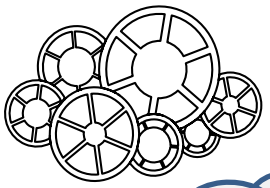
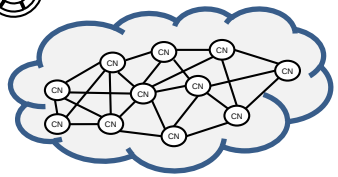
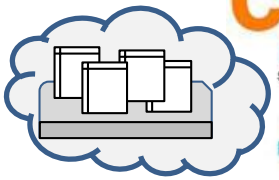
Big Data Infrastructure and Technologies
Lecture #1

Big Data Infrastructure
and Cloud based solutions for Big Data
by AWS, Microsoft Azure, Google

Yuri Demchenko, UvA







Outline

- Definitions
  - Big Data, Cloud Computing, Data Science
- Standardisation
  - NIST, CSA, DMTF, IDS, RDA, DAMA. IEEE
- Big Data platforms and tools
  - Big Data Storage, SQL and NoSQL, modern databases
  - Apache Hadoop Ecosystem
  - AWS, Google Cloud Platform, Azure
- DevOps and DataOps
  - Cloud automation and monitoring tools
  - Azure DevOps services and tools
- Optional: Data Markets and Data Exchange
  - Data properties as economic goods
- Discussion

This work is licensed under the Creative Commons Attribution 4.0 International License.
To view a copy of this license, visit http://creativecommons.org/licenses/by/4.0/

Yuri Demchenko,
Senior Researcher, Lecturer, UvA

- Graduated and PhD from National Technical University of Ukraine “Kiev Polytechnic Institute”
  - University of Amsterdam – since 2003
- Research areas
  - Big Data Infrastructure and Data Science platforms
  - Cloud architecture, cloud automation and DevOps
  - Cloud security and compliance
- Teaching courses (on campus and online)
  - Big Data Infrastructure and Technologies
  - Cloud powered Software Engineering and DevOps
  - Data Science Foundations, Professional Issues in Data Science
  - Security Engineering
- Recent projects
  - FAIRsFAIR: FAIR Principles in research data management
  - MATES: Digitalisation of the European Blue Economy
  - EDISON: Building the Data Science Profession for Europe
  - CYCLONE: Multi-cloud automation platform for cloud based applications
  - GEANT4 Research: Cloud aware networking infrastructure provisioning on-demand

Multiple aspects of Big Data

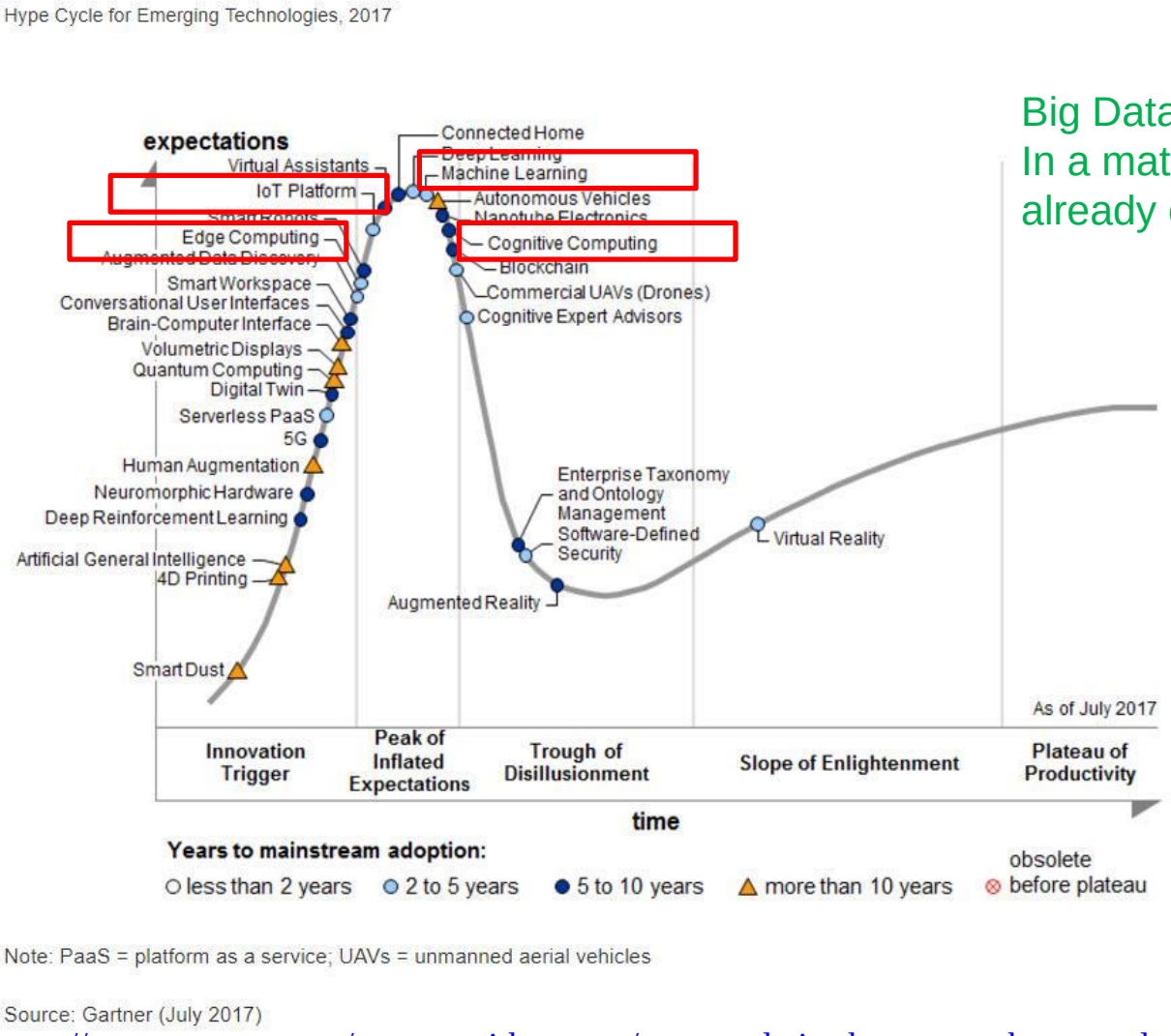
Big Data is a complex of technologies to enable handling of Big Data (storage, processing, transfer, security)

Big Data and multiple sources of data

- Social Media
- IoT
- Internet
- Science
- Industrial data
- Communication, voice

Data analytics blending with open and social media data

Gartner Technology Hypercycle (August 2017)



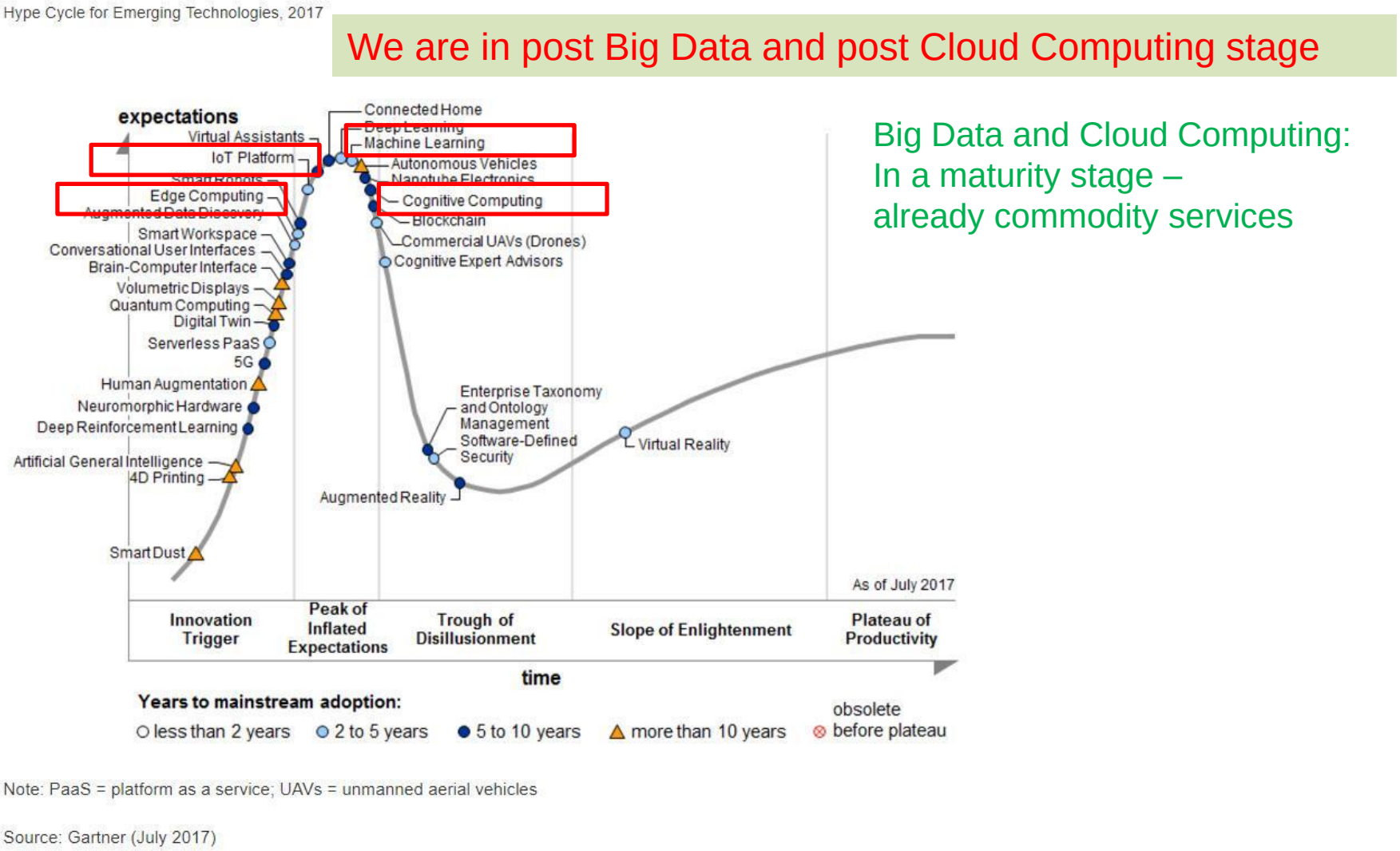
Big Data and Cloud Computing:
In a maturity stage –
already commodity services

Note: PaaS = platform as a service; UAVs = unmanned aerial vehicles

Source: Gartner (July 2017)

[ref] https://www.gartner.com/smarterwithgartner/top-trends-in-the-gartner-hype-cycle-for-emerging-technologies-2017/

Gartner Technology Hypercycle (August 2017)

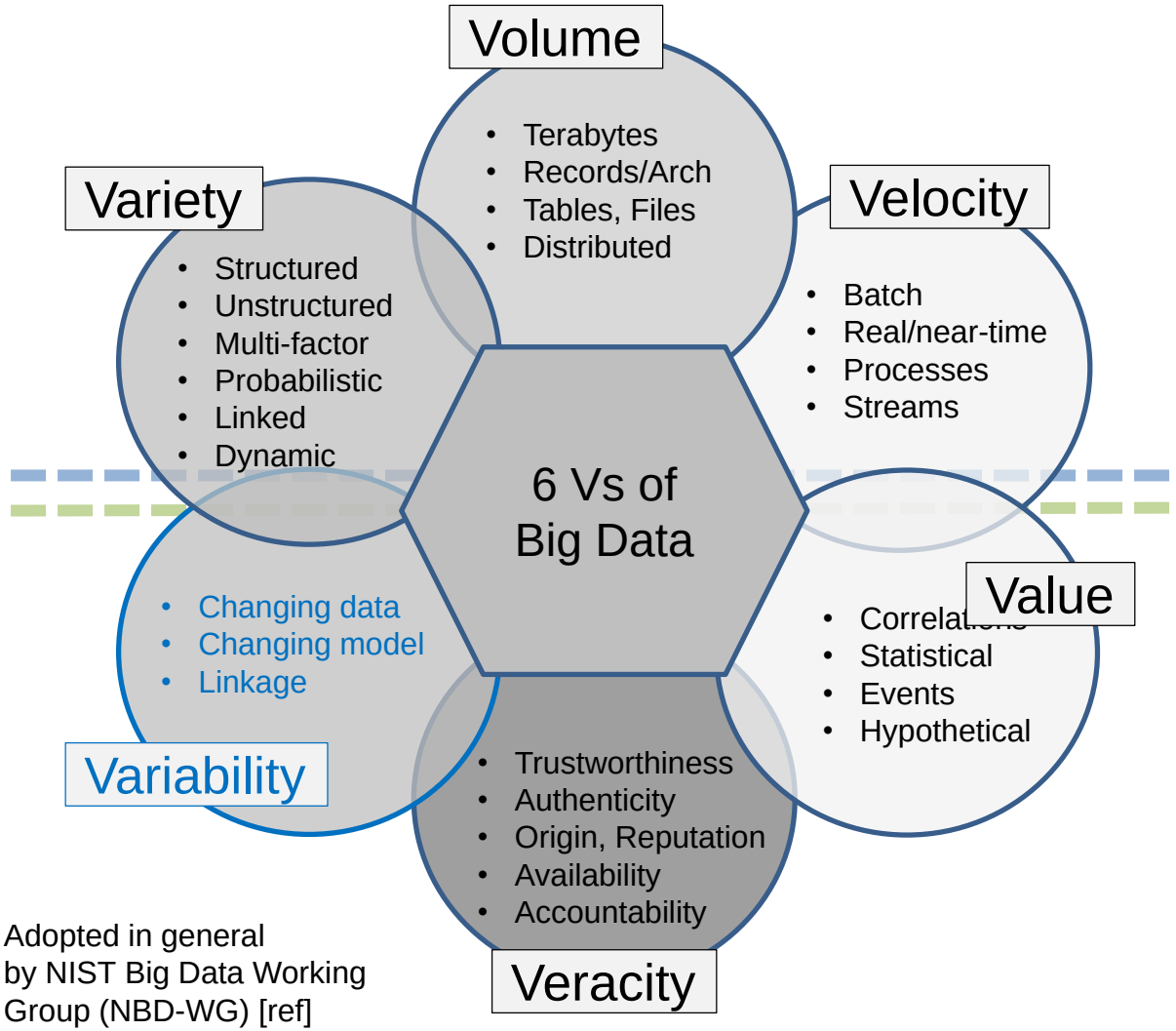


Note: PaaS = platform as a service; UAVs = unmanned aerial vehicles

Source: Gartner (July 2017)

[ref] https://www.gartner.com/smarterwithgartner/top-trends-in-the-gartner-hype-cycle-for-emerging-technologies-2017/

Big Data Properties: 6 (3+3) V's of Big Data



Adopted in general by NIST Big Data Working Group (NBD-WG) [ref]

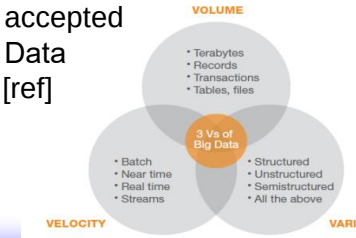
Generic Big Data Properties

- Volume
- Variety
- Velocity

Acquired Properties (after entering system)

- Value
- Veracity
- Variability

Commonly accepted 3V's of Big Data by Gartner [ref]



Big Data Definition: From 6V to 5 Parts (1)

(1) Big Data Properties: 5V
- Volume, Variety, Velocity, Value, Veracity
- Additionally: Data Dynamicity (Variability)

(2) New Data Models
- Data linking, provenance and referral integrity
- Data Lifecycle and Variability/Evolution

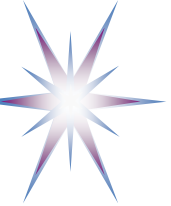
(3) New Analytics
- Real-time/streaming analytics, interactive and machine learning analytics

(4) New Infrastructure and Tools
- High performance Computing, Storage, Network
- Heterogeneous multi-provider services integration
- New Data Centric (multi-stakeholder) service models
- New Data Centric security models for trusted infrastructure and data processing and storage

(5) Source and Target
- High velocity/speed data capture from variety of sensors and data sources
- Data delivery to different visualisation and actionable systems and consumers
- Full digitised input and output, (ubiquitous) sensor networks, full digital control

Big Data technology drivers - Examples

- Modern e-Science in search for new knowledge
  - Scientific experiments and tools are becoming bigger and heavily based on data processing and mining
- Traditional data intensive industry
  - Genomic research, drugs development, Healthcare
  - High-tech industry, CAD/CAM, weather/climate, etc.
- AI and Industry 4.0
  - Data and Analytics are in foundation
- Network/infrastructure management
  - Network monitoring, Intrusion detection, troubleshooting
- Intelligence and security
- Consumer facing companies like Google and Facebook have driven many of the recent advances in Big Data efficiency
  - Facebook has 2.5 Bln daily users and still growing
  - Google handles number of search queries at 3.5 billion per day
  - Twitter handles some 500 million tweets per day count for 12 terabytes per day
    - Twitter data are widely used to add sentiments to market analysis and prediction
  - Power companies: process up to 350 billion annual meter readings to better predict power consumption
- Individually targeted online advertisement and campaigns



Cloud Computing, Big Data, Data Science

- Cloud Computing
  - Infrastructure/Platform/Software as a Service (IaaS/PaaS/SaaS)
  - Private, public, hybrid, community, federated
- Big Data
  - Technology domain to enable handling of Big Data (storage, processing, transfer, security)
  - Big Data properties and new data centric models
- Data Science (NIST)
  - **Data science** is the extraction of actionable knowledge directly from data through a process of discovery, or hypothesis formulation and hypothesis testing.
  - Data science combines concepts and methods from multiple disciplines to enable whole data lifecycle to bring value to business

Technology Maturity and Standardisation

- NIST: Cloud Computing (2008-2013) and Big Data (2013-now)
  - Cloud Computing Reference Architecture (CCRA)
  - Big Data Reference Architecture (BDRA)
- DMTF – Distributed Management Task Force
  - Cloud Information Model (CIM)
  - Open Virtualisation Format (OVF)
- CSA – Cloud Security Alliance
  - Cloud Compliance and Big Data Security Controls
- RDA – Research Data Alliance
  - PID, Data Factories, Data Registries
- DAMA – Data Management Association
  - Data Management Body of Knowledge (DMBOK)
- Industrial Data Space Association (IDS)
  - Industrial Data Space Architecture

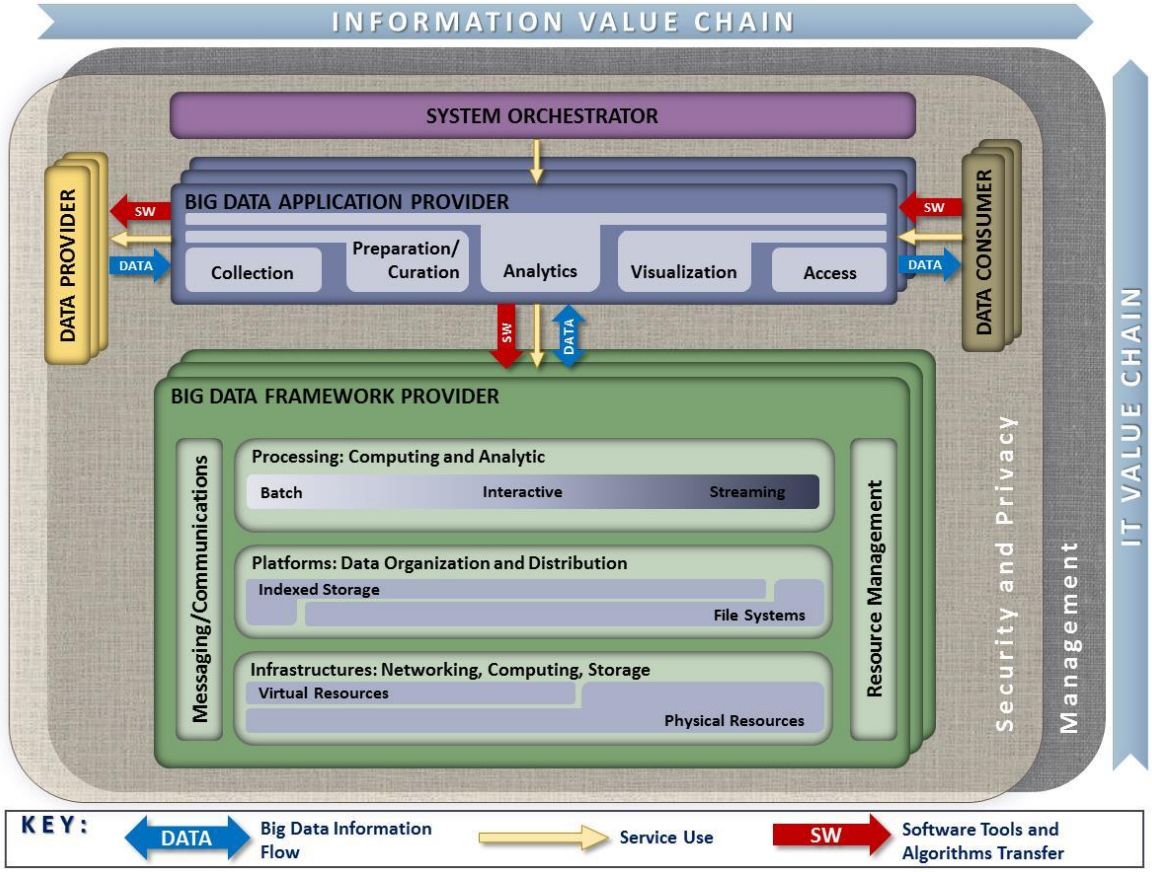
NIST Big Data Working Group (NBD-WG) and ISO/IEC JTC1 Study Group on Big Data (SGBD)

- NIST Big Data Working Group (NBD-WG) is leading the development of the Big Data Technology Roadmap - http://bigdatawg.nist.gov/home.php
  - Built on experience of developing the Cloud Computing standards
- Published as NIST Special Publication 1500 Volumes 1-7 in 2015

- New revision V2 to be published 2018 - https://bigdatawg.nist.gov/V2_output_docs.php
  - Volume 1: Definitions
  - Volume 2: Taxonomies
  - Volume 3: Use Case & Requirements
  - Volume 4: Security & Privacy
  - Volume 6: Reference Architecture
  - Volume 7: Standards Roadmap
  - Volume 8: Reference Architecture Interface
  - Volume 9: Adoption and Modernization
- NBD-WG defined 3 main components of the new technology:
  - Big Data Paradigm
  - Big Data Science and Data Scientist as a new profession
  - Big Data Architecture

The **Big Data Paradigm** consists of the distribution of data systems across horizontally-coupled independent resources to achieve the scalability needed for the efficient processing of extensive datasets.

NIST Big Data Reference Architecture (2018)



Main components of the Big Data ecosystem

- Data Provider
- Big Data Applications Provider
- Big Data Framework Provider
- Data Consumer
- Service Orchestrator

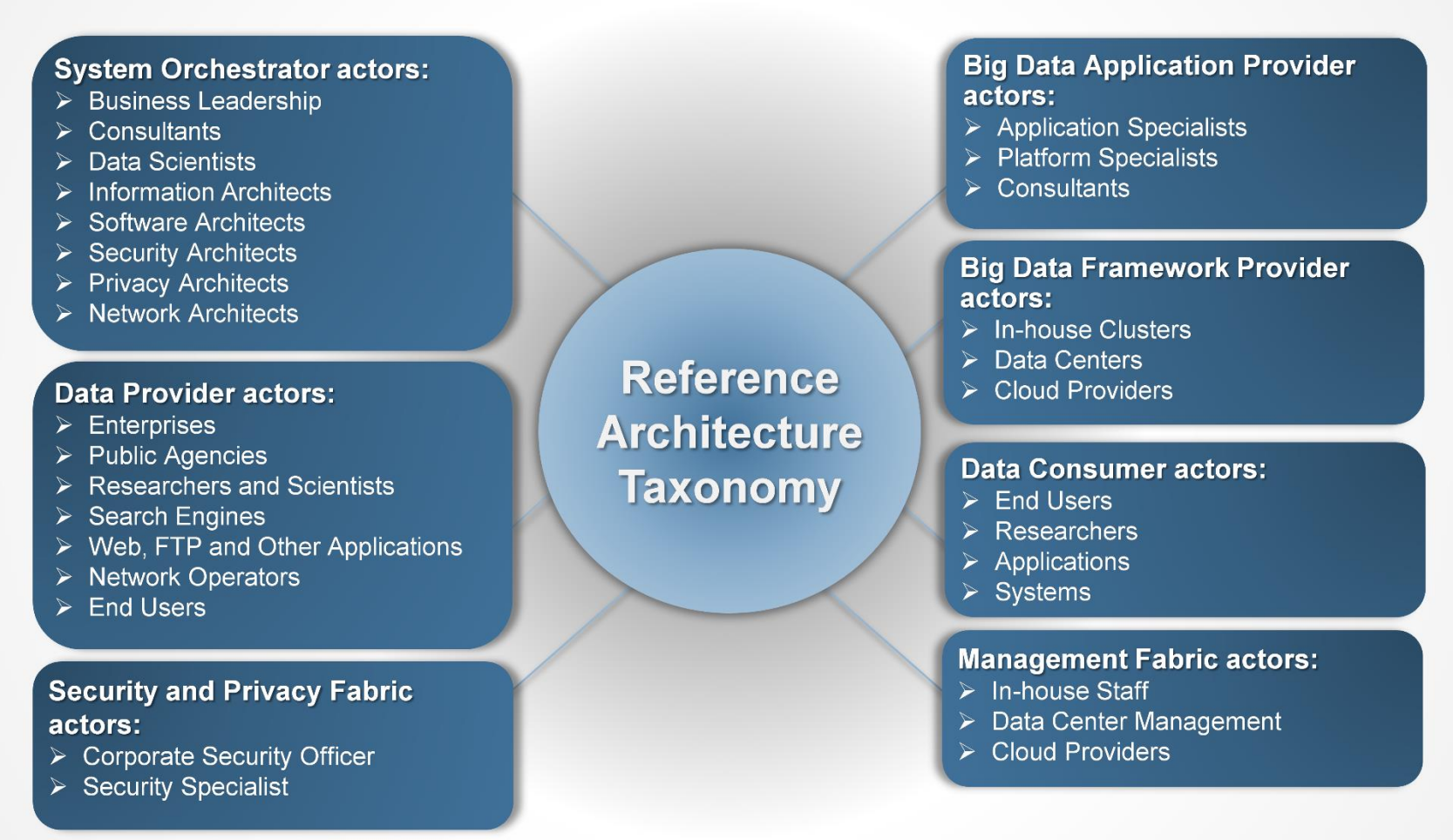
Big Data Lifecycle and Applications Provider activities

- Collection
- Preparation
- Analysis and Analytics
- Visualization
- Access

Big Data Ecosystem includes all components that are involved into Big Data production, processing, delivery, and consuming

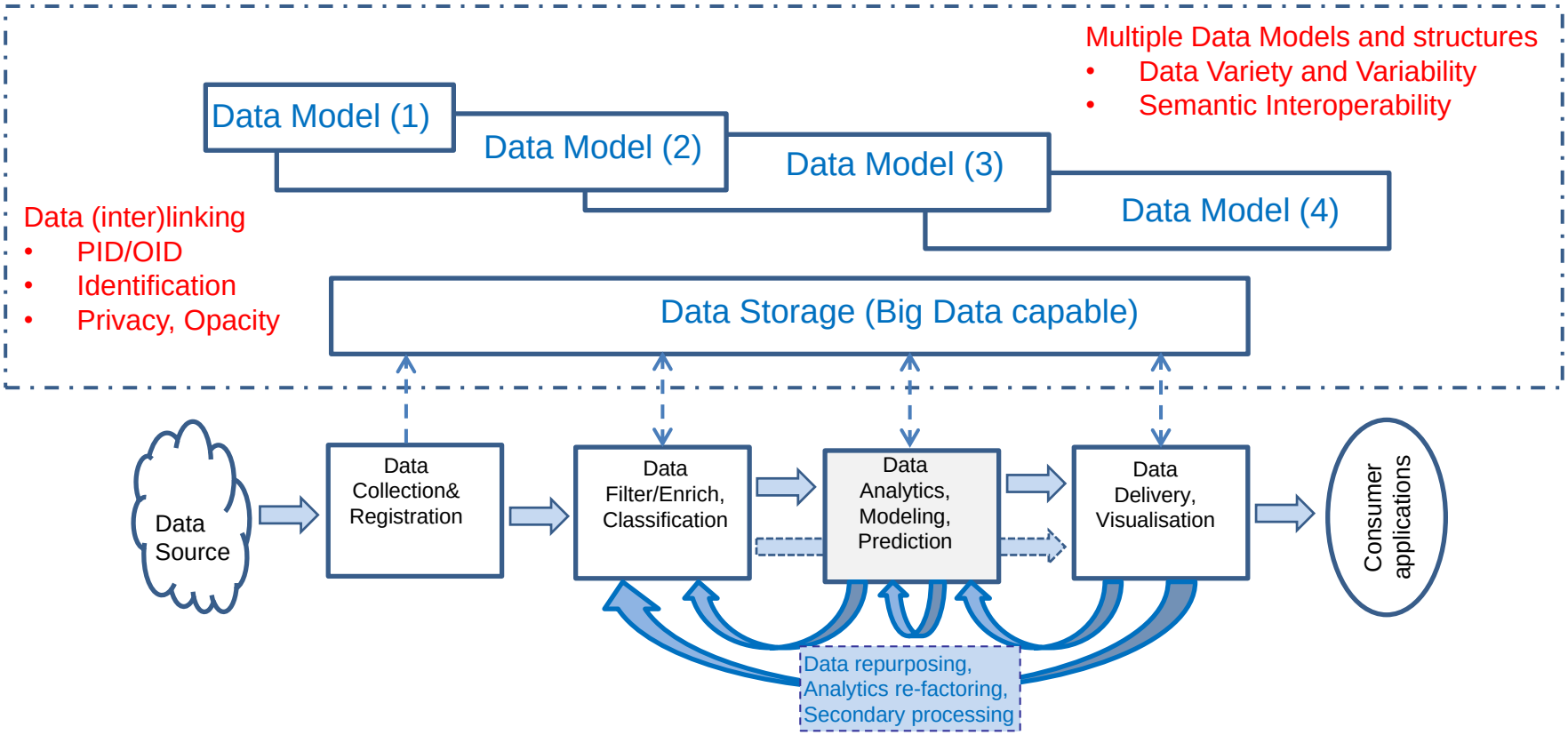
[ref] Volume 6: NIST Big Data Reference Architecture. http://bigdatawg.nist.gov/V1_output_docs.php

NIST Reference Architecture Taxonomy (2018)



[ref] Volume 6: NIST Big Data Reference Architecture. http://bigdatawg.nist.gov/V1_output_docs.php

Data Lifecycle/Transformation Model



- Data Model changes along data lifecycle or evolution
- Data provenance is a discipline to track all data transformations along lifecycle

- Identifying and linking data
  - Persistent data/object identifiers (PID/OID)
  - Traceability vs Opacity
  - Referral integrity

SQL and NoSQL

NoSQL definition (www.nosql-database.org):

- Next Generation Databases mostly addressing some of the points: being **non-relational, distributed, open-source** and **horizontal scalable**.
- Other characteristics apply: **schema-free, easy replication support, simple API, eventually consistent / BASE** (not ACID), a **huge data amount**, others

- ACID/SQL vs BASE/NoSQL
  - ACID Semantics: **A**tomic, **C**onsistent, **I**solated, **D**urable
  - BASE Semantics: **B**asically **A**vailable - **S**oft State - **E**ventual Consistency

NoSQL Distinguishing Characteristics

- Large data volumes
  - Web scale “Big Data”
- Scalable replication and distribution
  - Potentially thousands of machines
  - Potentially distributed around the world
- Queries need to return answers quickly
  - Not necessary precisely
  - Employing probabilistic search/decision
- Mostly query, few updates
- Asynchronous Inserts and Updates
- Schema – free (on write, schema applied by applications)
- Paradigm shift from ACID transaction properties to BASE
- CAP Theorem – NoSQL database types
- Open source development

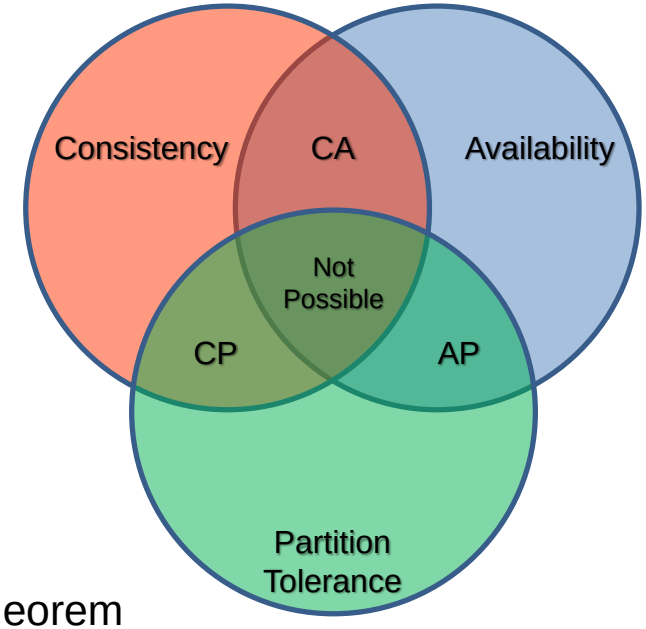
Brewer's CAP Theorem

Brewer's (CAP) Theorem (original formulation) [ref]

> *"There are three core systemic requirements that exist in a special relationship when it comes to designing and deploying applications in a distributed environment."*

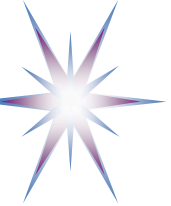
A distributed system can support only two of the following characteristics:

- Consistency
  - All nodes see the same data at the same time
- Availability
  - Node failures do not prevent survivors from continuing to operate
- Partition tolerance
  - The system continues to operate despite arbitrary message loss



[ref] http://www.julianbrowne.com/article/viewer/brewers-cap-theorem

The CAP Theorem Dilemma in Cloud Database and Storage



Main NoSQL Database Types and Existing Implementations

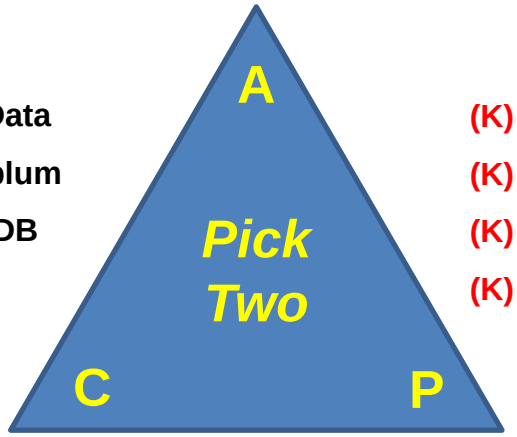
- **Column Store**: Each storage block contains data from only one column
  - BigTable by Google, Apache HBase, Cassandra
  - Work natively with HDFS and Hadoop
- **Document Store**: Store documents (in particular XML) made up of tagged elements
  - MongoDB, CouchDB, RaptorDB, CosmosDB
- **Key-Value Store**: Hash table of keys with arbitrary data as content/value
  - Memcached, Membase, Accumulo, Amazon DynamoDB
- **Graph Databases**: Store data in graph data in Triplestores or Quadstores
  - Neo4J, FlockDB, GraphDB

Visualizing the CAP Theorem and it's members

Configurable consistency:
Azure CosmosDB (K,C,D,G), AWS Aurora (R), Google Spanner (R)

Data Models:
(R) Relational (Comparison)
(K) Key Value
(C) Column-Oriented/Tabular
(D) Document-Oriented
(G) Graph

Availability
Each client can always read and write



CA

(R) RDBMs such as MySQL, Oracle, DB2, PostgreSQL, SQL Server, etc.
(R) Aster Data
(R) Greenplum
(D) CouchDB
(C) Vertica

AP

(K) Dynamo
(K) Voldemort
(K) Tokyo Cabinet
(K) KAI
(C) Cassandra
(D) SimpleDB
(D) CouchDB
(D) Riak

Consistency
All clients always have the same view of data

Partition Tolerance
The system works well despite physical network partitions

CP

(C) BigTable
(C) Hypertable
(C) Hbase
(D) MongoDB
(D) Terastore
(K) Scalaris
(K) BerkeleyDB
(K) MemcacheDB
(K) Redis
(R) VoltDB

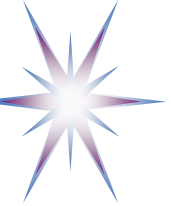
Modern Cloud Databases and CAP Theorem Challenges

- Google Spanner SQL database
- AWS Aurora Big SQL database
- Azure CosmosDB (NoSQL multi-data format database with underlying blob storage and HDFS)

Cloud Spanner - Mission Critical RDBMS

- SQL Database
- Best of both worlds
- Strong consistency and high availability worldwide

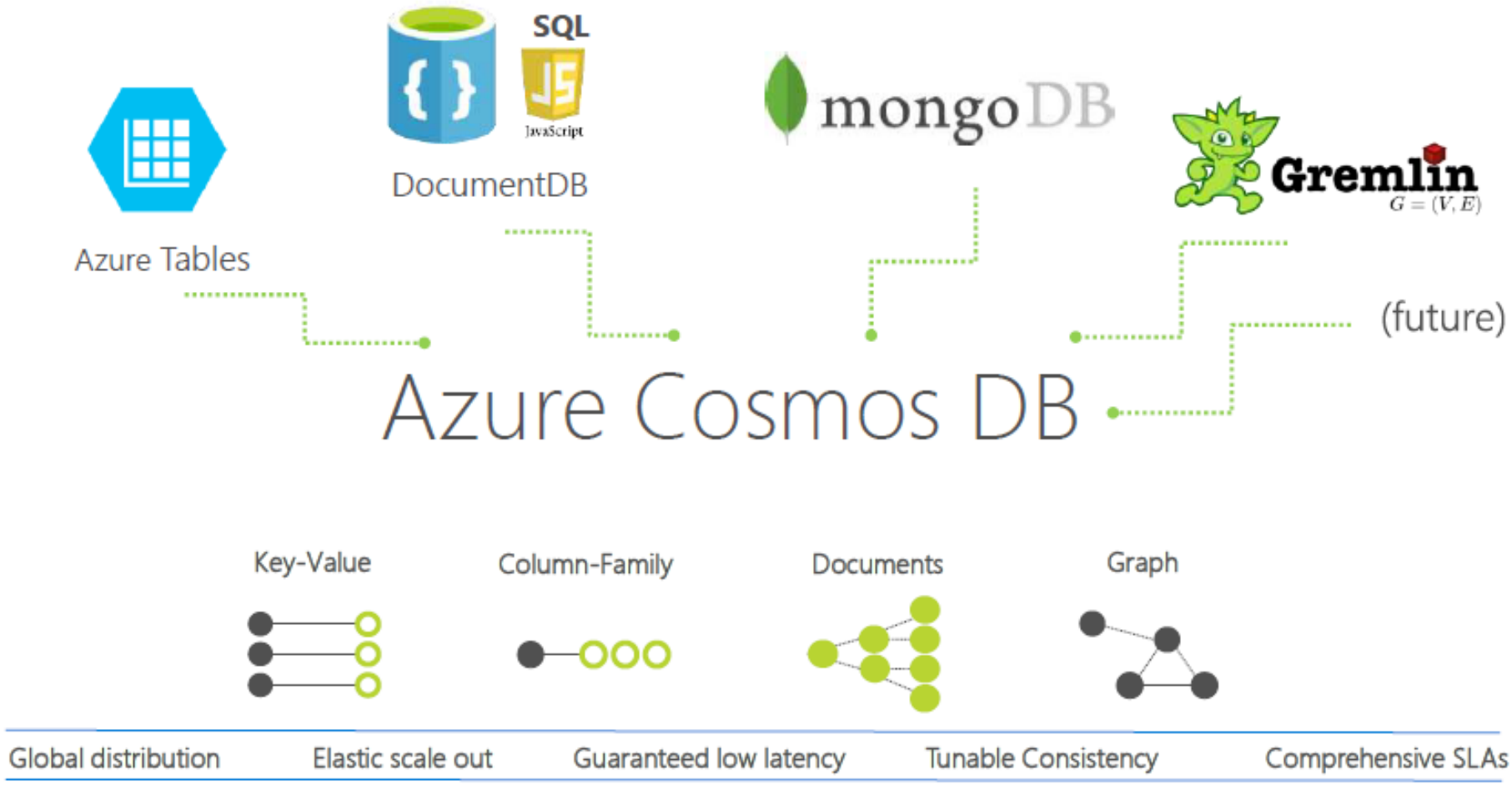
	CLOUD SPANNER	TRADITIONAL RELATIONAL	TRADITIONAL NON-RELATIONAL
Schema	✓ **Yes**	✓ Yes	✗ No
SQL	✓ **Yes**	✓ Yes	✗ No
Consistency	✓ **Strong**	✓ Strong	✗ Eventual
Availability	✓ **High**	✗ Failover	✓ High
Scalability	✓ **Horizontal**	✗ Vertical	✓ Horizontal
Replication	✓ **Automatic**	↻ Configurable	↻ Configurable



Amazon Aurora – Big SQL database

- High performance and scalability
- High availability and durability
- High security
- **MySQL and PostgreSQL** Compatible
- Fully managed
- Migration support

Azure CosmosDB (former DocumentDB) – Multi-model global distributed database



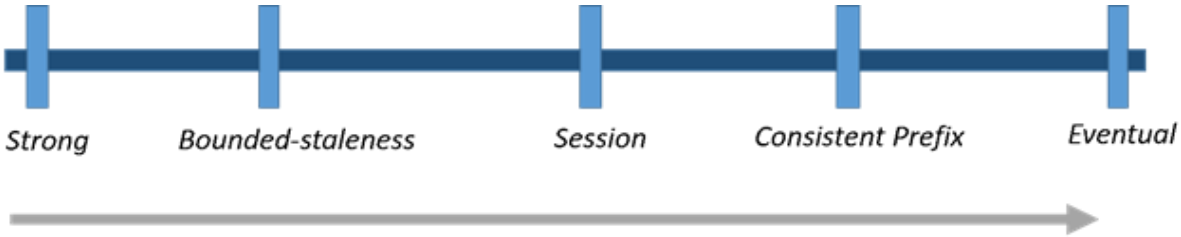
CosmosDB: Consistency and latency

Five Consistency Models

- Helps navigate Brewer's CAP theorem
- Intuitive Programming
  - Tunable well-defined consistency levels
  - Override on per-request basis
- Clear PACELC tradeoffs
  - Partition – Availability vs Consistency
  - Else – Latency vs Consistency

- Guaranteed low latency at P50 and P99 percentile
- Globally distributed with requests served from local region
- Write optimized, latch-free database
- Automatic Indexing

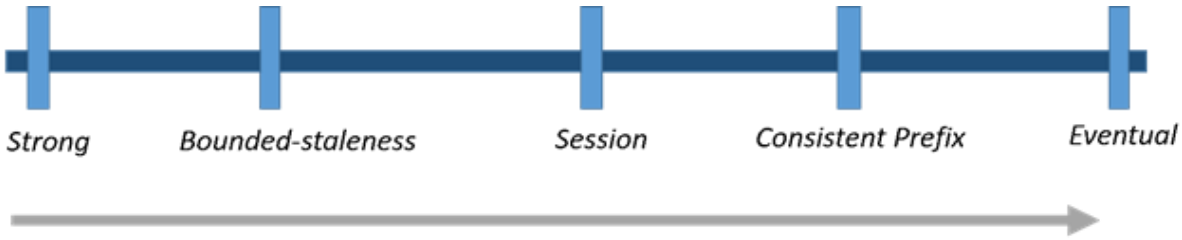
	Reads (1KB)	Indexed Writes (1KB)
P50	<2ms	<6ms
P99	<10ms	<15ms



CosmosDB: Consistency levels and guarantees

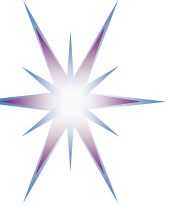
Consistency Level	Guarantees
Strong	Linearizability
Bounded Staleness	Consistent Prefix. Reads lag behind writes by k prefixes or t interval
Session	Consistent Prefix. Monotonic reads, monotonic writes, read-your-writes, write-follows-reads
Consistent Prefix	Updates returned are some prefix of all the updates, with no gaps
Eventual	Out of order reads

Lower latency, higher availability, better read scalability



MapReduce and Hadoop ecosystem

- MapReduce Computation Model
- Hadoop and components
- HDFS – Hadoop Distributed File System



MapReduce Programming Model

Map/Reduce is a programming model based on LISP that allows simple distribution of computing tasks across nodes. It includes two stages:

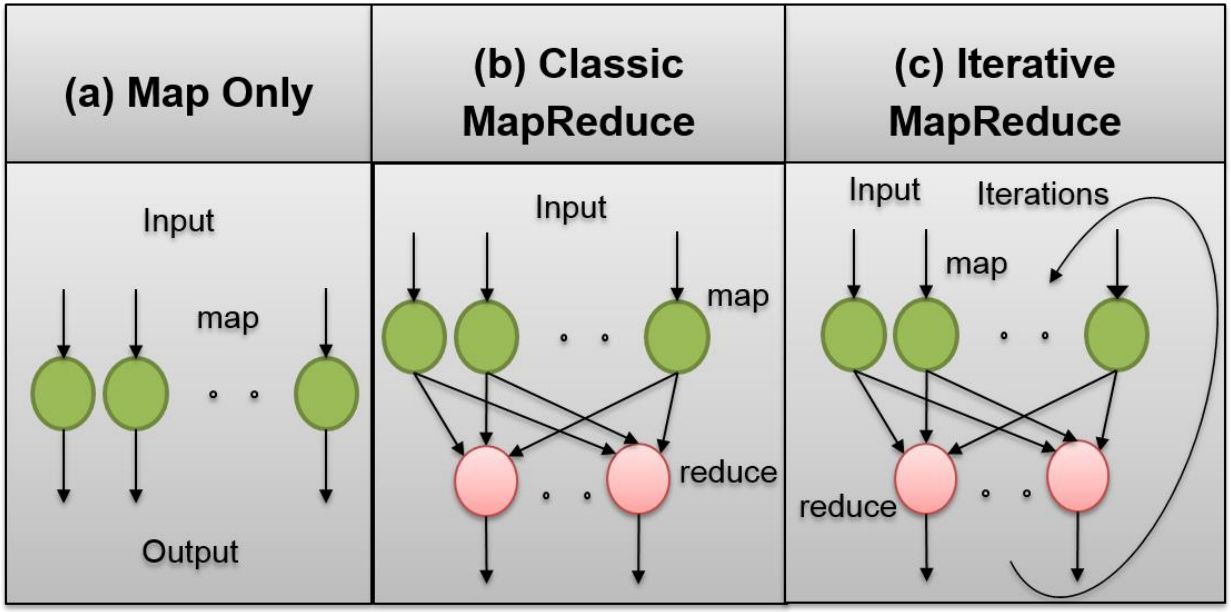
- **Map:** Perform a function on individual values in datasets to create a new list of values
- **Reduce:** Combine values from the new list to create a new value

Input and output are presented as key/value pairs. The following code fragment expresses the two operations map() and reduce() that run in parallel for two strings to execute word count:

```
map (in_key, in_value)
       list (out_key, intermediate_value)

reduce (out_key, list (intermediate_value))
       list (out_value)
```

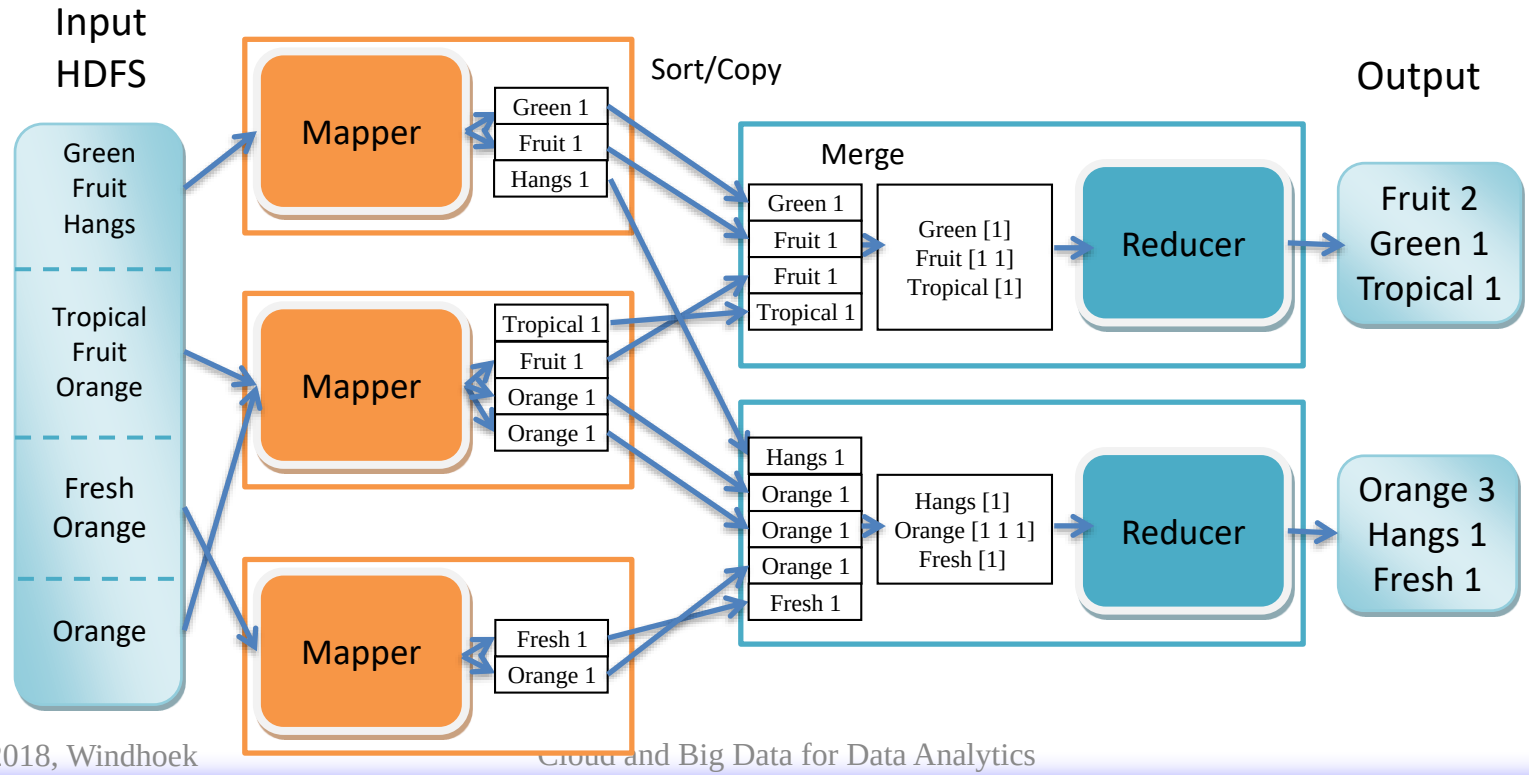
Three Forms of MapReduce



- The Map function produces an intermediate value for the intended output key, the Reduce function combines all intermediate values for a particular key.
- Data flow in different forms of MapReduce: (a) Map Only; (b) Classic MapReduce; (c) Iterative MapReduce.
- The classic MapReduce operates in a synchronous mode.
  - Input data are partitioned and multiple map() tasks are run in parallel.
  - After all map()s are complete, all intermediate values are combined for all unique keys by running multiple reduce() tasks in parallel.

Word Count Operations Allocation to Map/Reduce

Input strings		Mapper			Reducer	
		Green	1		Fruit	2
		Fruit	1		Green	1
Green fruit hangs	➔	Hangs	1	➔	Hangs	1
Tropical fruit orange	➔			➔	Tropical	1
		Tropical	1		Orange	1
		Fruit	1			
		Orange	1			



Hadoop Cluster operation on MapReduce

For Map machine

- Read content and prepares data from assigned portion of input data
- Feed data into Map function and saves result to local disk
- Notifies **Master** about (partially) completed work

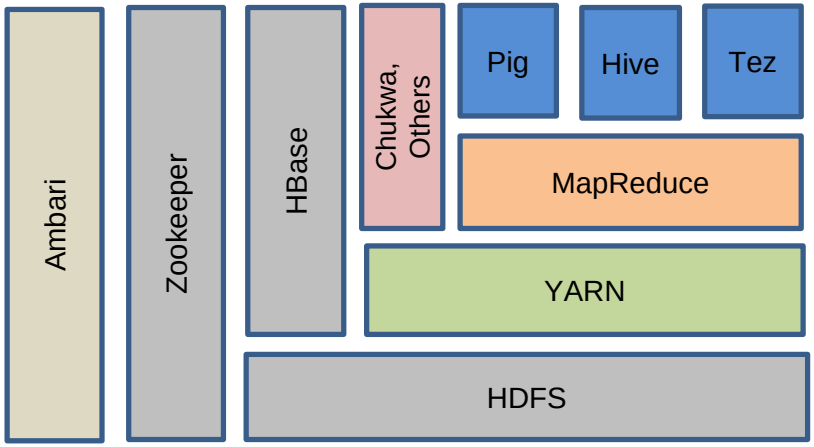
For Reduce machine

- Receive notification from Master about (partially) completed Map work
- Retrieve intermediate data from Map machine via remote read
- Sorts intermediate data by key (e.g. by key word)
- Iterate over intermediate data for each unique key and sends corresponding set through Reduce function
- Add result to final output file or dataset and write it to HDFS storage.

MapReduce and Hadoop

- Java based
- Designed for scalability
  - Up to TB data: distributed, not-consistent
- And not for speed
  - Even simple data query task will take seconds

Apache Hadoop (Release 2.2+)



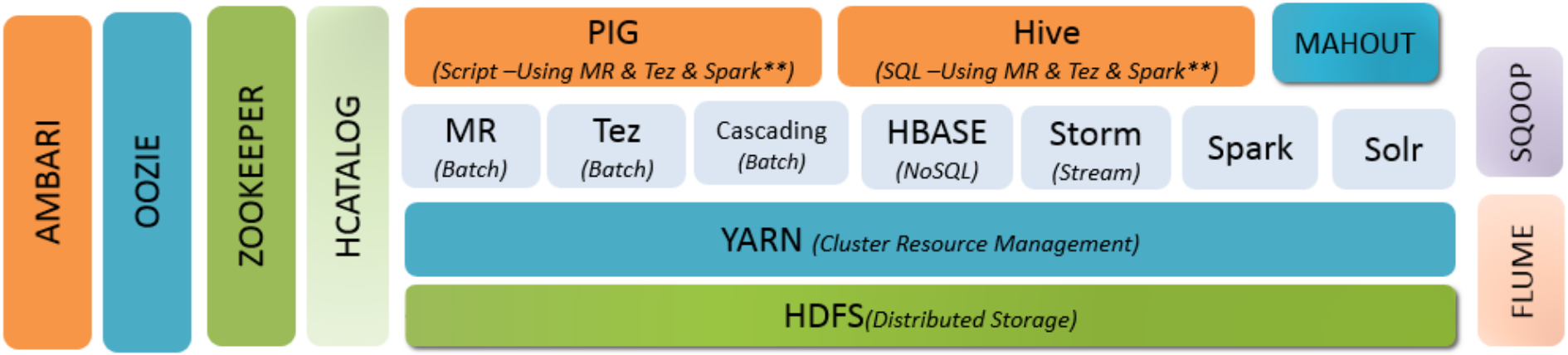
Apache Hadoop software stack includes the following main modules:

- **Hadoop Common**: The common utilities that support the other Hadoop modules and includes utilities and drivers to support different computer cluster and language platforms.
- **HDFS**: Hadoop Distributed File System optimized for large scale storage and processing of data on commodity hardware
- **Hadoop YARN:** A framework for job scheduling and cluster resource management.
- **Hadoop MapReduce:** A YARN-based system for parallel processing of large data sets.

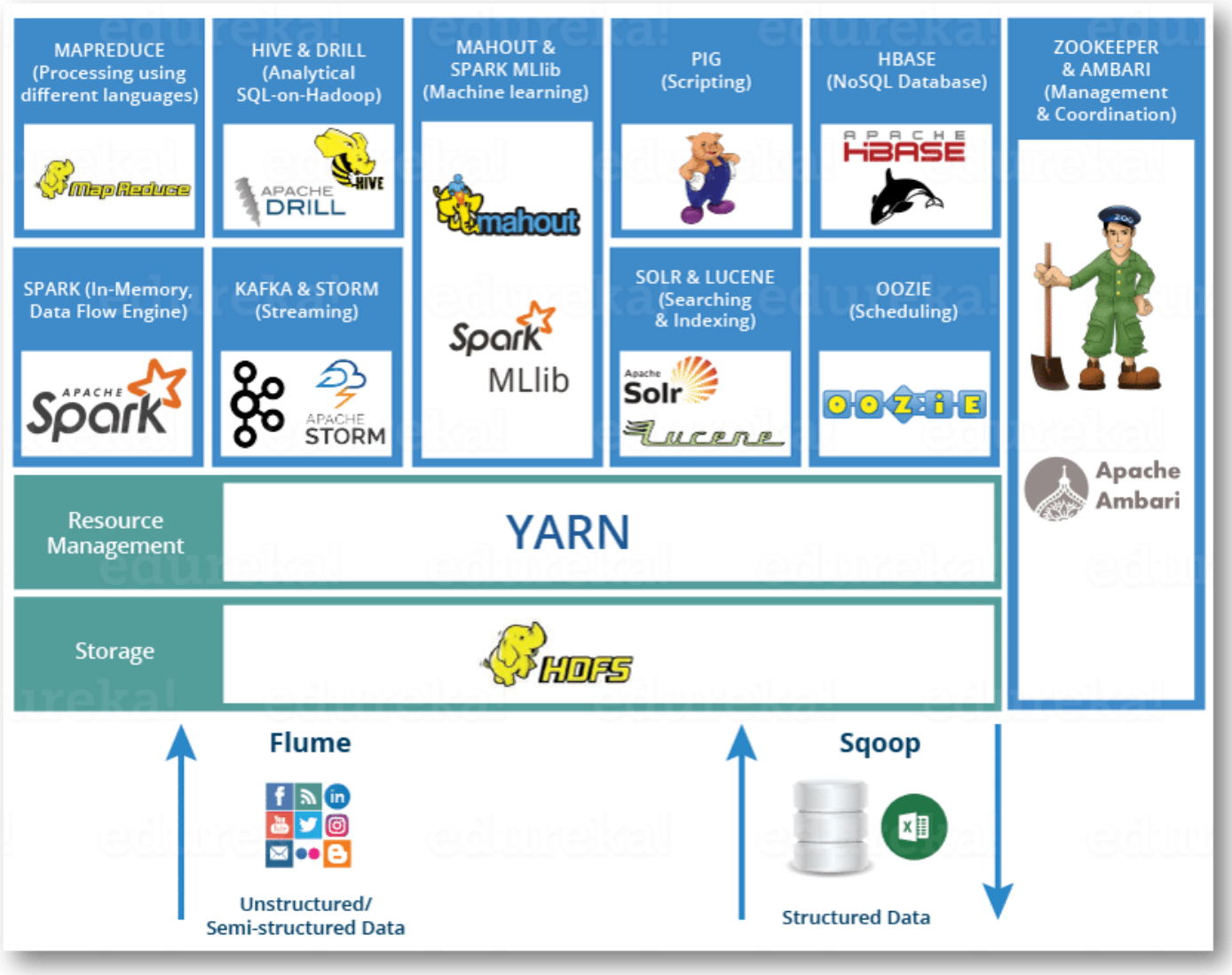
Other Hadoop-related projects at Apache include:

- **Hive:** A data warehouse system that provides data aggregation and querying.
- **Pig:** A high-level data-flow language and execution framework for parallel computation.
- **HBase:** A distributed column oriented database that supports structured data storage for large tables
- **Cassandra:** A scalable multi-master database protected against hardware failure
- **Tez:** A generalized data-flow programming framework, built on Hadoop YARN, which provides a powerful and flexible engine to execute an arbitrary DAG of tasks to process data for both batch and interactive use-cases.ooKeeper: A scalable coordination service for distributed applications.
- **Spark:** A fast and general compute engine for Hadoop data. Spark provides a simple and expressive programming model that supports a wide range of applications, including ETL, machine learning, stream processing, and graph computation.
- **Mahout:** A scalable machine learning and data mining library.
- **Avro:** A data serialization system that supports rich data structures
- **Chukwa:** A data collection system for managing large distributed systems.
- **Ambari:** A web-based tool for provisioning, managing, and monitoring Apache Hadoop clusters

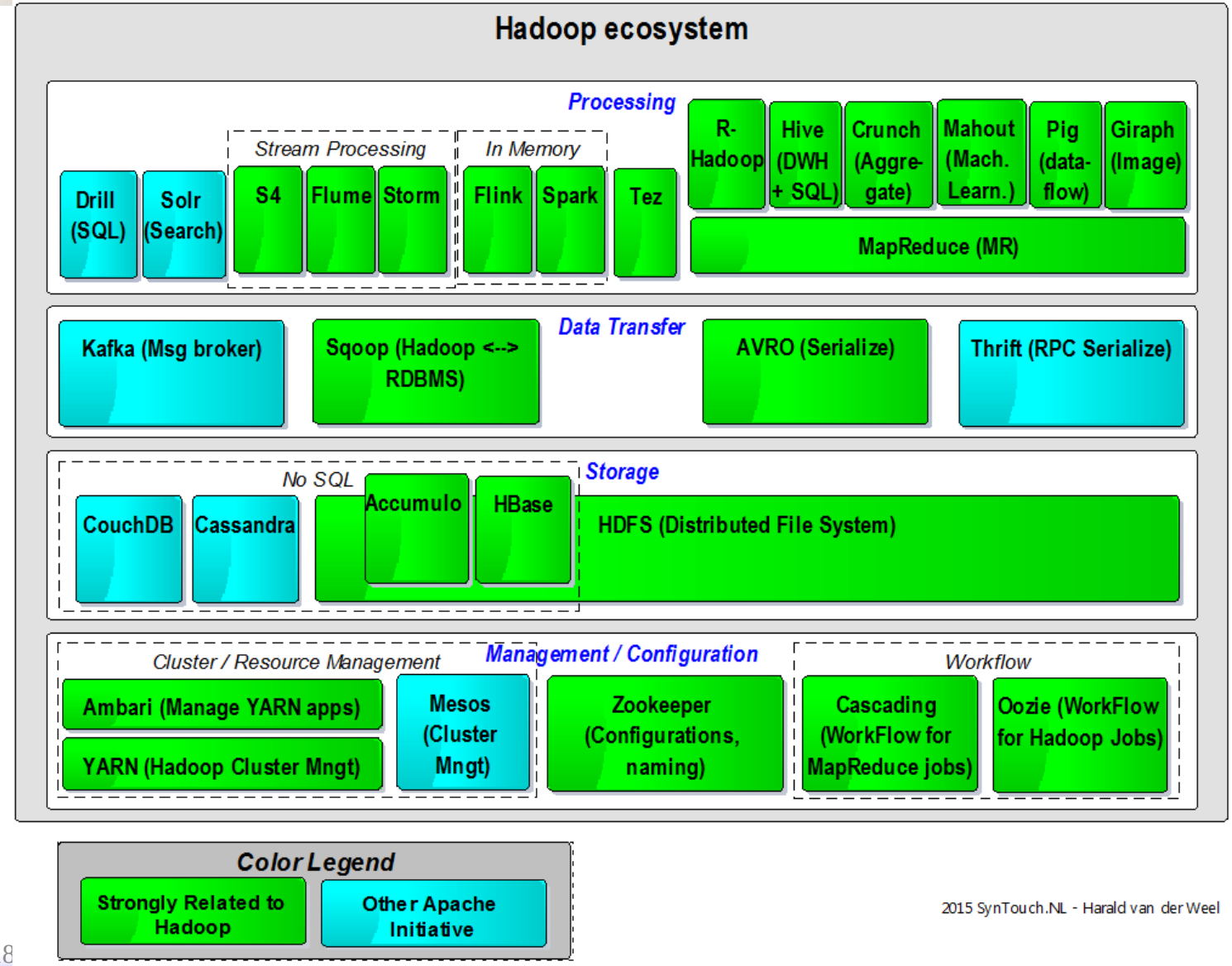
Hadoop Ecosystem – Layered view



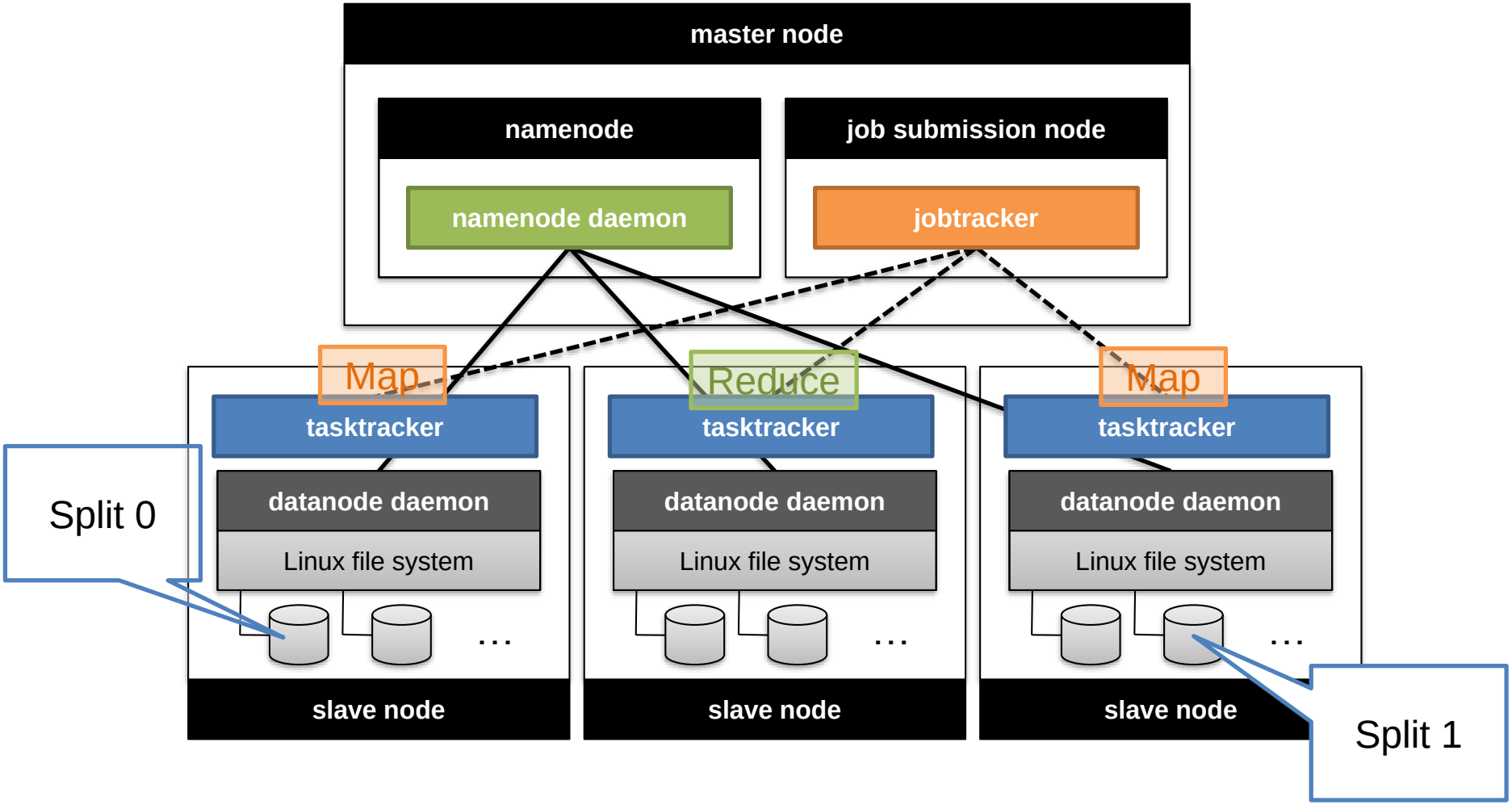
Zoo style Hadoop Ecosystem



Hadoop Ecosystem – Layered by functional groups



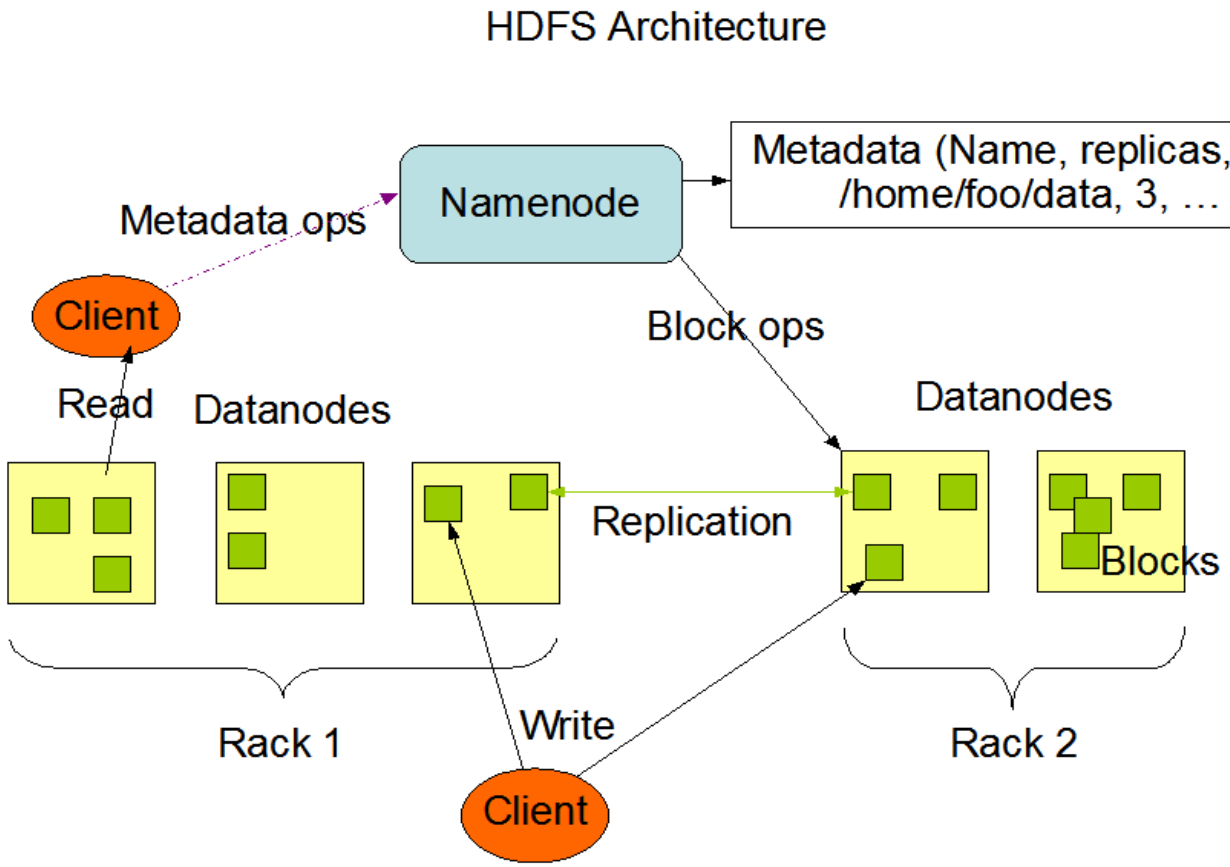
Hadoop Cluster Architecture – “Physical” view



Hadoop Cluster Architecture - Operation

- The Master node includes
  - Name Node that coordinates the data storage and files namespace in HDFS
  - Job Tracker that coordinates the MapReduce execution on the Worker nodes.
- The Master node manages the two main processes: storing data in HDFS and running parallel computation what include activities:
  - Initialise the cluster and split data and tasks according to number of available Workers
  - Send each Worker its part of data
  - Receive the results from each Worker
  - Re-execute task in case a Worker failed
  - Master coordinates data exchange between Map and Reduce machines
- The Slave (or Worker) node stores data and runs assigned map() or reduce() tasks.
  - Each Slave node runs both a Data Node that stores own part of data in HDFS and a Task Tracker daemon that communicates with and receives instructions from the Master node.
- Data partitioning and placement on Worker nodes is done to minimize data communication between general HDFS data storage and worker machines
  - Takes into account that Hadoop cluster can occupy few racks and data files can be split between nodes and racks
- The Client machine doesn't run any of MapReduce tasks but it communicates with the Master, loads data into cluster, submits MapReduce jobs, and retrieves results.

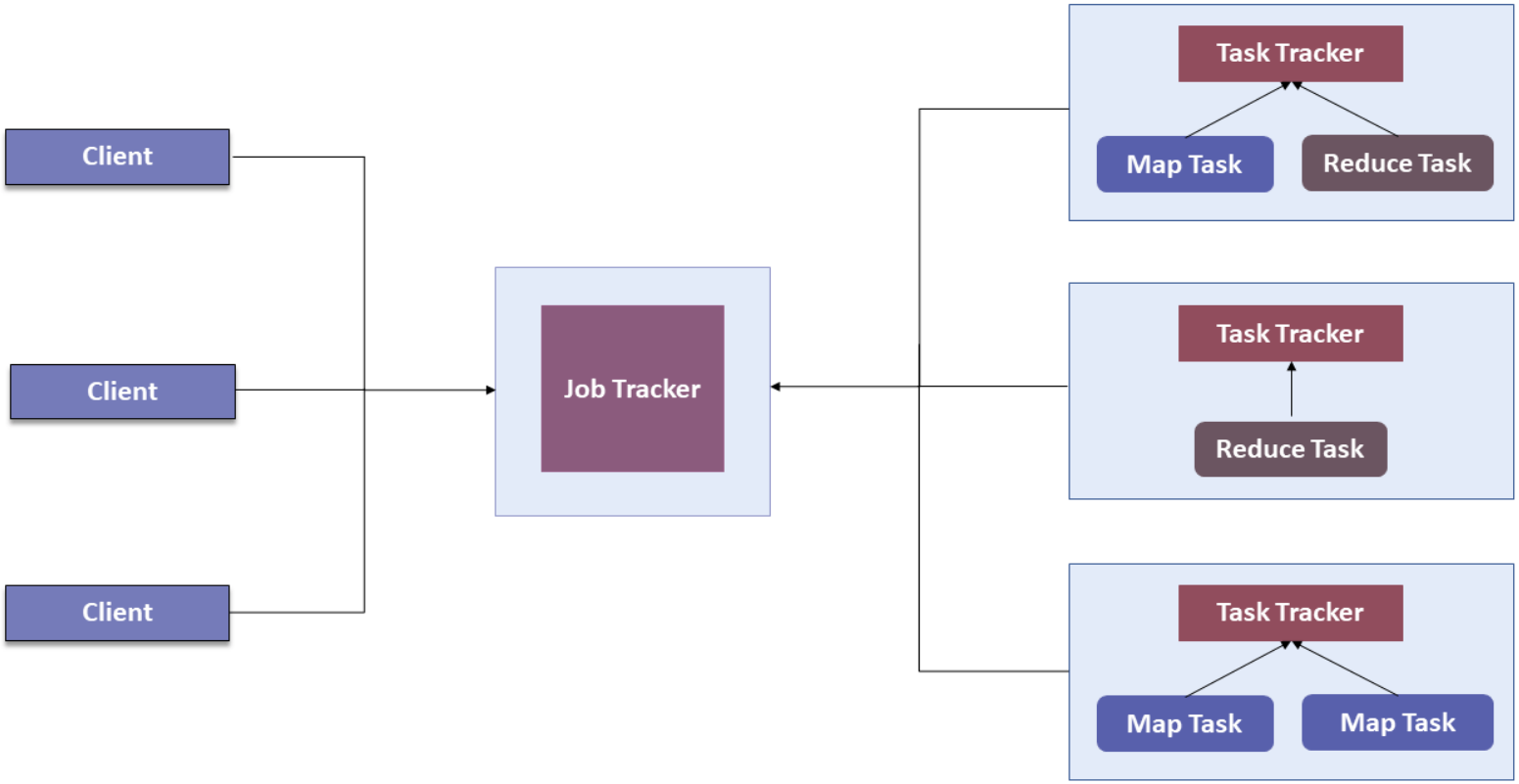
HDFS Architecture



- File content is split into blocks (default 128MB, 3 replica).
- NameNode maintains the namespace tree and the mapping of file blocks to DataNodes.
- Files and directories are represented on the NameNode by *inodes* (permissions, modification and access times, namespace and disk space quotas).
- Namespace is a hierarchy of files and directories.

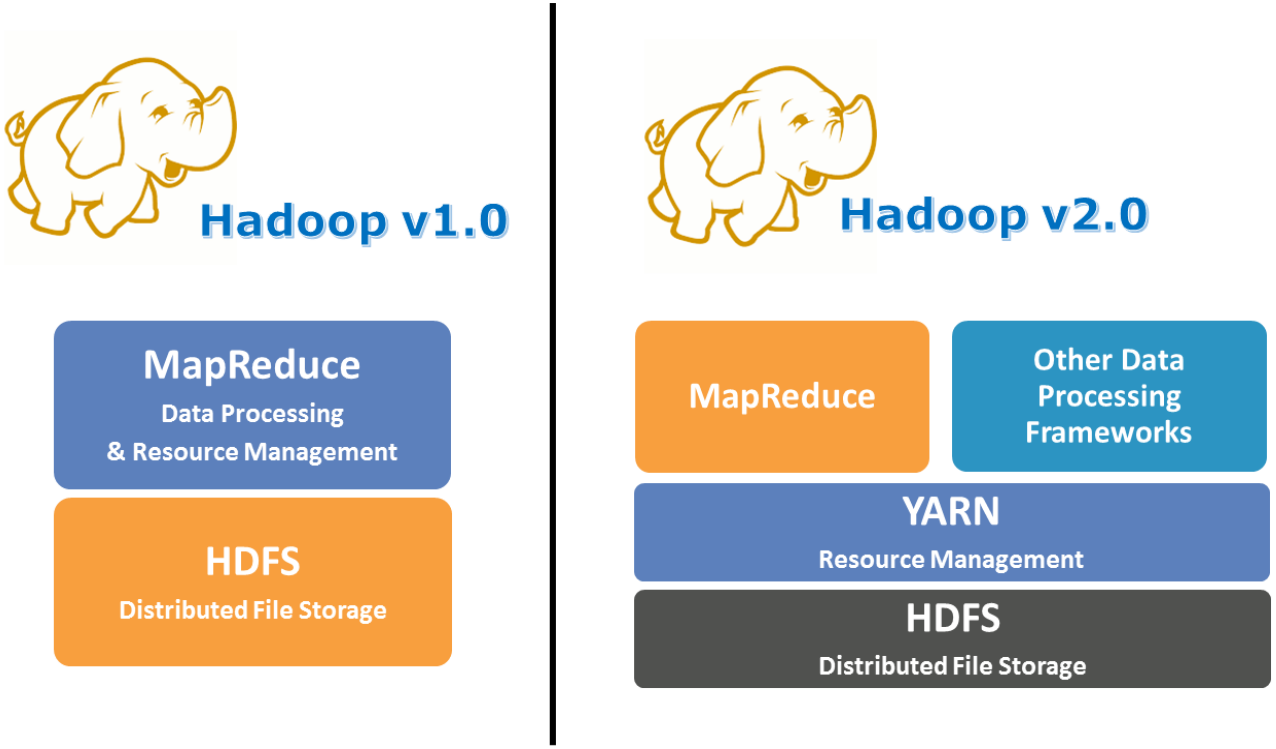
[ref] HDFS Architecture Guide. Apache Foundation. - http://hadoop.apache.org/docs/r1.2.1/hdfs_design.html

YARN vs MapReduce – MapReduce Hadoop v1.0



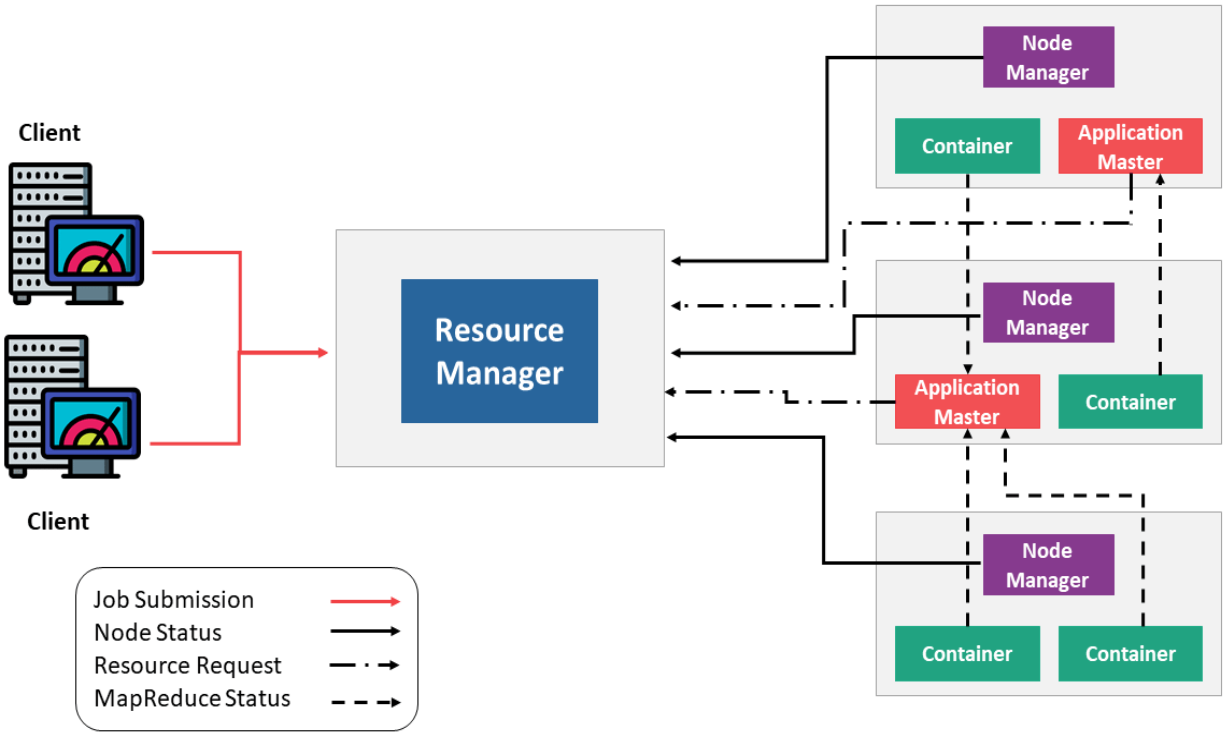
- Scalability issue MapReduce Hadoop v1.0
  - According tom Yahoo! – max 5000 nodes and 40,000 tasks running concurrently
- Utilization of computational resources is inefficient in MRV1

YARN since Hadoop v2.0



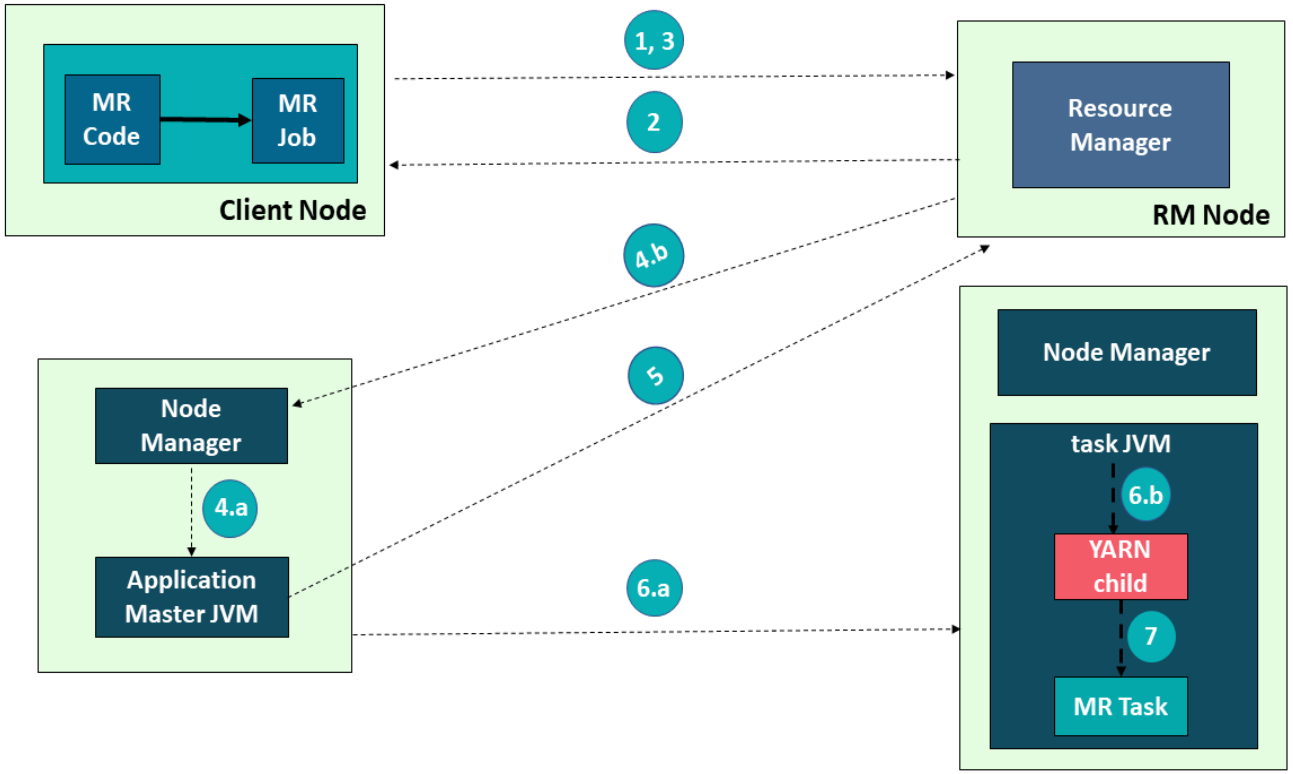
- YARN – Yet Another Resource Negotiator since Hadoop v2.0
- YARN performs all processing activities by allocating resources and scheduling jobs/tasks

YARN components



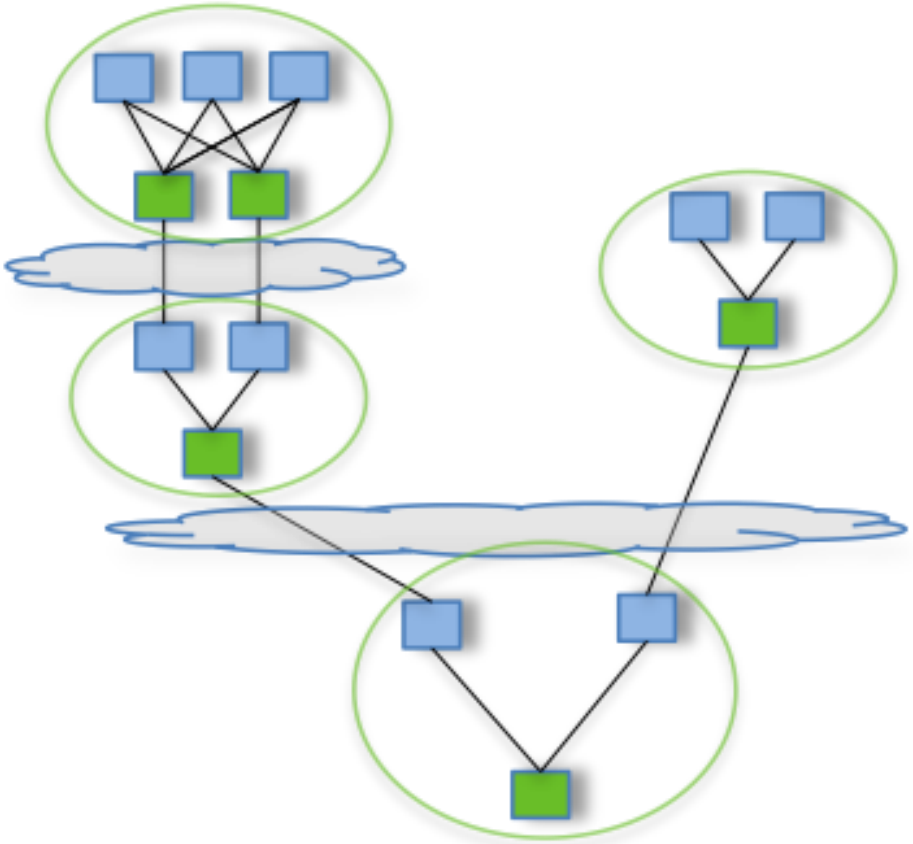
- **Resource Manager:** Runs on a master daemon and manages the resource allocation in the cluster.
- **Node Manager:** They run on the slave daemons and are responsible for the execution of a task on every single Data Node.
- **Application Master:** Manages the user job lifecycle and resource needs of individual applications. It works along with the Node Manager and monitors the execution of tasks.
- **Container:** Package of resources including RAM, CPU, Network, HDD etc on a single node.

Jobs Submission in YARN



1) Submit the job
2) Get Application ID
3) Get Application Submission Context
4 a) Start Container Launch
4 b) Launch Application Master
5) Allocate Resources
6 a) Container
6 b) Launch
7) Execute

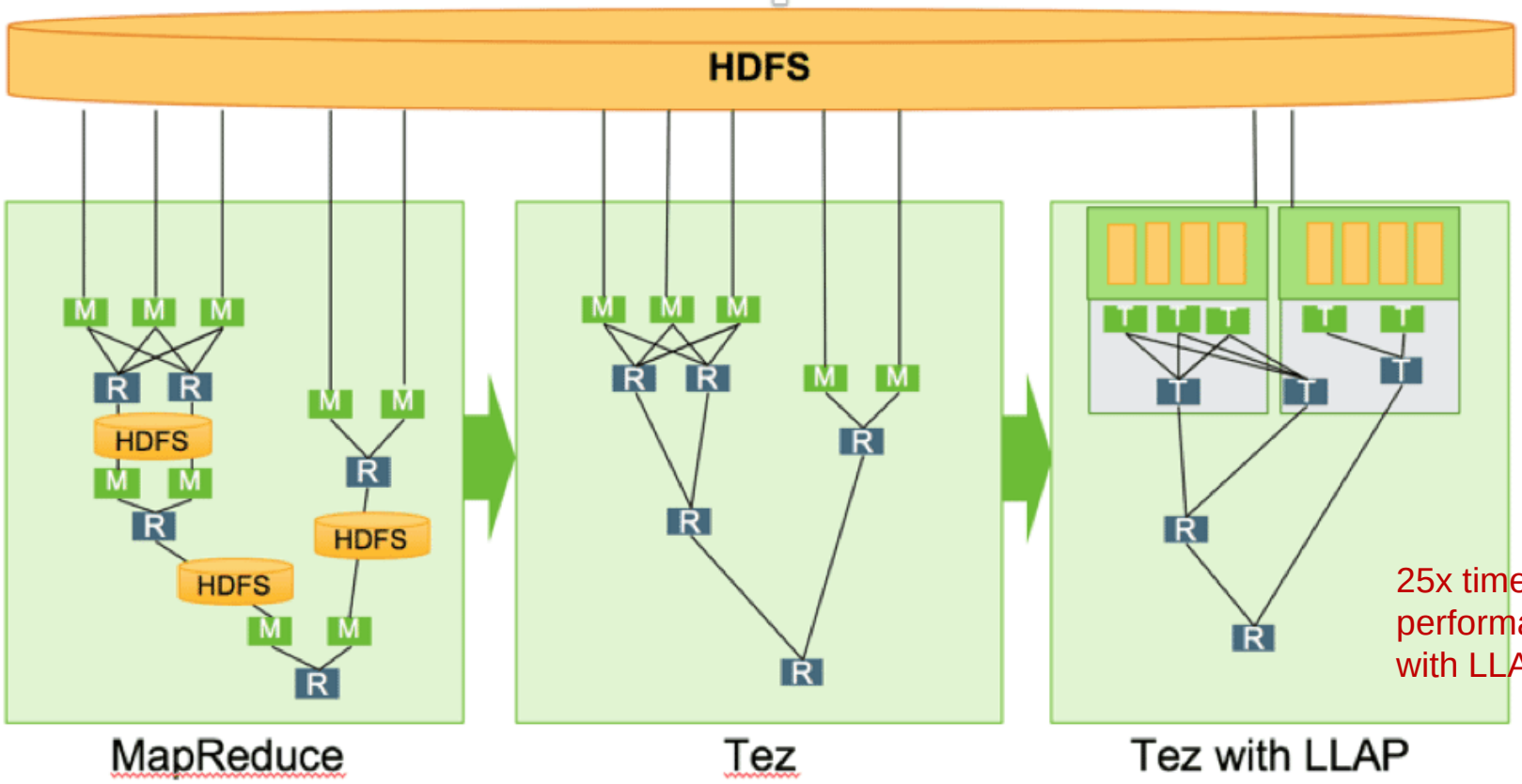
Apache Tez



Pig/Hive - MR

- Tez is application framework which allows for a complex directed-acyclic-graph of tasks for processing data
- Allows Hive and Pig tasks to run a complex DAG of tasks
- Tez can be used to process data, that earlier took multiple MR jobs, now in a single Tez job
- Reconfiguration at run time

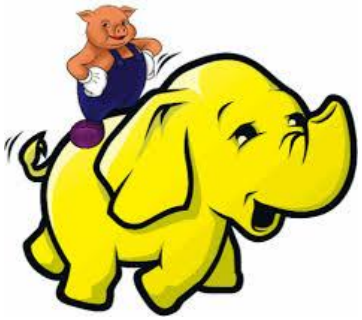
Apache Tez – Further improvement with LLAP by Hortonworks



- LLAP (Live Long and Process) combines persistent query servers and optimized in-memory caching that allows Hive to launch queries instantly and avoids unnecessary disk I/O.
- LLAP caches memory intelligently and it shares this data among all clients, while retaining the ability to scale elastically within a cluster. LLAP brings compute to memory (rather than compute to disk),
- Worker tasks run inside LLAP daemons, and not in containers

Hive, Pig Latin, Oozie

- Solution: Provide higher-level data processing languages
- Hive: Data warehousing application in Hadoop
  - Query language is HQL, variant of SQL
  - Tables stored on HDFS as flat files
  - Developed by Facebook, now open source
- Pig: Large-scale data processing system
  - Scripts are written in Pig Latin, a dataflow language
  - Developed by Yahoo!, now open source, Roughly 1/3 of all Yahoo! internal jobs
- Oozie
  - Server-based workflow scheduling system to manage Hadoop jobs.
  - Workflows defined as a collection of control flow and action nodes in a directed acyclic graph.
- Common idea:
  - Provide higher-level language to facilitate large-data processing
  - Higher-level language “compiles down” to Hadoop jobs



Summary and take away

- Big Data technologies target to support storing, processing and managing large and growing amount of data of large Volume, high Velocity and wide Variety
  - Big Data Infrastructure needs to support the whole Big Data lifecycle including data collection, filtering, processing, visualization, and storing
- MapReduce is a programming model for parallel data processing that is commonly used for Big Data processing and analytics
- Hadoop is a designated cluster solution for MapReduce
  - Hadoop is a batch processing system that implement paradigm of moving processing to data
  - Apache Hadoop is an Open Source software stack that includes a number of products to support the whole Big Data processing ecosystem

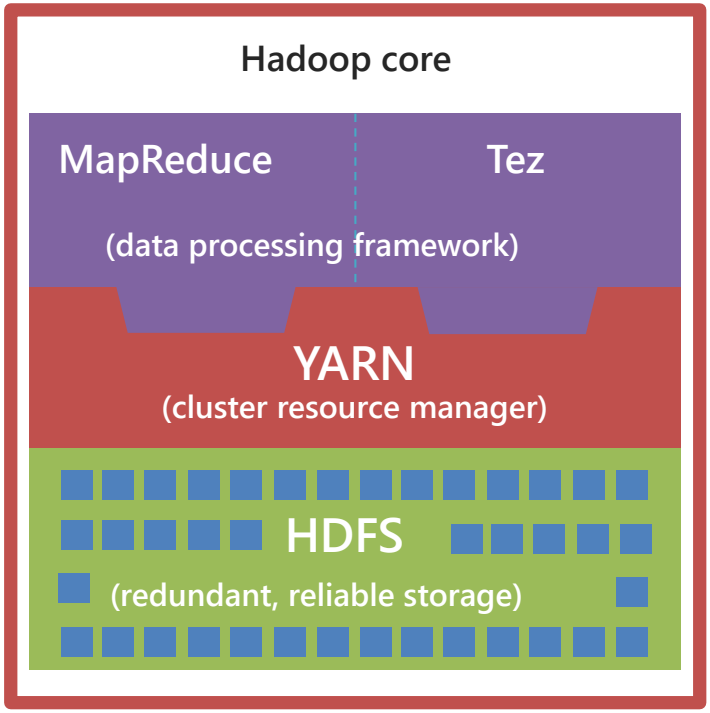
Cloud based Storage for Big Data

- Hadoop Distributed File System (HDFS)
- Data Lakes
- Large Scale Databases with controlled consistency

Revisiting: Hadoop – Core components

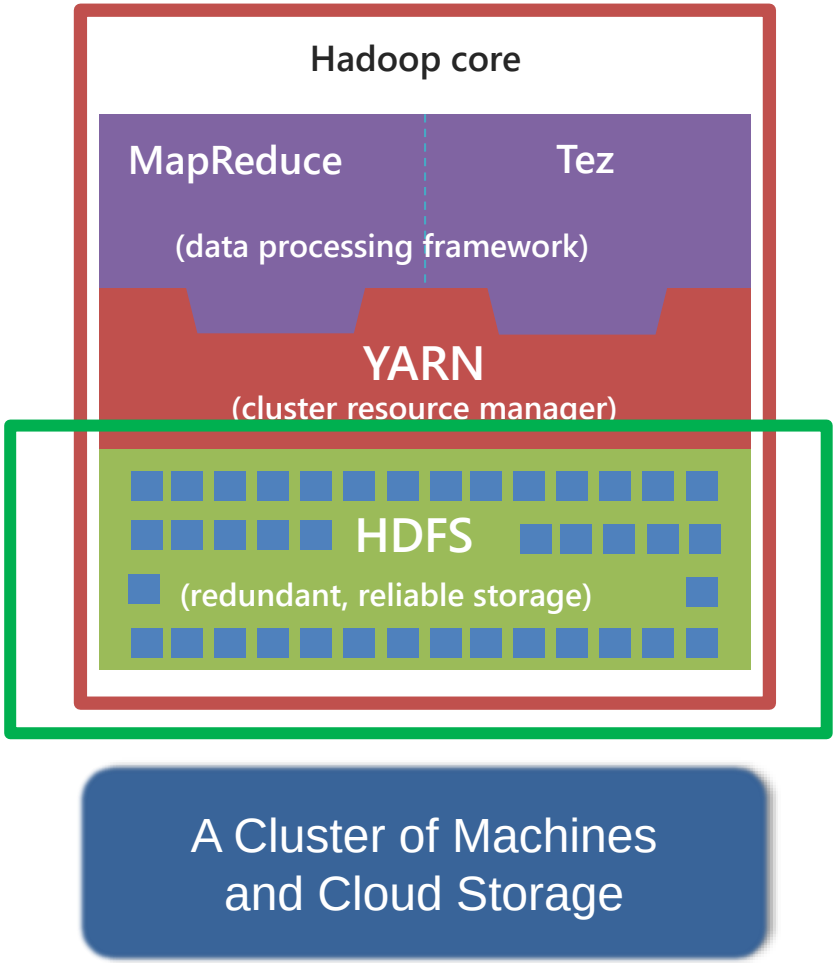
Hadoop is a highly reliable, distributed, and parallel programming framework for analyzing big data

- A Java-based, open source Apache project
- Capable of running on a variety of hardware platforms, including clusters of commodity hardware
- The Hadoop core includes:
  - A scalable, reliable file system (HDFS)
  - A framework that enables development of programs based on MapReduce (MR) or directed acyclic graph (DAG) model
  - YARN, a distributed resource manager that allocates and controls access to resource of cluster manager
- In addition to the core, Hadoop has a rich ecosystem that supports SQL/NoSQL, streaming, real-time, and interactive applications



Hadoop Distributed File System (HDFS)

- A scalable distributed file system for large scale data analysis
- A part of the Open Source Apache Hadoop suite
  - The primary storage used by Hadoop MapReduce applications
- Can run on commodity hardware assuring high fault-tolerant
- HDFS is platform layer for Data Lakes



Data Lakes - Definition

Data Lake allows an organization to store all of their data, structured and unstructured, in one, centralized repository.

- Since data can be stored as-is, there is no need to convert it to a predefined schema and you no longer need to know what questions you want to ask of your data beforehand.

A Data Lake should support the following capabilities:

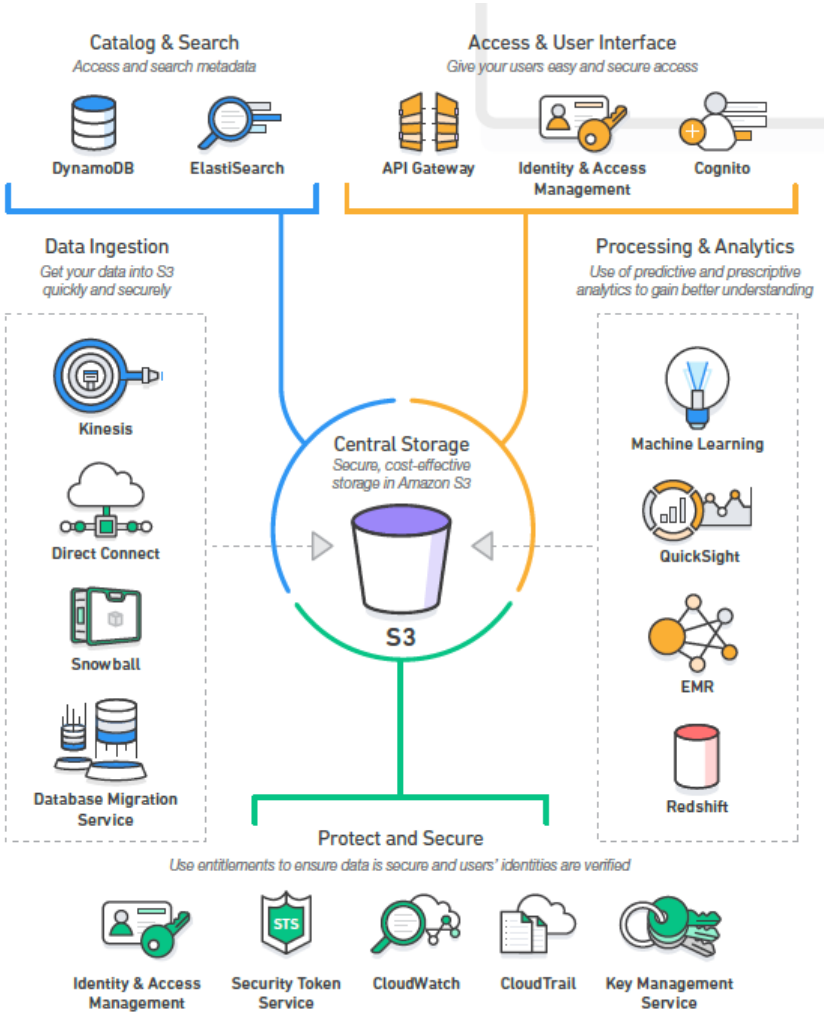
- Collecting and storing any type of data, at any scale and at low costs
- Securing and protecting all of data stored in the central repository
- Searching and finding the relevant data in the central repository
- Quickly and easily performing new types of data analysis on datasets
- Querying the data by defining the data's structure at the time of use (schema on read)

Data Lake layers

- **Raw data layer** – Raw events are stored for historical reference. Also called staging layer or landing area
- **Cleansed data layer** – Raw events are transformed (cleaned and mastered) into directly consumable data sets. Aim is to uniform the way files are stored in terms of encoding, format, data types and content (i.e. strings). Also called conformed layer
- **Application data layer** – Business logic is applied to the cleansed data to produce data ready to be consumed by applications (i.e. DW application, advanced analysis process, etc). Also called workspace layer or trusted layer

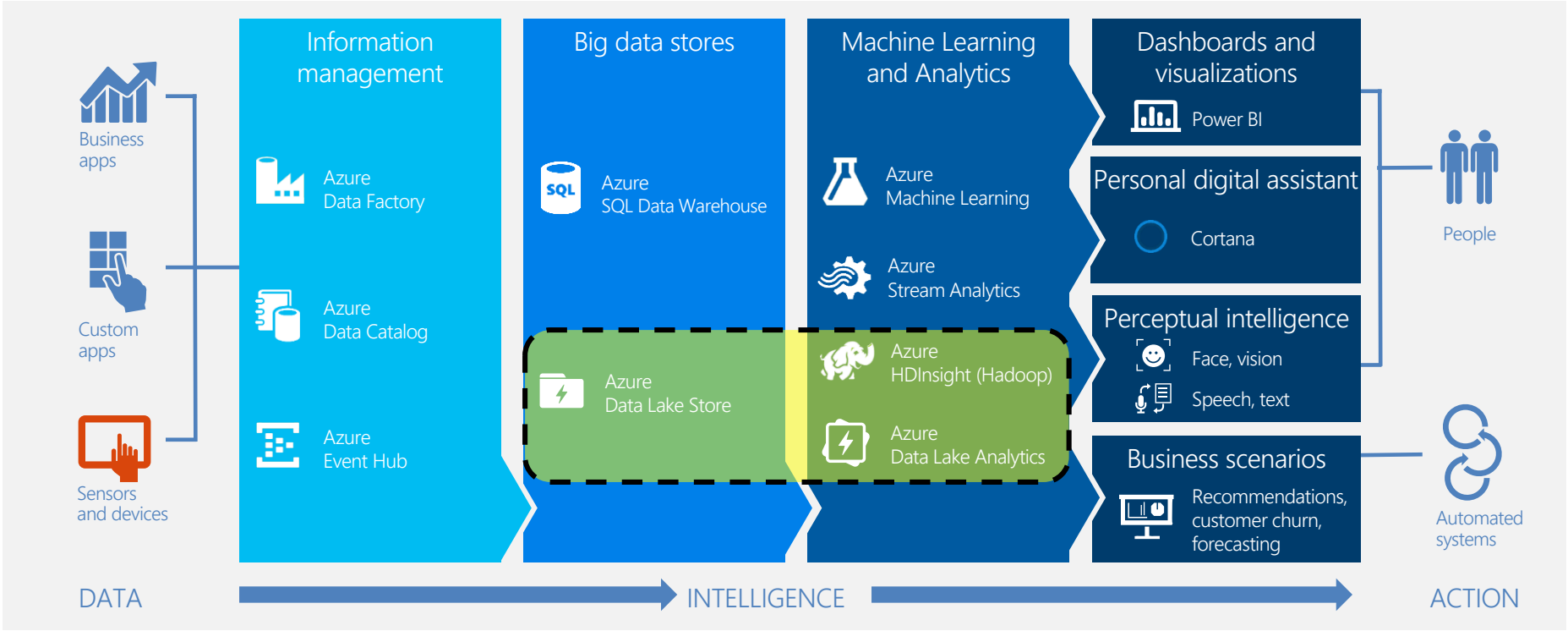
Still need data governance so your data lake does not turn into a data swamp!

AWS Data Lake Storage Architecture



- A Data Lake solution on AWS, at its core, leverages Amazon Simple Storage Service (Amazon S3) for secure, cost-effective, durable, and scalable storage.
- Data Management with DynamoDB
- Data Analysis with AMR

Azure Data Lake

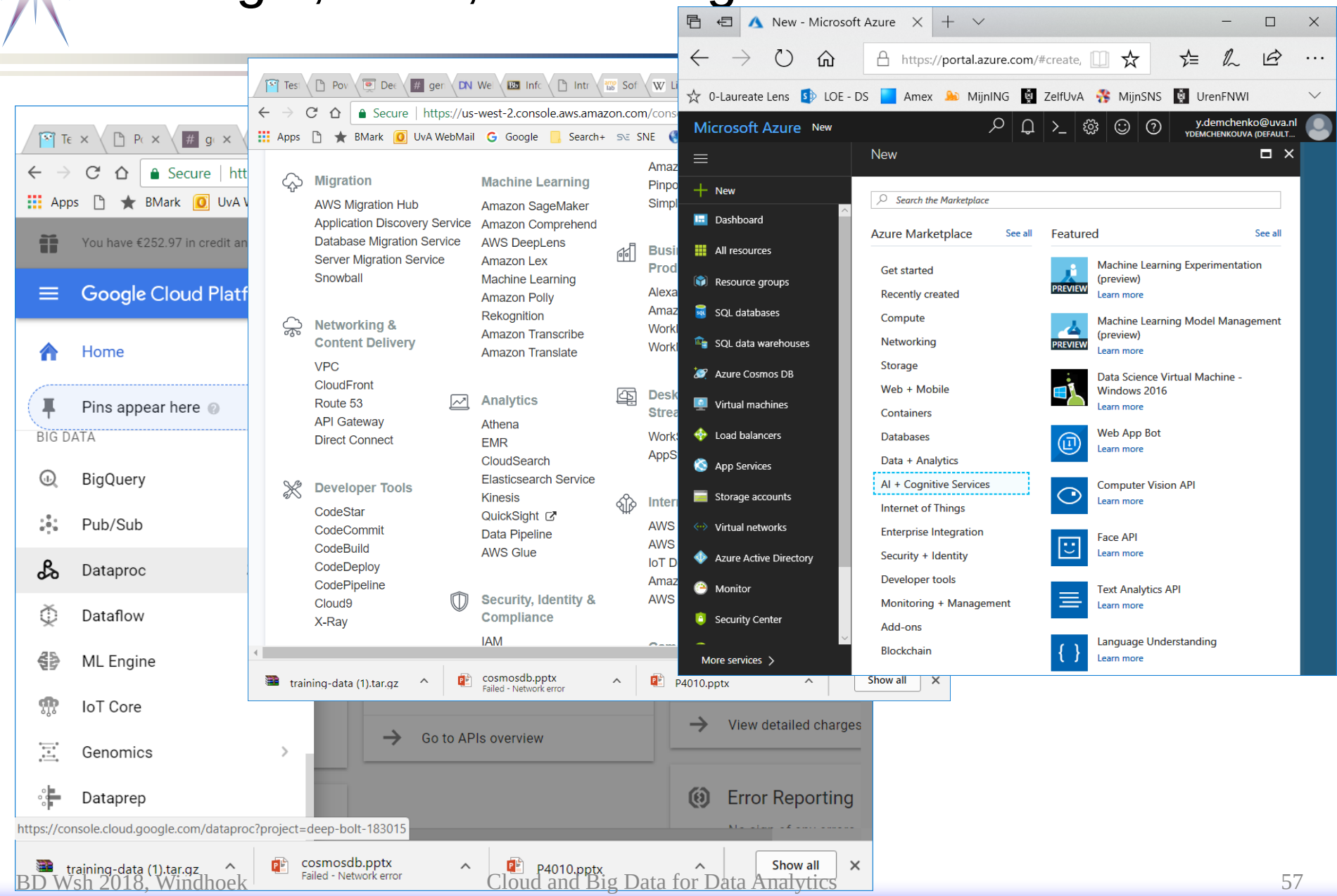


- Azure Data Lake Analytics is a part of Cortana Analytics Suite

Cloud based Big Data Platforms and Solutions

- Amazon Web Services (AWS)
- Google Cloud Platform (GCP)
- Microsoft Azure

Google, AWS, Azure Big Data Stacks



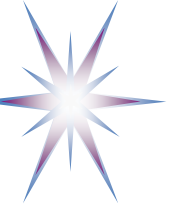
Amazon EC2 HPC oriented AMI instances

- Amazon EC2 offers dedicated computation optimized VMs and “cluster instances” that deliver better performance to HPC users
  - Two Cluster Compute instances that provide a very large amount of CPU coupled with increased network performance (10GigE). Instances come in two sizes,
    - Nehalem-based “Quadruple Extra Large Instance” (eight cores/node, 23GB of RAM, 1.7TB of local storage)
    - Sandy Bridge-based “Eight Extra Large Instance” (16 cores/node, 60.5GB of RAM, 3.4TB of local storage).
  - Besides the Amazon cluster instance, users may select from one of several preconfigured clusters
  - Additionally, two other specialized instances for HPC clusters
    - Cluster GPU instance that provides two NVidia Tesla Fermi M2050 GPUs with proportionally high CPU and 10GigE network performance.
    - High-I/O instance that provides two SSD-based volumes, each with 1024GB of storage.
  - Pricing can vary depending upon on-demand, scheduled, or spot purchase of resources
    - Quadruple Extra Large Instance is US$ 1.3/hour (US$ 0.33/core·hour)
    - Eight Extra Large Instance is US$ 2.4/hour (US$ 0.15/core·hour)
    - Cluster GPU instance is US$ 2.1/hour, and the High I/O instance is US$ 3.1/hour
- For example
  - Small usage case (80 cores, 4GB of RAM per core, and basic storage of 500GB) would cost US$ 24.00/hour (10 Eight Extra Large Instances)
  - Larger usage case (256 cores, 4GB of RAM per core, and 1TB of fast global storage) would cost US$ 38.4/hour (16 Eight Extra Large Instances)
  - Once created, the instances must be provisioned and configured to work as a cluster by the user
- Total cost depends on compute time, total data storage and data transfer cost
  - Amazon does not charge for data transferred into EC2 but has a varying rate schedule for transfer out of the cloud; additionally, there are EC2 storage costs

AWS Cloud Big Data Services

AWS Cloud offers the following services and resources for Big Data processing:

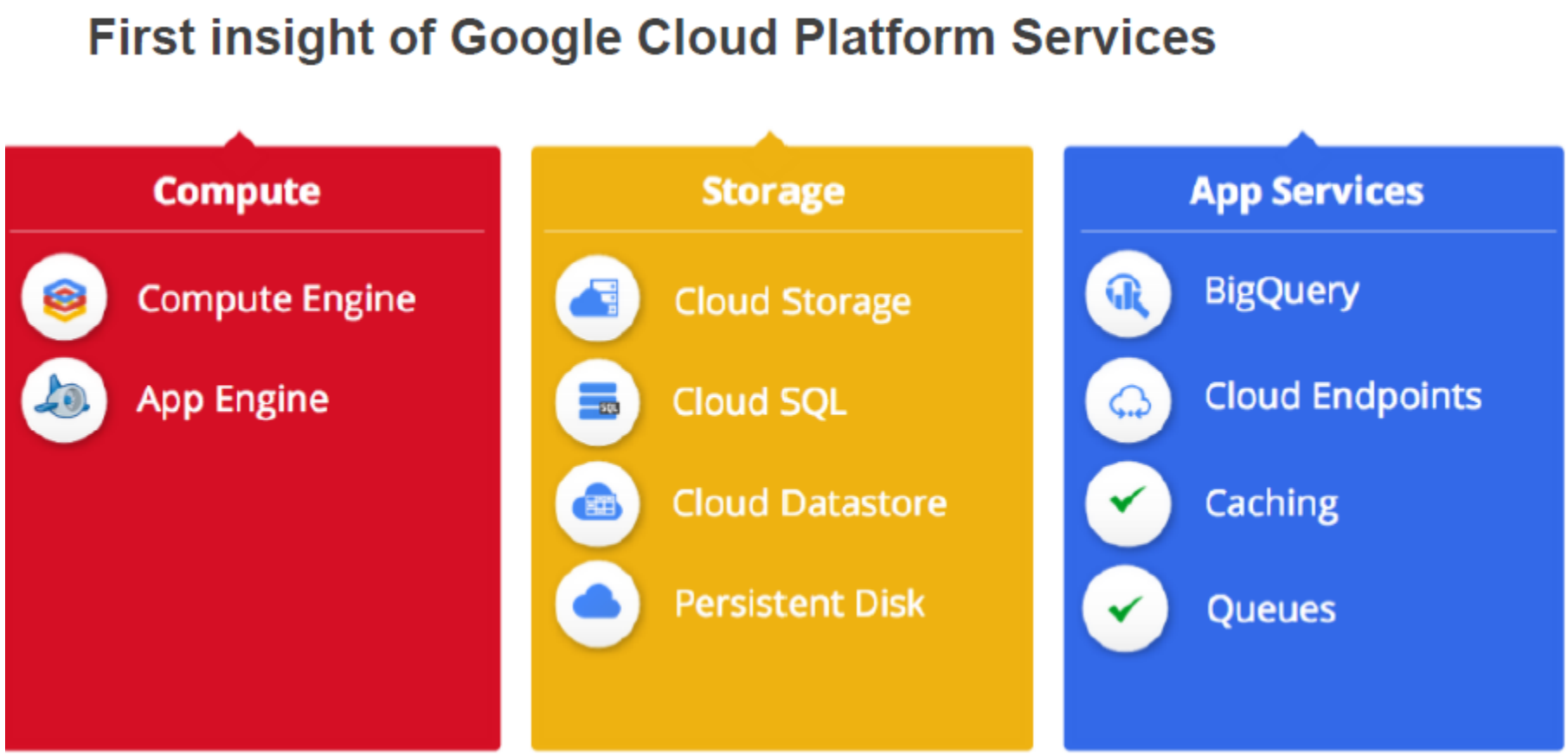
- EC2 Virtual Machine (VM) instances for HPC optimized for computing (with multiple cores) and with extended storage for large data processing.
- **Amazon Elastic MapReduce (EMR)** provides the Hadoop framework on Amazon EC2 and offers a wide range of Hadoop related tools.
- **Amazon Kinesis** is a managed service for real-time processing of streaming big data (throughput scaling from megabytes to gigabytes of data per second and from hundreds of thousands different sources).
- **Amazon DynamoDB** highly scalable NoSQL data stores with sub-millisecond response latency.
- Amazon Aurora scalable relational database.
- Amazon Redshift fully-managed petabyte-scale data warehouse in cloud at cost less than $1000 per terabyte per year.
- Amazon Machine Learning
- Machine Learning (Artificial Intelligence) based services (Lex, Translate, Recognition, etc.)



Google Cloud Platform (GCP)

- Compute
  - AppEngine
  - Google Functions (serverless with Node.js)
- Storage: Static, sharing, backup, for applications and computation
  - Cloud Spanner SQL database
- Big Data
  - BigQuery – Hadoop Data Warehouse
- Machine Learning services
  - Translate
  - Prediction
- Cloud endpoints

Google Cloud Platform (GCP) structure



Google App Engine (**GAE**) is a Platform as a Service (PaaS) cloud computing platform for developing and hosting web applications in Google-managed data centers.

Java
Python
PHP
GO

Google's Go:

- Go is an Google's open source programming environment.
- Tightly coupled with Google App Engine.
- Applications can be written using App Engine's Go SDK.

The sandbox:

- All hosted applications run in a **secure environment** that provides limited access to the underlying operating system.
- **Sandbox isolates the application** in its own secure, reliable environment.
- Limitations imposed by sandbox (for security):
  - An application can only access other computers over internet using the provided URL fetch and email services.

Free ???

Yes, free for up to 1 GB of storage and enough CPU and bandwidth to support 5 million page views a month. 10 Applications per Google account.

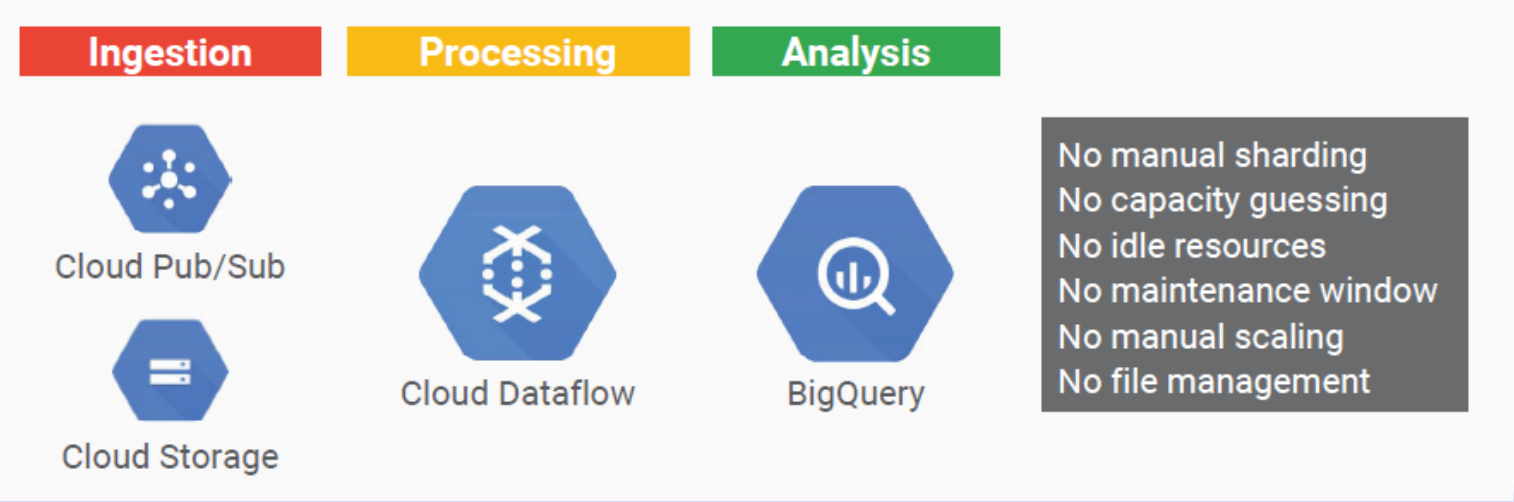
Machine Learning Focus

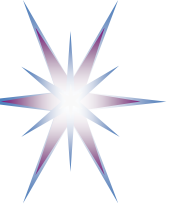
- Machine Learning embedded across most products
- Multiple Tensorflow ML models in use
  - Portable TensorFlow models
- Key models exposed via APIs (Democratizing Machine Learning)
  - Cloud Video Intelligence API
  - Cloud Vision API
  - Cloud Natural Language API
  - Cloud Translation API
  - Cloud Speech API

- Acquired Kaggle in 2017 - Data Science Enthusiasts

Google Machine Learning

- Support all stages of ML workflow
- Dataprep: Serverless platform for all stages of the analytics data lifecycle



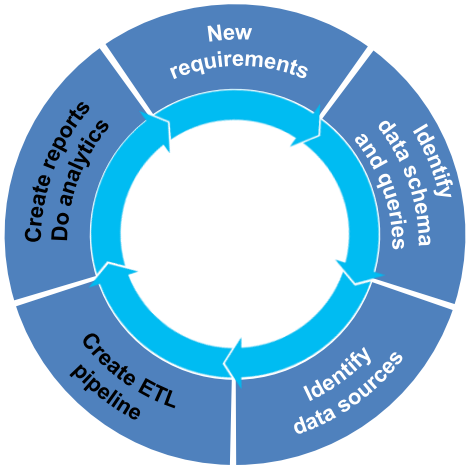


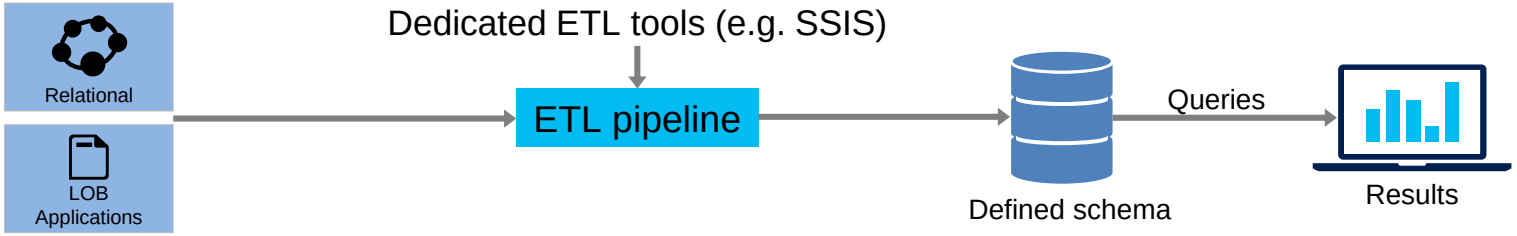
Microsoft Azure Big Data Services

- New Big Data data-centric thinking
- Azure Data Lakes
- HDInsight

Traditional business analytics process

1. Start with end-user requirements to identify desired reports and analysis
2. Define corresponding database schema and queries
3. Identify the required data sources
4. Create a Extract-Transform-Load (ETL) pipeline to extract required data (curation) and transform it to target schema (‘*schema-on-write*’)
5. Create reports, analyze data

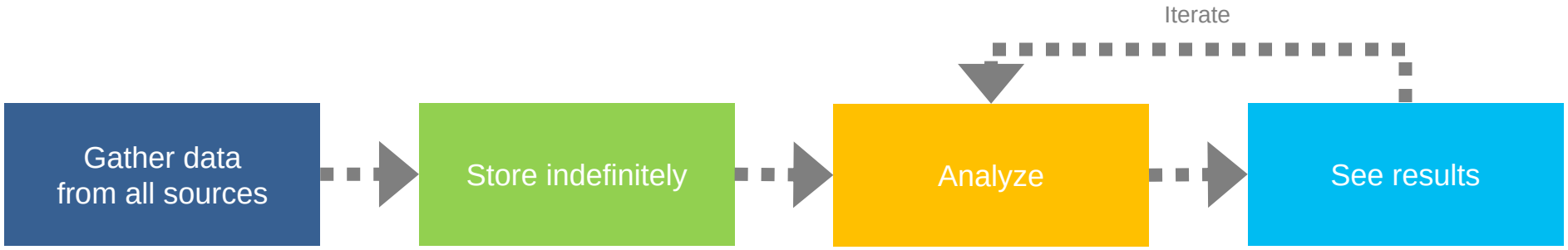




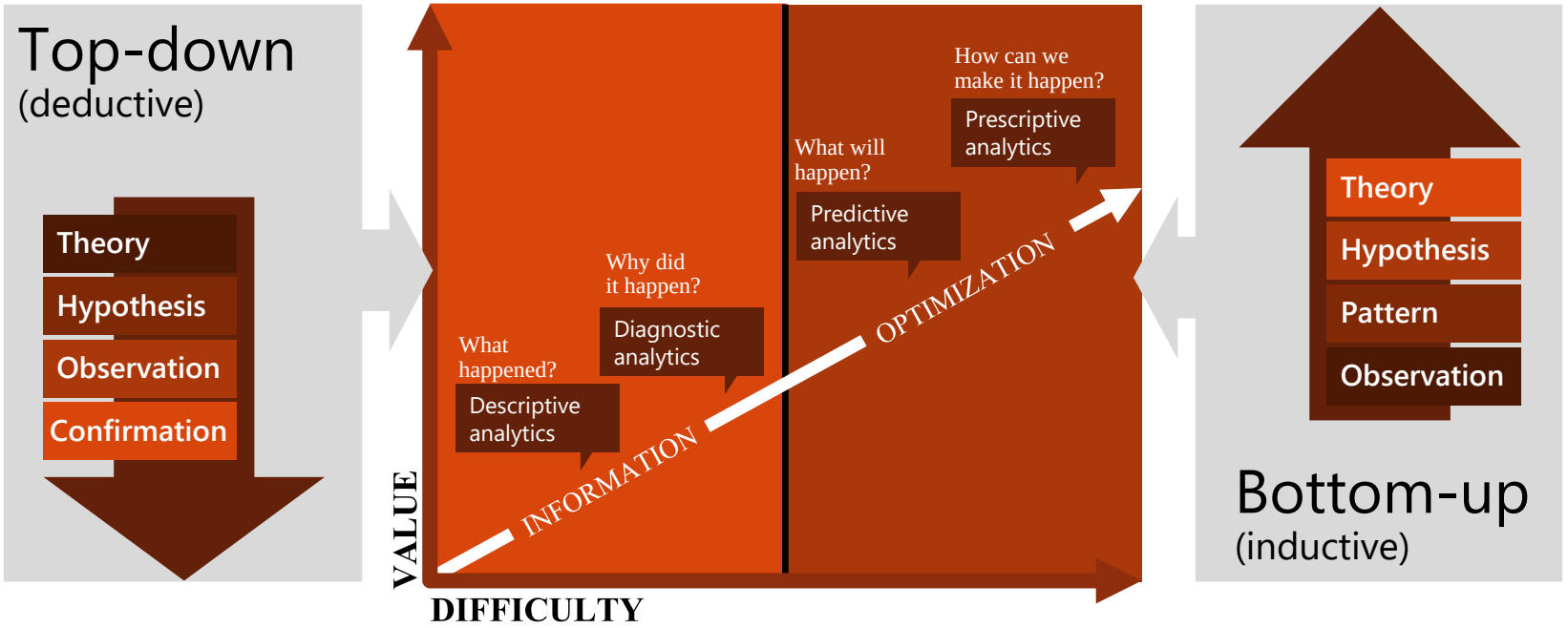
All data not immediately required is discarded or archived

New Big Data thinking: All data has value

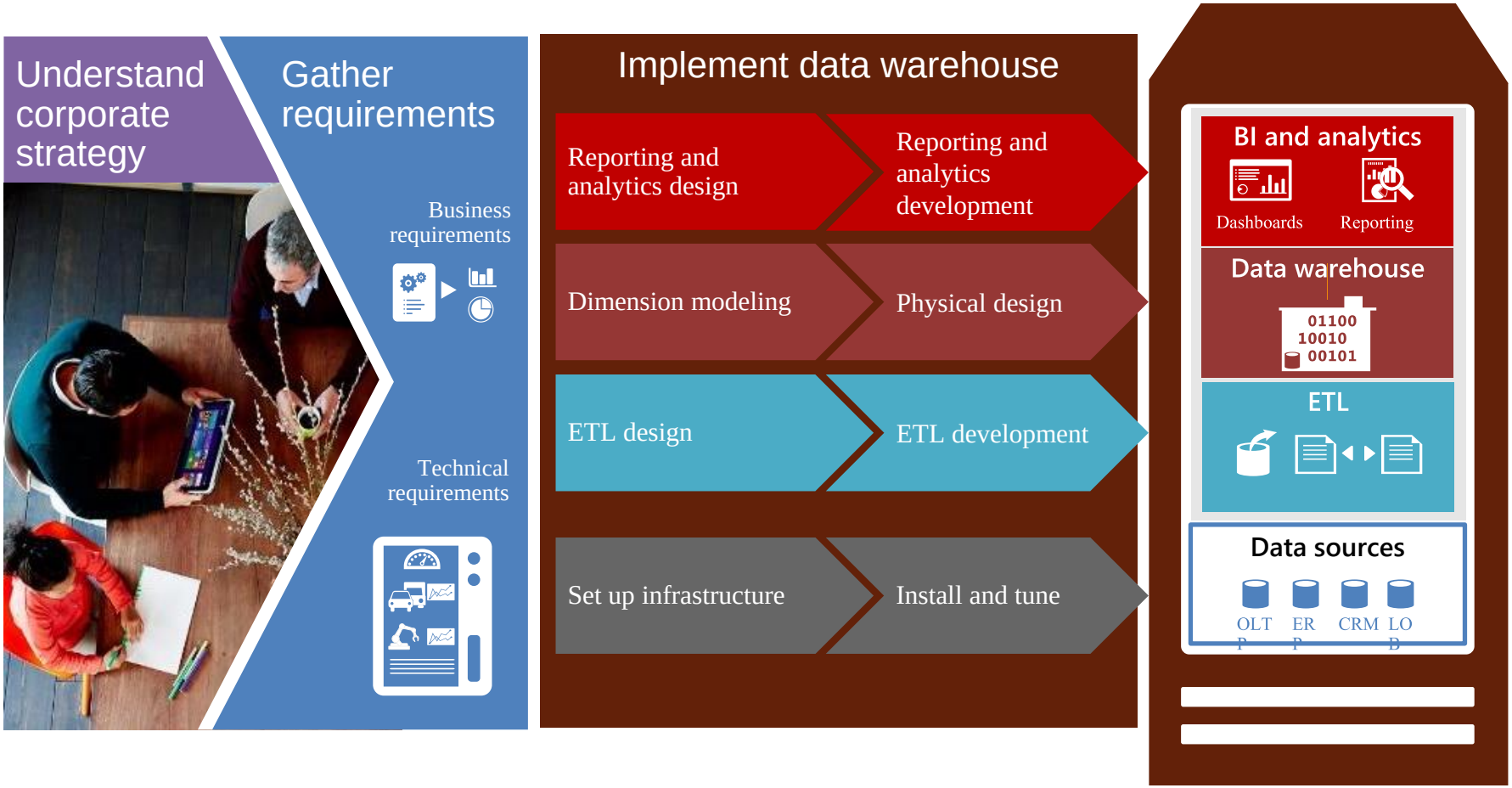
- All data has potential value
- Data hoarding
- No defined schema—stored in native format
- Schema is imposed and transformations are done at query time *(schema-on-read).*
- Apps and users interpret the data as they see fit



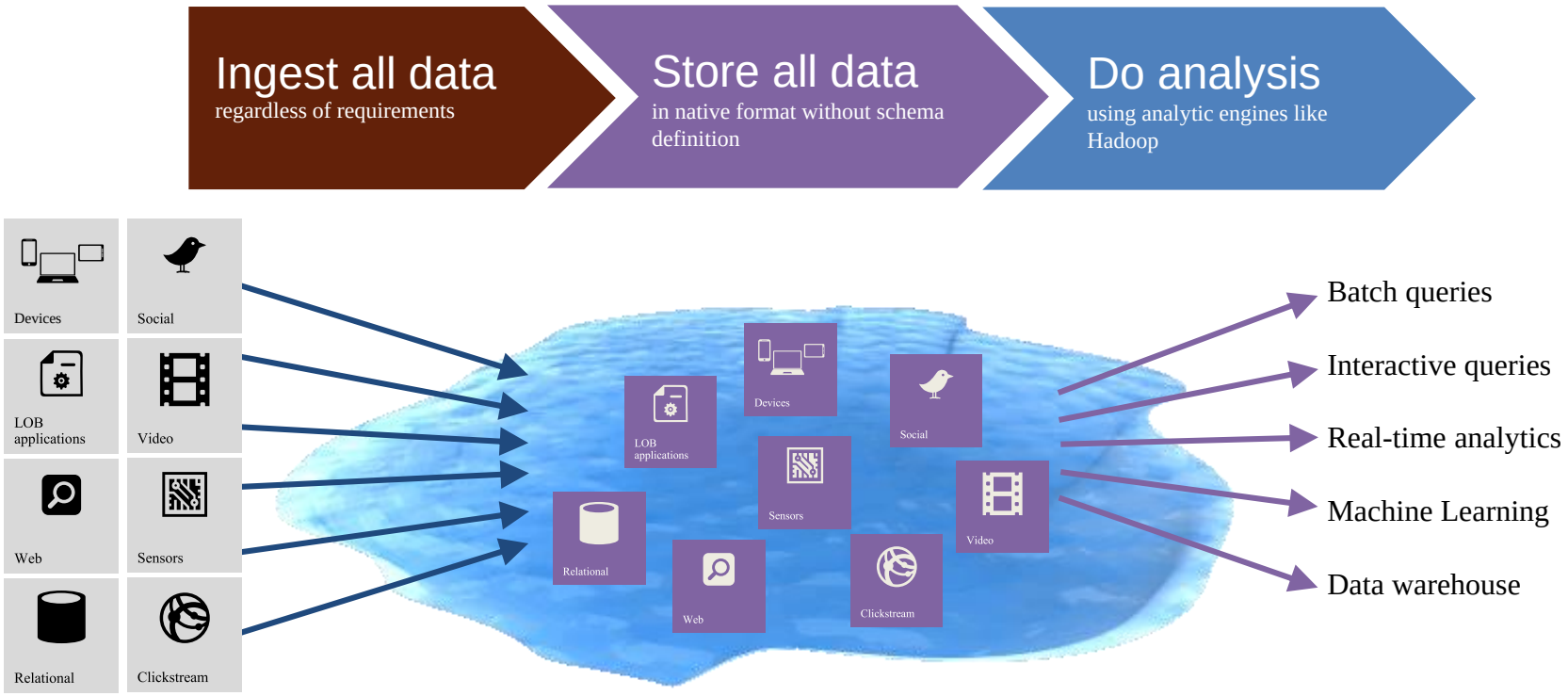
Two approaches to information management for analytics: Top-down and bottom-up



Data Warehousing uses a top-down approach

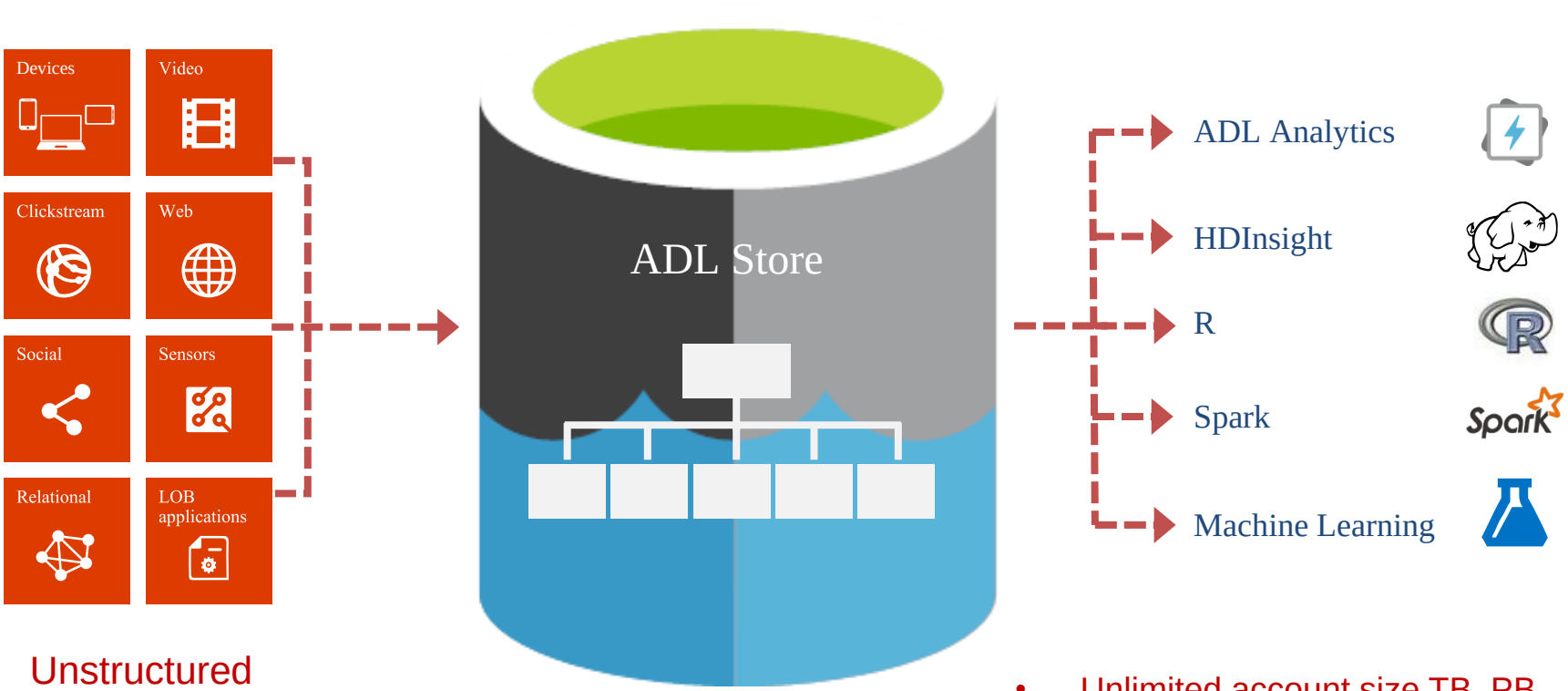


The Data Lake uses a bottom-up approach



What is Azure Data Lake (ADL) Store?

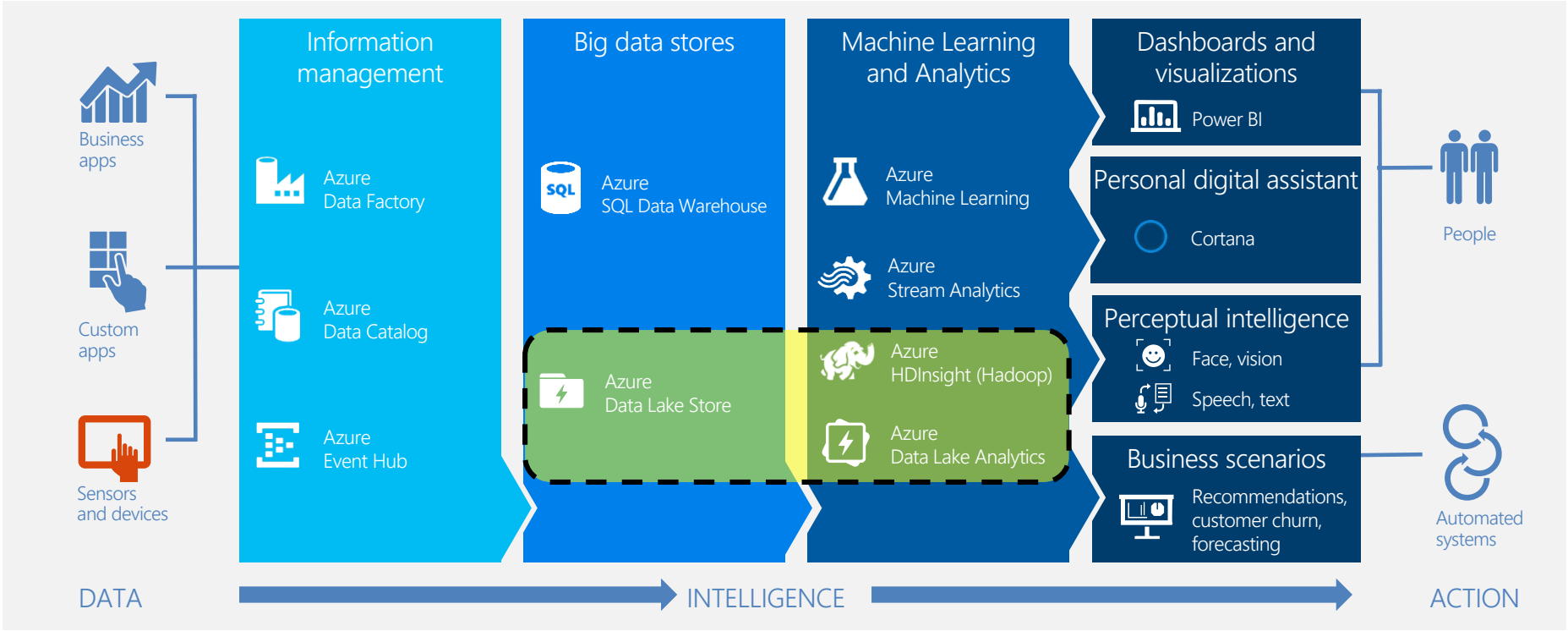
- A highly scalable, distributed, parallel file system in the cloud specifically designed to work with multiple analytic frameworks



- Unstructured
- Semi-structured
- Structured

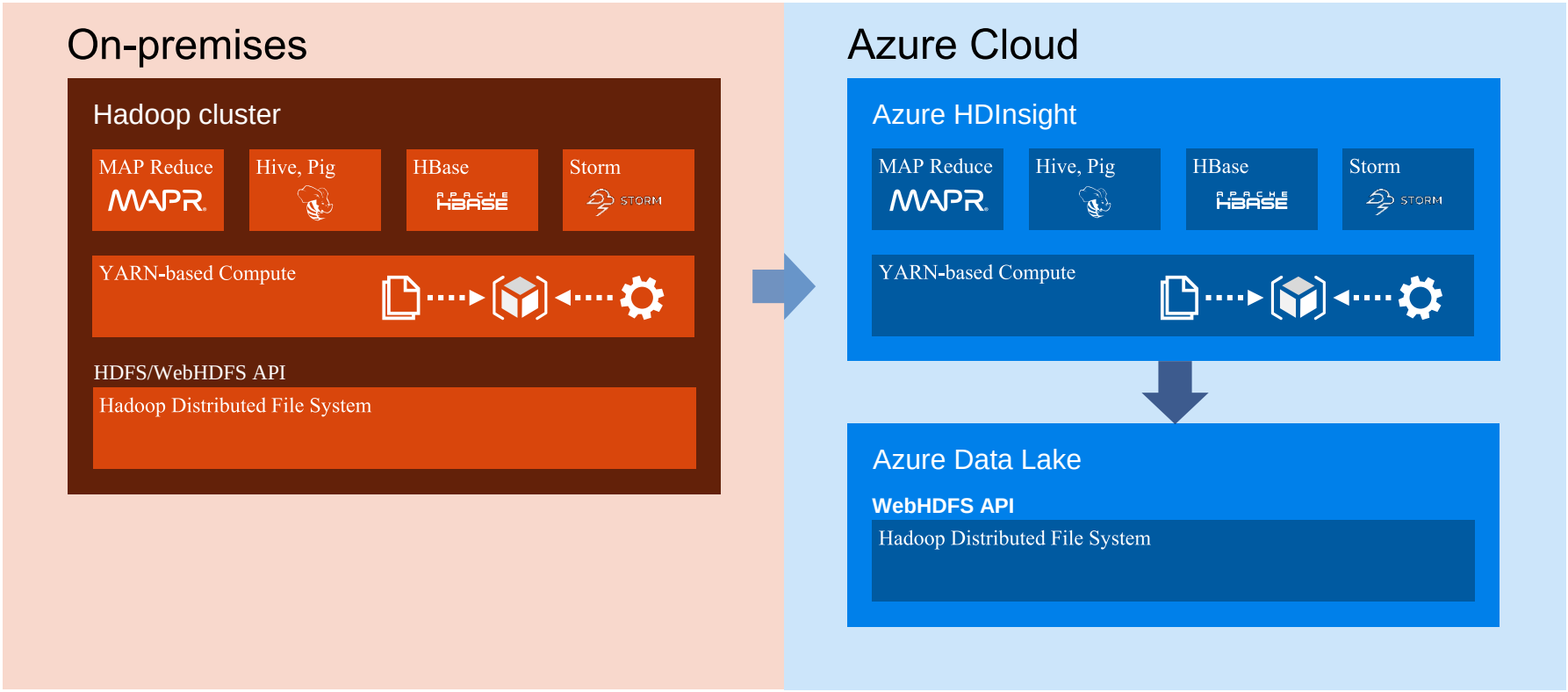
- Unlimited account size TB, PB
- Individual files size from gigabytes to petabytes
- No limits to scale

Azure Data Lake



- Part of Cortana Analytics Suite

Hybrid Data Lake Model in Azure



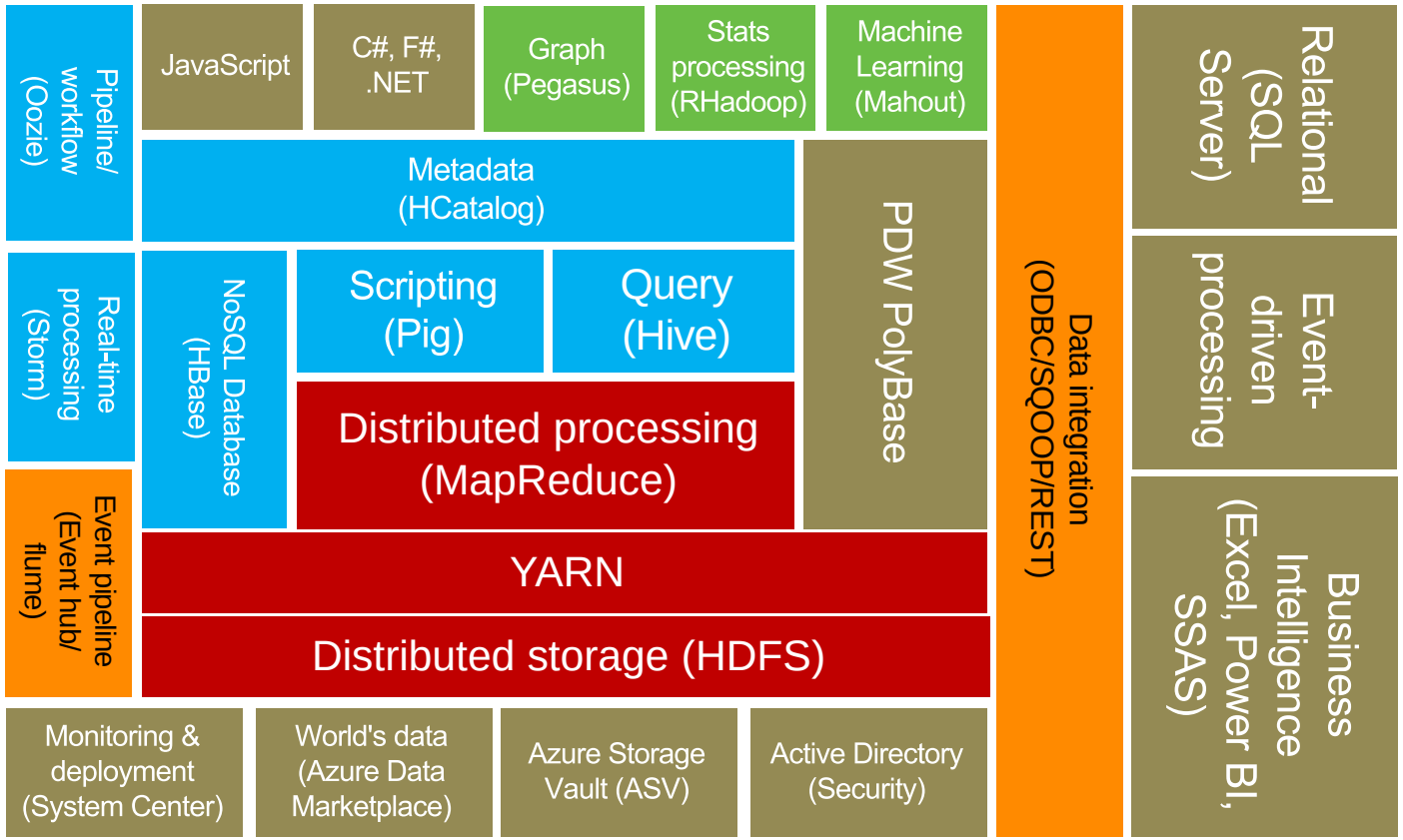
Azure HDInsight – What is it?

A standard Apache Hadoop distribution offered as a managed service on Microsoft Azure

- Based on Hortonworks Data Platform (HDP)
- Provisioned as clusters on Azure that can run on Windows or Linux servers
- Offers capacity-on-demand, pay-as-you-go pricing model
- Integrates with:
  - Azure Blob Storage and Azure Data Lake Store for Hadoop File System (HDFS)
  - Azure Portal for management and administration
  - Visual Studio for application development tooling

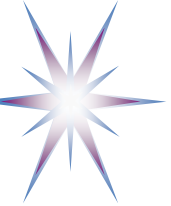
In addition to the core, HDInsight supports the Hadoop ecosystem

HDInsight and Hadoop ecosystem (2017)



Legend
Red =
Core Hadoop
Blue =
Data processing (Hadoop)
Gray = Microsoft integration points and value adds
Orange =
Data movement
Green =
Packages
Azure
Infrastructure components

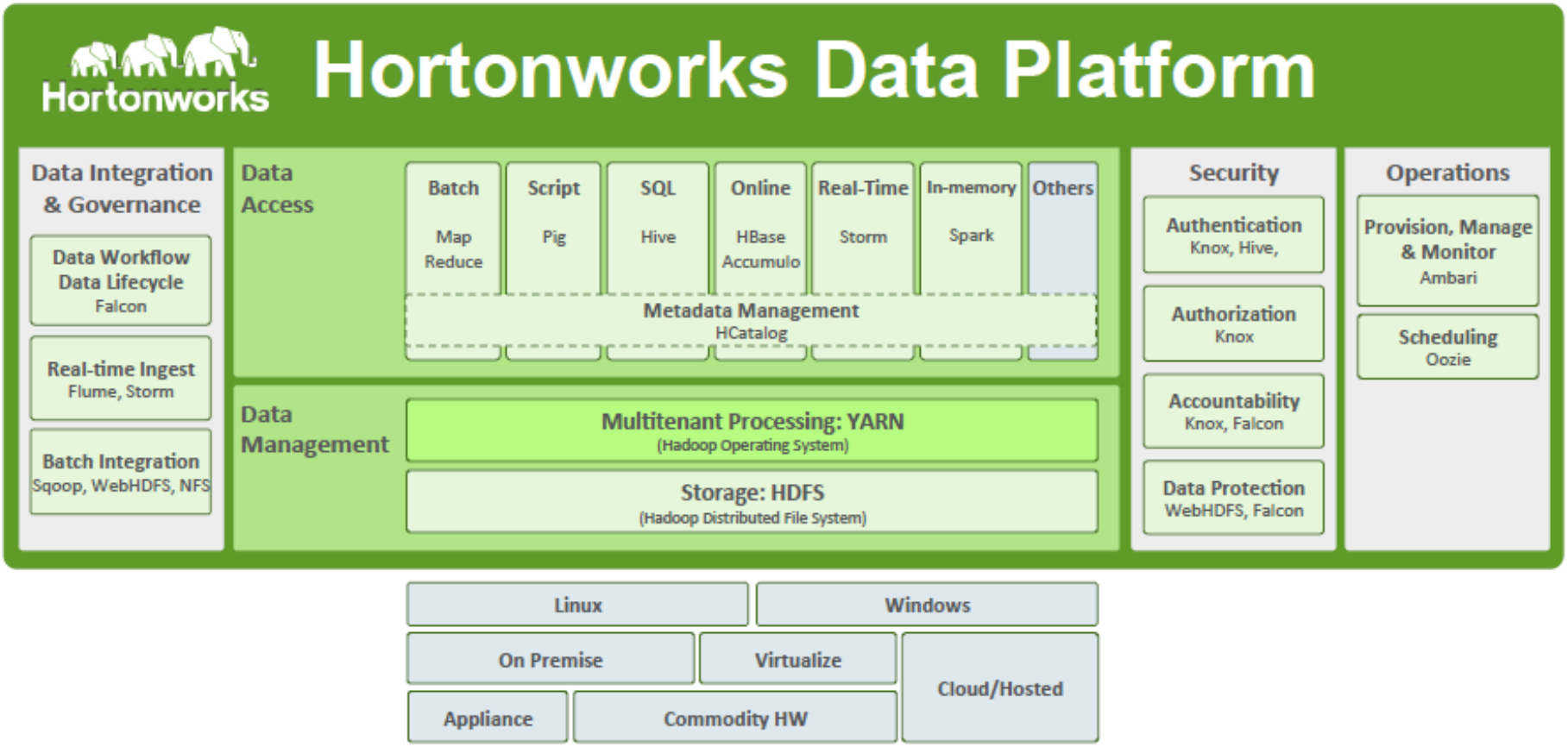
HDInsight supports Mahout, HBase, Storm, Hive



Hortonworks Data Platform (HDP)
http://hortonworks.com/

- HDP delivers a single integrated Hadoop platform for enterprises
  - Provides a data platform for multi-workload data processing across an array of processing methods including batch and interactive to real-time
  - Supports key capabilities of an enterprise data platform: Governance, Security and Operations
  - YARN and Hadoop Distributed Filesystem (HDFS) are the core components of HDP
- YARN is treated as datacenter OS and supports multiple access methods (batch, real-time, streaming, in-memory, and more) on a common data set
  - YARN is the architectural center of Hadoop that allows to process data simultaneously in multiple ways
  - Allows creating multi-tenant data analytics applications
- HDP runs natively on Linux and Windows OS
  - HDP provides the basis for Microsoft's HDInsight Service meaning complete portability of data is retained on-premise and in the cloud
  - Available in integrated hardware from Teradata
- Hortonworks provides a simple starters solution Hadoop Sandbox
  - Hortonworks Sandbox is a single-node implementation of Hadoop based on the Hortonworks Data Platform that includes all the typical components found in a Hadoop deployment

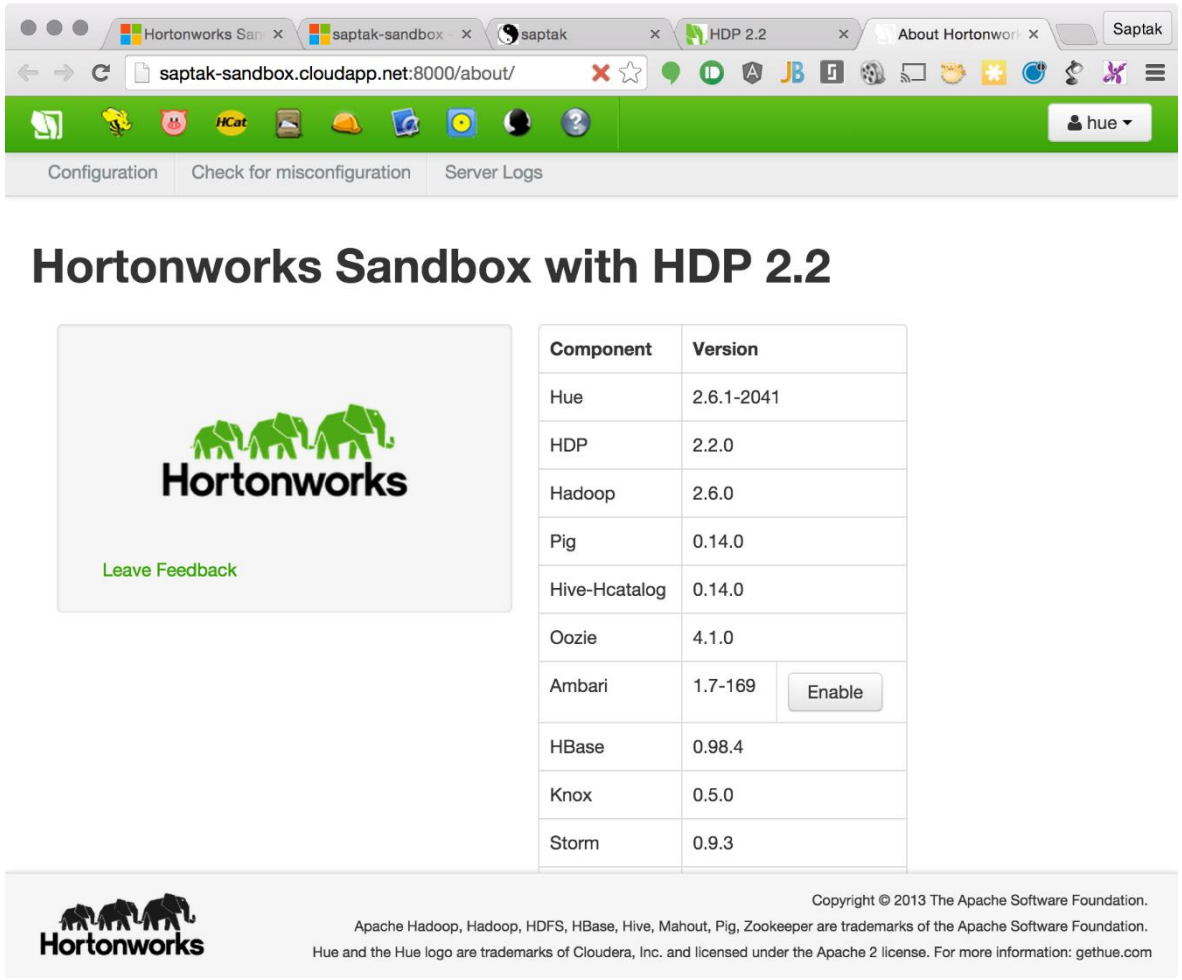
Hortonworks Data Platform Architecture [ref]



- HDP includes the most recent developments of the Open Source Hadoop suite
- Can run on Linux and on Windows OS
- Can be deployed on premises on dedicated cluster and on cloud as a hosted application

[ref] http://hortonworks.com/hdp /

HortonWorks Sandbox VM

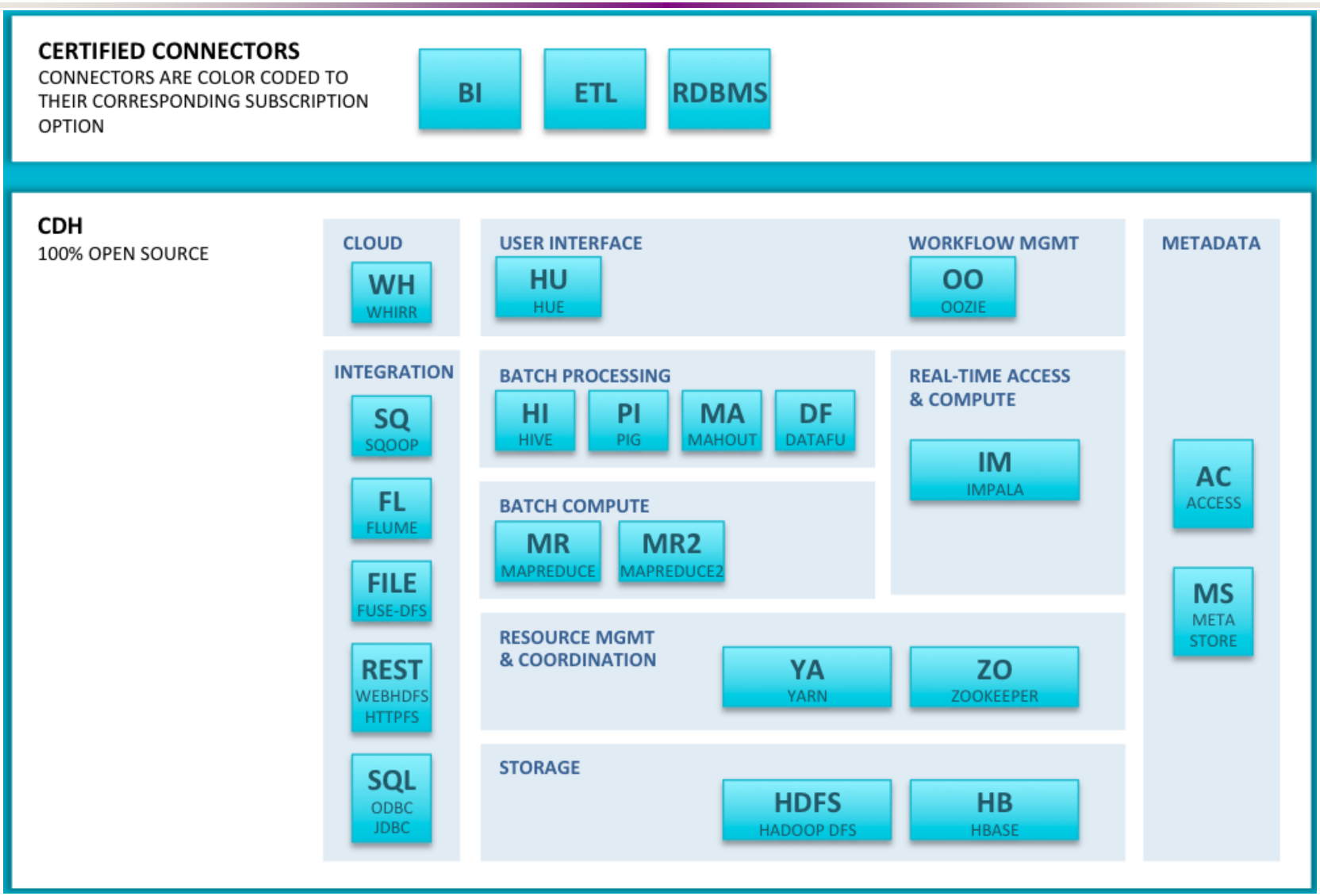


Simple starters solution Hadoop Sandbox

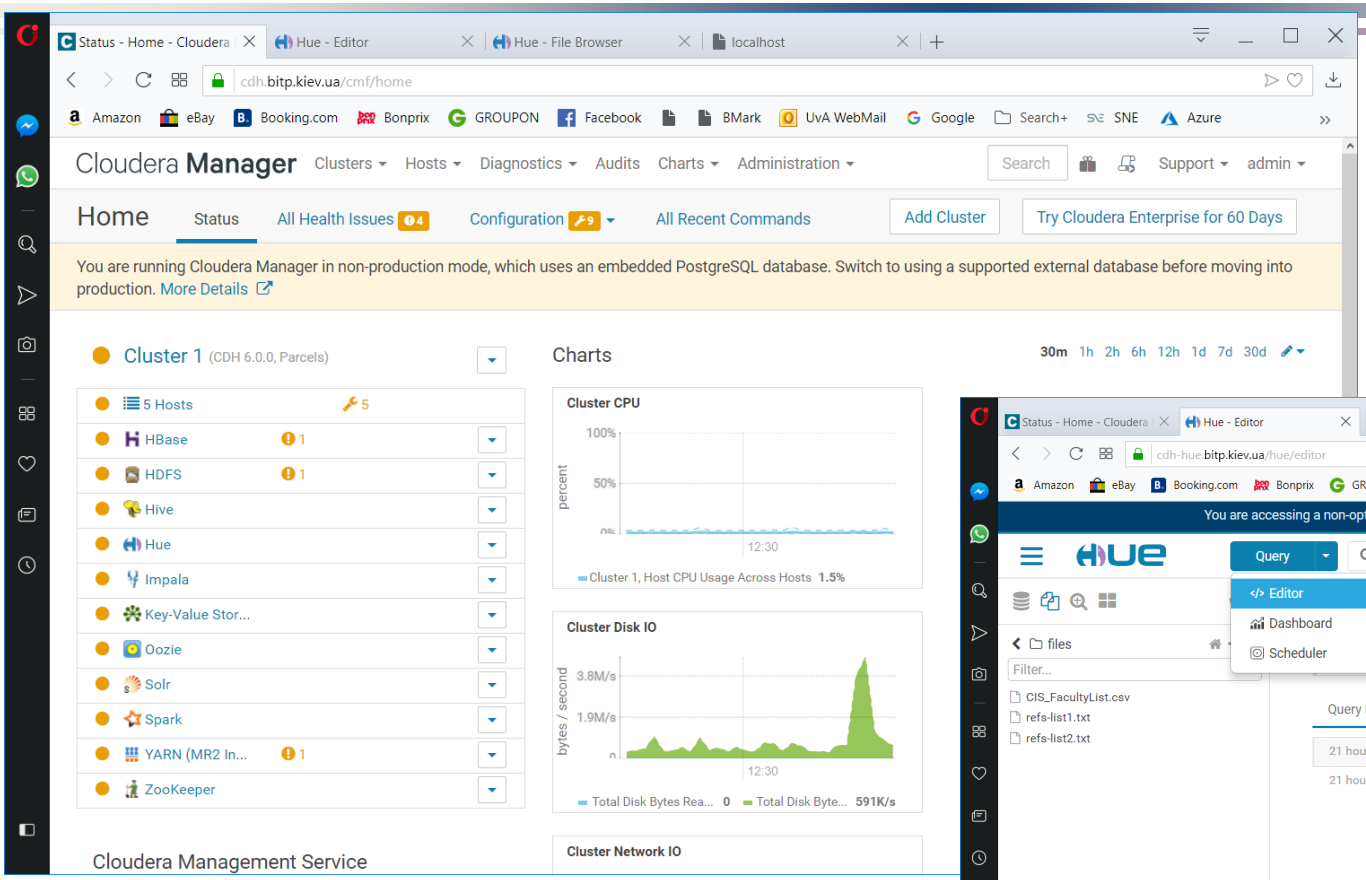
- Hortonworks Sandbox is a single-node implementation of Hadoop based on the Hortonworks Data Platform
- Includes all the typical components found in a Hadoop deployment

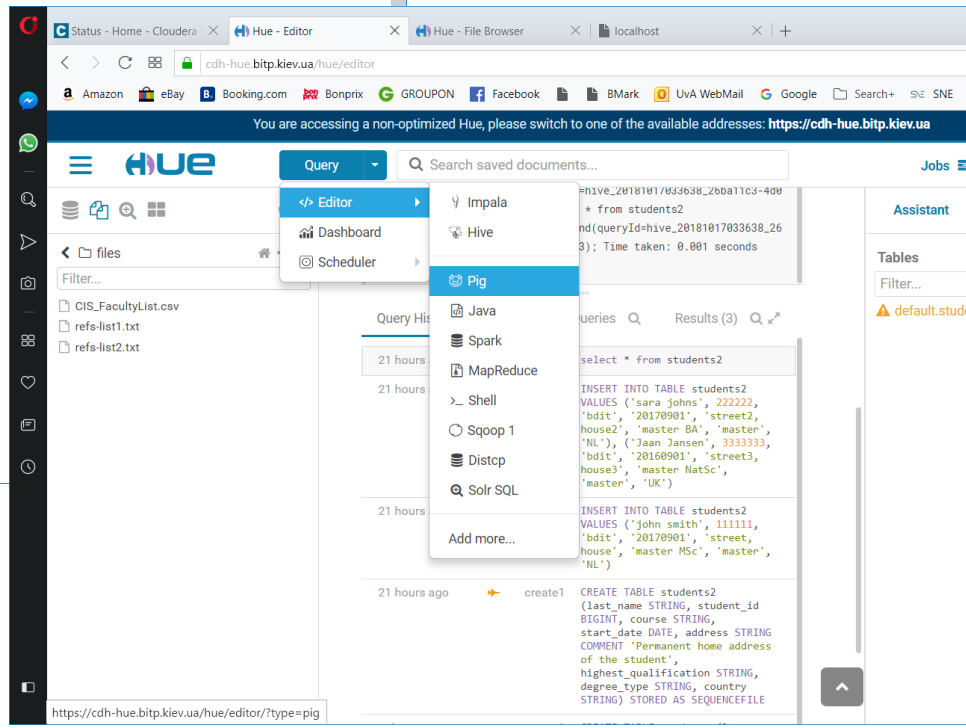
Cloudera Hadoop Cluster Architecture

https://www.cloudera.com/downloads/quickstart_vms/5-13.html



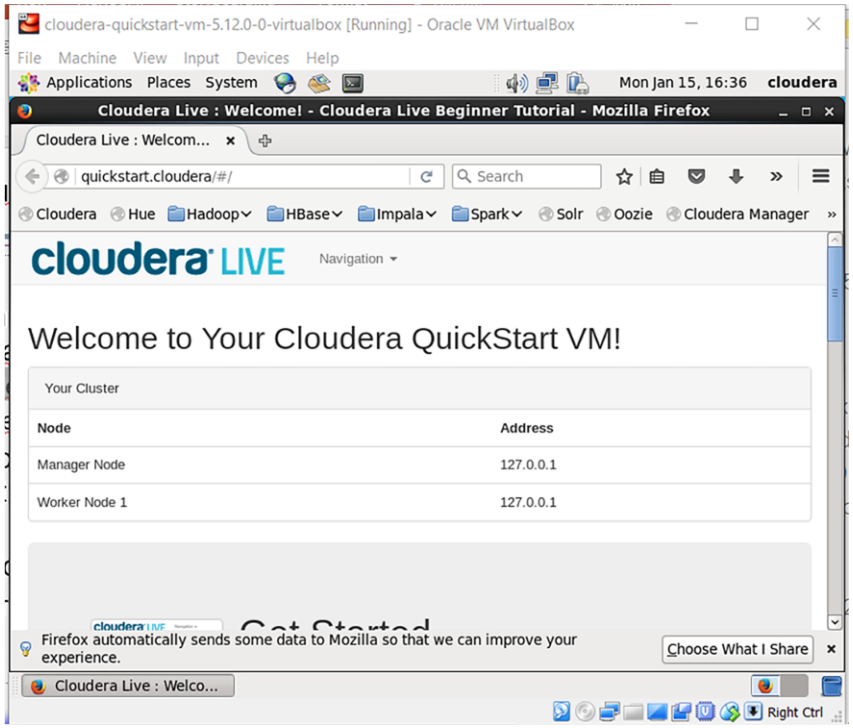
Cloudera Hadoop cluster on cloud





Cloudera Quickstart VM for VirtualBox

https://www.cloudera.com/downloads/quickstart_vms/5-13.html

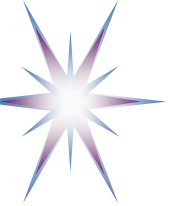


Accounts

- Once you launch the VM, you are automatically logged in as the cloudera user. The account details are:
  - username: cloudera
  - password: cloudera
- The cloudera account has sudo privileges in the VM. The root account password is cloudera.
- The root MySQL password (and the password for other MySQL user accounts) is also cloudera.
- Hue and Cloudera Manager use the same credentials.

Summary and take away

- Cloud is a platform of choice for Big Data and Data Analytics applications and tasks
- Hadoop is a standard de facto platform for Big Data Analytics
- All major CSP provide variety of Big Data Analytics services: AWS, Azure, GCP
- HDFS is a commonly recognised storage for Big Data and scalable data processing
- Data Lakes is new model for Big Data and ELT (Extract – Load – Transfer) processes



Additional Materials

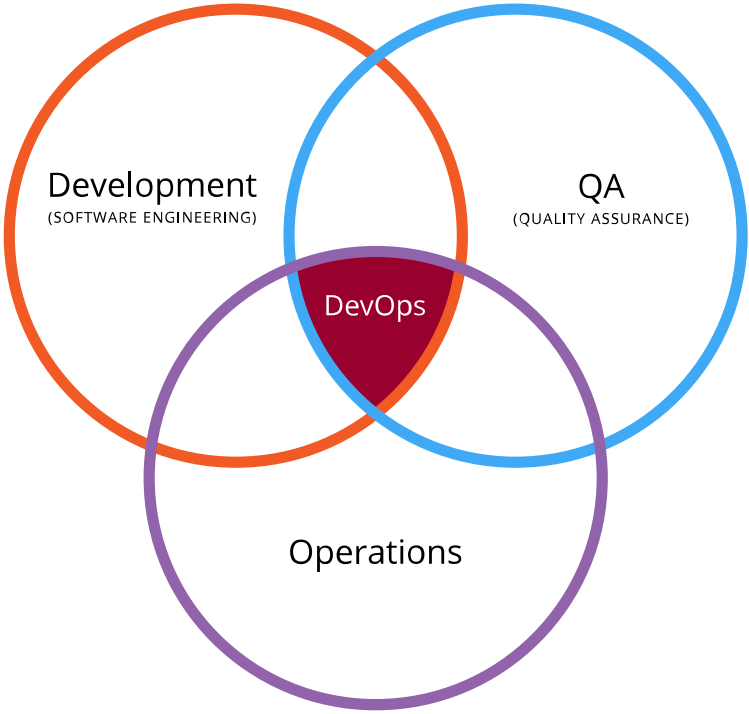
- DevOps and DataOps
- Data Markets and Data as economic goods

DevOps

DevOps is the practice of operations and development engineers participating together in the entire service lifecycle, from design through the development process to production support.

DevOps Essentials

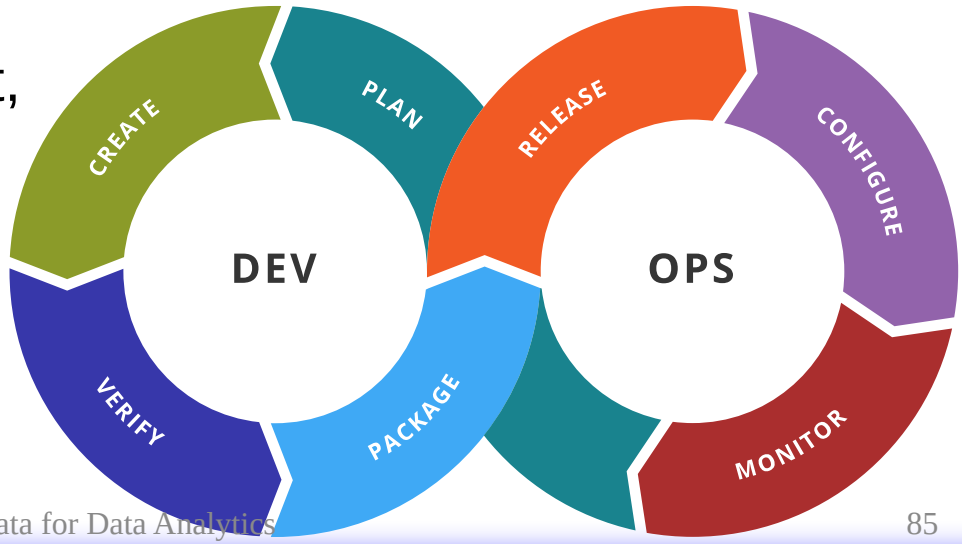
- Better Software, Faster time to market
- Movement Comes from Open Source
- Synergy of **Development and Operations**
- Covers the *entire* Application LifeCycle



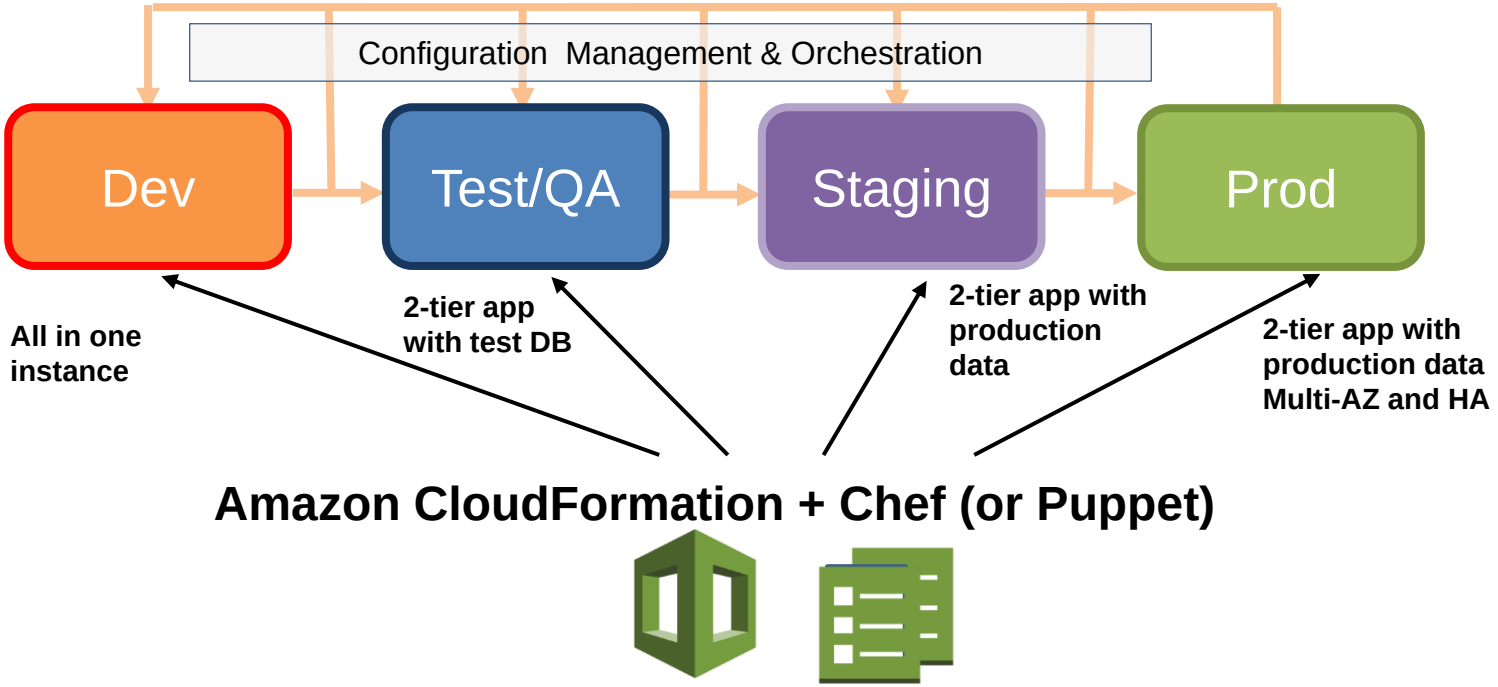
DevOps Toolchain

DevOps is a cultural shift and collaboration between development, operations and testing, enabled by DevOps toolchain

- **Code** — Code development and review, version control tools, code merging
- **Build** — Continuous integration tools, build status
- **Test** — Test and results determine performance
- **Package** — Artifact repository, application pre-deployment staging
- **Release** — Change management, release approvals, release automation
- **Configure** — Infrastructure configuration and management, Infrastructure as Code tools
- **Monitor** — Applications performance monitoring, end–user experience



Cloud-powered Services Development Lifecycle



- Easily creates test environment close to real
- Powered by cloud deployment automation tools
  - To enable configuration Management and Orchestration, Deployment automation
- Continuous development – test – integration
  - CloudFormation Template, Configuration Template, Bootstrap Template
- Can be used with Puppet and Chef, two configuration and deployment management systems for clouds

[ref] Building Powerful Web Applications in the AWS Cloud" by Louis Columbus
http://softwarestrategiesblog.com/2011/03/10/building-powerful-web-applications-in-the-aws-cloud/

Azure DevOps Services
(since 10 Sept 2018, former VSTS)

Azure DevOps Services is a cloud service for collaborating on code development.

- Azure Pipelines
  - CI/CD that works with any language, platform, and cloud.
- Azure Repos
  - Unlimited cloud-hosted private Git and TFVC repos for your project.
- Azure Boards
  - Work tracking with Kanban boards, backlogs, team dashboards, and custom reporting.
- Azure Test Plans
  - All-in-one planned and exploratory testing solution.
- Azure Artifacts
  - Maven, npm, and NuGet package feeds from public and private sources.
- Built-in wiki for sharing information with DevOps team

Azure DevOps Processes

Continuous integration (CI)

- Take advantage of continuous integration to improve software development quality and speed. When you use Azure DevOps or Jenkins to build apps in the cloud and deploy to Azure, each time you commit code, it's automatically built and tested—so bugs are detected faster.

Continuous delivery (CD)

- Ensure that code and infrastructure are always in a production-deployable state, with continuous delivery. By combining continuous integration and infrastructure as code (IaC), you'll achieve identical deployments and the confidence you need to manually deploy to production at any time.

Continuous deployment with CI/CD

- With continuous deployment, you can automate the entire process from code commit to production if your CI/CD tests are successful. Using CI/CD practices, paired with monitoring tools, you'll be able to safely deliver features to your customers as soon as they're ready.

DataOps

- DataOps (data operations) is an emerging discipline that brings together DevOps teams with data engineer and data scientist roles to provide the tools, processes and organizational structures to support the data-focused enterprise.
- DataOps is a new approach to the end-to-end data lifecycle, which applies new processes and methodologies to data analytics.
- Agile software development helps deliver new analytics faster and with higher quality.
- DevOps automates the deployment of new analytics and data.
- Statistical process controls, used in lean manufacturing, test and monitor the quality of data flowing through the data-analytics pipeline.

Data Exchange and Data Markets

- Next EU Horizon2020 Programme Call on Data Markets Infrastructure
  http://ec.europa.eu/research/participants/portal/desktop/en/opportunities/h2020/topics/ict-13-2018-2019.html

- BoF on Data property as economic goods at IDW2018 & RDA12 Plenary on 5-8 Nov 2018 in Gaborone Botswana
  https://www.rd-alliance.org/bof-data-properties-economic-goods-rda-12th-plenary-meeting

Data Exchange and Data Markets

- IoT is considered as a key use case and a facilitator for Data Markets
  - Potentially many consumers for centrally or locally operated IoT infrastructure
  - IoT networks create valuable data that can be used for multiple purpose
  - IoT data can be exchanged and traded
- Open and Public data
- Current trading models are simple and in most cases are free or by subscription
- Making data economical goods would require new operational models, infrastructure and tools

Modern data architecture vs Data Market

Characteristics of modern data architecture

1. Customer-centric
2. Automated
3. Smart
4. Adaptable, Agile
5. Cloud based, Elastic
6. Collaborative
7. Governed
8. Secure, Trusted

Characteristics of emerging data markets

1. Customer-centric
2. Automated
3. Smart
4. Regional/sectoral specialised
5. Cloud powered/integrated
6. Collaborative
7. Governed
8. Secure, Trusted
9. Auditable
10. Transparent
11. Commoditised/Monetised
12. Combining data and algorithms (as part of containers)

Data Properties as Economic Goods

STREAM data principles for industrial and commoditised data

- **[S] Sovereign**
- **[T] Trusted**
- **[R] Reusable**
- **[E] Exchangeable**
- **[A] Actionable**
- **[M] Measurable**

- Other data properties: Important ***to commoditise*** data
  - Quality, Valuable, Auditable/Trackable, Brandable, Authentic
  - Interoperable, Findable, Accessible, not-Rival, Composable
  - Ownership and IPR

- Leverages FAIR principles for research data
  - Findable – Accessible – Interoperable - Reusable

Data Market Architecture components

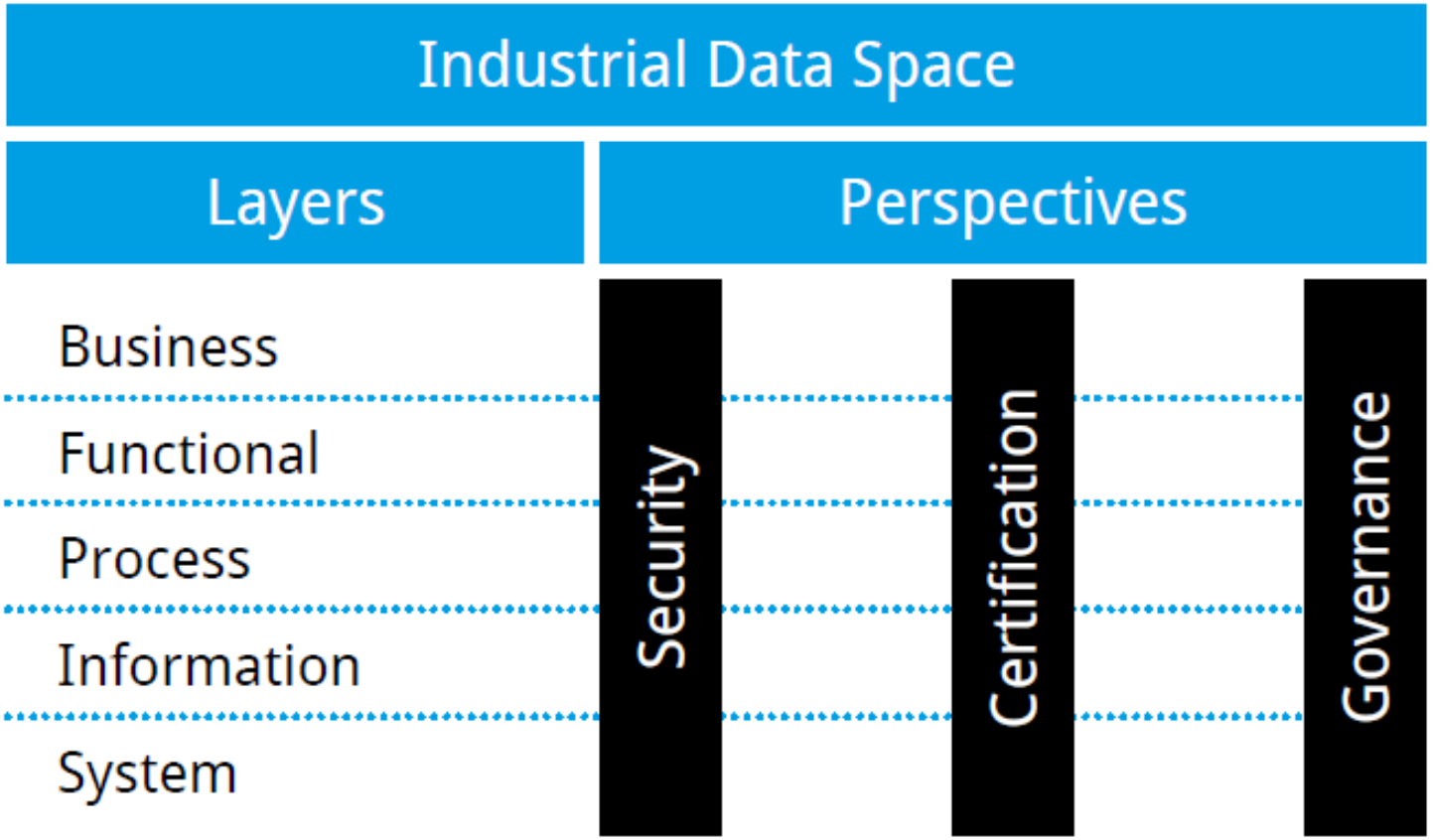
- **Data Source** (producer/seller/publisher) – Supply side
- **Data Target** (consumer, buyer, subscriber) – Demand side
- **Data Broker**
  - Data/Value Broker
  - Trust Broker or Trusted Introducer
- **Directory or Catalog service**
  - Including data (quality) ranking
  - Including API link or repository
- **Data Exchange**
  - Infrastructure component vs peer-to-peer customer network
  - Provenance and transactions control issue
- **Data Storage**/Cache Data Delivery Network
  - Data Lake (HDFS based and SQL/NoSQL)
  - Caching for traffic optimisation in DDN
- **Open/Public Data** access and storage
  - Can be stored to offer as part of DM storage
  - Facilitate quality of data analytics
- Data Transfer vs Data Access
  - Scenario: App container vs Data container
  - Container security using Intel TXT or SGX technology
- (Optional) Secure data processing
  - Used for data quality inspection (e.g. P.I.D., or IP), auditing, composition
  - Data preparation/conditioning

Re-using experience of Internet eXchange, Cloud eXchange and Financial Exchanges

International Data Space Association

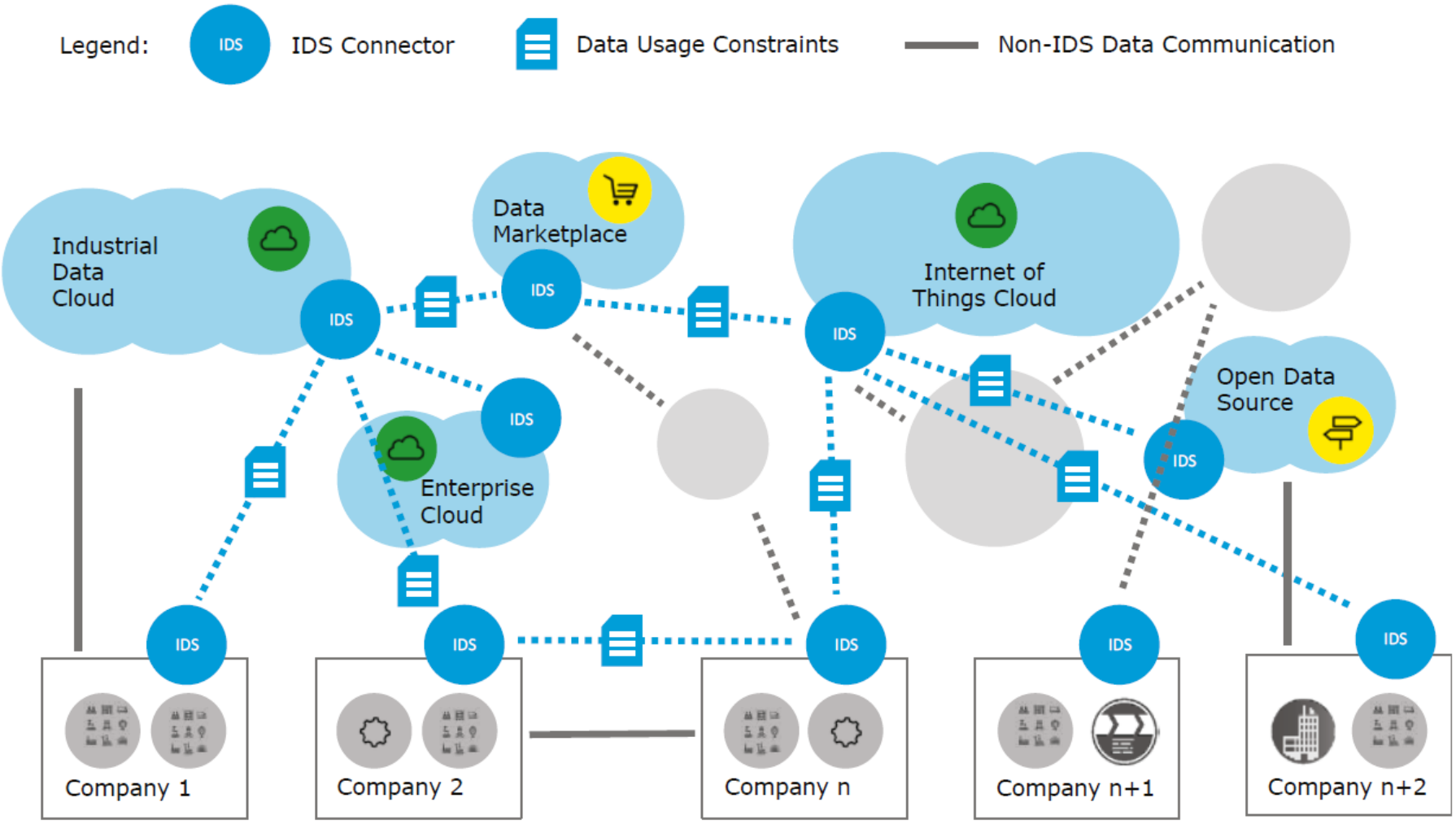
- Started 2016 as Industrial Data Space initiative (supported by German project)
- Re-defined as International Data Space Association (IDSA)
  - Published International Data Space Architecture Version 2.0 (2018)
  - Whitepaper and use cases
- Associated H2020 projects
  - Boost4.0 – Big Data for Factories (20 Mln (100 Mln private), 3yrs, 50 partners, 16 countries)
  - MIDIH – Manufacturing Industry Digital Innovation Hub (22 partners, 12 countries)
    - Services: technological, business, skills building
    - Open calls
  - Close cooperation with FIWARE Foundation (cloud like infrastructure resulted from Future Internet program)
    - Positions itself against IoT and Open-Data solutions in the areas of smart cities, Industry 4.0 and agriculture
- Ongoing active outreach developments
  - Data Sovereignty and Secure Data Exchange model
  - Data Markets
  - Trusted platforms

General Structure of IDS Architecture



- Specification defines functionalities by layers
- Details are sufficient to define processes, functional components and API

Cloud based IDS infrastructure for Data Exchange and Trading



- IDS Connector is the main functional component
- No specifically defined common infrastructure services