TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

Macroarea di Ingegneria
Dipartimento di Ingegneria Civile e Ingegneria Informatica

Introduction to Big Data

Corso di Sistemi e Architetture per Big Data

A.A. 2025/26

Valeria Cardellini

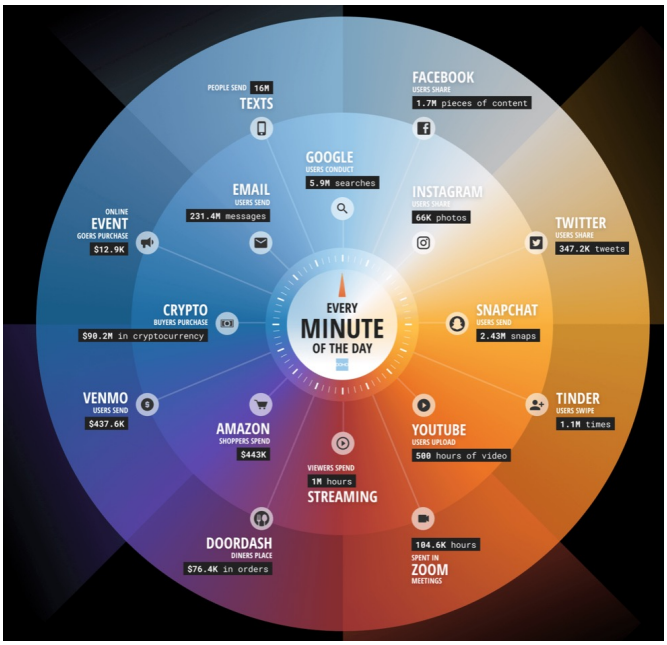
Laurea Magistrale in Ingegneria Informatica

Why Big Data?

How much data is created every single minute of the day?

Global Internet population in Jan. 2025: 6.04 billion (70% of world population)

1 billion in 2005



Source: https://www.domo.com/data-never-sleeps

How much data?

- Big data volume: from Terabytes to Zettabytes
  - How big is a Zettabyte?
  - 1 ZB = 2^{70} B = 2^{40} GB ≈ 10^{21} B
    - Recall that 2^{10} = 1024 ≈ 10^{3}
- Zettabytes of data generated by 2025

  181 Zettabytes (181x2^{70} ≈ 181x10^{21}) …

  ≈ 181,000 Exabytes (181,000x10^{18}) ...

  ≈ 181,000,000 Petabytes (181,000,000x10^{15}) ...

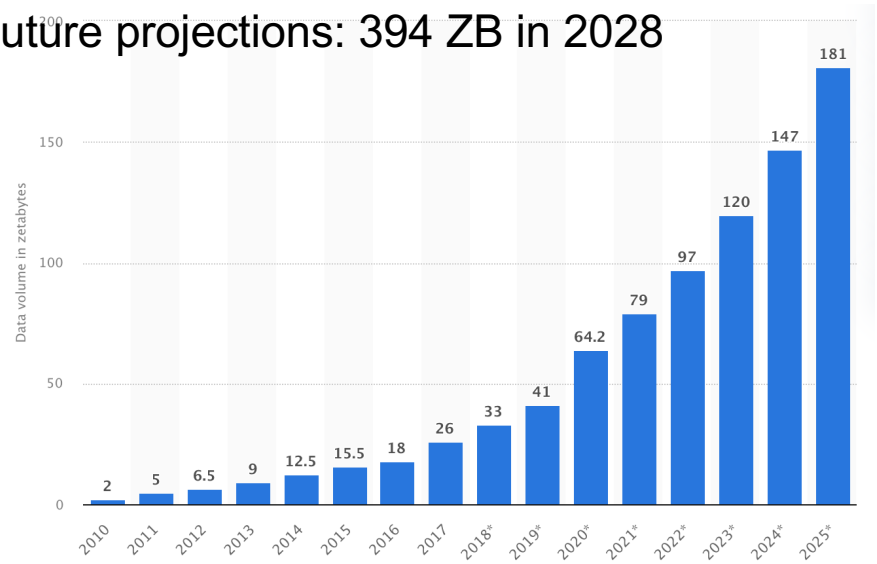
  ≈ 181,000,000,000 Terabytes (181,000,000,000x10^{12}) …

  ≈ 181,000,000,000,000 Gigabytes (181,000,000,000,000x10^{9}) …

  ≈ 181,000,000,000,000,000,000,000 bytes!
- Bigger than Zettabytes? Yottabytes!
  - 1 YB = 2^{80} B ≈ 10^{24} B

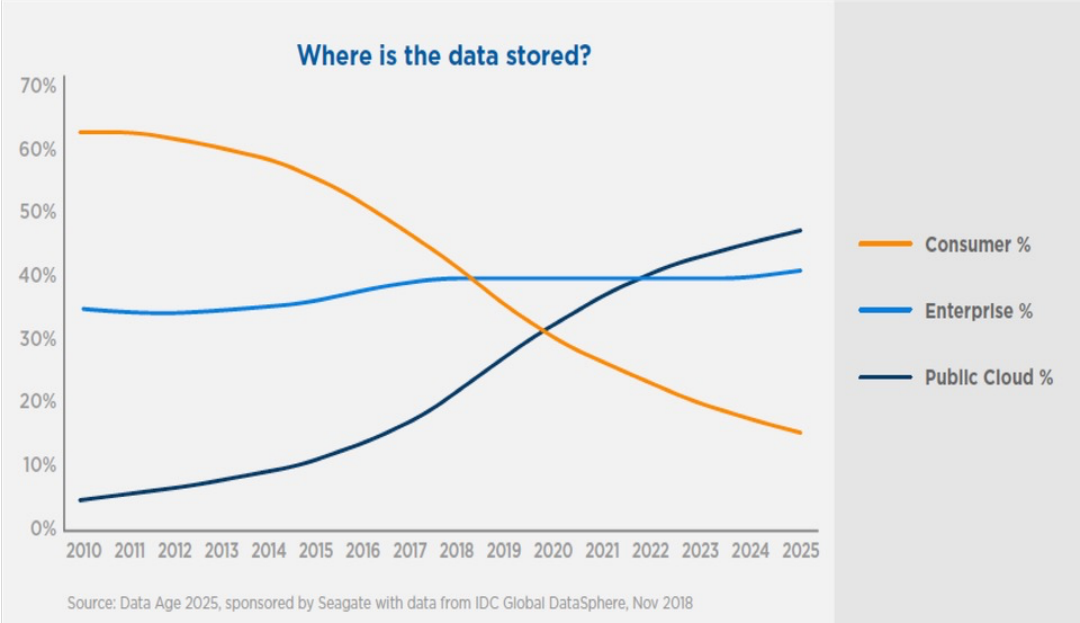
How much data?

- The explosion of data
  - 2013: 90% of all data in the world generated in the previous 2 years
  - From 2010 to 2025: 90x growth in data volume
  - Future projections: 394 ZB in 2028



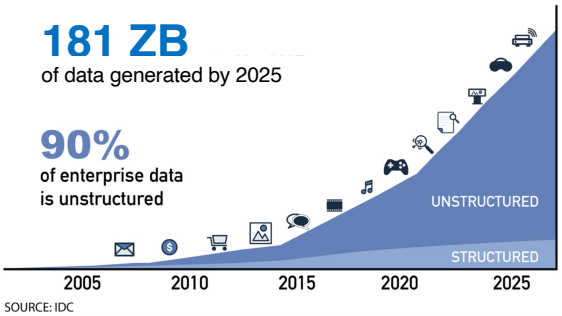
Where is data stored?

- The majority of data is now stored in Cloud data centers



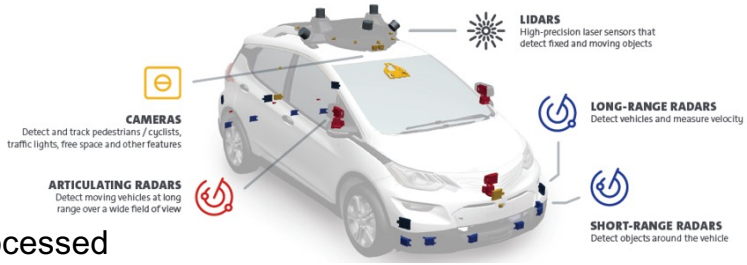
Big data driving factors

- Why Big Data is growing fast
  - Smartphones: 7.5 billion smartphone user, each user generates ~25 GB of data per month
  - Social networks
  - Internet of Things (IoT)
  - AI: the new "engine", from analyzing data to generating data
- Exponential growth in unstructured data
  - Unstructured data: information that does not have a fixed, organized structure
    - E.g., social media post, photo



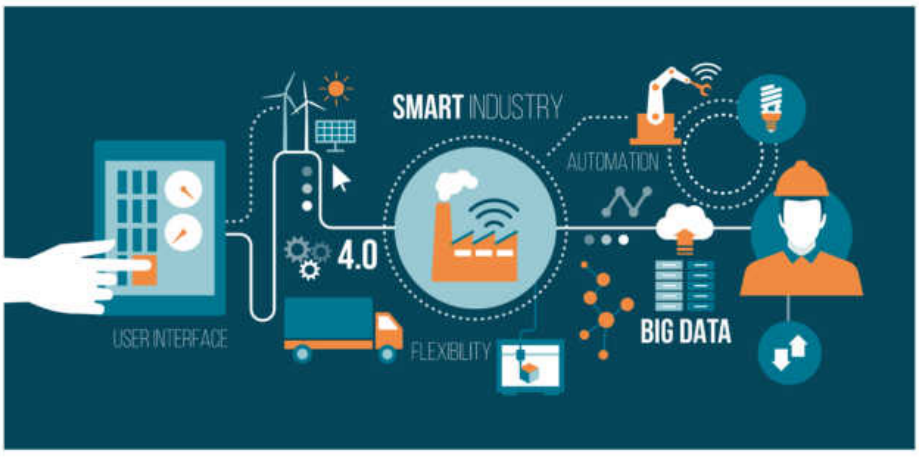
IoT and data explosion

- IoT is everywhere and largely contributes to increase Big Data challenges
  - Proliferation of data sources: by 2025 over 21 billion IoT devices
- Example: self-driving cars
  - A single car can generate 1 TB of data per hour
  - Where to process and store?
    - Collect everything is unfeasible
    - Process and discard
      - 99% of data is processed at the edge (in the car)
      - Only 1% of data is sent to the cloud (high-value events, corner cases)



IoT impact: Industrial IoT

- Industrial Internet of Things (IIoT): network of physical objects, systems, platforms and applications that contain embedded technology to communicate and share intelligence with each other, with external environment and with people



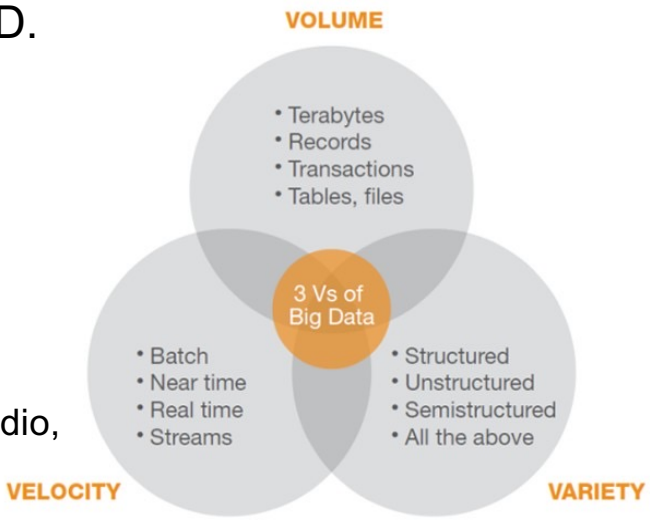
Big Data definitions

- Several definitions
  - "Big data primarily refers to data sets that are too large or complex to be dealt with by traditional data-processing software." (Wikipedia, 2026)
  - "Big data is mostly about taking numbers and using those numbers to make predictions about the future. The bigger the data set you have, the more accurate the predictions about the future will be." (Anthony Goldbloom, Kaggle's founder)
  - Big data is high volume, high velocity, and/or high variety information assets that require new forms of processing to enable enhanced decision making, insight discovery and process optimization. (Gartner, 2012)

- Scale changes everything!
  - Methodologies, tools, architectures

Dimensions for Big Data

- The 3V model (defined by D. Laney in 2001)
- Volume: data size
  - Challenging to store and process
- Variety: data heterogeneity
  - Different data types (text, audio, video, …) and degree of structure (structured, semi-structured, unstructured data)
- Velocity: data generation rate and analysis rate



Dimensions for Big Data

- Additional Vs include:
- Value: Big data can generate significant competitive advantages
  - "The bigger the data set you have, the more accurate the predictions about the future will be" (A. Goldbloom)
- Veracity: refers to quality and reliability of data, including issues related to uncertainty, accuracy, and authenticity of data
- Visualization: to represent large and complex datasets visually

Visualization

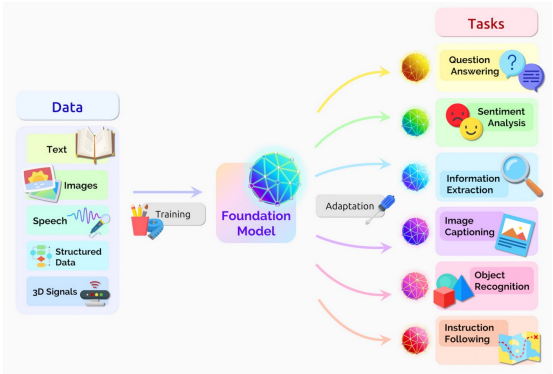
- Presentation of data in a pictorial and graphical format
- Why? Our brain processes images 60,000x faster than text
- Example:

Why Big data matters

- Data available in digital form
  - Global data continues to grow (23% growth from 2024 to 2025)
- We can store and process massive amounts of data
  - Storage: high-capacity disks (e.g., 20 TB for 400 €)
  - Computing: over 40 years of Moore's law plus multicore architectures
- New approaches require even larger datasets
  - E.g., transformer models
    - GPT-4: hundreds of billions of training parameters, hundreds of terabytes of training data

Transformers and foundation models

- Transformer
  - Type of neural network that learns context and meaning by tracking relationships in sequential data (like words in this sentence)
- Foundation models (aka large pretrained models, e.g., GPT-4)
  - Large-scale AI models trained on massive unlabeled data that can be adapted for many tasks
  - Learn general knowledge first, which can then be fine-tuned for specific applications



"The sheer scale and scope of foundation models over the last few years have stretched our imagination of what is possible" https://crfm.stanford.edu/report.html

The downside: environmental costs

- Rapidly rising energy consumption
  - 2022: data centers used ~1-1.3% of global electricity, cryptocurrency mining 0.4%
  - 2026 estimate: data centers will consume ~2% of global electricity
    - Similar to Japan
- The hidden cost of AI
  - A single ChatGPT query (~0.3 Wh) uses 10x more power than a Google search
- Is this sustainable?
  - The carbon gap: AI demand is growing faster than green grids can be built
  - Water impact for server cooling

https://www.nytimes.com/2023/10/10/climate/ai-could-soon-need-as-much-electricity-as-an-entire-country.html , NYT, Oct. 2023

Examples of Big Data applications across sectors

- Retail/customer analytics
  - Increase customer retention and loyalty
- Predictive maintenance for Industry 4.0
  - Detect anomalous machine states
- Crime prevention
  - Analyze crime patterns and trends
- Healthcare
  - Diagnose and treat genetic diseases, improve patient care
- Finance
  - Anticipate customer behaviors, optimize strategies
- Education
  - Improve learning outcomes; design personalized courses
- Space science
  - Support astronomical discoveries

Batch vs. real-time analytics

- **Batch analytics**: analyze datasets collected over a period of time that have been already stored
  - Goal: process large volumes efficiently
  - Our focus: batch processing engines

- **Real-time analytics**: analyze high-velocity, continuous data streams as soon as they arrive without (or before) storing them
  - Goal: obtain insights immediately or very quickly after data ingestion
  - Our focus: stream processing engines

Examples of real-time analytics

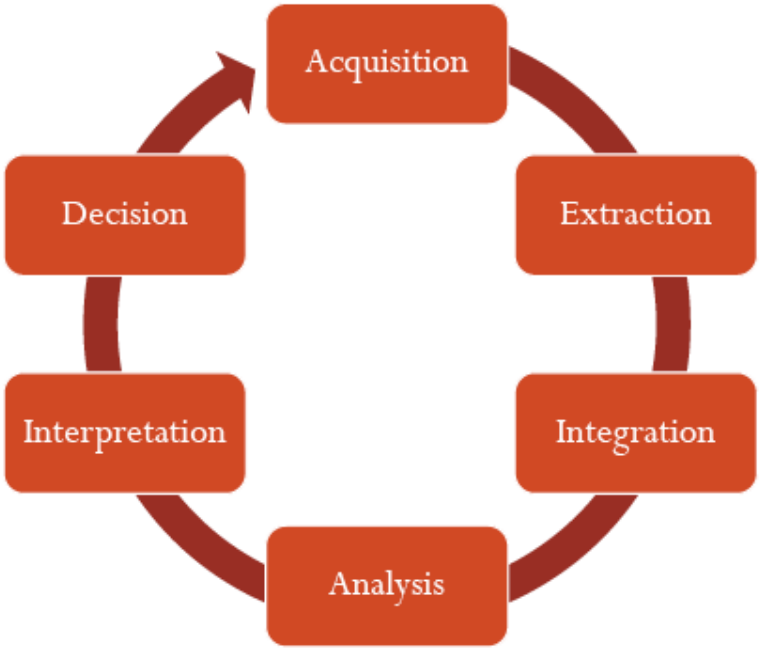
- Grand Challenge at DEBS conference https://debs.org/grand-challenges/
  - Energy consumption: analyze high-volume sensor data for energy measurements (DEBS 2014)
  - Taxi trips: analyze high-volume geospatial data streams from NYC taxi trips (DEBS 2015)
  - Social networks: Identify posts triggering the most activity and large communities involved in topics (DEBS 2016)
  - Maritime transportation: predict destinations and arrival times of ships (DEBS 2018)
  - Financial market data: compute trend indicators and detect patterns used by traders to decide on buying or selling (DEBS 2022)
  - Industrial manufacturing: real-time monitoring of defects in additive manufacturing to detect temperature anomalies and potential defects (DEBS 2025)

Examples of real-time analytics

- Medicine
  - Track epidemic outbreaks, prevent diseases through wearable health technologies
- Security
  - Detect fraud and DDOS attacks
- Urban traffic management
  - Address traffic congestion and lack of parking; optimize public transportation

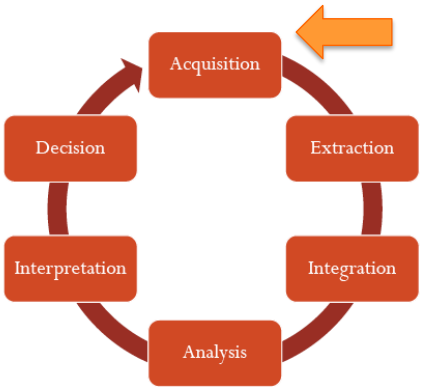
The Big Data process

- 6 stages of the Big data analytics lifecycle



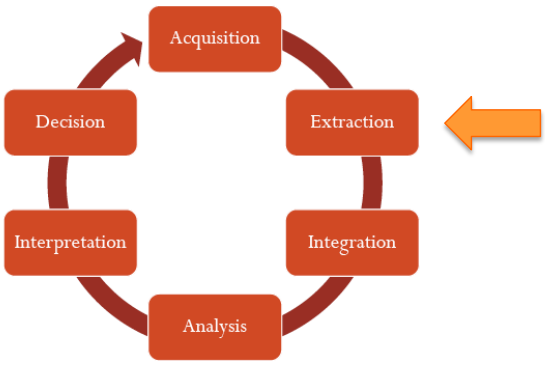
The Big Data process

- Acquisition
  - Gather and select data
  - Filter data
  - Generate metadata
  - Manage data provenance
    - Track origin, history, and transformations of data
    - Ensure compliance with regulations (e.g., GDPR)



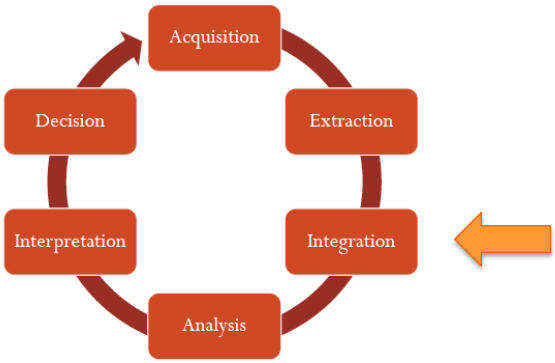
The Big Data process

- Extraction
  - Requires:
    - Data transformation: convert into formats used by Big data processing frameworks
    - Data normalization
      - Standardize data and avoid duplication
    - Data cleaning
      - Remove corrupted or inaccurate data (e.g., outliers)
      - Impute missing data using data imputation technique
    - Data aggregation
      - Combine data from multiple sources (e.g., different data providers)



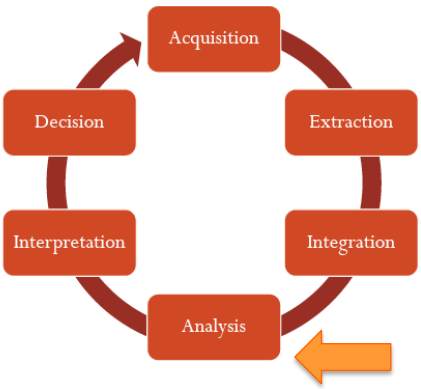
The Big Data process

- Integration
  - Combine data from multiple sources into a unified dataset
  - Requires:
    - Standardization
    - Conflict management
    - Reconciliation
    - Mapping definition
      - Define correspondences between different data schemas or attributes



The Big Data process

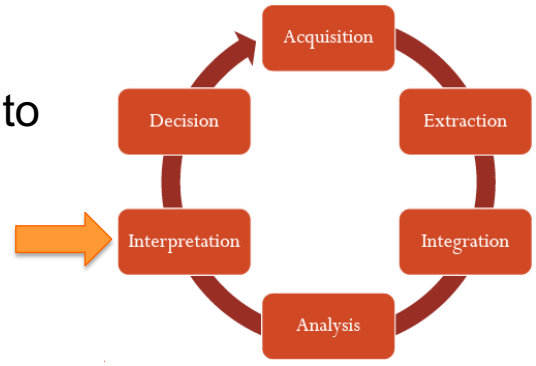
- Analysis
  - Extract insights and knowledge from data
  - Requires:
    - Data analytics techniques
      - Statistics
      - Data mining
      - Machine learning
      - Visualization



In this course we focus on Big Data systems and architectures rather than on data analysis techniques

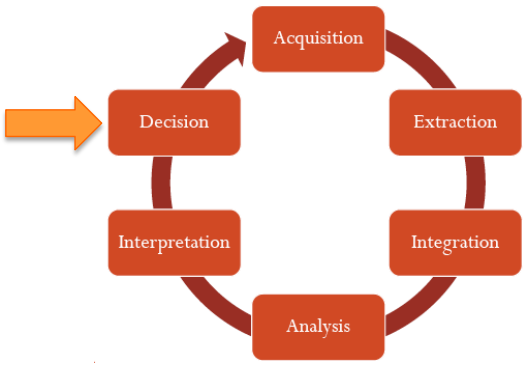
The Big Data process

- Interpretation
  - Transform analytical results into meaningful insights and decisions
  - Requires:
    - Knowledge of domain
    - Knowledge of data provenance
    - Identification of relevant patterns
    - Process flexibility



The Big Data process

- Decision
  - Use insights from data analysis to support actions and strategic choices
  - Requires:
    - Managerial skills
    - Continuous process improvement (loop iteration)

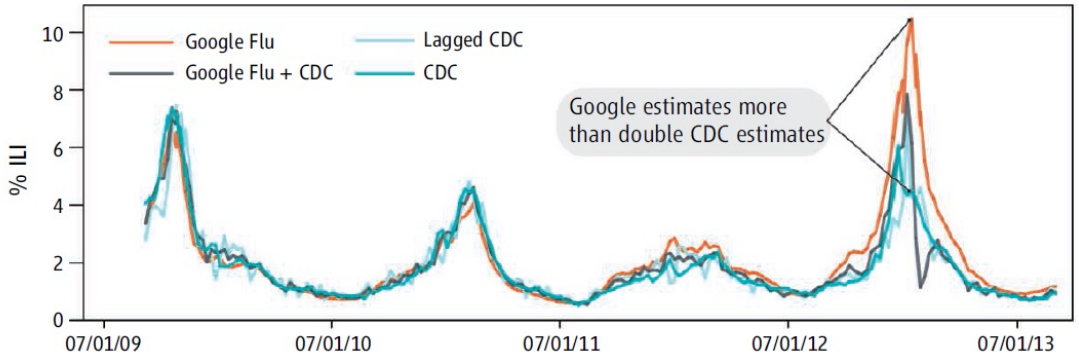


Risks and challenges of Big Data

- Effectiveness of data analysis
- Performance
  - Efficiency
  - Scalability and elasticity
    - Scale linearly as workloads and data volumes increase
  - Fault tolerance
  - Sustainability
    - Data volumes grow faster than energy efficiency of computing systems
- Heterogeneity
  - Data formats, processing environment, network latencies, …
- Flexibility
- Privacy and security
- Costs

Effectiveness of Big data analysis

- A famous example of inaccurate analysis
- Google Flu Trends
  - Used search query data to estimate flu activity
  - Sometimes highly inaccurate
    - 2011-2013: consistently overestimated flu prevalence
    - 2012-2013: predicted about twice
  - Lesson: large datasets do not guarantee accurate predictions



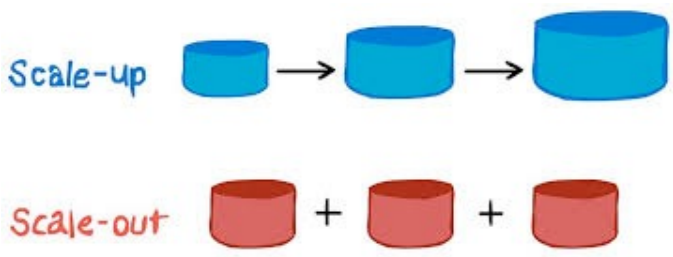
Lazer et al., "The Parable of Google Flu: Traps in Big Data Analysis". *Science,* 2014 https://gking.harvard.edu/files/gking/files/0314policyforumff.pdf

Taming performance: distribution and replication

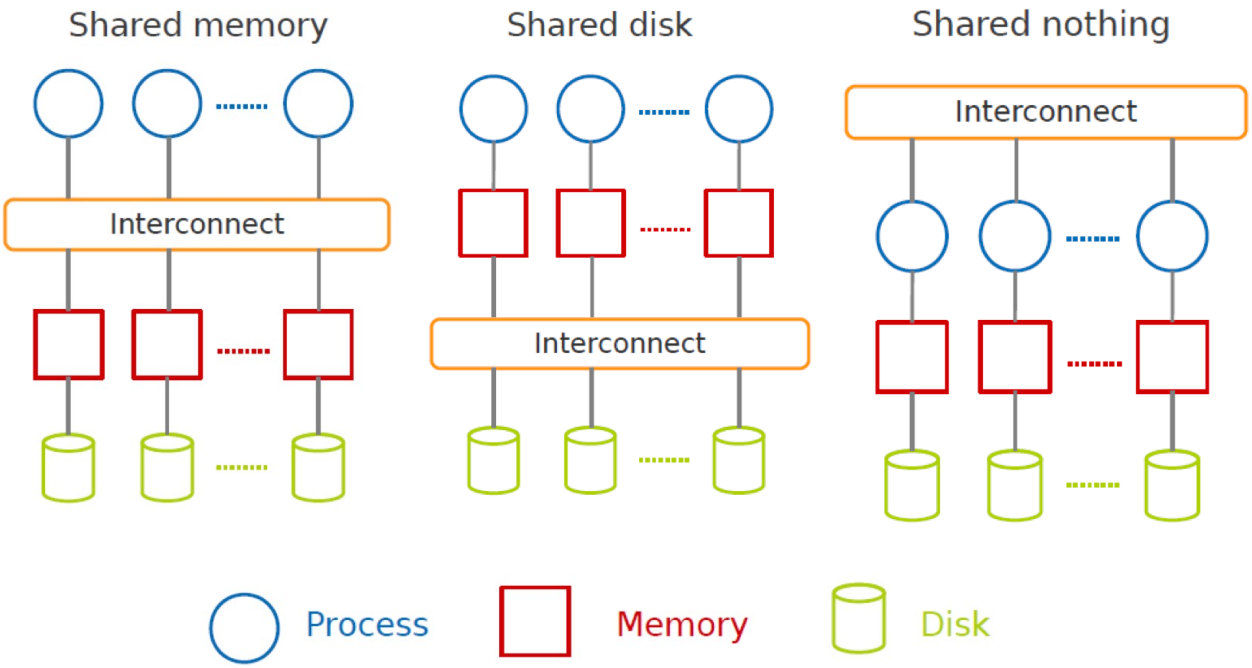
- Distributed architecture for Big Data
  - Common solution: cluster of commodity hardware, including Cloud
  - Scale out (horizontally), not up (vertically)
  - Challenges: elasticity and edge data processing
- Distributed processing
  - Shared-nothing model
  - New programming paradigms, e.g., functional programming
- Resource replication
  - Well-known solution for fault tolerance
  - Eventual consistency (CAP theorem)

Scale out vs. scale up

- Different ways to address the need for more computing and storage
- Scaling up (vertical scalability): upgrade to a more powerful server or VMs
  - E.g., add CPU cores, RAM
- Scaling out (horizontal scalability): add more servers or VMs to share the workload



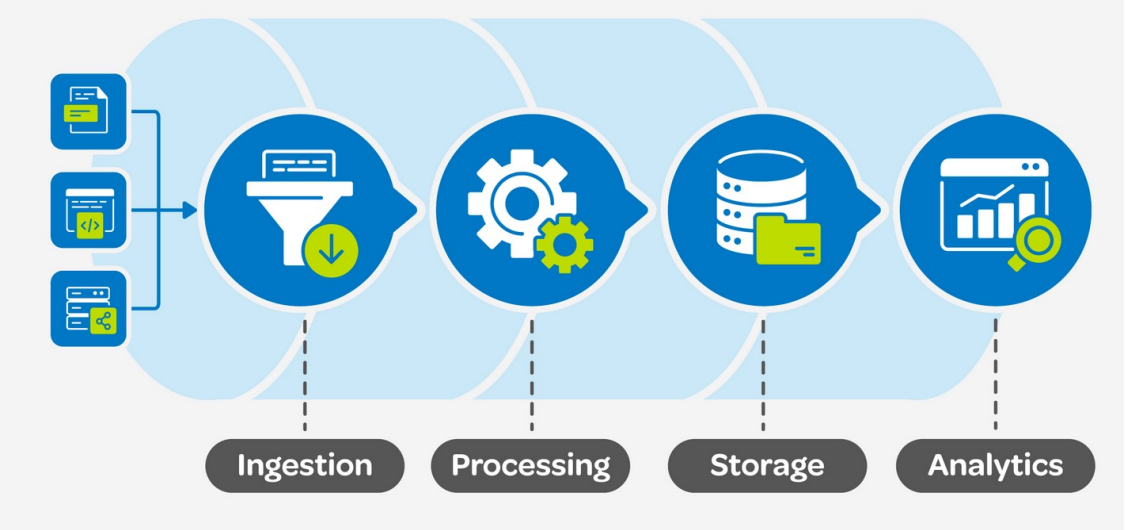
Shared nothing vs. other parallel architectures



DeWitt and Gray, "Parallel database systems: the future of high performance database systems", *Comm. ACM*, 1992
https://dl.acm.org/doi/pdf/10.1145/129888.129894

Data pipeline architecture

- A series of processes that collect, process, and deliver data from sources to storage or analytics systems



Data pipeline architecture: stages

- Ingest data
  - Capture raw data from various sources
  - Distributed file systems, object stores, message queues
- Process data
  - Transform, clean, filter, and enrich data for analytics or downstream use
  - Batch and stream processing frameworks
- Store data
  - Store processed data for later retrieval and analysis
  - Databases, data stores, data lakes, data warehouses
- Analyze data
  - Perform analytics, machine learning, and reporting on processed data
  - BI dashboards, ML frameworks, …

Where to process Big Data

- Traditional on-premises cluster of servers
  - Compute nodes are organized in racks
    - 8-64 nodes per rack
  - Multiple racks for large clusters
    - Nodes within a rack: high-bandwidth Gb network
    - Racks connected via a higher-level network or switch
    - Intra-rack bandwidth > inter-rack bandwidth

✓ Full control over hardware and software environment

✗ Must manage infrastructure: acquire, install, configure, …

Where to process Big Data

- The Cloud way: using Cloud analytics services
- Examples
  - Amazon EMR, Google Dataproc, Azure HDInsight: provide Spark and other frameworks as fully managed services

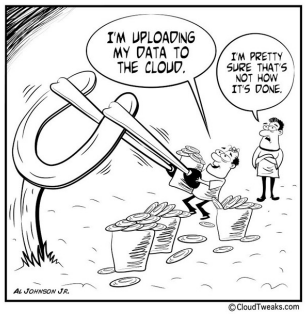
✓ Cloud scalability and elasticity

✓ No infrastructure and platform management

✗ Data transfer to the Cloud can be slow or costly

  - Latency is not zero!
  - Network bandwidth (minor)

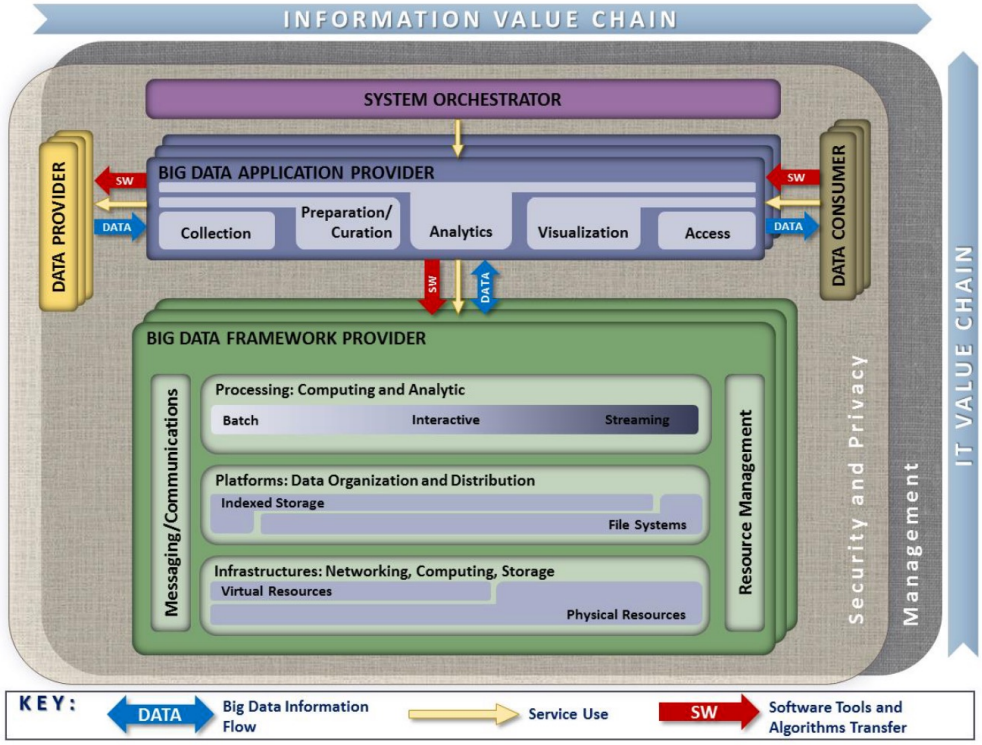
✗ Data security and privacy



Where to process Big Data

- The new scenario: Compute continuum
  - Progressive convergence of Cloud, Fog and Edge computing, resulting in a continuum of resources
  - Not limited to Cloud data centers, includes micro data centers located at network edges
  - Move data processing and analytics closer to data producers and consumers

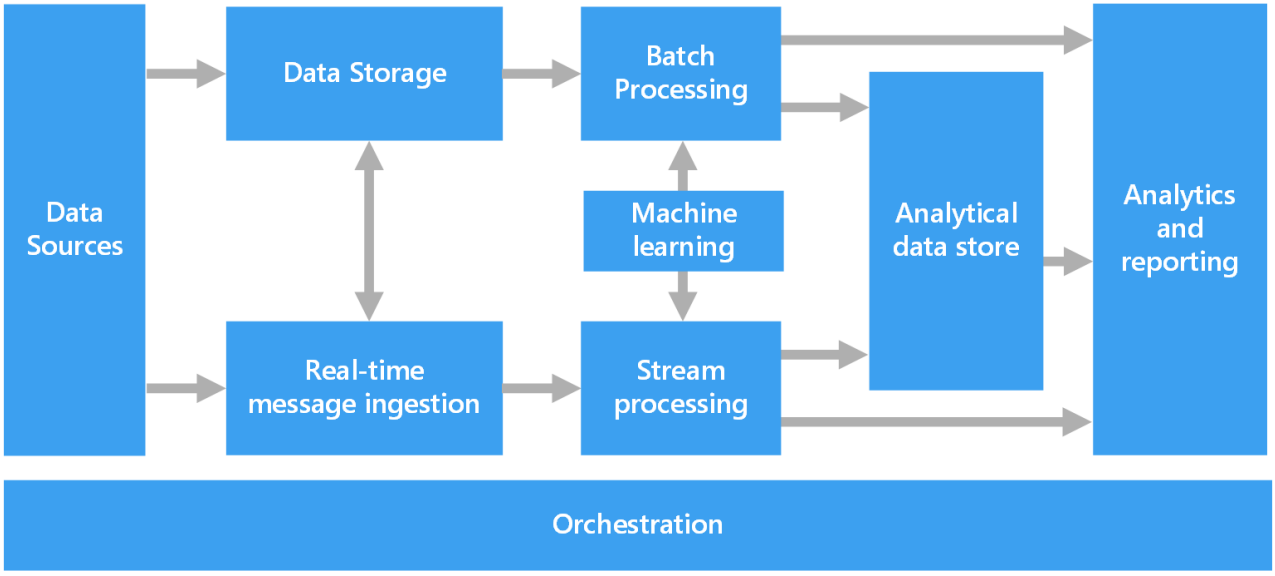
NIST Big Data reference architecture



https://doi.org/10.6028/NIST.SP.1500-6r2

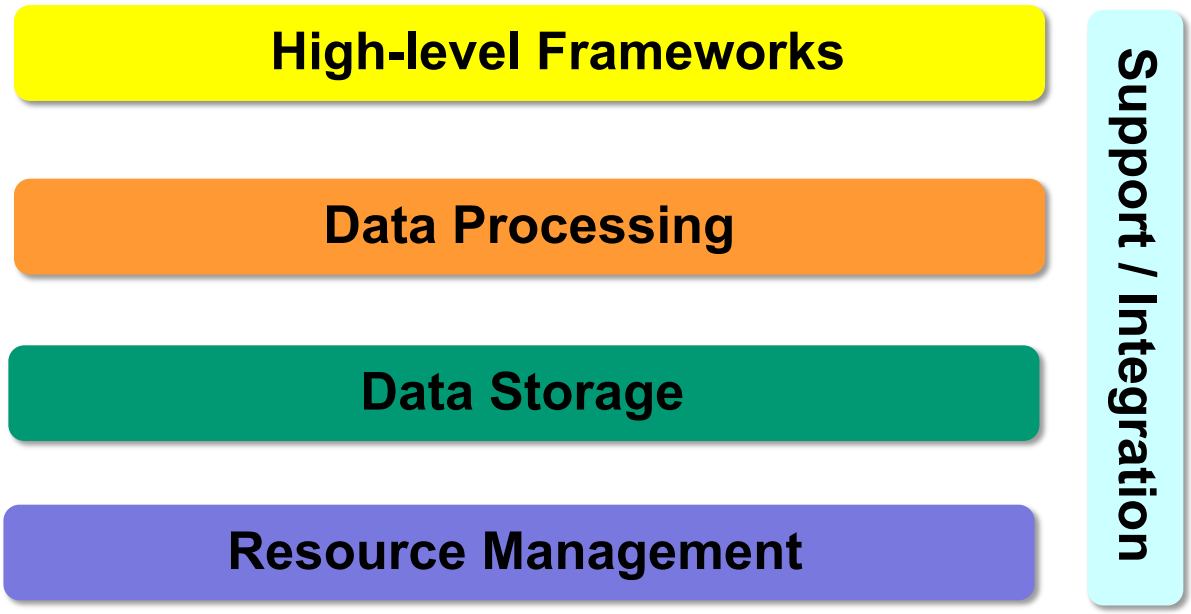
Big data architecture

- Designed to handle ingestion, processing, and analysis of Big data
- Components



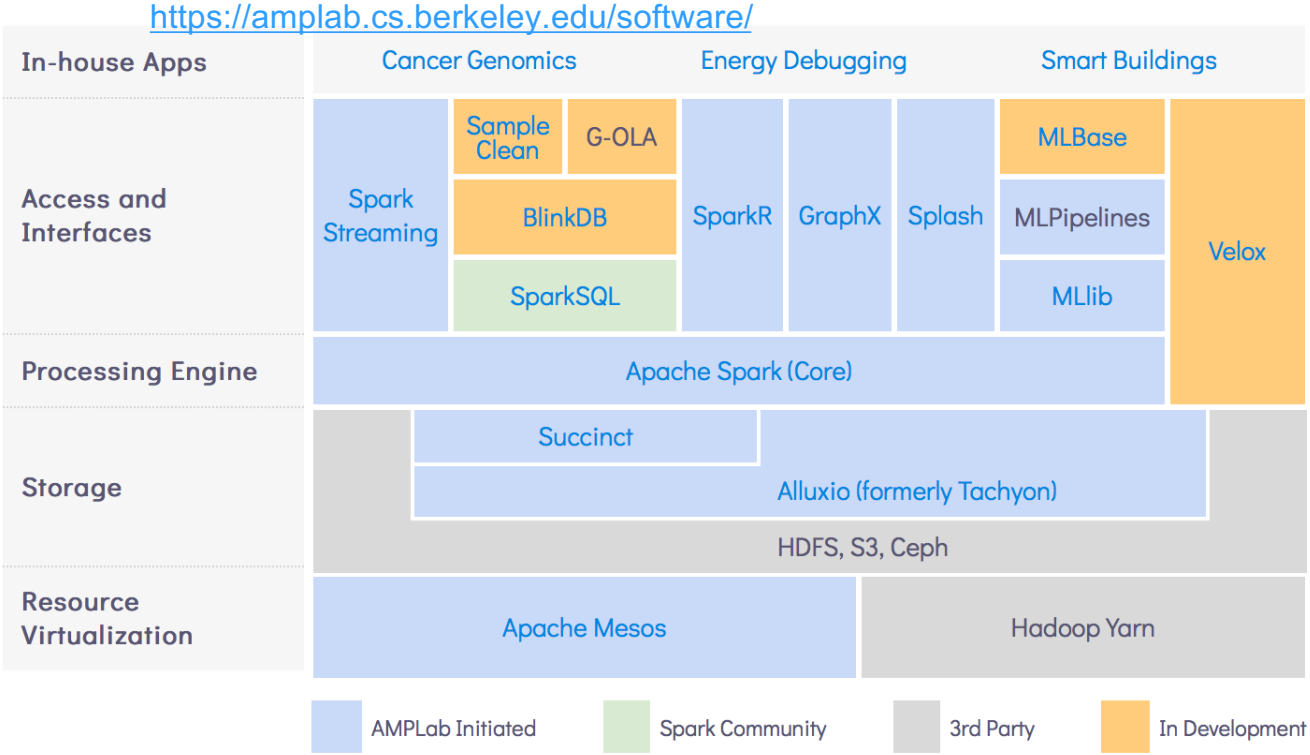
https://docs.microsoft.com/en-us/azure/architecture/data-guide/big-data/

Our Big Data stack



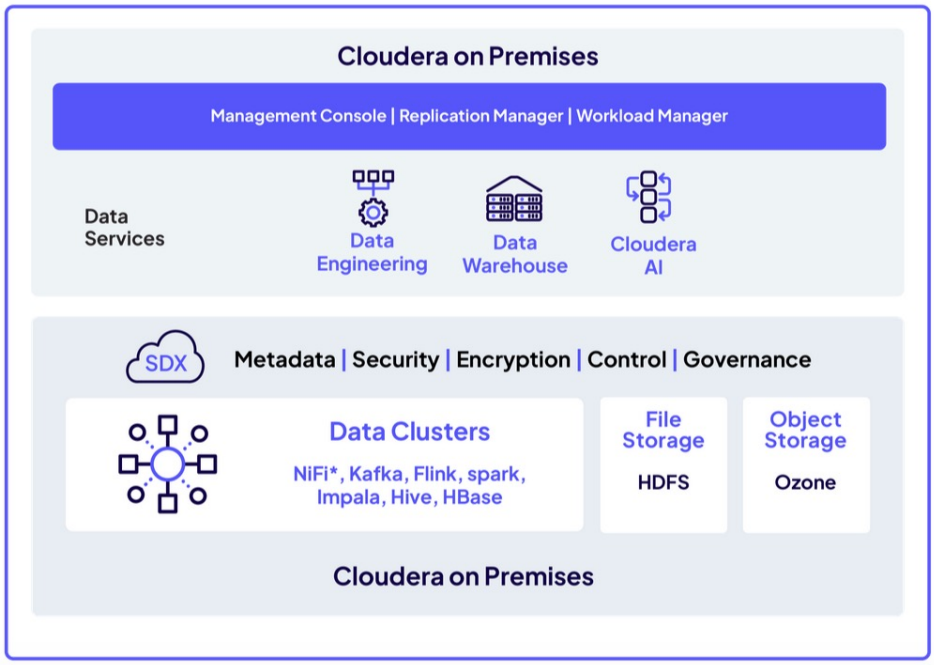
Example of Big Data stack: BDAS

- BDAS: Berkeley Data Analytics Stack
  https://amplab.cs.berkeley.edu/software/



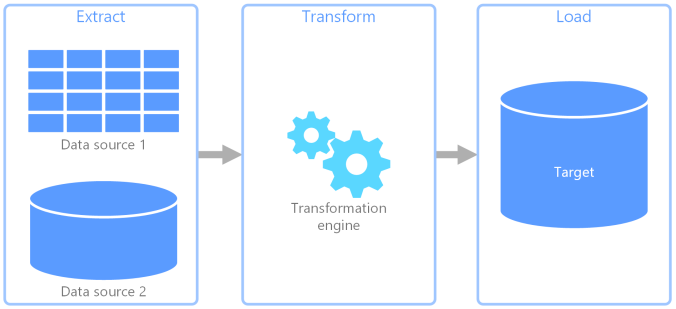
Example of Big Data stack: Cloudera

https://www.cloudera.com/products/cloudera-data-platform/private-cloud.html



Approaches for data ingestion and integration: ETL

- Extract, Transform, and Load (**ETL**)



  - ➢ *Extract* data from multiple sources
  - ➢ *Transform* data into a usable and trusted resource for storing
    - ➢ Often staged in a temporary staging database before final loading
  - ➢ *Load* transformed data into the target system (DB or data warehouse, DWH)

https://databricks.com/it/glossary

Data warehouse

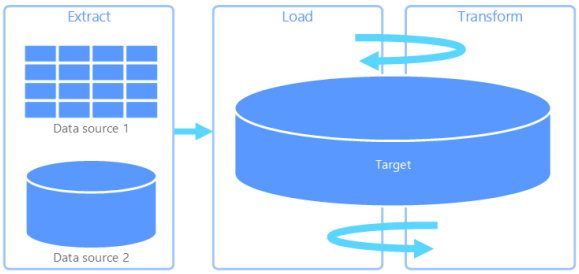
- Relational DBMSs were sufficient for decades for business operational needs (On-Line Transaction Processing, OLTP)
  - Typically, writing/updating single rows of data for many concurrent users (e.g., banking, e-commerce)
  - E.g., record that Maria bought a jar of Nutella
- Analytical shift (1990s): emergence of On-Line Analytical Processing (OLAP)
  - Complex, long-running queries for decision support
  - E.g., show the total exports of Nutella to France
- Data warehouses introduced to integrate structured data from multiple RDBMS under a single umbrella
  - Allow an organization's collection of RDBMS to be queried and viewed as a whole

Data warehouse: pros and cons

- Architectural evolution of data warehouses
  - 1st generation: ran on expensive on-premises hardware appliances (e.g., Teradata, early Vertica)
    - Tight coupling of storage and compute
  - 2nd generation: cloud architectures based on separation of compute and storage (e.g., AWS Redshift, Snowflake, Google BigQuery)
- Pros and cons
  - ✓ Integration of multiple data sources
  - ✓ Optimized for read access
  - ✓ Fast ad hoc analytical queries
  - ✓ Data audit, governance, and lineage
  - ✗ Limited support for unstructured or raw data
  - ✗ Expensive proprietary hw and sw (e.g., proprietary data formats)
  - ✗ Older architectures are difficult to scale

Approaches for data ingestion and integration: ELT

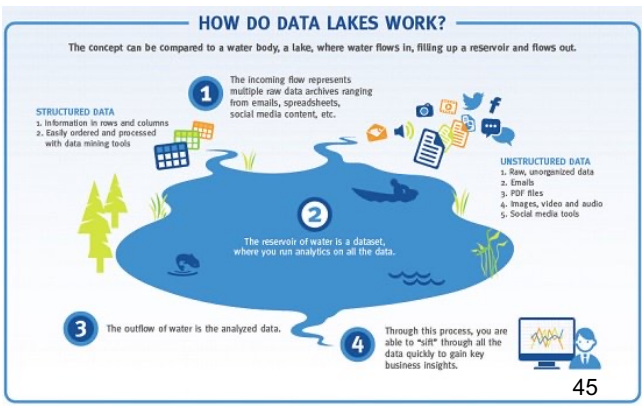
- Extract, Load, and Transform (**ELT**)



  - ➢ *Extract* data from different sources
  - ➢ *Load* data into a **data lake**, where data is held in original format
  - ➢ *Transform* data using the processing capabilities of target system
- Pros:
  - No need for separate transformation engine
  - Data transformation and loading can happen in parallel
  - More effective when speed is critical
  - Works well when target system is powerful enough to transform data efficiently

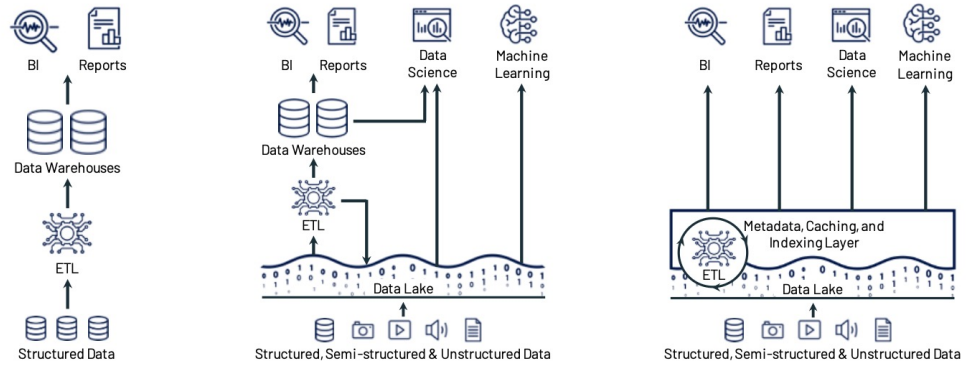
Data lake

- Method of storing data in a system or repository in its native, raw format
  - Facilitates collocation of data in a variety of formats (structured, semi-structured, unstructured)
  - Raw format: as-is from source, without prior transformations or integration (e.g., logs, posts, DB extracts, images, video)
  - Data is store using files or object blobs in distributed file systems or object stores
- Designed for quickly changing data
  - Data that arrive quickly or data with evolving types
- Schema-on-read
  - Data is stored in its raw format
  - Schema is applied only when data is read



New architectural paradigm: data lakehouse

- Data lakehouse: data lake + data warehouse
- A new architectural unification strategy
  - Integrates data lake and data warehouse to improve performance, scalability, flexibility, and cost-effectiveness and eliminate data silos and ETL processes
  - Unifies all data to simplify data engineering processes
  - Supports business intelligence (BI) and ML workloads on all data



(a) First-generation platforms. (b) Current two-tier architectures. (c) Lakehouse platforms.

Data lakes vs data lakehouses

- Data lakes
  - Rely on distributed file systems or object stores
  - Typically, do not provide ACID transactions or table-level metadata
  - Store raw data in various open formats (e.g., JSON, CSV, Parquet, ORC)
  - Schema is applied at read time (schema-on-read)
- Data lakehouses
  - Add a transactional metadata layer between storage and the query engine
  - Use table formats (Apache Iceberg, Delta Lake, Hudi)
    - Define table-level metadata
    - Track file locations, schema evolution, snapshots, partitions
  - Enable ACID transactions, schema evolution, and efficient querying

Data warehouse vs lakes vs lakehouses

	Data warehouse	Data lake	Data lakehouse
Data type	Structured	All	All
Schema	Schema-on-write	Schema-on-read	Schema enforcement + evolution
Performance	Optimized for queries and BI	Flexible, may require transformation at read time	High performance for BI, ML, and analytics
Cost	High	Low	Cost-effective (decoupled and elastic)
Data processing	ETL, batch	ELT, batch/streaming	Unified, batch and streaming
Flexibility	Low (rigid)	High (open)	High (open formats, multi-engine)
Use cases	BI, reporting	Data exploration, ML training on raw data	BI + ML + real-time analytics
Examples	Teradata, AWS Redshift	Amazon S3, HDFS, Apache Ozone	Databricks, Dremio, Apache Hudi

References

- Wing, The data life cycle, *Harvard Data Science Review*, 2019 https://hdsr.mitpress.mit.edu/pub/577rq08d/release/4
- Armbrust et al., Lakehouse: A new generation of open platforms that unify data warehousing and advanced analytics, *CIDR '21*, 2021 https://mail.vldb.org/cidrdb/papers/2021/cidr2021_paper17.pdf