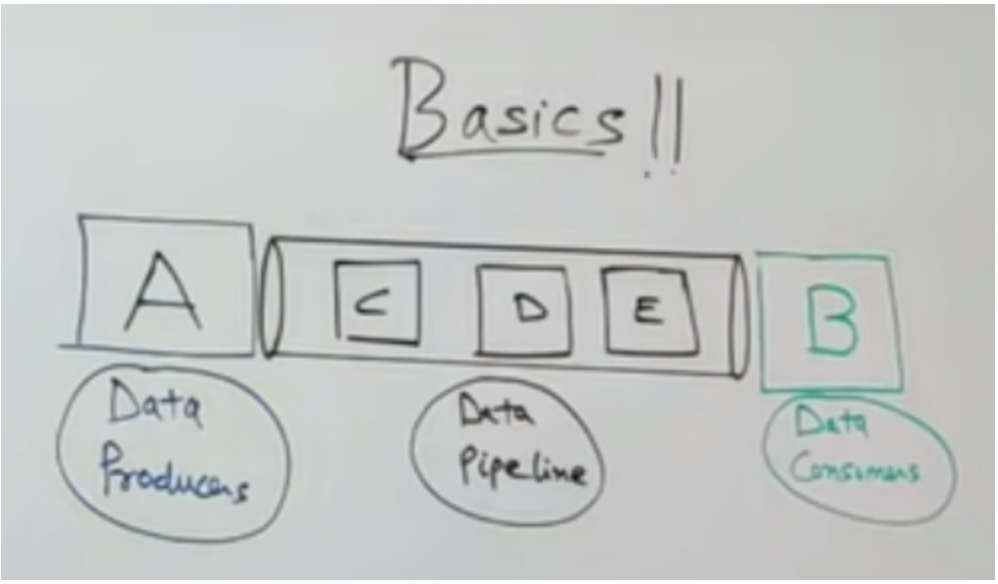
Data Engineering

Big data refers to data that is so large, fast and complex that it’s difficult or impossible to process using traditional tools like a data warehouse.

Data Pipeline



Data Cleansing

Data Governance

Data Enrichment

Data Processing

Data Pipeline vs ETL

ETL is a type of data pipeline

Example: Processing POS (point of sale) data from a retailer daily

ETL =

- Take batch data at midnight
- Store in database

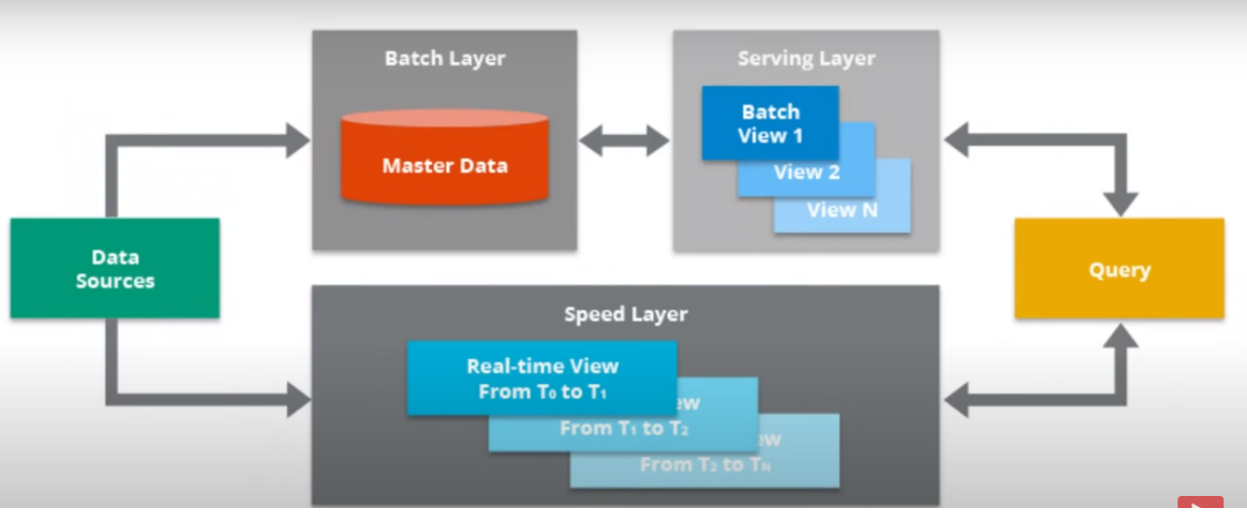
- Run reports

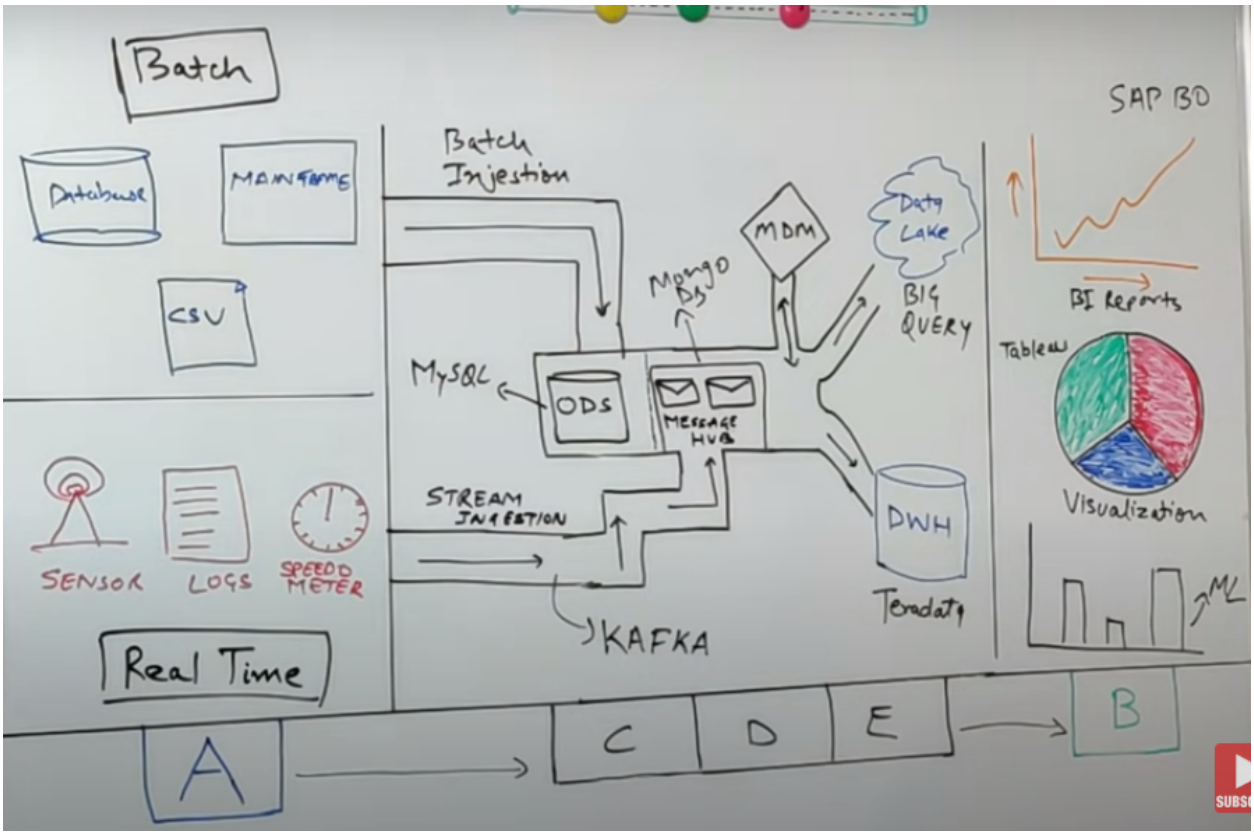
Data Pipeline =

- Real-Time Access
- Search Product inventory
- How frequently is stock replenished
    - Stock Count Once An Hour
- How Capable are you at meeting the demand of the customers

Two Types of Pipeline

- Real-Time Streaming
- Batch Data Streaming
- Lambda Architecture (both)





Big Data

We might have heard the term Big Data before, but what does it mean? Below is a high-level definition:

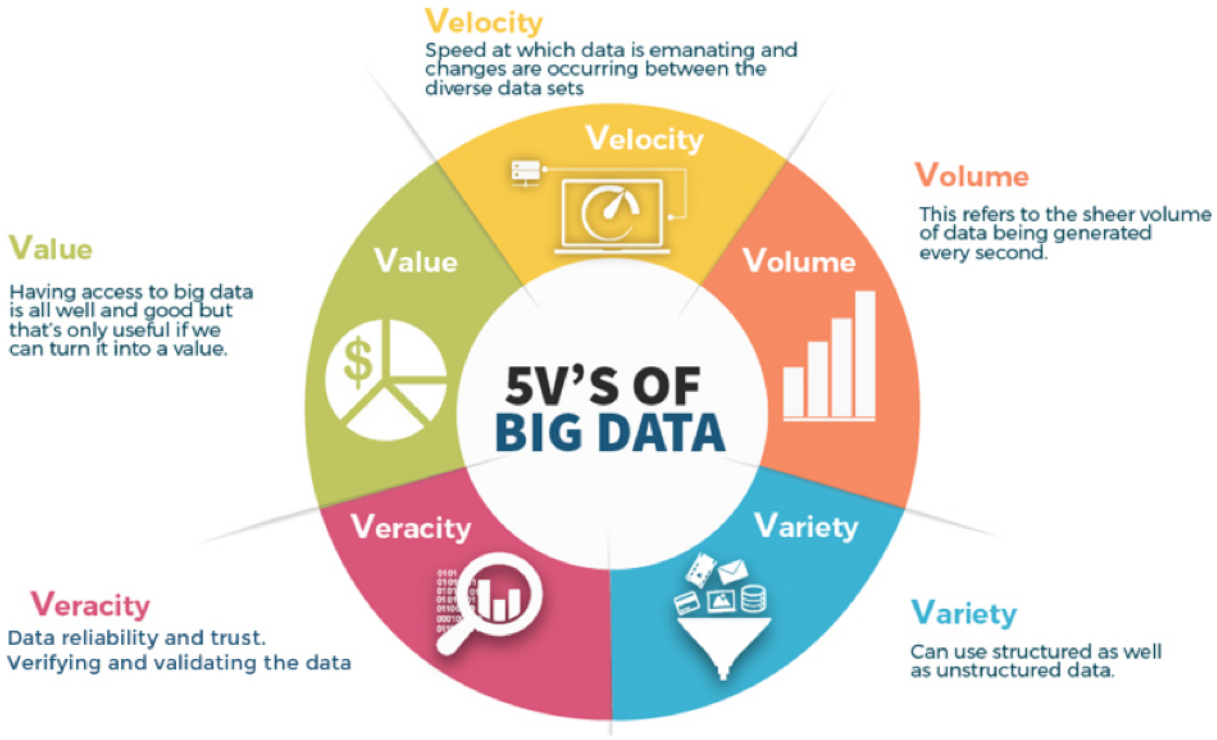
> Big data refers to data that is so large, fast and complex that it's difficult or impossible to process using traditional tools like a data warehouse.

Traditional methods of data storage included storing data locally on a computer, on a server, mainframe system or in a database. Over time, due to the rapid evolution in data production, these traditional approaches are no longer able to cope with the new reality.

Accordingly, a new group of tools and technologies were introduced starting mid to late 2000s. The logic behind these new tools was to switch the mentality from storing and analyzing small quantities of high-value data in a structured format using expensive

systems such as a data warehouse (DWH) to being able to capture and store **all** raw data (both structured and unstructured) in a central repository, and then running various operations and transformations on this comprehensive dataset using tools that were open-source and inexpensive.

Hence, these new big data tools were developed to address the below challenges called the "V's" of big data:



- **Volume**:
  - The quantity of data is growing, and so is the number of data production sources (such as from the Internet of Things devices and machines)
- **Variety**:
  - The format of produced data is also evolving. Now we have:
    - Structured: Table-like data organized into rows and columns. This used to be the standard format for data storage for decades.
    - Unstructured: More modern data types including text documents, emails, videos, images, and audio.

- **Velocity**:
  - The speed at which the data is arriving (batch initially and real-time more recently)
- **Veracity**:
  - The quality of the data (inconsistencies, uncertainties, empty data records etc.)
- **Value**:
  - How valuable is the captured data? How does it increase its business value?

The first true Big data technology was the introduction of Apache Hadoop in 2006. This was a revolutionary step, as it finally enabled the capture, storage and analysis of structured and unstructured data in one centralized location called the data lake. Hadoop also provided the capabilities to manipulate, transform and explore the data using tools such as - Java MapReduce - HiveQL (which is a SQL-like language) - Pig Latin (which is a scripting language similar to Bash)

Data Ecosystem

- The modern data infrastructure consists of various concepts, tools and components to be able to handle the various types of data being generated.
- These tools and components are arranged in groupings, with each layer performing a specific activity on the data before passing it on to the next layer
- The 6 main layers in the data ecosystem include:

▼ Data Storage

> Data storage, especially within the context of big data, is a compute-and-storage architecture that can be used to collect and manage huge-scale datasets and perform real-time data analysis.

> In this layer, data generated from sources is captured and persisted. Several alternatives are available for this approach,

> each tailored for specific types of data and related use cases. The most common types of enterprise data storage components include:

▼ Data Lakes

> A data lake is a centralised repository that allows you to store all your structured and unstructured data in one location. This type of storage approach enables easy scaling to cope with the increasing quantity of data.

▼ HDFS

> HDFS is a distributed file system that can store large data sets. It's designed to be installed on cheap commodity hardware. At the time of its release, HDFS offered a much cheaper alternative over expensive databases and data warehouses to store data. HDFS is one of the major components of Apache Hadoop, the others being MapReduce and YARN. It also provides the capability to store both structured and unstructured data types.

▼ Cloud Storage

> Cloud storage is a model of computer data storage in which the digital data is stored in logical pools, said to be on "the cloud".

> Cloud storage is a cloud computing model that stores data on the Internet through a distributed computing provider that manages and operates data storage as a service. It’s

> delivered on demand with just-in-time capacity and costs and eliminates buying and managing data storage infrastructure. This gives users agility, global scale and durability, with “anytime, anywhere” data access.

> Under the hood, cloud-based systems store their data on a very large number of very powerful server machines located in a single location called a *data centre*. Such data centres are typically owned and managed by a hosting company (such as Amazon). This approach to data storage is quickly becoming the dominant trend in the industry and is replacing the traditional approach of companies having to purchase and maintain expensive servers and storage devices.

> The most popular types of cloud storage are:

- Amazon S3
- Microsoft Azure and OneDrive
- Google Firebase

▼ HBASE

> HBase is an open-source, non-relational (NoSQL), distributed columnar data store modelled after Google's Bigtable. It's designed to handle big data in real time.

> Hbase has many connectors to integrate it with the various other tools in the big data ecosystem and is a widely used data store in the industry.

▼ Casandra

> Apache Cassandra is another popular free, open-source, distributed, wide-column data store

> It's designed to handle big data easily by using a distributed network of servers. Cassandra provides high-availability, and its architecture has no single point of failure.

▼ MongoDB

> MongoDB is another popular free, distributed, non-relational data store used to handle document-oriented data

> MongoDB is mainly designed to store JSON-like documents efficiently. The tool provides a flexible schema model, so this unstructured data can easily be stored and analysed in real-time.

▼ Data Warehouses

> Data warehouse (DWH) systems were traditionally the de-facto standard to store and analyse structured corporate data. Structured data is that which can be stored into tables using rows and columns (such as data stored in an Excel sheet).

> A data warehouse (DWH), also known as an enterprise data warehouse (EDW), is a system used for reporting and data analysis and is considered a core component of the

> traditional business intelligence architecture. DWHs are central repositories of integrated and structured data from one or more disparate sources. They store current and historical data in one single place that are used for creating analytical reports. These types of systems are expensive to maintain and usually store high-value data.

These advanced tools rely on SQL logic to store, process and analyse data. There are several brands with popular data warehouse systems, which include:

- Oracle Data warehouse
- Microsoft Data warehouse
- Amazon Redshift
- SAP
- IBM Db2 Warehouse
- Snowflake (the latest tool)

▼ Data Acquisition

> This layer involves tools and applications to ingest data from different sources and move the data throughout the ecosystem

> Data acquisition is the process of collecting the data from various sources and moving it from point of origin to the target destination.

> Depending on the type of data being produced, and the frequency of its production, we can classify this layer into two main types:

▼ **Batch Data:**

> This is the acquisition of a vast amount of data at rest, in its entirety and at the same time. This approach is usually implemented at regular intervals (such as once per day or once a week) on a full dataset, which includes the historical data plus any new incremental data. An example of batch data is moving a dataset of the entire list of customers for a certain company.

▼ **Real-time Data:**

> This is the acquisition of continuous data while it's in motion in very short, near-instantaneous intervals as the data arrives from its source. An example of real-time data is mobile application data that is constantly turned on, such as GPS locations.

▼ **Kafka**

> Apache Kafka is a free, novel and open-source platform designed to handle big data streaming. It can connect to multiple data produces and consumers, and is able to handle up to trillions of data events daily.

> Streaming data is that which is continuously created by one or more data sources. An example of this is Internet of Thing (IoT) devices such as sensors and smartphones. In a typical enterprise, we'll have hundreds or thousands of such devices, and they all send data records simultaneously. In order to properly capture and process

> this constant flow of data, a streaming platform needs to be properly configured to handle the data sequentially and incrementally.

Kafka is one such platform that provides the following functions:

- Publish and subscribe one or more data streams
- Store records in the same order in which they were created
- Process streams of records in real-time as the data is being ingested and transported

▼ Flume

> Apache Flume is a tool for collecting, aggregating, and transporting big data streams efficiently. It uses a distributed model which provides reliability and robustness.

> Flume is easily customisable and can be configured to ingest data from one or more data sources and feed it to various types of data consumers. Although it was very popular a few years ago, its popularity is currently in decline as its being replaced by Apache Kafka.

▼ Data Processing

> This layer involves the tools that perform various operations and transformations on the data itself

> Data processing is, generally, the automated manipulation of raw data and transforming it to produce meaningful information. This is the layer that handles the ETL/ELT operations on the data.

> Data processing is handled by specialized tools, with the vast majority of these tools being open-source and oftentimes free to download and use.

> The most popular industry-ready frameworks are Hadoop (including Yarn), Spark, Flink and Storm. The choice of which framework to use, and even which component within that tool to deploy (as they all come with various modules) is determined by an enterprise system Architect. There are several criteria that support determining the tool of choice. Some of these criteria include:

- Is the incoming data batch or streaming?
- What is the type of data being captured and stored (structured, unstructured?)
- Where will the data be stored?
- What transformations are required on the data?
- What is the quality of the data?
- What are we expected to do with the data after cleaning and transforming?

> Once the data passes through the required processing steps, it's ready to be accessed via the required stakeholders as it's now in a cleaned and integrated state.

▼ Apache Hadoop

> Hadoop is an open-source framework that allows for the distributed processing of large data sets across clusters of computers using customisable programming models.

> It is designed to be able to scale-up from a single node/server to any number of machines, with each machine offering local computation and storage in addition to dividing larger jobs into smaller tasks that can run in parallel across the various nodes.

> Hadoop was designed to tackle big data, and especially unstructured data types which traditional relational database systems couldn't handle. It mainly focuses on batch data processing using an on-disk approach. In the later versions of Hadoop, support was provided to integrate newer components that could handle real-time data (such as Apache Storm).

▼ **Apache Spark**

> Apache Spark is a powerful, unified data processing engine which is currently the most widely used tools by top enterprises worldwide such as Netflix, Yahoo and eBay. Due to its flexibility, it can process both batch and real-time data efficiently.

> Spark was originally developed by the University of California, Berkley in 2009. It has the capability to perform data processing activities on massive amounts of structured and unstructured data by leveraging its in-memory, distributed computational model.

> Spark performs data computations in-memory (as opposed to Hadoop's on-disk approach), and thus can be up to

> 100X faster than Hadoop for certain types of data processing activities.

> Spark was also designed to handle real-time data, something which Hadoop wasn't able to handle at the time.

▼ Apache Storm

> Apache Storm is another free and open-source distributed real-time data processing platform

> Apache Storm was one of the first real-time data processing engines introduced that could integrate with Apache Hadoop. Storm is not as complicated as Hadoop to set up, and can be used with several programming languages.

> Typical use cases for Storm are those that require constant real-time data processing, such as:

- Real-time data analytics
- Fraud detection
- Continuous machine learning
- Extract, Transform and Load

> Apache Storm is also very fast. One study recorded the processing power to be 1 million tuples processed every second per node.

▼ Apache Flink

> Apache Flink is another popular open-source, distributed data processing framework that can process large data streams at-scale in real-time

> Flink can operate in several types of cluster environments and performs data computations in-memory (similar to Apache Spark). It can handle bounded or unbounded data streams, and is designed to be robust and fault-tolerant.

▼ Data Access

> This layer involves components that are used to explore, query and analyse data

> Data access refers to the capability to interact with the data once it's been through the data processing steps.

> In the data access layer, the aim is to expose the cleaned, integrated and prepared data to the downstream systems and various stakeholders. This can be done via several tools, depending on our objective. For instance, if we are querying the data for analytics purposes, then we'll use tools such as HiveQL, Spark SQL or Python. If the goal is to create predictive models, we can use a tool like R, Pandas, and Numpy to create and run these algorithms.

▼ Hive

> Apache Hive is a data warehouse software project built on top of Apache Hadoop for providing data queries and analysis.

> Hive gives an SQL-like interface to query data stored in various databases and file systems that integrate with Hadoop.

▼ Knime

> KNIME, the Konstanz Information Miner, is a free and open-source data analytics, reporting and integration platform.

> KNIME integrates various components for machine learning and data mining through its modular, GUI-based data pipelining "Building Blocks of Analytics" concept.

▼ Snowflake

> Snowflake is a recently introduced big data warehouse built on top of cloud infrastructure (such as AWS or Microsoft Azure)

> The Snowflake architecture allows storage and computing to scale independently, so users can pay for storage and computation separately. This technology is becoming quite popular in industry due to the benefits it provides.

▼ Data Management

> This layer includes tools that organise and manage the jobs and workflows which execute code and applications based on certain conditions

> Data management is the process of managing and synchronizing data based on requirements.

> In the data management layer, the objective is to determine what to do next with the data after access is provided. For instance, it may be required that certain jobs need to run hourly or daily. In this case, we'll use tools like Oozie or Airflow to create and schedule such jobs and to provide the required parameters.

> If the requirement is to move or copy the data into another system, we can use Kafka to transport the data to the target system. If the requirement is to store the data in an enterprise data warehouse (EDW), we can use a tool like Sqoop to place the data there directly.

> Finally, if the need is to create various dashboards and graphs, we can feed the data to the next (and final) layer, which is the data visualisation layer.

▼ Apache Airflow

> Apache Airflow is a workflow engine initially created by Airbnb. It easily automates, schedules, and runs complex data pipelines. It will make sure that each task of the data pipeline will get executed in the correct order and each task gets the required resources.

- Airflow also provides a user interface that can monitor each task's status and allows users to deal with any errors or bugs
- It's easy to use for data engineers who have expertise using Python

- It's free and open-source with an active community
- The tool provides many ready-to-use connectors so that you can work with Google Cloud Platform, Amazon AWS, Microsoft Azure, etc.

▼ Apache Oozie

> Apache Oozie is a system for workflow scheduling designed to manage Hadoop jobs. The tool is normally hosted on servers that run numerous big data workflows on a regular basis.

> Workflows in Oozie are defined as a collection of control flow and action nodes in a directed acyclic graph. Control flow nodes define the beginning and the end of a workflow as well as a mechanism to control the workflow execution path.

▼ Talend

> Talend is an open source data integration platform. It provides various software and services for data integration, data management, enterprise application integration, data quality, cloud storage and big data.

▼ Data Visualisation

> This layer is responsible for visually representing data that's been prepared for reporting to business stakeholders

> Data visualisation is the process of displaying data in charts, graphs, maps, and other visual forms. It is used to help people easily understand and interpret their data at a glance,

> and to clearly show trends and patterns that arise from this data.

> After the raw data has been through the previous 5 steps, it's now ready to be presented to business leaders and executives. It's quite common for non-technical professionals to refer to dashboards and graphs to analyze how the business is performing, track metrics and key performance indicators (KPIs) and monitor the day-to-day operations of the company. Actually, in reality, most data-related projects are initiated to at least partially support dashboards and visualisation tools for the top-level business leaders.

> In this layer, a tool like Tableau can be leveraged to represent the information we have in an easy-to-understand graphical format. Some common types of charts include:

- Pie charts
- Bar charts
- Time series charts
- Graphs

▼ Tableau

> Tableau is a powerful and increasingly popular data visualisation tool used by top-tier companies for business intelligence and real-time report creation

> It helps users to unlock insights in raw data by visualising it in an easy to interpret format. Tableau uses an intuitive

> drag-and-drop interface to create advanced dashboards. This allows non-technical users to create and use customised dashboards.

> Data analysis with Tableau is fast, and the tool provides a wide variety of customisable dashboards and charts which can connect to a range of data storage tools in the back-end.

▼ Microstrategy

> Microstrategy is a business intelligence software, which offers a wide range of data analytics capabilities

As a suite of applications, it offers:

- Data discovery
- Advanced Analytics
- Data visualisations
- Embedded BI
- Banded Reports and Statements.

> Similar to Tableau, Microstrategy can connect to big data storage tools like Hive, data warehouses, relational systems, flat files, web services and a host of other types of sources to provide data for visual charts and dashboards. Although Microstrategy offers powerful features, it has a steeper learning curve over Tableau.

▼ DataWrapper

> Datawrapper is a free, online data visualisation tool that can create charts, maps and tables via a user-friendly graphical user interface.

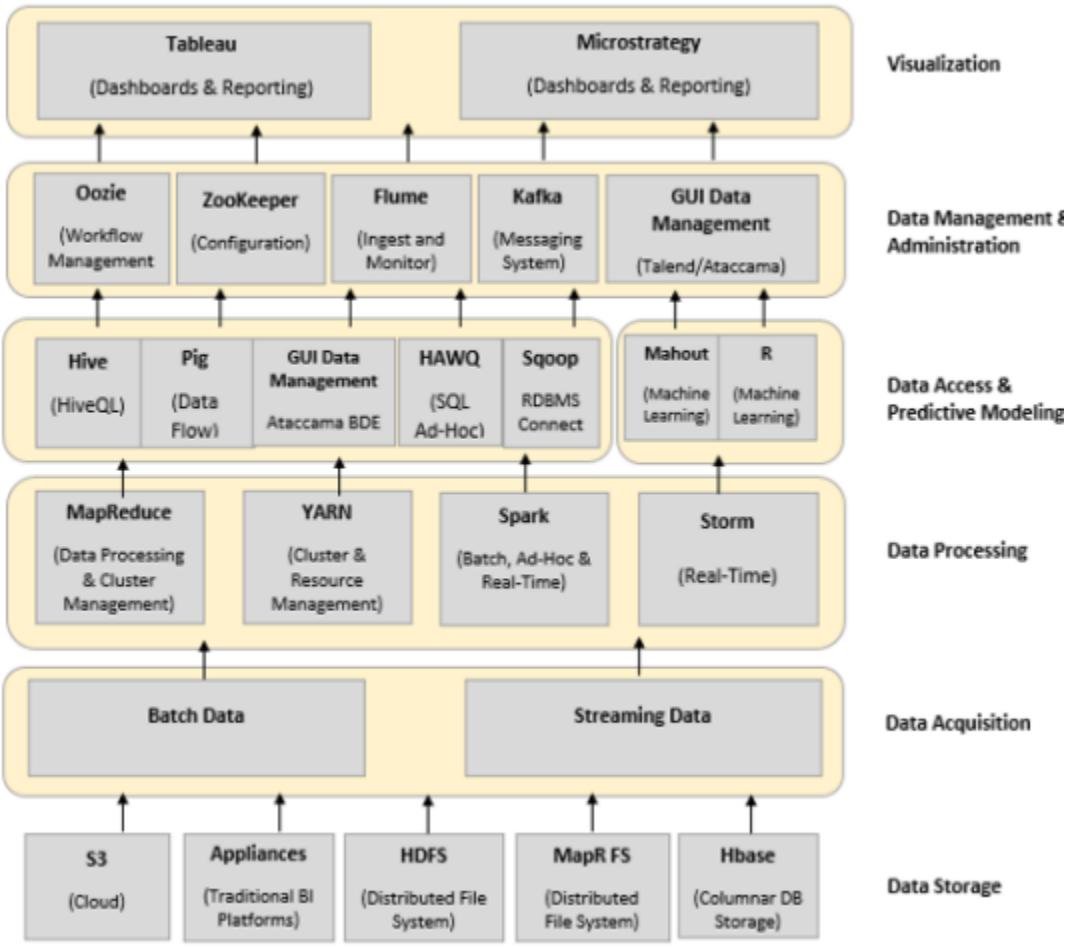
> Datawrapper provides the ability to create three kinds of visualisations: maps, charts and tables.

▼ **Lumify**

> Lumify is a big data integration, analysis, and visualisation platform.

> Lumify is another tool that allows users to explore connections and discover relationships in their data via a wide range of data visualisation options.

Here is an example diagram showing how these layers and their corresponding tools fit in the big picture:



Talend is one interface for all data analysis in either batch/real-time/streaming modes

Containerization

There are a host of other types of tools that don't explicitly fall under the above 6 layers we discussed throughout this notebook. These tools generally either perform functions across more than one layer or provide certain features or benefits to the entire ecosystem and all its layers. One such type of technology is called *containerisation*.

Containerisation can be thought of as the evolution of virtual machines (VMs). Containerisation is defined as a form of operating system virtualisation, through which applications are run in isolated user spaces called containers, all using the same shared operating system (OS). Containers are more lightweight than virtual machines, and typically have their own file system, applications, and share of resources (memory, CPU

et.). This type of technology is rapidly being adopted by industry due to the benefits they provide. Currently, the most popular containerisation tools in the market are:

▼ **Kubernetes**

> Kubernetes is an automated, portable, and open-source system for container orchestration. It's widely used for software deployment, scaling, and management.

> It was originally designed by Google. The tool can be used to automatically and reliably run a wide variety of tools and software in parallel. Although it requires some setting up beforehand, containers can easily be deployed any number of times across different computer nodes.

▼ **Docker**

> Docker is another containerisation technology that leverages virtualisation to provide pre-packed software that can be easily deployed and used on servers and on the cloud

> Each container is isolated from other containers, and each comes bundled with the required tools, libraries and configuration files needed to operate independently.

Data Fabric

> A data fabric is a novel data management architecture that serves as an integrated layer (fabric) of data and connecting processes. It can optimise access to distributed data and

> intelligently curate and orchestrate it for self-service delivery to data consumers.

> Data fabric is a very modern approach to managing the data within large organizations. With a data fabric, you can elevate the value of enterprise data by providing users access to the right data just in time, regardless of where it is stored. A data fabric architecture is agnostic to data environments, data processes, data use and geography, while integrating core data management capabilities. It automates data discovery, data governance and consumption, delivering business-ready data for analytics and AI.

Top-performing enterprises are data-driven. However, several challenges block them from fully exploiting all data:

- Lack of data access.
- Numerous data sources and data types.
- Data integration complexities.

> Research shows that up to 74% of data is not analyzed in most organizations and up to 82% of enterprises are inhibited by data silos.

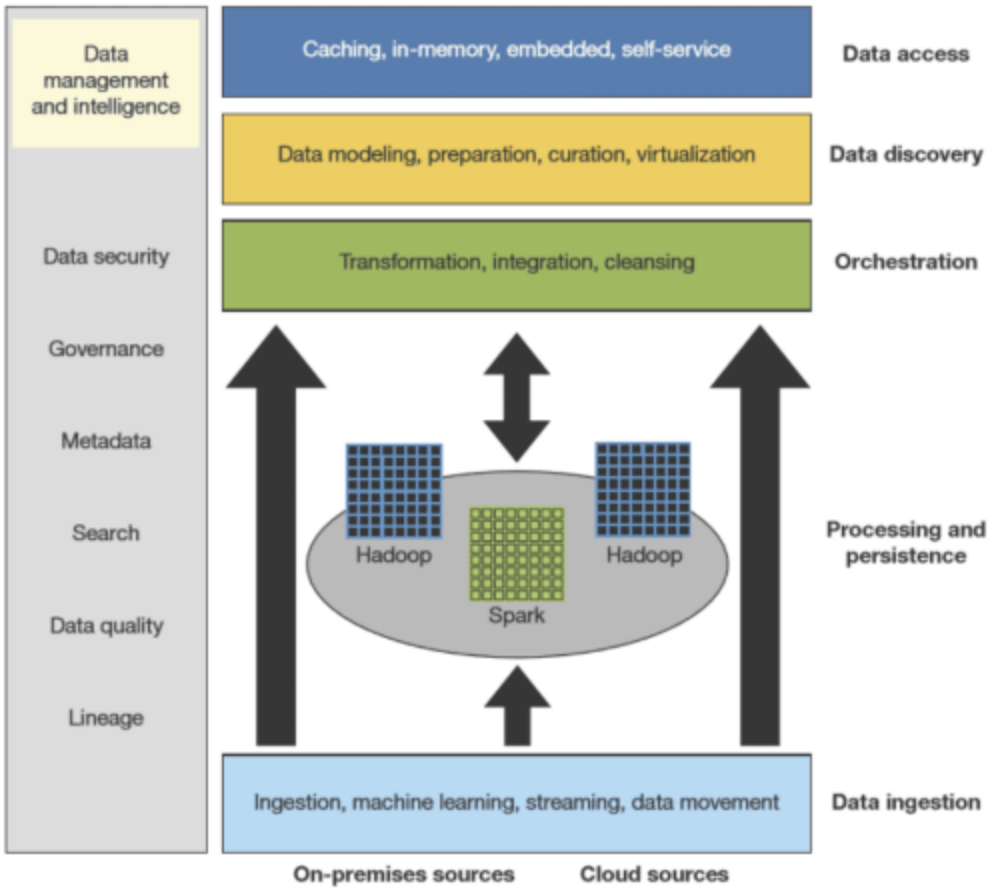
> With a data fabric, business users and data scientists can access trusted data faster for their applications, analytics, AI and machine learning models, and business process automation, helping to improve decision-making and drive digital transformation. Technical teams can use a data fabric to radically simplify data management and governance in complex hybrid and multi-cloud data landscapes while significantly reducing costs and risk.

> Data fabric enables a permanent and scalable mechanism for businesses to consolidate all its data under the umbrella of one unified platform. It leverages storage and processing power from multiple heterogeneous nodes to enable enterprise-wide access to all data assets of an enterprise. According to Forrester, a big data fabric assists enterprises to "…quickly ingest, transform, curate, and prepare streaming and batch data to support a real-time trusted view of the customer and the business."

Furthermore, big data fabric enables companies to:

- Effectively consolidate data assets with on-premises and Cloud data sources, for a complete view of enterprise-wide information.
- Gain access to the latest data in real-time.
- Easily onboard new big data systems and retire legacy systems, while keeping business systems running continuously without disruption.
- From a problem-solving perspective, data fabric overcomes the challenges of insufficient data availability, the unreliability of data storage and security, siloed data, poor scalability, and reliance on underperforming legacy systems.

Below is what a typical data fabric ecosystem would look like in a global company:



- Big data can be defined using the 5 V's: volume, velocity, variety, veracity and value
- Big data is different from traditional data, as it comes in both structured and unstructured formats and the data arrives in various velocities in accelerating quantities
- Traditional relational database and data warehouse models were well suited for structured data with low volumes, however, such systems are expensive to maintain, and can't handle unstructured data. This is why they are currently being augmented with more modern big data tools.
- The modern data ecosystem is composed of 6 layers, namely: storage, acquisition, processing, modelling, management and administration.
- Data must flow sequentially from the first layer (data storage) to the last layer (visualisation) and each layer performs a specific function on the data.
- Each of the 6 layers of the modern data ecosystem has specific tools that have become the industry standard in the data storage layer, cloud data storage is

quickly becoming the dominant trend in the industry

Data fabric is not a single technology, it is a group of tools in a pipeline that make the most of big data in a data estate. It makes use of the AI Ladder:

▼ AI Ladder

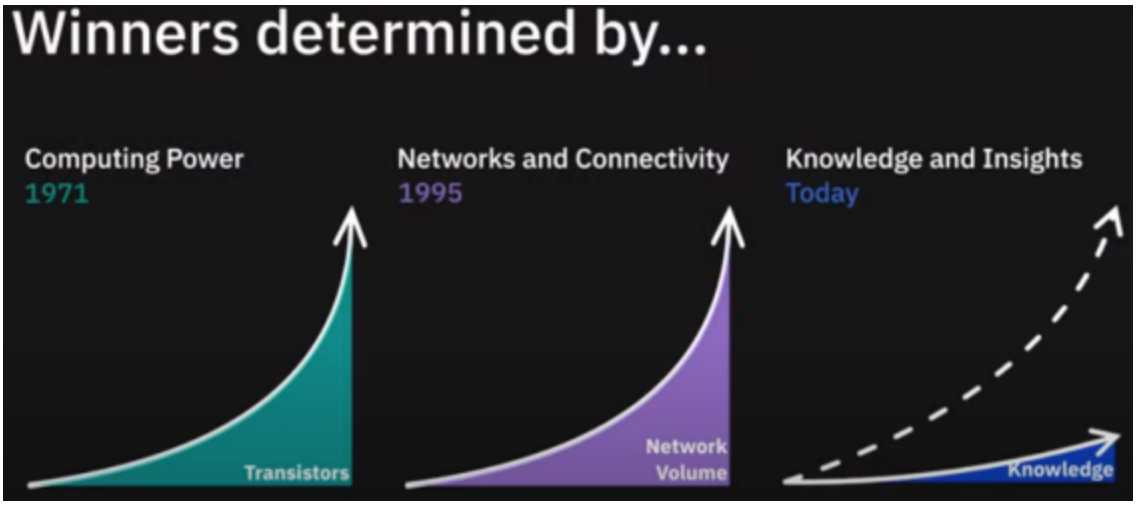
Collect

Organise

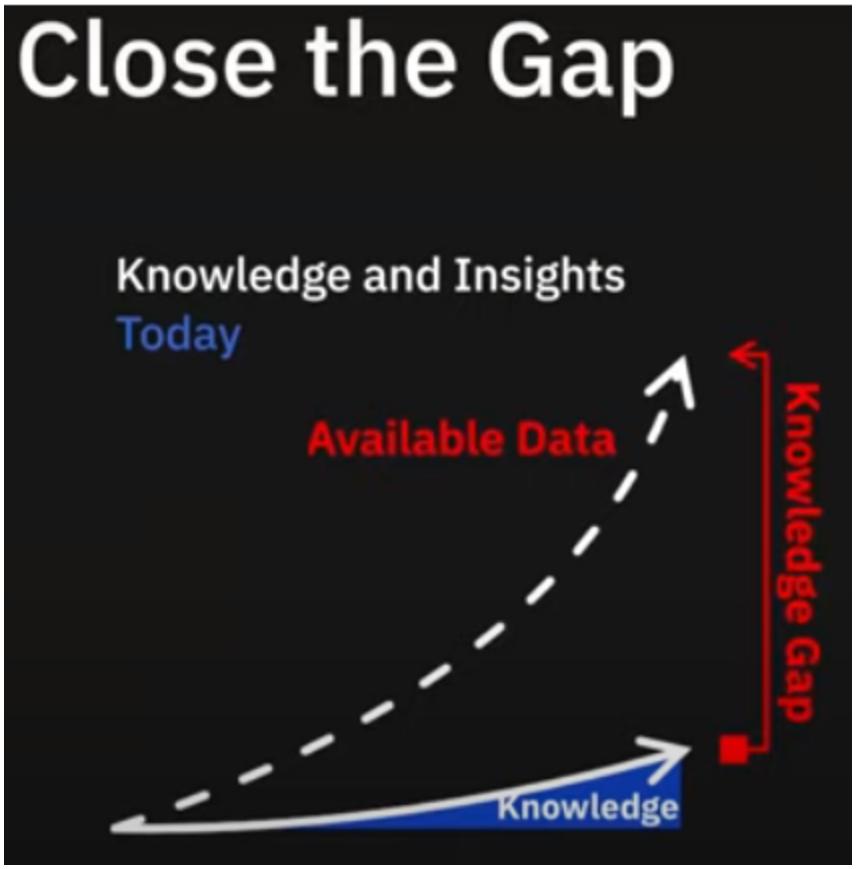
Analyse

Infuse

> Understanding data fabric is important because we are entering a new era of knowledge



> The biggest challenge is to close the gap between the available data and how much of it turns into knowledge/insights


Close the Gap
Knowledge and Insights
Today
Available Data
Knowledge Gap
Knowledge

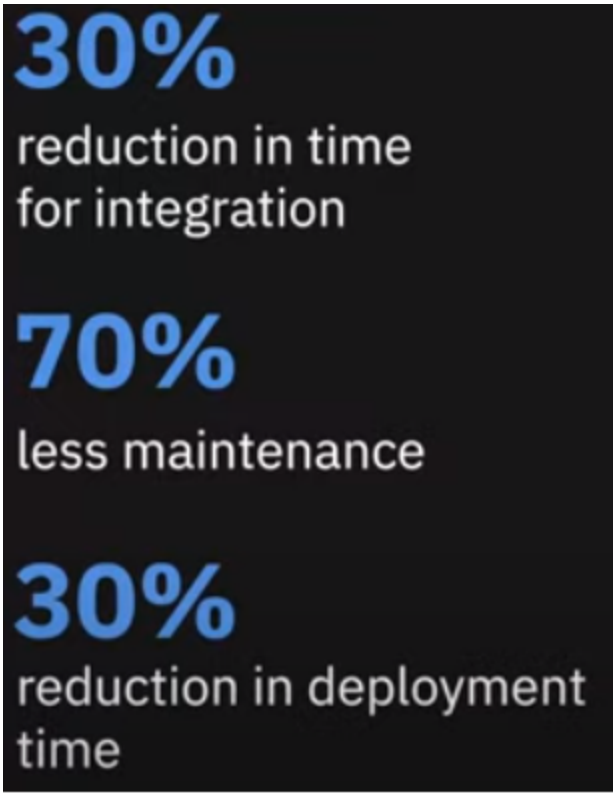

60-73%
of enterprise data goes unused
30%
of IT dev time is spent just trying to make interfaces work together

3 key areas

Universal Data Access: Allows access to all Sources | Types | Domains, especially in a multi-cloud system

Creates Efficiencies: Thinks holistically throughout the AI ladder to be able to process data faster

Enforceable Policies: The right people have access to knowledge at the right time



Talend Data Fabric

Now information comes from various sources (premise, cloud, sensors, internet of things, customers/partners) it has become more complicated

Talend connects all of these systems together to:

- Stop Fraud
- Increase Retention
- Personalise Offers
- Predict Insurance Risks

- Optimise Crop Yields
- Maximise Cross-Sell

Allowing you to:

- Big Data Integration
- Data Integration
- Cloud Integration
- Application Integration
- Master Data Management

Talend is one interface for all data analysis in either batch/real-time/streaming modes

---

Example Use Case

Employee Wages

- Data Storage = S3 = Stored on a cloud
- Data Acquisition = Batch Data = Shifts worked updates once a day at midnight
- Data Processing = Spark
- Data Access = R = By HR
- Data Management = Kafka
- Data Visualisation = Exel

Issues: Duplicate Data | Timeliness of source data | Lack of developer knowledge by the end user

Point Of Sale

- Data Storage = S3 = Every hour batch data is updated and stored on the cloud
- Data Acquisition = Batch Data = Sales and replenishment
- Data Processing = Spark = Profits, Losses and Next Delivery
- Data Access = R = Product Managers
- Data Management = Kafka

- Data Visualisation = Tableau

Issues: Duplicate data | Users in store not updating on time | Human error

Spark vs Hadoop

	Spark	**Hadoop**
	Frameworks are written in scala that organises information in clusters. Core functionality and extensions	Big Data framework that stores/processes data in clusters
	Faster than Hadoop.	Based on Mapreduce a model that can process multiple data nodes simultaneously
	Integration - works with many extensions and databases	Fault Tolerance - replicates each node automatically so if one cluster is down the rest of the structure is fine
	Supports other Apache clusters	Namenodes - always runs active nodes that step in for technical errors
	written with high-level operators = reduced lines of code	Built in modules - YARN (framework for cluster management) Hadoop Ozone (saving objects)
Architecture	Computations are conducted in memory. Results go to HFDS and go to the resilient distributed data set. Processed in parallel = better performance speed	Files are distributed into Hadoop file system, split into blocks and replicated (in case of failure). Mapreduce algorithm to track resources and allocate queries. Results are sent back to HFDS to make new data blocks to optimize
Performance	Better for smaller and faster apps	Better for apps relying on ability and reliability
Languages	More versatile	Java-based
Cost	Better quality/price	Open Source with added expenses
Security	Off by default	
Compatibility	Python / R / Java / Scala	Python / R / Java / Scala Supports more languages
Usability	More user-friendly - uses multiple APIs and has many extensions	Limited add-ons - APIs are less intuitive

Uses	Good for handling big data in real-time / ML / personalization / real-time marketing	Good for large networks of enterprises / scientific computation / predictive platforms