elasticsearch Service
	Dashboards and visualizations	Amazon QuickSight
	Visual data preparation	Amazon Glue DataBrew

Data movement	Real-time data movement	Amazon Managed Streaming for Apache Kafka (Amazon MSK) | Amazon Kinesis Data Streams | Amazon Kinesis Data Firehose | Amazon Kinesis Video Streams | AWS Glue
Data lake	Object storage	Amazon S3 | AWS Lake Formation
	Backup and archive	Amazon S3 Glacier | AWS Backup
	Data catalog	AWS Glue | AWS Lake Formation
	Third-party data	AWS Data Exchange
Predictive Analytics and Machine Learning	Frameworks and interfaces	AWS Deep Learning AMIs
	Platform services	Amazon SageMaker

Azure Data Lake and Analytics Products

Azure Analysis Services

Enterprise-grade analytics engine as a service

Azure Data Lake Storage

Massively scalable, secure data lake functionality built on Azure Blob Storage

Azure Databricks

Fast, easy, and collaborative Apache Spark-based analytics platform

Azure Synapse Analytics

Limitless analytics service with unmatched time to insight

Data Factory

Hybrid data integration at enterprise scale, made easy

Event Hubs

Receive telemetry from millions of devices

Log Analytics

Collect, search, and visualize machine data from on-premises and cloud

R Server for HDInsight

Predictive analytics, machine learning, and statistical modeling for big data

Azure Data Explorer

Fast and highly scalable data exploration service

Azure Data Share

A simple and safe service for sharing big data with external organizations

Azure Stream Analytics

Real-time analytics on fast moving streams of data from applications and devices

Data Catalog

Get more value from your enterprise data assets

Data Lake Analytics

Distributed analytics service that makes big data easy

HDInsight

Provision cloud Hadoop, Spark, R Server, HBase, and Storm clusters

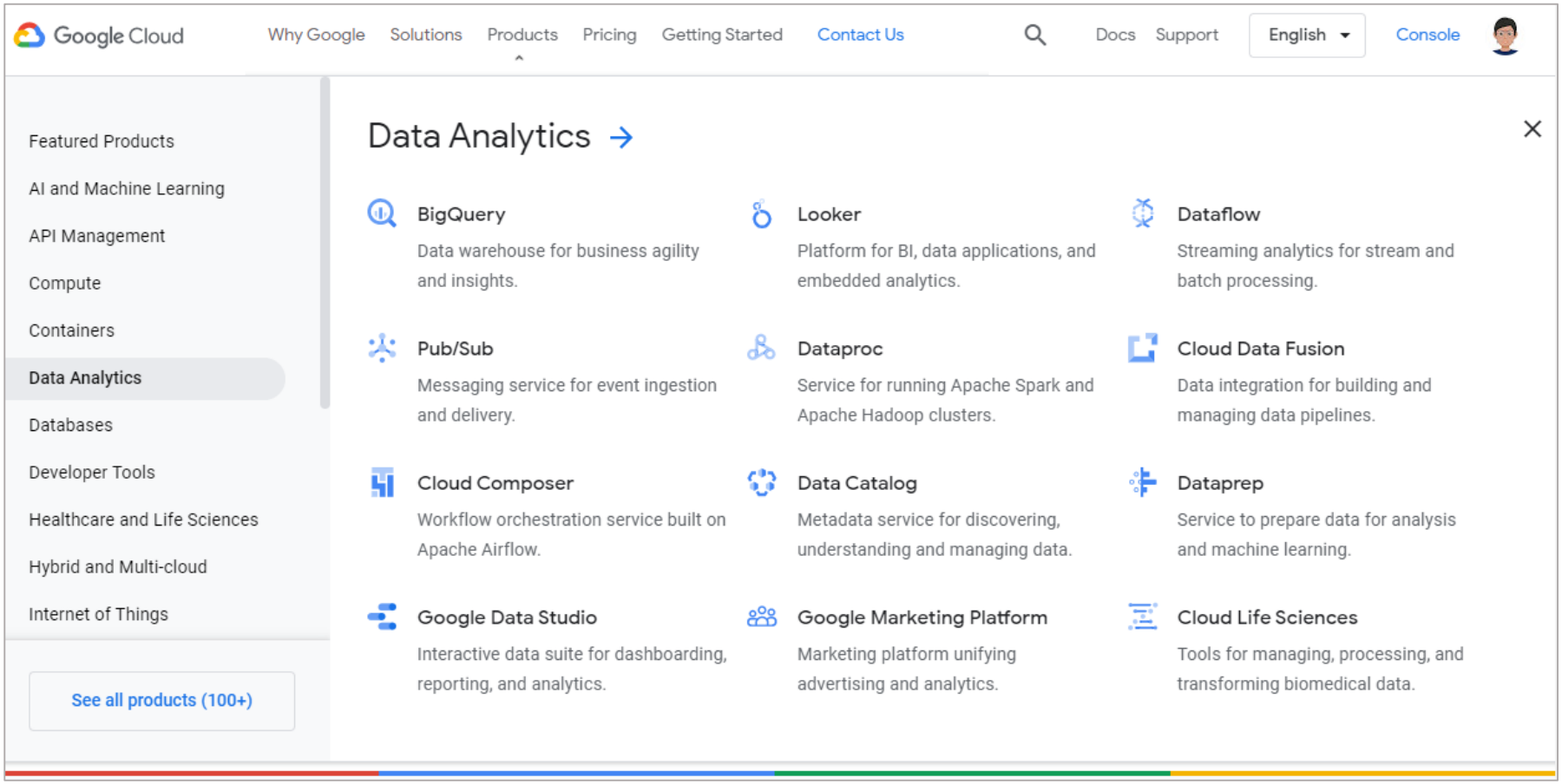
Power BI Embedded

Embed fully interactive, stunning data visualizations in your applications

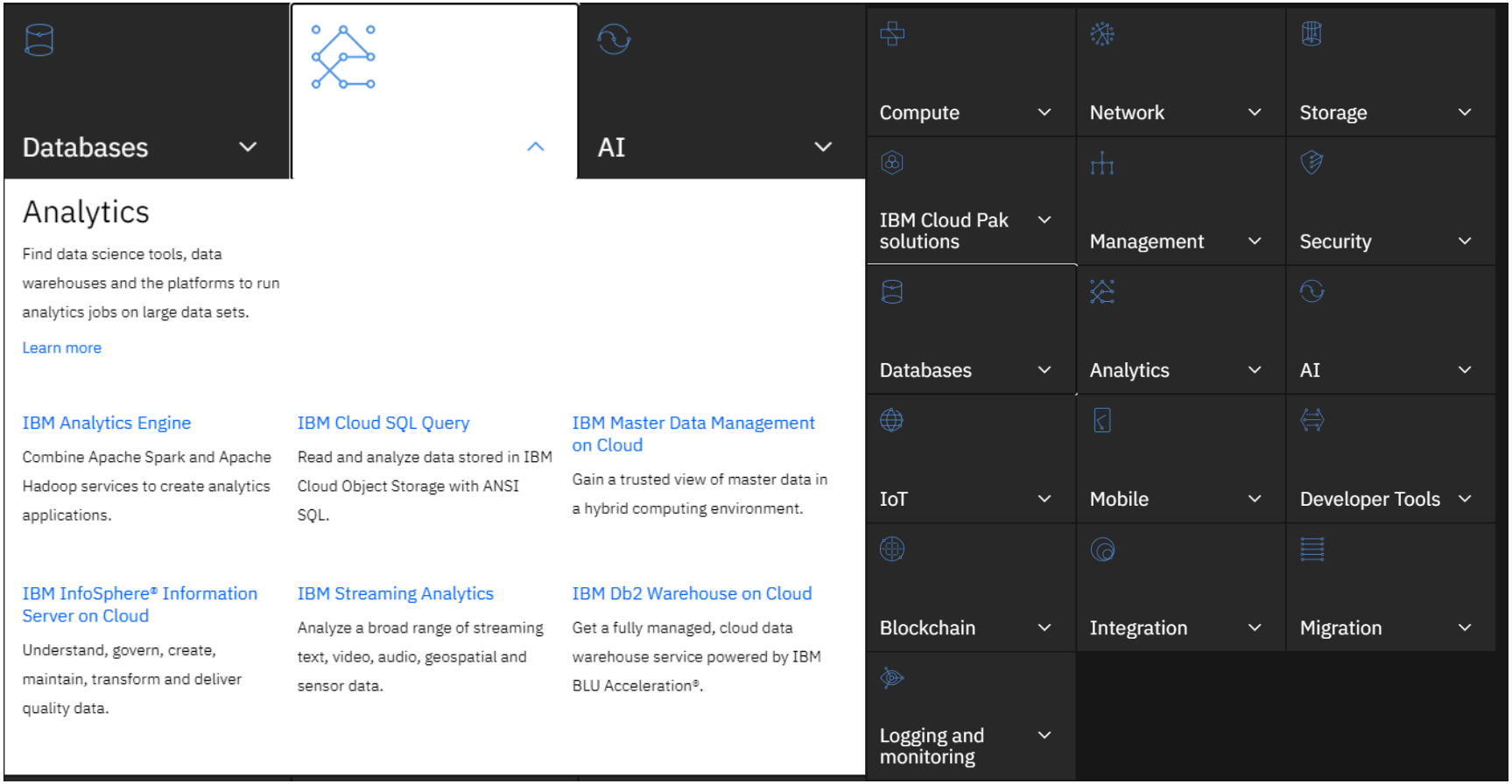
Azure Purview PREVIEW

Maximize business value with unified data governance

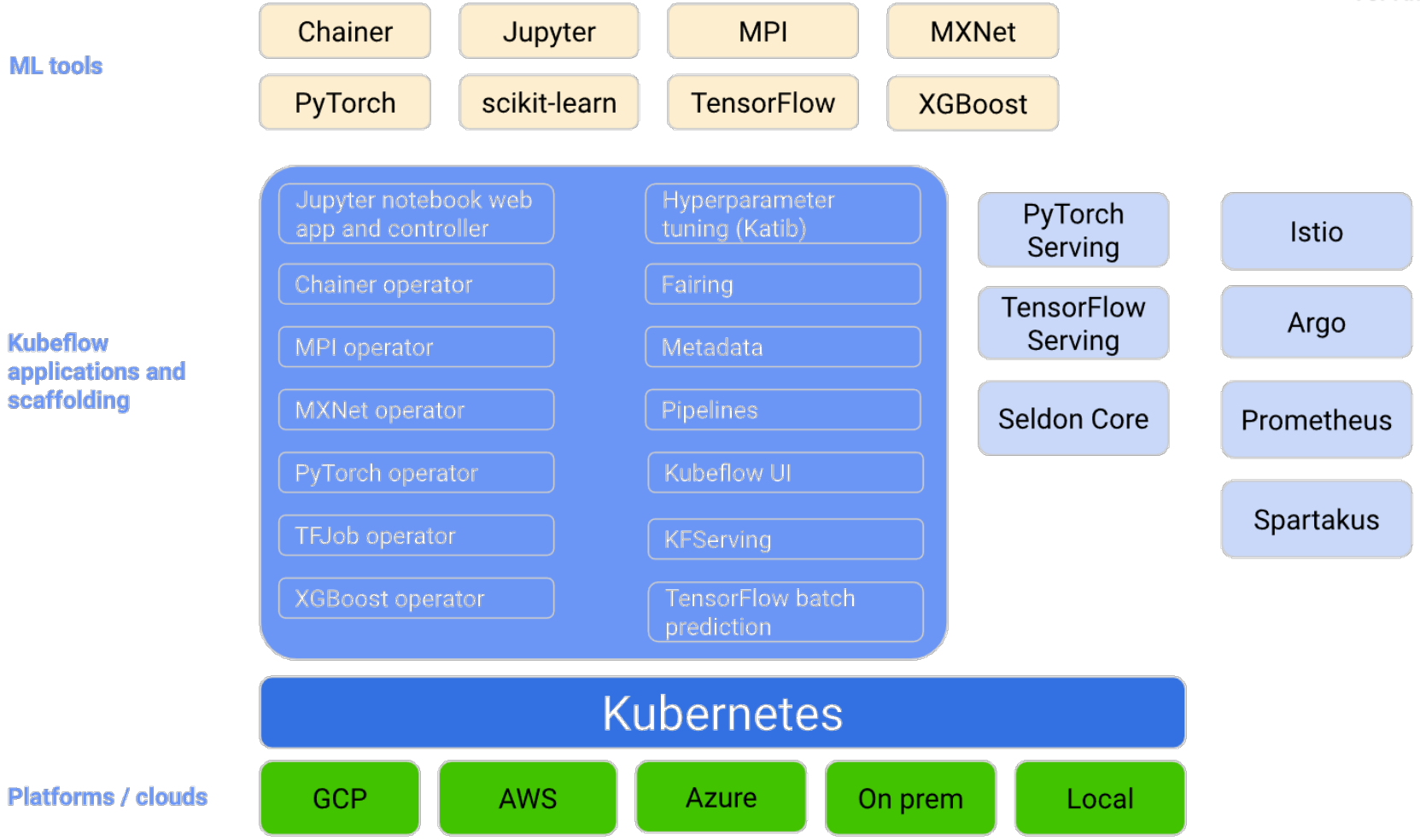
GCP Analytics Products



IBM Analytics Products

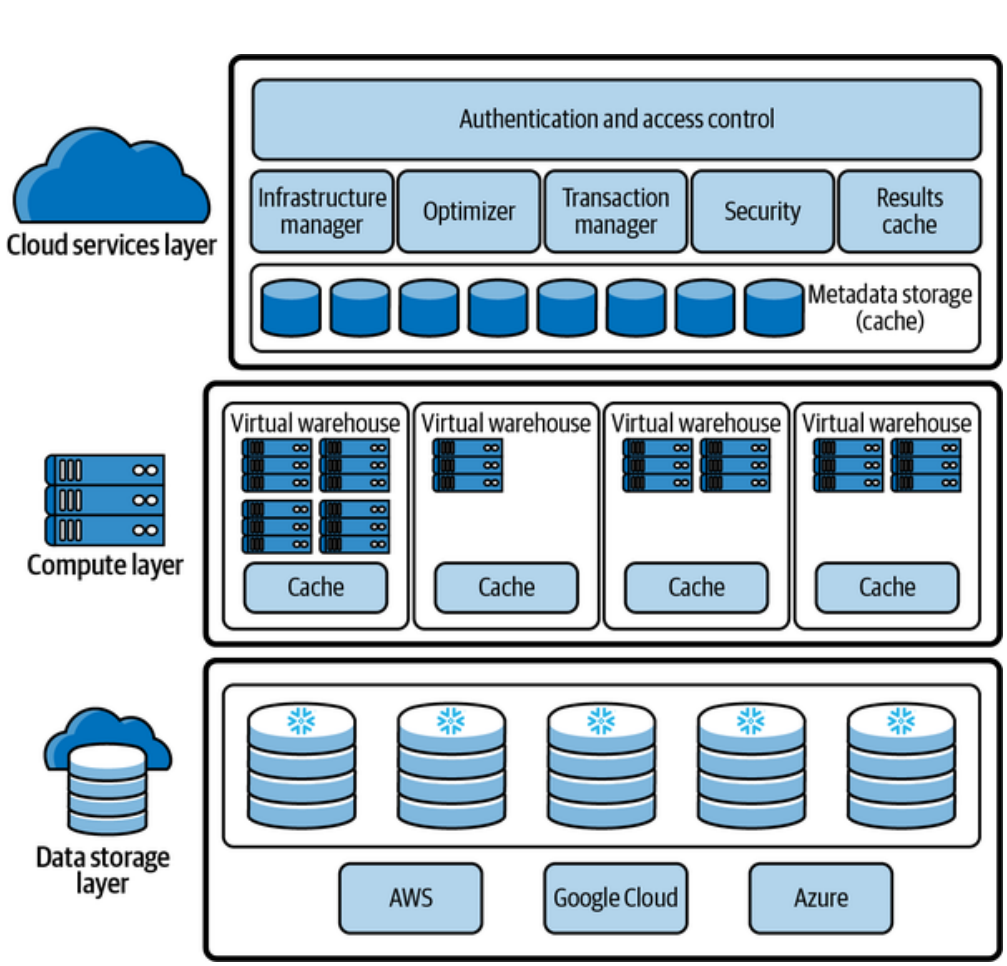


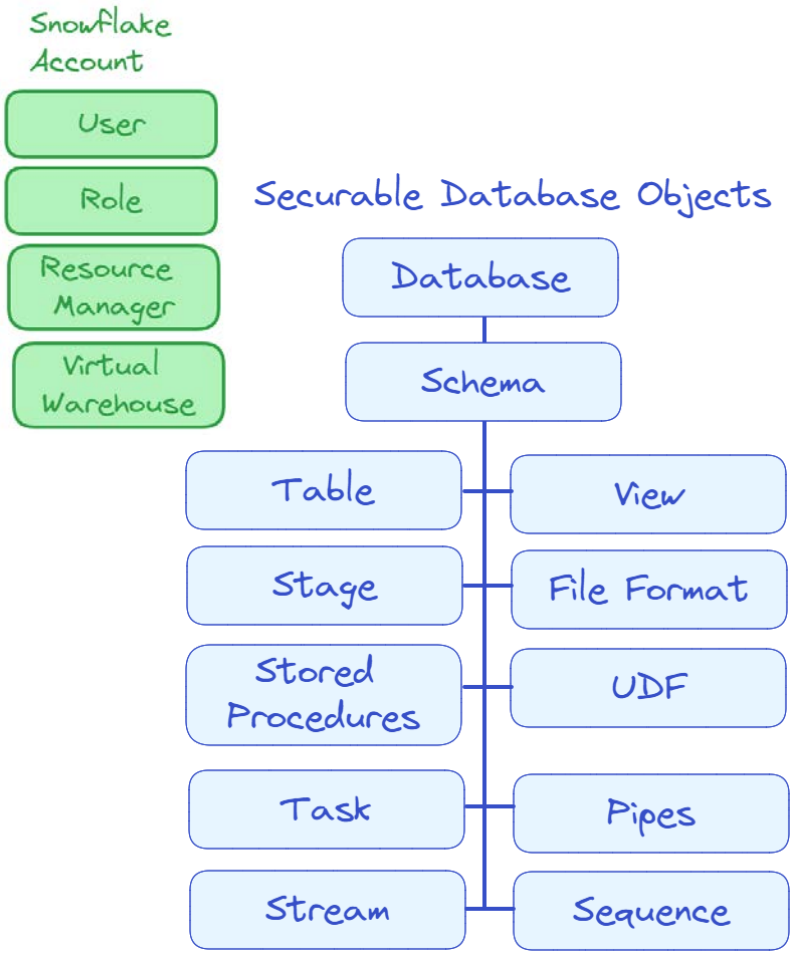
Kubernates : Kubeflow Architecture



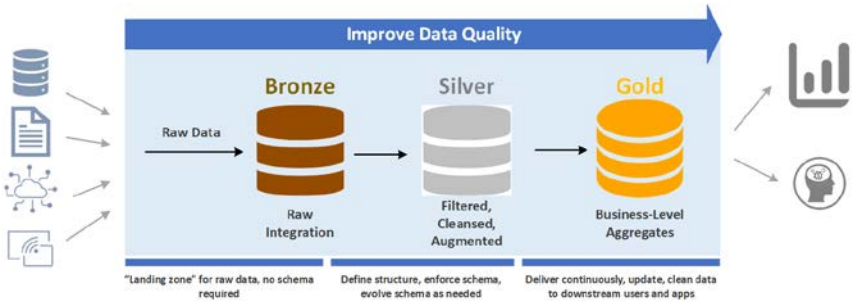
https://www.kubeflow.org/docs/images/kubeflow-overview-platform-diagram.svg

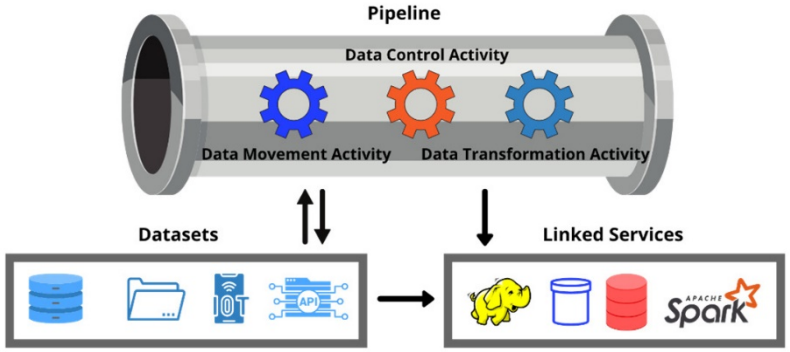
Snowflake's hybrid columnar architecture



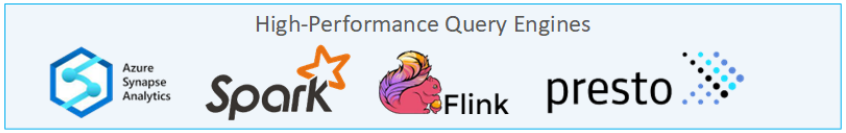


Delta Lakehouse Layered Architecture

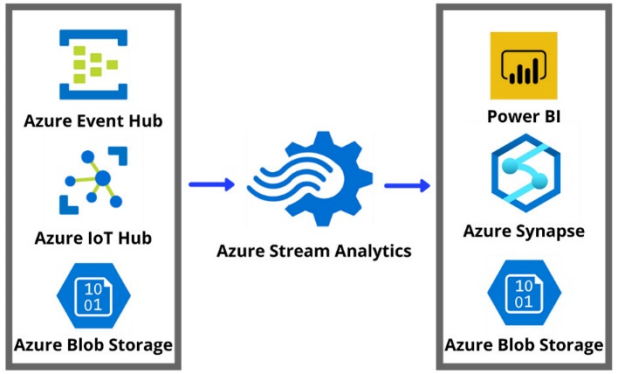




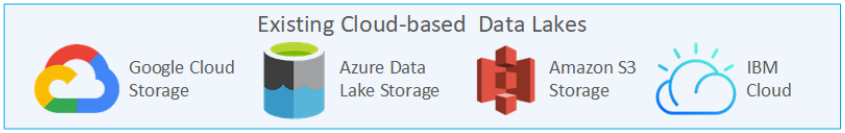
One platform for all Data use cases



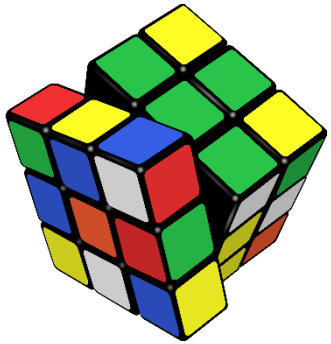
High performance query engines

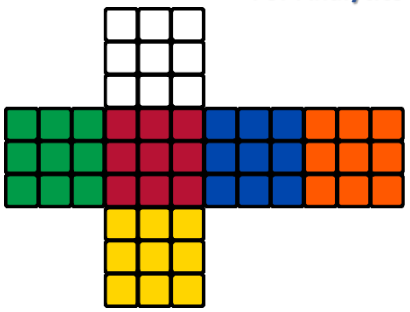


Open, Transactional layer for curated data



Scalable, low-cost Cloud-based Data Storage





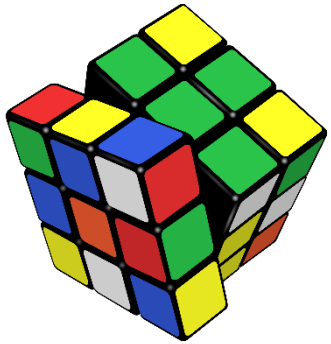
Summary

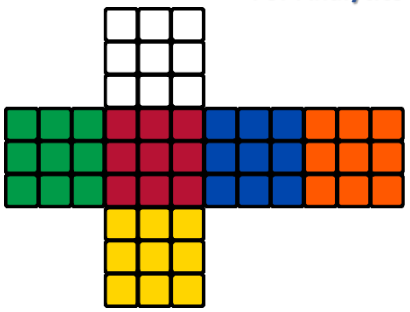
A very large part of space-time must be investigated, if reliable results are to be obtained.

Alan Turing

Summary of Essential Points

- **Data architecture** translates business needs into data and system requirements and seeks to manage data and its flow through the enterprise.
- **Data Engineering** focuses on data preparation and management. The data engineering lifecycle breaks data problems into key stages
- Core **Hadoop** includes HDFS for storage and YARN for cluster resource management
- The Hadoop ecosystem includes many components for:
  - Date Ingesting (Flume, Sqoop, Kana)
  - Data Storage (HDFS, HBase)
  - Data Processing (Spark, Hadoop MapReduce, Pig)
  - Data Modelling as tables for SQL access (Impala, Hive)
  - Data Exploration (Hue, Search)
  - Data Protection (Sentry)
  - Service Programming (Avro, Zookeeper, Thrift)
  - Data Visualization (Zeppelin, Tableau)
- New data systems are hosted and managed using **Kubernetes**.





References

A very large part of space-time must be investigated, if reliable results are to be obtained.

Alan Turing

References

- Frank J. Ohlhorst, Big Data Analytics: Turning Big Data into Big Money, Published by John Wiley & Sons, 2012.
- David Feinleib, "Big Data Bootcamp: What Managers Need to Know to Profit from the Big Data Revolution", Published by Apress, 2014.
- L. Bass, P. Clements, R. Kazman, Software Architecture in Practice. 3rd edition Addison-Wesley; 2013.
- Software Architecture for Big Data and the Cloud by Bruce Maxim; Maritta Heisel; Rami Bahsoon; Nour Ali; Ivan Mistrik Published by Morgan Kaufmann, 2017
- Eadline, Douglas. *Hadoop 2 Quick-Start Guide: Learn the Essentials of Big Data Computing in the Apache Hadoop 2 Ecosystem*. Addison-Wesley Professional, 2015.
- Swizec Teller, Hadoop Essentials, Publisher: Packt Publishing, Release Date: April 2015, ISBN: 9781784396688.