DATA TECHNOLOGIES AND SERVICES

LECTURER: Martina Šestak, Ph.D.

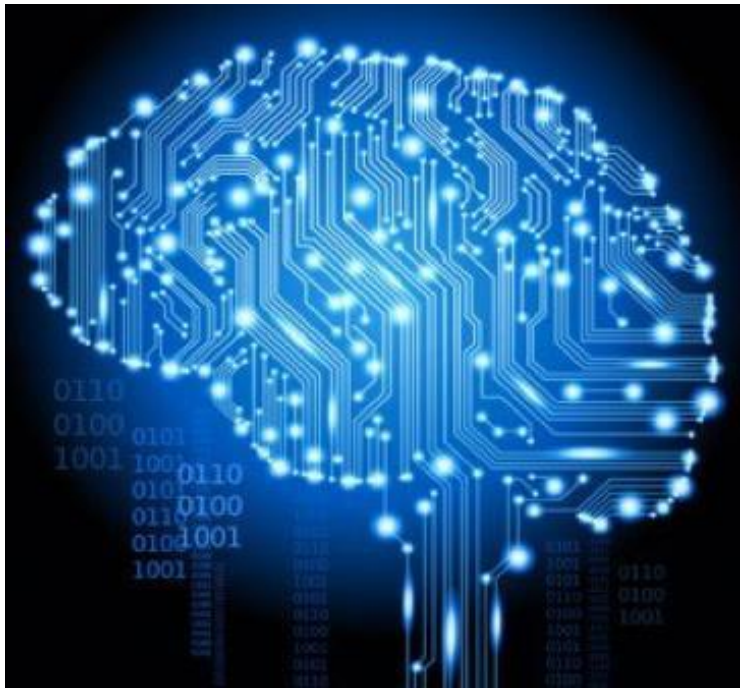
martina.sestak@um.si
Office: G2-1N.10
Office hours: Tuesday, 9AM-10AM
(send email beforehand)

DATA MANAGEMENT NOWADAYS…

- How do we ingest or consume the data?
- Where and how do we store it? How can we ensure as well as enforce data quality?
- What operations do we perform on the data?
- How can these operations be efficient?
- How do we manage data scalability, variety, velocity, and access?
- How can we enforce security and privacy at each stage of data modeling?

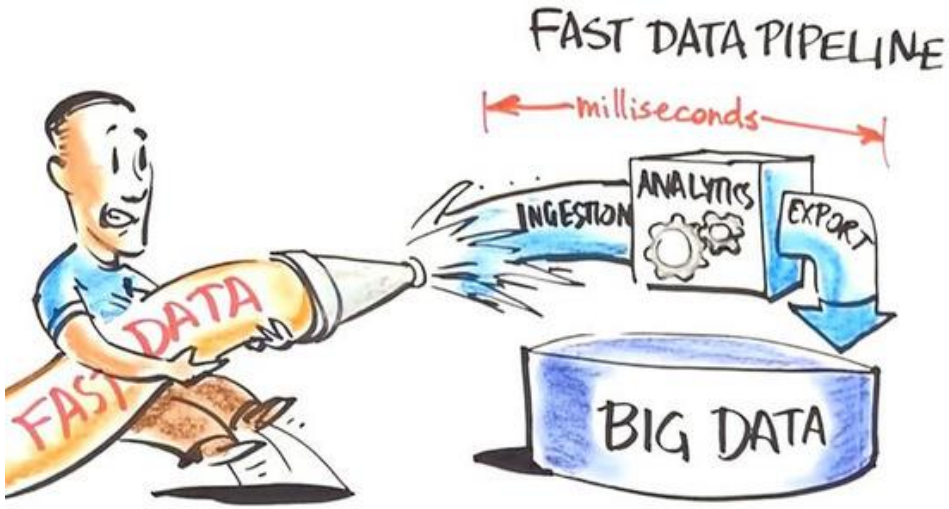
TRANSITION TO SMART DATA

- Changing data from assets to value
- Data analytics, predictive modelling and visualization are becoming crucial to the businesses' existence
- Big data analytics allows the detection of hidden
- patterns and uncertain correlations
- The growing focus on smart data versus big data can be attributed to the growing use of artificial intelligence
- Smart data is data in which intelligent algorithms have detected patterns



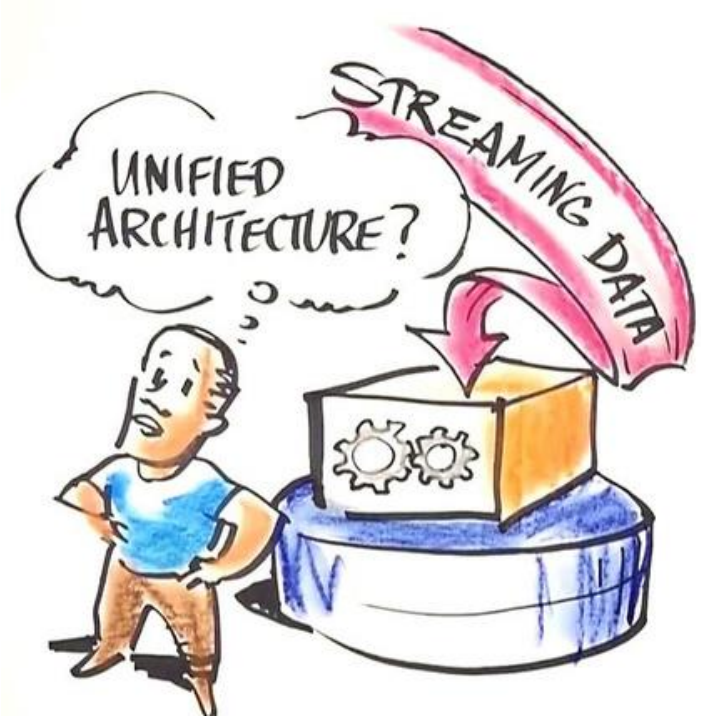
TRANSITION TO FAST DATA

- Fast data is data in motion
- They are described with velocity
- Streaming analytics
- The aim is to quickly collect and mine structured and unstructured data enabling activities to be carried out
- Fast data often flows in streams into data systems

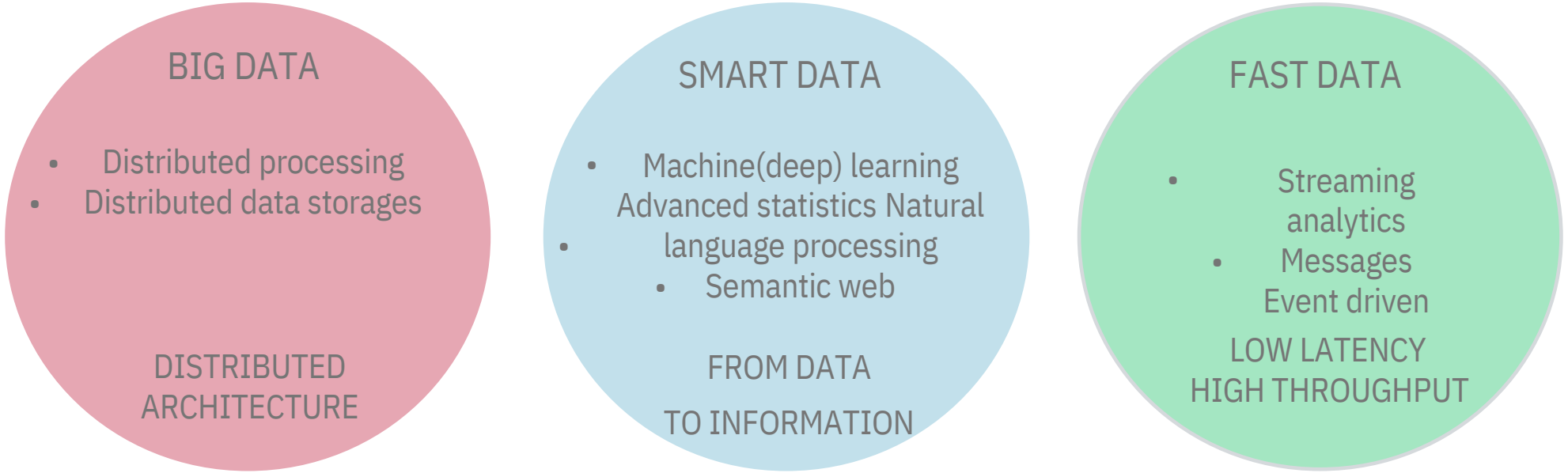


TRANSITION TO FAST DATA

- Fast data processing
  - "Ingestion" -gets millions of events per second
  - “Decision” -executes a data-driven decision at each event
  - “Real-time analysis” -enables automatic decision-making and provides insight into the trends



BIG DATA, SMART AND FAST DATA



BIG DATA

- Huge amounts of data that cannot be stored, processed and analyzed in the traditional way

- Example: platform X –millions of tweets (Xs) and history search

(6V) OF BIG DATA

Veracity

Variety

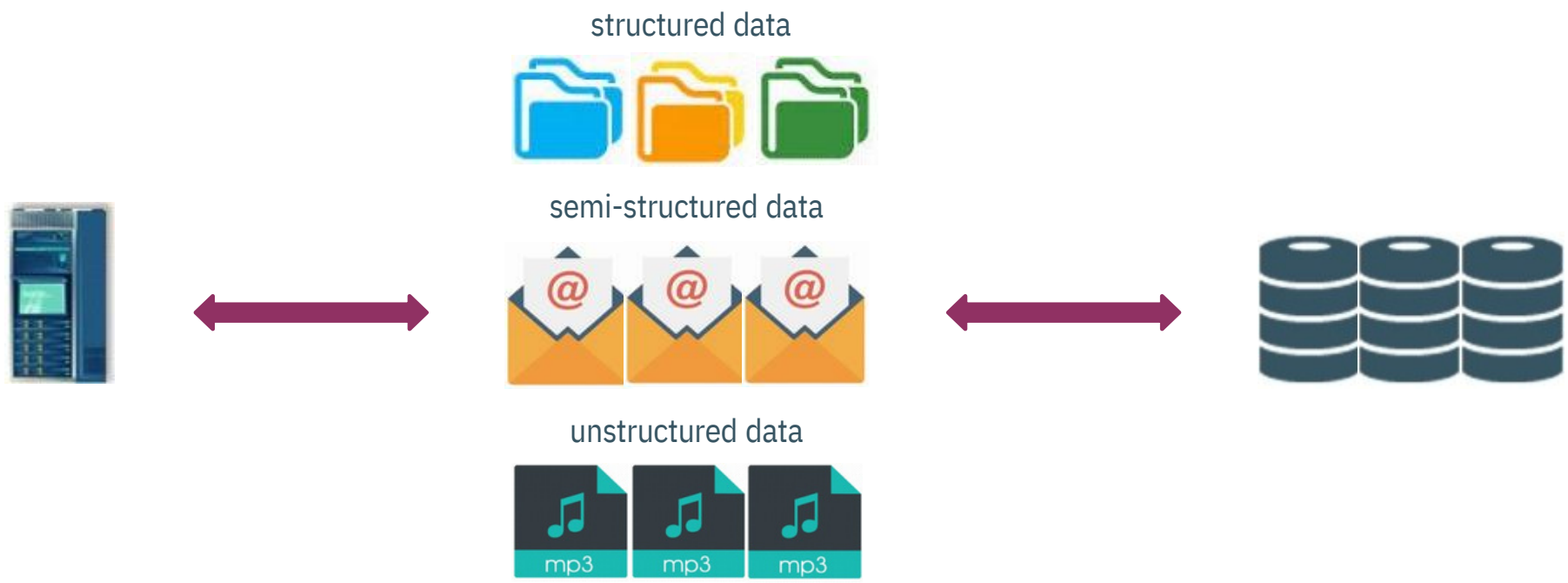
Velocity

Volume

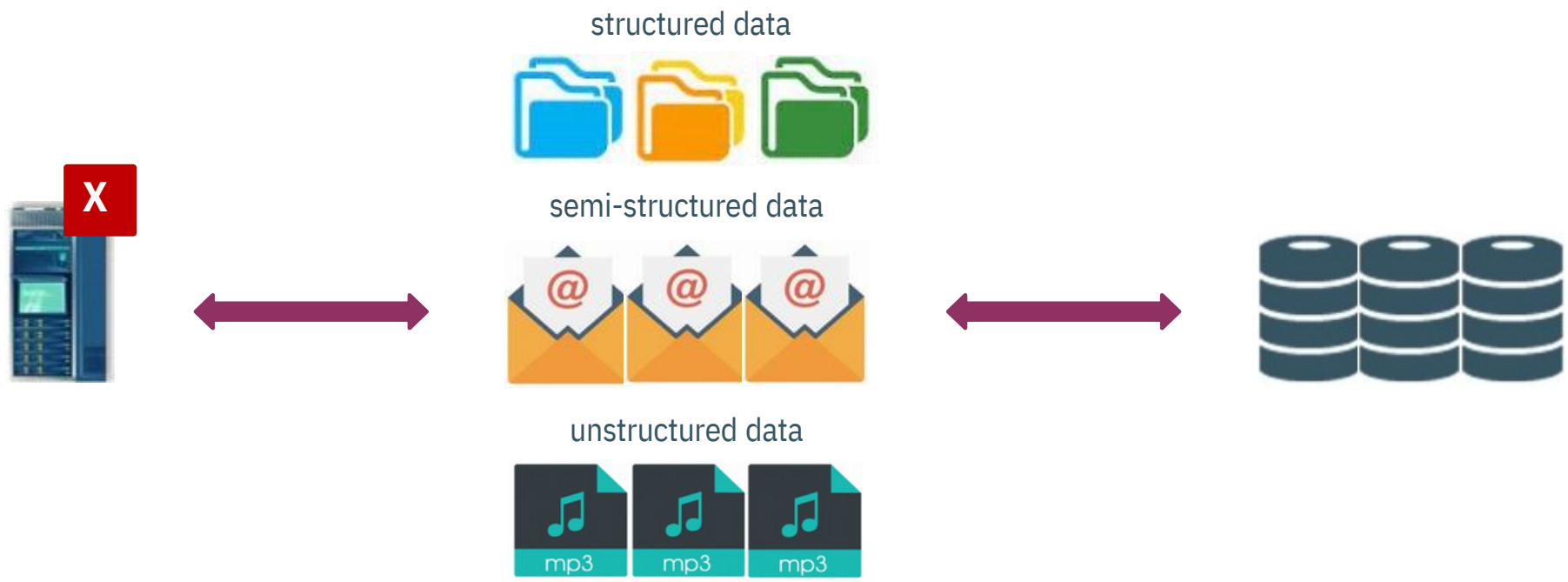
Value

Variability

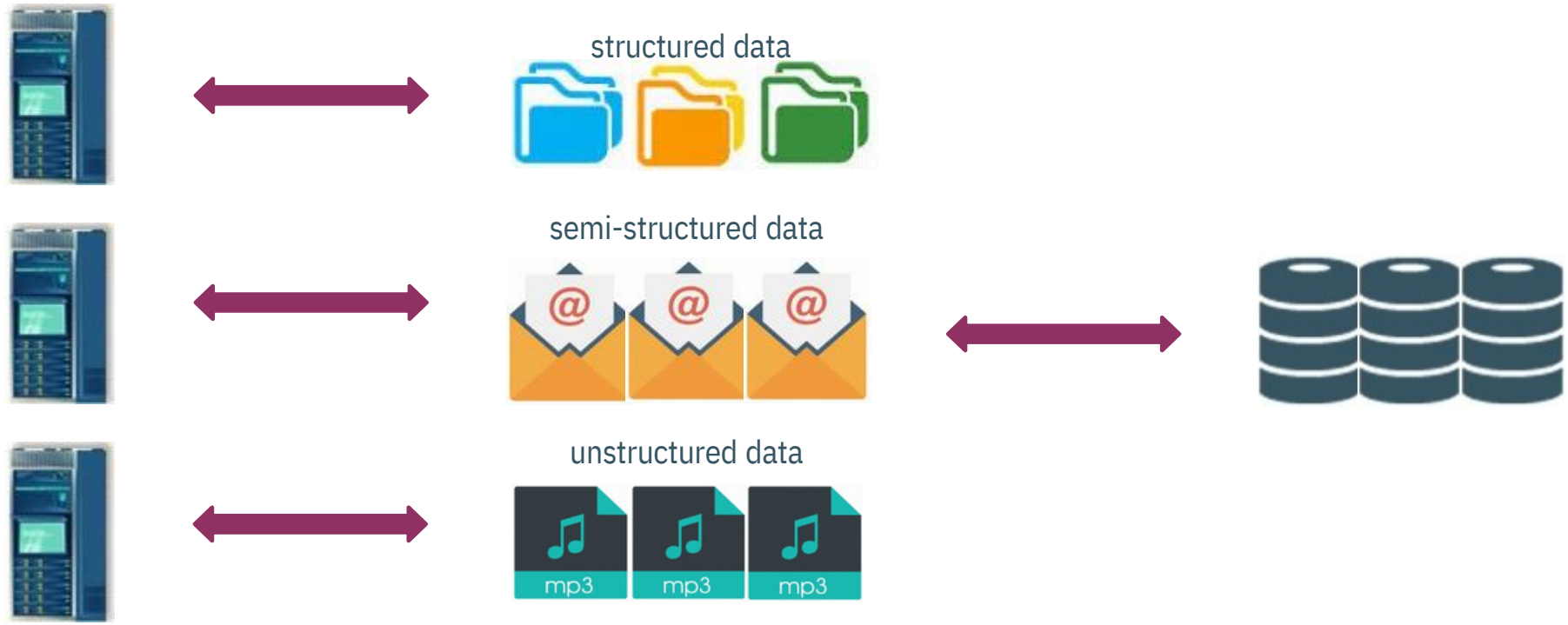
BIG DATA


structured data
semi-structured data
unstructured data
mp3
mp3
mp3

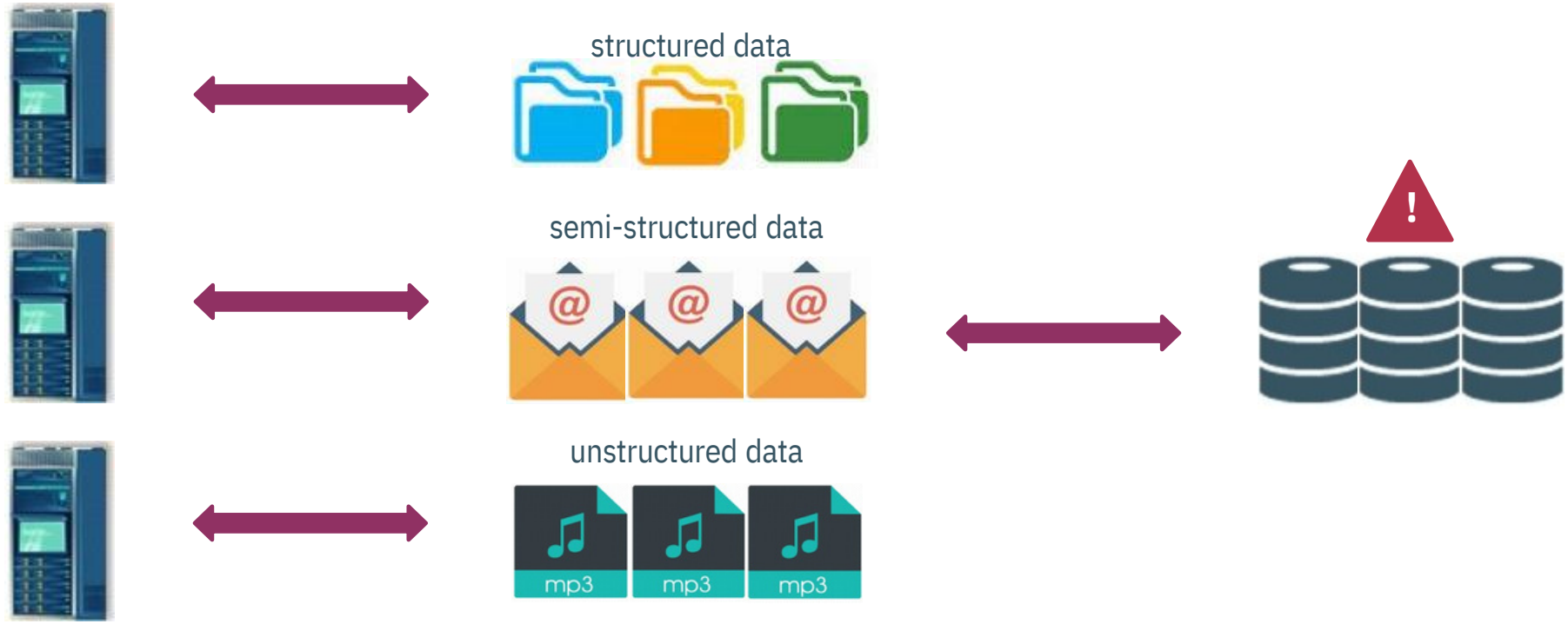
BIG DATA


structured data
semi-structured data
unstructured data
mp3
mp3
mp3

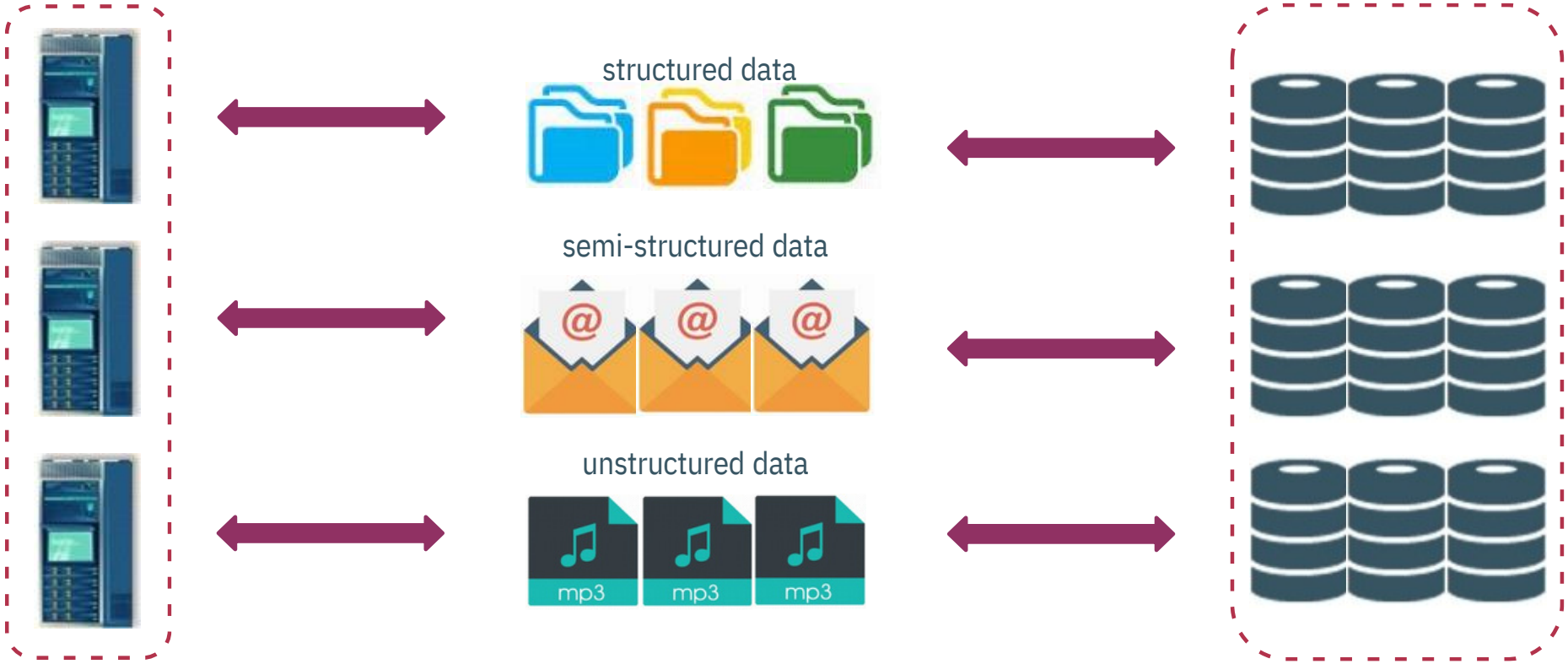
BIG DATA


structured data
semi-structured data
unstructured data
mp3
mp3
mp3

BIG DATA


structured data
semi-structured data
unstructured data
mp3
mp3
mp3

BIG DATA



We are not waiting for data access or data processing!

WHAT IS NEEDED IN THE BIG DATA ERA?

Scalability

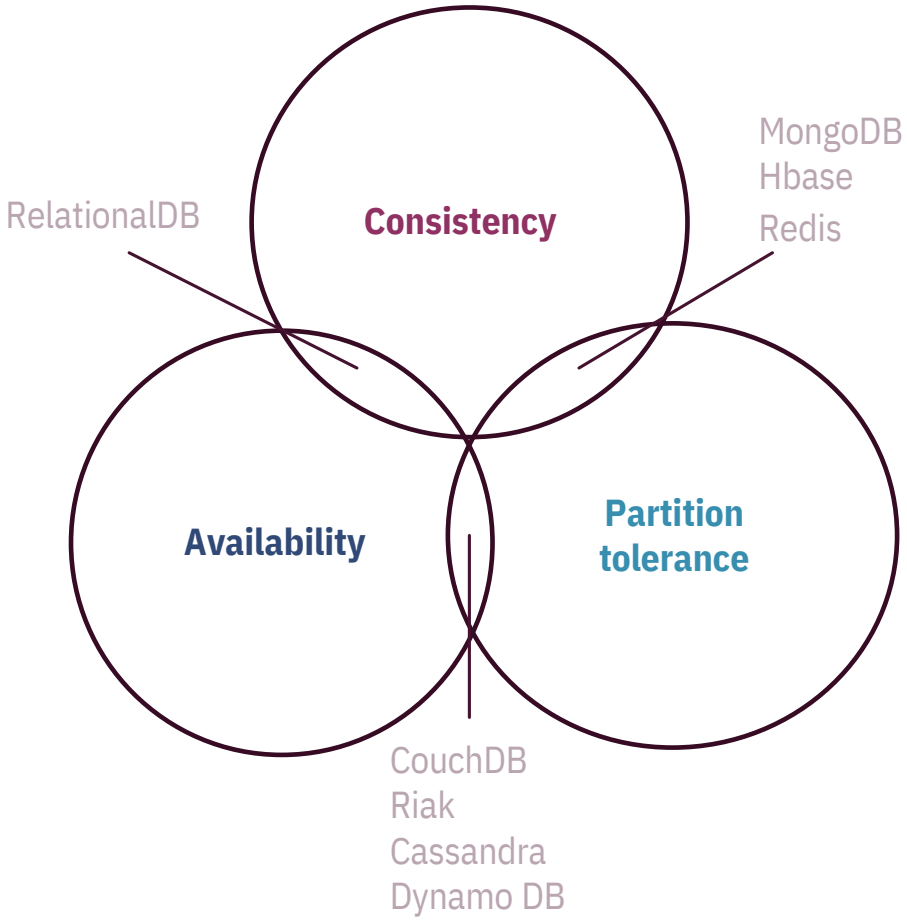
Vertical or horizontal? By increasing the amount of data, we reach a point where we cannot or no longer make sense to increase the capacity of the environment

Availability
Flexibility → reliability

Increase efficiency → high performance

DISTRIBUTED DATABASES

Brewer‘s theorem

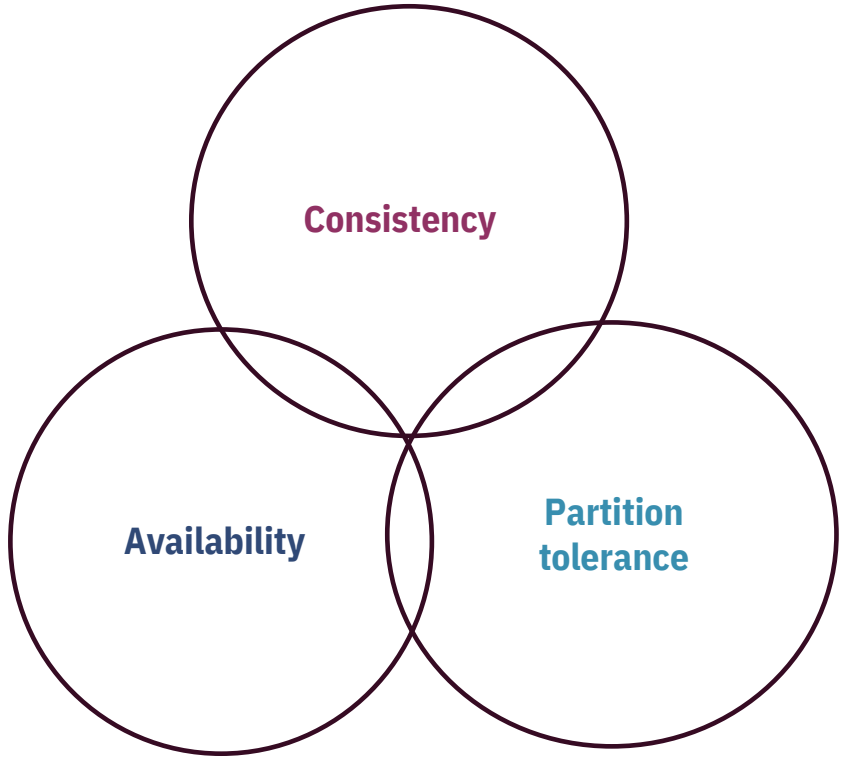


- Named CAP theorem (Consistency, Availability, Partition Tolerance)
- Any distributed data system can provide at most 2 out of 3 guarantees at once!
- Consistency
  - All nodes always have access to the same data
- Availability
  - Every request receives a (non-error) response, without the guarantee that it contains the most recent write
- Partition tolerance
  - The ability of the system to operate despite of the failure of part of the system

DISTRIBUTED ENVIRONMENTS

CAP theorem

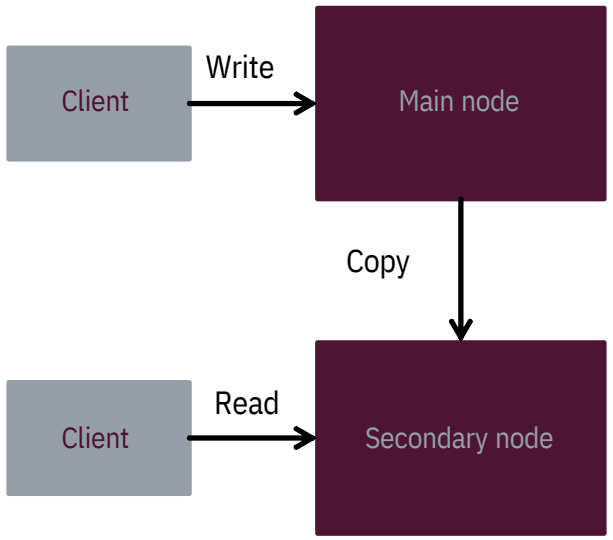
- Applies only in cases where the connection between nodes in the cluster fails
  - The more reliable the connection, the less likely the CAP effect is to occur
  - We need to ensure partition tolerance in a distributed database
  - Availability and consistency issue in case of network failure
  - Decide on the appropriate approach according to the requirements



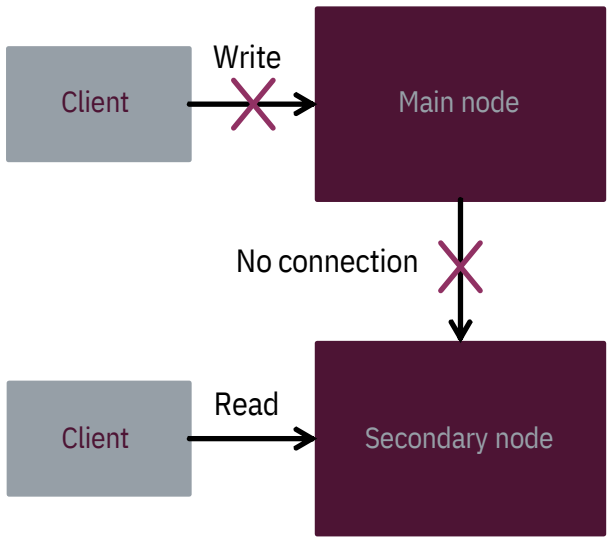
DISTRIBUTED ENVIRONMENTS

CAP theorem

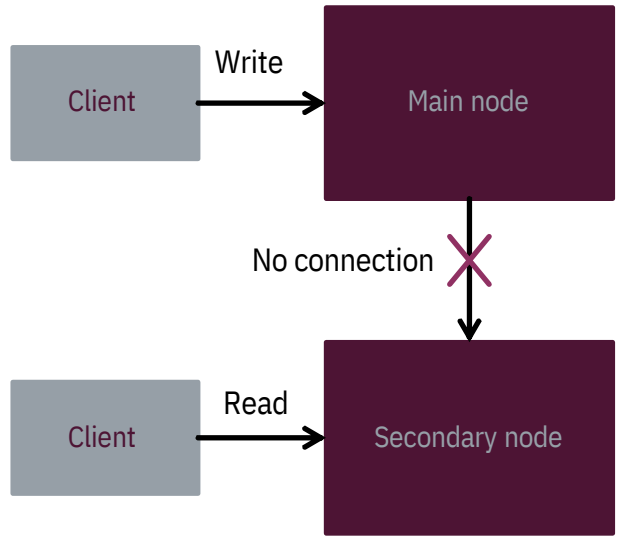
Normal functioning



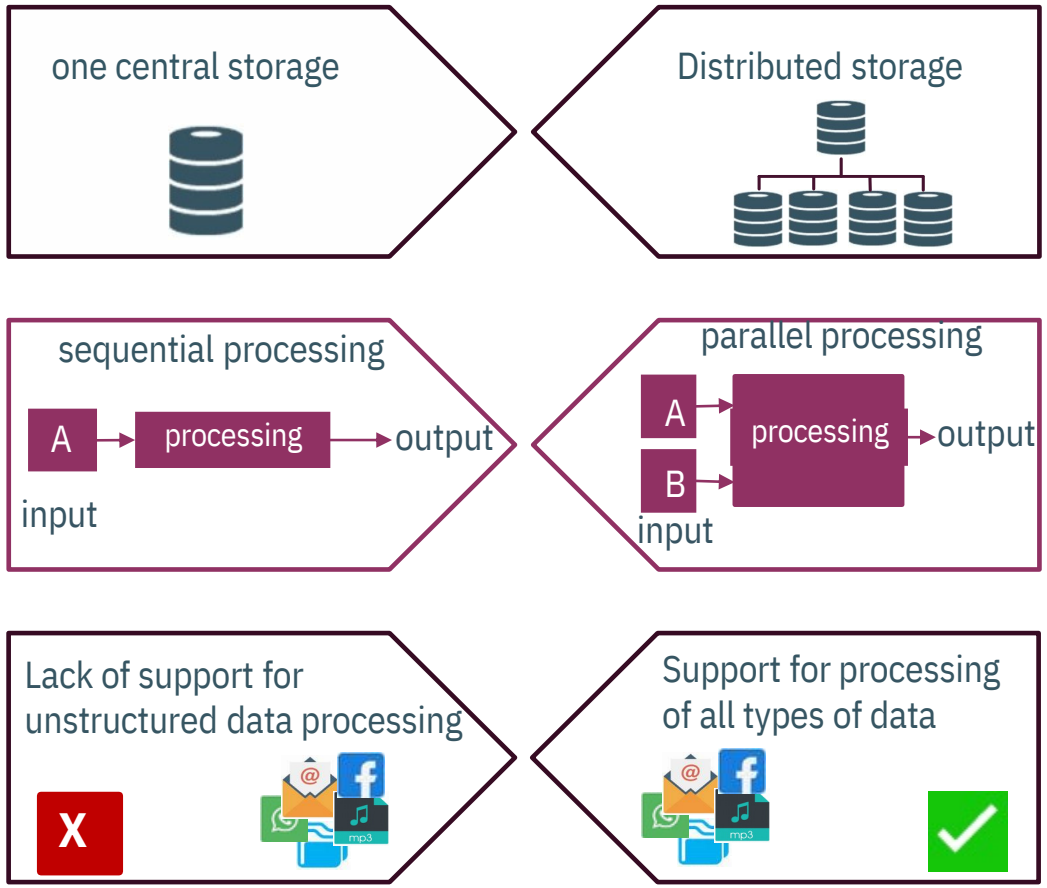
High consistency



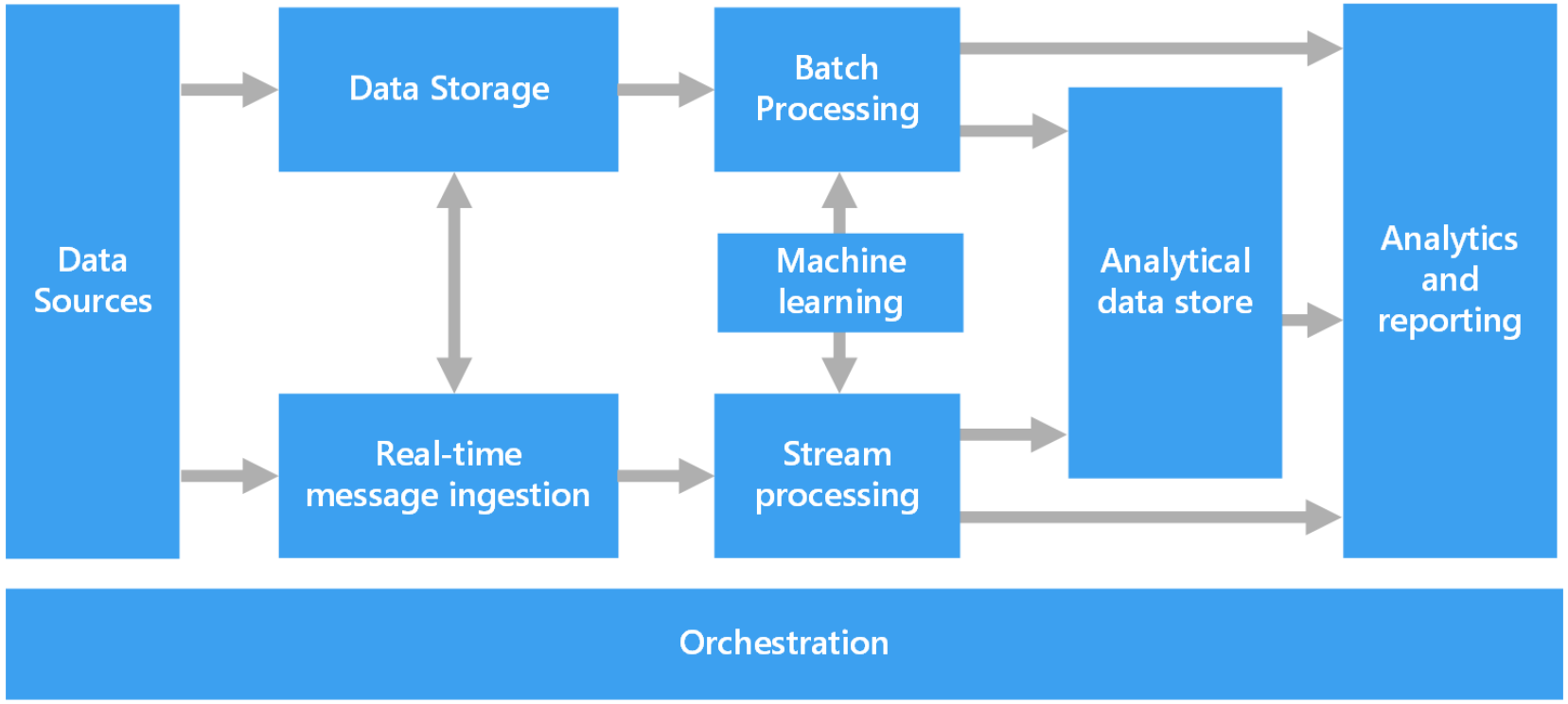
High availability



BIG DATA –CHALLENGES AND SOLUTIONS



BIG DATA ARCHITECTURE



https://learn.microsoft.com/en-us/azure/architecture/databases/guide/_images/big-data-pipeline.png

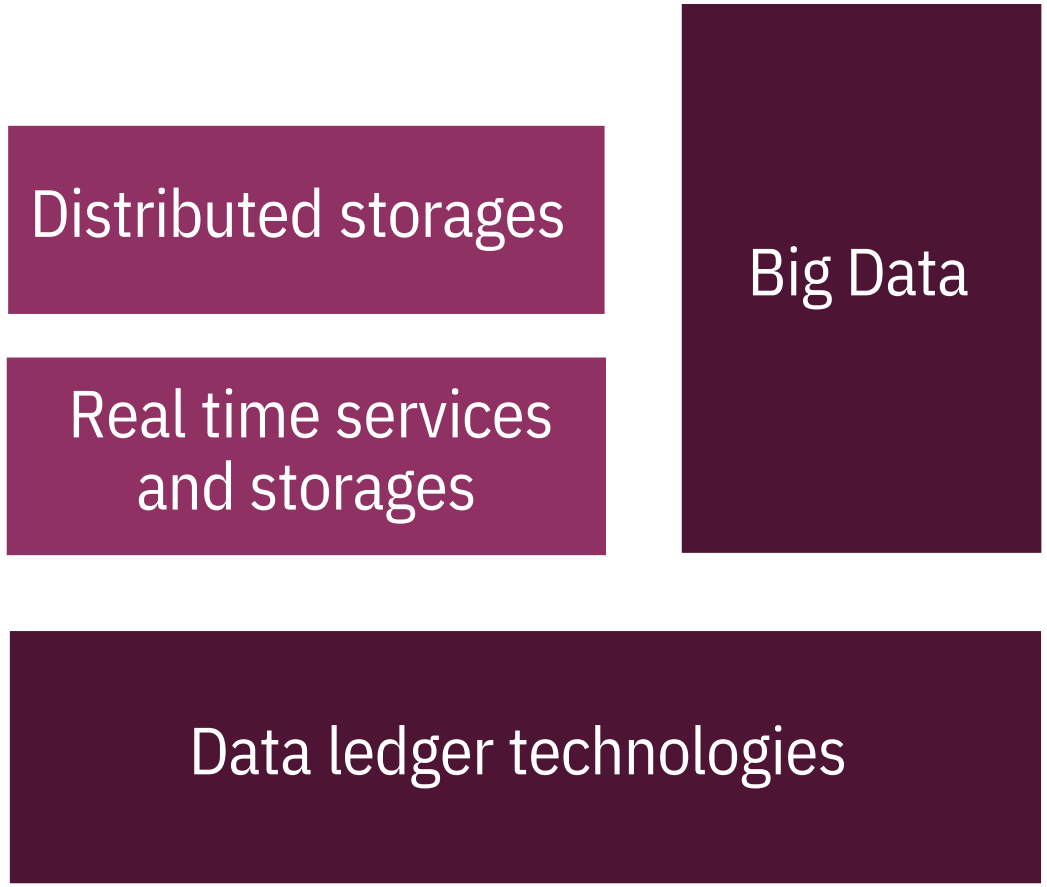
DATA STORAGE ARCHITECTURES

Architectures and platforms for Big Data management:

- Cloud data storage
- NoSQL storage

- In-memory databases (NewSQL)
- Data streaming technologies

- Blockchain technology
- Smart contracts
- IPFS, Swarm



ACID vs BASE DATA SYSTEMS

ACID model

- (A) Atomicity
  - Each transaction is treated as a single "unit"
- (C) Consistency
  - The data cannot leave the database in an inconsistent state
- (I) Isolation
  - At the moment of execution, one transaction has no impact on another
- (D) Durability
  - Once a transaction has been committed, it will remain committed even in the case of a system failure

ACID vs BASE DATA SYSTEMS

BASE MODEL

- (BA) Basically available
  - The system is guaranteed to be available in event of failure
- (S) Soft-state
  - The state of the data could change without application interactions due to eventual consistency
- (E) Eventually consistent
  - Data may be temporarily inconsistent

- The primary task of BASE systems is

  AVAILABILITY

  - In a distributed database, it means the resistance to failure of one node -"partition tolerance"

- Allow data to be stored at any time, even if the data will not be synchronized for some time
- Loosen up the rules
  - We can produce a report even if the system knows that not all data is synchronized
- An optimistic approach
  - They anticipate that the data will be consistent over time

COMPATIBILITY OF APPROACHES

- It is possible to combine the approaches
  - Use ACID model where continuous consistency is essential
  - Use BASE model where system availability and speed are more important
  - Some solutions allow changing between ACID / BASE settings

RELATIONAL DATABASES

- Enable REPLICATION of data
  - Copy of the entire DB
  - Negatively affects writing operations
  - Improves reading speed (reading from secondary systems)

- Master-slave architecture

- What happens in a distributed environment?
- How do we achieve *horizontal scalability*?

  E. g. transfer of funds between banks on different continents

  - Requires extensive resource locks
  - Providing multiple copies of the same data

RELATIONAL DATABASES – BENEFITS AND DRAWBACKS

- It is easy to create views that combine data from a large number of tables
- Determining the start and end point of a transaction and committing the transaction is easy
- All tables must be on the same server for queries to be effective -scalability limitation
- Transactions slow down the system
- It is easy to set business rules and enforce them when creating tables
- It is possible to specify data access authorizations
- It is difficult to support variable data and data that contains exceptions
- **Weak support for unstructured data**

RELATIONAL DATABASES – BENEFITS AND DRAWBACKS

- Relational systems based on a single processor do not grow with business requirements
  - Fast and efficient capture of large amounts of data!!

- Relational systems cannot keep up with rapid business changes
  - E. g. storing some additional data for a smaller number of customers
    - If we expand the table, we get a large amount of “empty” records from other customers
    - If we do not expand the table, we get a join with another table
    - What happens to availability and responsiveness during ALTER TABLE?

TABLE JOINS PERFORMANCE

- Table joins become complicated with a large number of tables
- Querying from several nodes
  - Transferring large amounts of data to a node that aggregates data
  - Mitigating using data replication–but RDB is not supporting automatic synchronizations
    - Sometimes the responsibility is even transferred to the application layer

AVAILABILITY OF RELATIONAL DB

- Data replication
- Logs

- Recovery
  - Loading a backup
  - Import logs

NOSQL DATABASES

- Non-relational oriented architecture
- Scale-out scalability
- Efficiency
- Availability
- Structural flexibility
  - Support for unstructured data without schemas

Key-value	Document	Wide-column	Graph DB	Multi-model
redis	mongoDB.	cassandra	Neo4j	ArangoDB
riak	CouchDB	APACHE HBASE	INFINITEGRAPH	DATASTAX

NOSQL DATABASES

- „Not only SQL“
- Not denying SQL or relational databases!!
- Data servers are not based on relational databases
- They are not scheme-based – they allow data storage and querying without creating an E-R model
- Support multiple formats(graphs, columns, documents)
- Not based on JOIN instruction(access to data without JOIN instruction)
- Run on multiple processors (default operation in multiprocessor environments)
- Use common computer components (widely available processors with separate memory and disk)

SCALABILITY OF NOSQL STORAGE SOLUTIONS

- Horizontal or scale-out scalability
- Partitioning of data–SHARDING

SHARDING

- Parts are shards
- Shards are stored into a larger number of computers (clusters)
- Each shard may have a copy of a particular part of the other shard
- By increasing capacity, we have as little impact on availability as possible
- Allows distribution of reading/ writing operations between systems

KEY-VALUE STORES

- Allow to assign a value to each key
- Not necessary to have a schema
- Key must be specified and is usually a string
  - Any text, logical path to file, REST service address, SQL queries
- The value is usually an object of any size (BLOB), it can be stored in the form of different data types

KEY-VALUE STORES

- Search is enabled only by key and not by value!
  - Although they cannot search by values, they can understand the datatype of value (eg.string, list, etc.)
  - They have a built-in simple query language (only three functions are supported: insert, select and delete)
  - There is an API –less platform dependent
    - Lower cost of replacing the selected solution It
    - simplifies ensuring reliability as well as scalability
- Storage is indexed by key, so searching by key is very efficient
  - Almost insensitive to storage size
- Ensuring reliability and scalability!

KEY-VALUE STORES

- No need for structures / schemas
- No JOINs
- No primary, foreign keys
- Simple -very fast
- Highly scalable -almost linear with data size
- Portable because of simplicity

KEY-VALUE STORAGE USE CASES

- Session management on a large scale
  - Multi-player Online Gaming
  - User session details and preferences
- In-memory data caching to speed up applications by minimizing reads and writes
- Real-time recommendations and advertising

KEY-VALUE STORAGES

- Twitter, GitHub, Weibo, Pinterest, Snapchat, Craigslist, Digg, StackOverflow, Flickr

- weather.com, Uber, Booking.com, BestBuy, Comcast Xfinity, CNN.com, Bet365

GRAPH DATABASES

- Sequence of nodes and edges (connections) that together represent a graph
  - Nodes, relationships, properties
- „triple stores“ because of structure node-connection-node

- Useful for all problems where there are many complex connections between objects
- Used in applications
  - That analyse the relationships between objects or
  - require node traversal

GRAPH DATABASES

- Relationships are visualized by graphs
  - Use-cases
    - Social relations between people
    - Routes on the path
    - Network topology
- Nodes typically represent real-world objects —e.g. nouns
  - People
  - Organizations
  - Phone numbers
  - Websites
  - Computers

GRAPH DATABASES

We can perform complex queries that focus on the relationships (connections) between objects (nodes)

- What is the shortest path between two nodes in a graph?
- Which nodes have neighbors with certain properties?
- If we choose two nodes in a graph, how similar are their neighbors?
- What is the average interconnectivity between the different points of the graph?

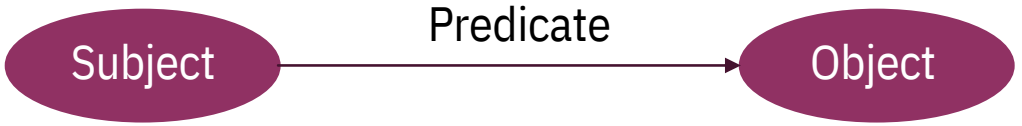
- Graph DBs understand that two nodes with the same data represent the same node
- With the help of internal identifiers, they can combine entire networks with each other
- Queries are extremely efficient and fast
  - Small node size
  - Built-in caching-reduce the number of I / O operations

GRAPH DATABASES SCALABILITY

- Complicated scalability due to node connections
  - Scalability can be achieved in reading and querying
  - Writing requires complex implementation
- Graph DBs vary depending on the terms used and the types of queries they support

IDENTIFYING NODES

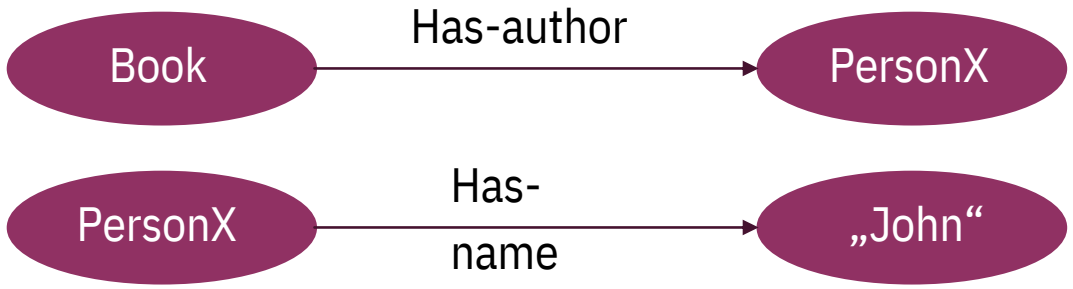
- Using triples



- The terms 'source', 'link' and 'destination' may differ
- In RDF: subject-predicate-object

IDENTIFYING NODES

- Triple describes the fact



- Descriptions are independent and can be stored in different databases
- If the URI of the subject Person is the same, we can determine that the author of the book is "John"

IDENTIFYING NODES

- Triple describes the fact

- Then we can answer the question: "Does the book have any author named John?„
- Normally, nodes are also linked with metadata (dates, authorizations, etc.)

GRAPH DATABASE USE CASES

- Connections analysis (social networks, phone calls, e-mail)
  - Crime detection in call analysis
  - Money laundering detection (Panama papers, Pandora papers)

- Advantages
  - NoSQL solutions for data with many connections

They are not suitable for data that does not have connections

They require enough memory to be able to store all connections during graph analysis

We can use document storage and graph storage (ArangoDB) at the same time

GRAPH DATABASES

- eBay, Walmart, Airbus, Boeing, JPMorgan, UBS, Volvo, Daimler, Toyota, Marriott, AccorHotels, Orange, AT&T

- X

- Target, Red Hat

WIDE-COLUMN STORAGES

- Important NoSQL solutions because they are scalable for large amounts of data
  - Designed for parallel processing of large amounts of data in a distributed network
  - Support for MapReduce and Hadoop
- They combine a row and a column that can be used as a key
- The RDB must read the entire row, although sometimes only certain information from that row is needed (e.g. listing all student names)
- Column storage automatically creates vertical partitioning of tables –it cuts an X-column table at the physical level into X tables with a single-column
- There is no visible difference for the user, since the same principles apply for RDBs

WIDE-COLUMN STORAGES

- Store data in columns instead of rows

Id	Name	Surname	Address
1	John	Mark Win	Unknown 1
2	Joan		Unknown 2

Id
1
2

Name
John
Joan

WIDE-COLUMN STORAGE ADVANTAGES

- Stores the data of one column in the same location on disk
  - Same way as relational store rows together
- Improved analytical queries using aggregation functions
- Support the SQL interface for data access
- Significantly reduces the time required to read data such as:
  - All titles, all surnames, average grade, oldest citizen, etc.
- High data compression
- Tables with an unlimited number of columns
- High scalability and availability in distributed systems
- Not based on joins -not normalized tables
- Built-in automatic failover

WIDE-COLUMN STORAGE DISADVANTAGES

- Slow inserts
- Slow updates and deletes
- Not adequate for frequent queries of a type:
  - All information about a particular student
- Not adequate for small amounts of data
- Require at least 5 nodes
  - Due to availability most replicate data on at least 3 nodes

WIDE-COLUMN STORAGES – USE CASES

- Mainly for analytical queries
- Business intelligence systems
  - OLAP solutions
- Data warehouses
- Systems that require huge tables and many rows
- GIS systems
  - GIS of buildings in which each row is latitude and each column is longitude

    Not every m2 of space has a building -so a lot of data will be blank RDBs are not efficient when storing a lot of empty values

CHARACTERISTICS OF WIDE-COLUMNS TORAGES

- Scalability
  - In a well-designed system, a linear proportion can be achieved between the amount of additional data and the amount of new nodes in the cluster
  - The ability to distribute the query across nodes because no JOIN operation is required
    - By carefully determining row and column keys, the system has enough data for retrieving data directly from the node where it is stored

CHARACTERISTICSOF WIDE-COLUMNSTORAGES

- Effective communication
  - Lower replication costs
  - Arbitrary distribution of data by nodes, so it is possible to set up a system with high availability
- The scheme is not defined in advance, but:
  - We need to know the groups of columns (families) in advance!
    - Row and column keys can be arbitrary and added at any time
- They are not suitable for small amounts of data
  - They need at least 5 nodes
    - Due to availability most replicate data on at least 3 nodes

WIDE-COLUMN STORAGES

- Netflix, Apple, AT&T, Facebook, eBay, Coursera, Bank of America, PayPal, PBS, Yahoo, Intel, IBM, Visa

- Cognite, Moloco, YouTube, Gmail, ostale Google aplikacije

- Airbnb, Alibaba Group, Adobe, Pinterest, Sears, Yahoo

DOCUMENT STORAGE

- Flexible, powerful and advanced NoSQL solutions
- Key-value storages do not support searches by value, while document storages have the opposite approach
  - The key may exist
    - It is not necessary that it is visible or to use it

DOCUMENT STORAGE

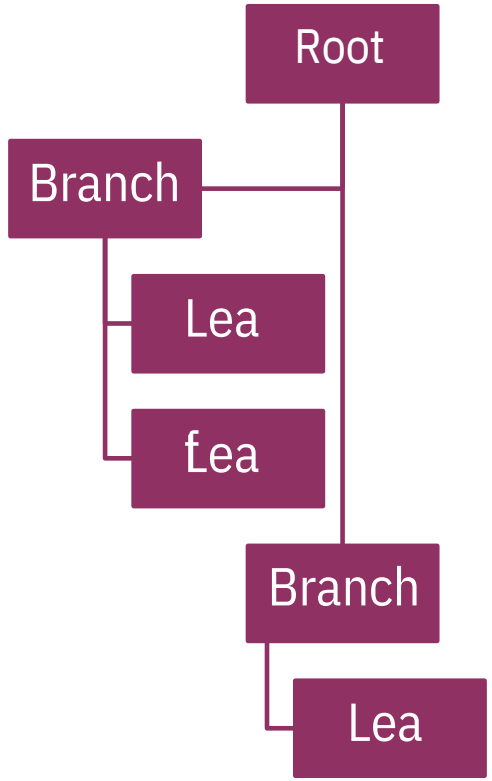
- A document storage can fast find a subset of documents by value
  - Without reading all the documents
- Example
  - If we want to print one page of a book, we do not have to load the entire book into memory

DOCUMENT STORAGE

- They enable the storage and processing of partially structured and unstructured data Using XML,
- JSON or other formats Access usually with HTTP and RESTful API
- The entire document represents one record
- Documents can have a different schema
  - Nevertheless, we can make indices
  - Nevertheless, the API can remain simple

DOCUMENT STORAGE

- They have a tree structure
  - The data is usually stored as tree leaves
  - Similar to XML and XML navigation



DOCUMENT COLLECTIONS

- Most repositories combine documents into collections
  - They are similar to the directory structure on a file system
  - To navigate the document hierarchy
  - To logically combine similar documents
  - To store business rules
- Collections may contain other collections and trees may have subtrees

DOCUMENT STORAGE

- Example A

```
{
    „StudentNumber“ : „1276172“,
    „Name“ : „John“, „Surname“ :
    „Mark“,
}
```

- Example B

```
{
    „ StudentNumber“ : „78237382“,
    „Name“ : „Merry“,
    „Surname“ : „May“,
    „Address“ : {
        „Street“ : „Unknown4“
    }
}
```

DOCUMENT STORAGE

- The document identifier can be specified separately from the document, when inserting and reading the document
- Good practice
  - Each document should have an internal identifier that is part of the document

DOCUMENT STORAGE - ADVANTAGES

- No schema
  - Especially in online solutions, where the content changes over time
- Depending on the solution, in some cases it is possible to partially change the document
- Since there is no concept of tables, the search is simple
  - Remember - even though there is no structure, we can determine the indices!

DOCUMENT STORAGE

- Bosch, Telefonica, GAP, Nokia, eBay, Adobe, Cisco, SAP, Facebook, Paypal, Business Insider, Shutterfly

- IBM, GrubHub Inc

- Thomson Reuters, Oxford University

WHY NOSQL?

- Do we use it only for scalability issues?
- It has also many other interesting features:
  - Not necessary usage of schema
  - Shorter development time (notSQL!)
  - Velocity
    - If responsiveness is important (also for mobile devices!) The difference is already between milliseconds and a few tens of milliseconds
  - Future scalability
    - If we run into problems, we know in advance whatwillbethe solution

NOSQL IS NOT A SOLUTION FOR ALL SCENARIOS

- Banks?
- Stock market?

- In some cases, a system that does not support ACID properties is not acceptable.

- NoSQL does not solve all problems and does not represent a universal solution for data storage.

EXAMPLE: NETFLIX

Challenges of managing data:

- Large volume of data
- Diverse data types
- Real-time requirements

- Highly scalable architecture
- Support for multiple data centers and data replication in several directions
- Linear performance
- Transparent fault detection and recovery
- Flexible, dynamic schema
- Ensured data security
- Adjustable data consistency

NETFLIX

cassandra

Amazon EC2

Back-end system

DATABASE ARCHITECTURE OPTIONS

Traditional relational DB

- SQL query language
- ACID model
- Limited horizontal scalability
- Strict schema

NewSQL databases

- Taking advantage of hybrid architecture capabilities
- Fast, scalable and consistent

NoSQL databases

- Renounce SQL and ACID model
- Based on BASE model, horizontal scalability
- Lack of consistency

DBMS POPULARITY RANKING

Rank			DBMS	Database Model
Oct 2024	Sep 2024	Oct 2023		
1.	1.	1.	Oracle	Relational, Multi-model
2.	2.	2.	MySQL	Relational, Multi-model
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model
4.	4.	4.	PostgreSQL	Relational, Multi-model
5.	5.	5.	MongoDB	Document, Multi-model
6.	6.	6.	Redis	Key-value, Multi-model
7.	7.	↑ 11.	Snowflake	Relational
8.	8.	↓ 7.	Elasticsearch	Multi-model
9.	9.	↓ 8.	IBM Db2	Relational, Multi-model
10.	10.	↓ 9.	SQLite	Relational

Source: db-engines.com

KEY ELEMENTS FOR CHOOSING NOSQL STORAGE

- **Key-value storages**
  - Persistent data sharing between multiple processes or micro services in an application
  - A simple schema
  - High read / write speed without frequent updates
  - No complex queries involving multiple keys or joins
- **Graph databases**
  - In-depth relationship analysis for proximity calculation, fraud detection
  - Complex queries to determine the relationships between data points
  - Requirement to detect patterns between data points
  - Storing the properties of each data point and connections between them
- **Wide-Column storage**
  - Collecting a very large amount of data for analytics
  - Extreme writing speed and lower reading speed
  - Retrieving data by columns using a row key
  - No direct querying patterns, complex indices or high-level aggregates
- **Document storage**
  - Flexible schemawith complex querying
  - JSON / BSON or XML data formats
  - Complex indices
  - High performance and balanced writing-reading ratio

WHICH DATA STORAGE TO USE?

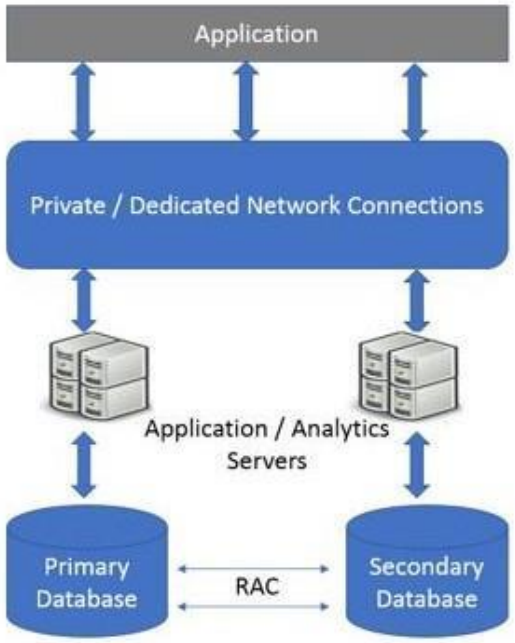
- **What is the data structure?**
- **How will we access the data?**
- **What is the ‘temperature’ of data?**
- **How much will the solution cost?**

- Fixed schema→SQL, NoSQL
- No schema → NoSQL, search index
- Key/value → In-memory DB, NoSQL
- Graph → Graph DB

- Simple connections → NoSQL
- Table join, transactions, SQL →SQL
- Graph traversal → Graph DBs

DATA STORAGE IN CLOUD PLATFORM

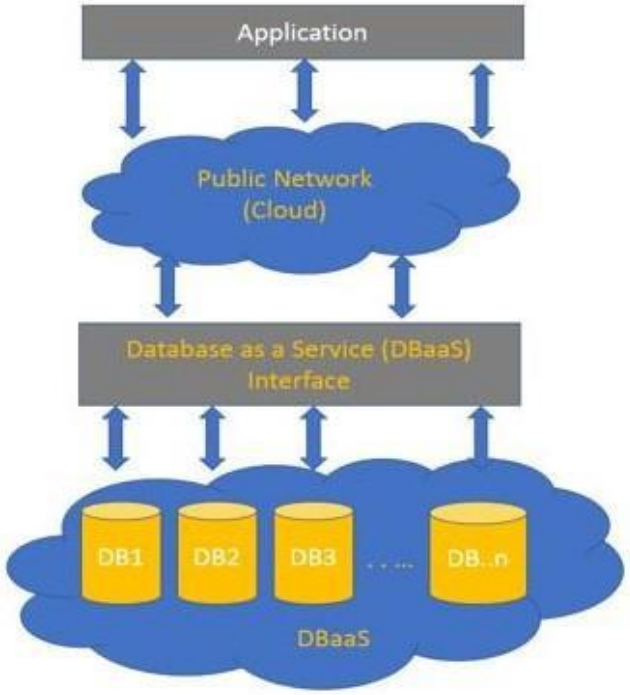
Traditional databases



- On-premise Database Servers
- Application/ Analytics
- Servers to host the applications
- Dedicated or private connections to connect to the database
- Company owns all the servers
  - Must bear the entire cost of procurement, installation and maintenance of the database

DATA STORAGE IN CLOUD PLATFORM

Cloud databases(DBaaS)



- Hosted by various service providers
- Available over Public Cloud Network
- Company rents a service
  - Saves on the procurement, installation and maintenance cost of the database

DATA STORAGE IN CLOUD PLATFORM

Key features

- A database service built and accessed via a cloud platform
- It provides many of the same features as traditional storage with the added flexibility of cloud computing
- Users install software on a cloud infrastructure to implement the database
  - No purchase of dedicated hardware
- Can be managed by the users or by the provider (managed and offered as a service)
- Support for relational databases and NoSQL data storages
- Access via a web interface or provider‘s API

DATA STORAGE IN CLOUD PLATFORM

ADVANTAGES of using Cloud platforms

- 70 % faster movement to market
  - The service is available -there is no delay in the delivery and deployment of infrastructure
  - The company uses the service directly and hosts its applications with the provider
- 80 % more applications launched
  - Multiple applications are available hosted on the cloud–we take advantages of the applications availability
- 75-85 % reduction of infrastructure costs
  - All costs (infrastructure, networking costs, maintenance) are taken care of by the cloud service provider–the company only uses the services

DATA STORAGE IN CLOUD PLATFORM

ADVANTAGES of using DBaaS/ Cloud database

- Ease of access
  - User can access from anywhere using a provider‘s API or web interface
- Scalability
  - Storage capacities can be adjusted/ expanded on run-time to accommodate changing needs(during Peak times, ahead of deadlines)
    - Highly scalable – near infinite data storage capacity
      - The need to manage large amounts of data
- Disaster recovery
  - In the case of a natural disaster, device failure or power outage, the data is kept secure through backups on remote servers

DATA STORAGE IN CLOUD PLATFORM

ADVANTAGES of using DBaaS/ Cloud database

- Hardware independance
  - Cloud service provider cover the maintenance and infrastructure aspects
- Cost effectiveness (Value for money)
  - Companies are not worrying about operational costs and expensive upgrades
  - Multiple configurations available
    - Pay only for what you actually use and for the time you use it
- The latest technology available
  - Companies are not preoccupied about purchasing new technologies and upgrading infrastructure, There is no need to hire dedicated staff
- Security
  - Service provider takes care of the security aspect and invests in the best available solutions

DATA STORAGE IN CLOUD PLATFORM

DISADVANTAGES of usingDBaaS

- No direct access control to the database
  - If something goes wrong, we are helpless
- No control on the physical safety of the servers
  - In the case of a natural disaster at the server location or system failure, we must face a downtime– if not data loss
- No control over sensitive data
  - We depend on the cloud database server management
- Cost effectiveness for small or mid-size DBs?
  - Maintaining their own DB servers (DB size< 1 TB) become cost-effective

DATA STORAGE IN CLOUD PLATFORM

Cloud database environment models

- Traditional cloud model –virtual machine deployment
  - Company purchase virtual machine instances from a CSP
  - DB run on the cloud
  - Company responsible for DB control and management

- Database-as-a-Service(DBaaS)
  - Company purchase access toa database service
  - DB runs on the service provider‘s infrastructure, which is responsible for any failures
  - Company can focus on operations, development and business goals

DATA STORAGE IN CLOUD PLATFORM

Comparison between on-premises traditional DB and cloud DB

Measures	On-Premises database	Cloud database/ DBaaS
Reliability	Reliable and private	More reliable but not necessarily private
Scalability	Limited scalability	Unlimitedly scalable
Speed	Faster, but may fail in any point of time (hardware failure)	Faster and will be up always
Deployment	Takes time	Within no time
Cost effectiveness	Lots of capital required	Pay only for what you use, highly cost effective, no overhead cost involved
Maintenance	High cost(all cost to the company)	No cost
Setup cost	Covered by company	Covered by provider, company only pays for the service
Security	Highly secure and controlled	Highly secured as per provider

DATA STORAGE IN CLOUD PLATFORM

DATA STORAGE IN CLOUD PLATFORM

Amazon Web Services (AWS)

- Market leader in the DBaaS space
- It offers supplementary data management services
  - Redshift – data warehouse

  - Data Pipeline –data integrating service for easier data management

ADVANTAGES

- Lots of features, easy to use, good support and documentation

DISADVANTAGES

- Not to customizable, downtimes as per Amazon‘s schedule

- RelationalDB service(RDS)
- Runs on
  - Oracle, MS SQL Server, MySQL, PostgreSQL, MariaDB

- NoSQL DB
- High availability, data durability
- Automatic replications of workloads

- NoSQL DB (key-value, document)
- High efficiency, unlimited scalability
- Automatic replications of workloads

DATA STORAGE IN CLOUD PLATFORM

MS Azure

- Cloud computing platform for VM creation, building and running web-based applications, smart client applications, and XML web services
- It has the biggest and strongest global infrastructure with 55 regions
- Offers the biggest range of software
  - It allows creating an ecosystem that has the same roots and a single platform to solve problems and issues

ADVANTAGES

- Comprehensive solution, good security, strong ecosystem

DISADVANTAGES

- Not the best customer support, not user friendly

Supportfor MS SQL Server, PostgreSQL, MySQL, MariaDB

Supporting relational data model

NoSQL DB service with open source APIs fo
MongoDB and Cassandra

DATA STORAGE IN CLOUD PLATFORM

Google Cloud Platform

- More and more businesses of different sizes are adopting its solutions
- Broad open-source compatibility allows to scale while doing more with analytics and integrations

ADVANTAGES

- Comprehensive documentation, good for small and big companies

DISADVANTAGES

- Not yet at the level of big three providers (AWS, Oracle, Azure)

Supportfor MySQL in PostgreSQL

Google Cloud SQL

Supporting relational data model

Cloud Spanner

NoSQL DB

Cloud Datastore

Cloud Bigtable

Data warehouse in cloud

Google BigQuery

DATA STORAGE IN CLOUD PLATFORM

MongoDB Atlas

- Popular open-source NoSQL database
- powerful scaling, sharding and automation capabilities
- Available in AWS, MS Azure, Google Cloud Platform

mongoDB. Atlas

ADVANTAGES

- Strong support community, quick installation, flexibility

DISADVANTAGES

- NoSQL only, challenging for new/inexperienced developers