Mahidol University
Wisdom of the Land

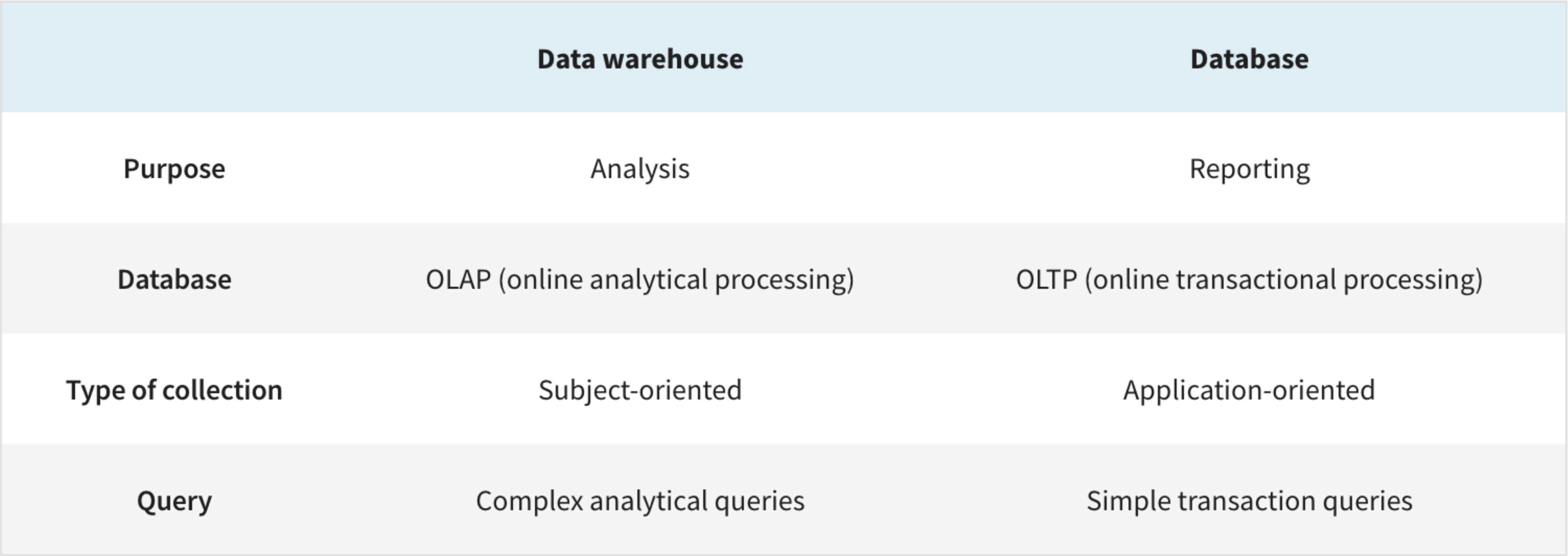
BIG DATA PROCESSING

EGCI 466

File vs DBMS

- File
  - Data redundacy
  - Inconsitency
  - Isolation
  - 1 task/ program
  - Data integrity
  - Atomicity of updates

Database vs Datawarehouse



	Data warehouse	Database
Purpose	Analysis	Reporting
Database	OLAP (online analytical processing)	OLTP (online transactional processing)
Type of collection	Subject-oriented	Application-oriented
Query	Complex analytical queries	Simple transaction queries

DBMS: Advantages

- Declarative query languages
- Data independence
  - Separate itself from applications
- Efficient access through optimization
  - The system automatically finds and efficient way to access data

DBMS: Advantages

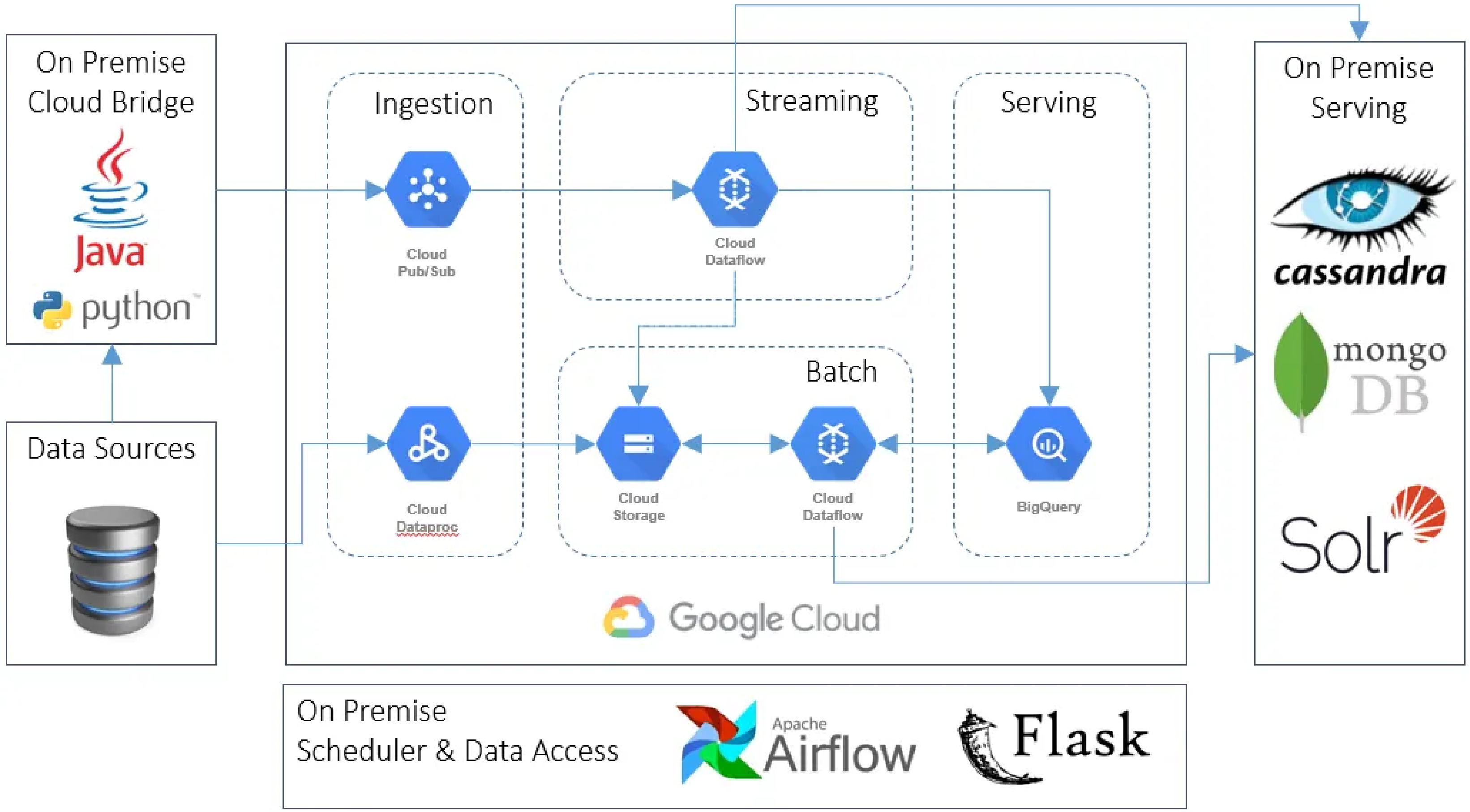
- Data integrity and security
  - keep accuracy and consistency
  - Failure recovery
- Concurrecy
  - Many people can simultaneously access datas without conflict

Types of DBMS

- Relational Database (ACID)
  - MySQL
  - Postgred

- Non-Relational Databases (BASE)
  - Amazon dynamo
  - Mongo DB
  - Cassandre
  - Neo4j

Database in the flow



Database to use

- Consistency
- Structured data (Fixed schema)
- Transaction
- Joins

RDBMS & non-RDBMS (No-SQL)

- High performance
- Unstructured data (Flexible schema)
- Availablity
- Easy Scalability

No-SQL not a de facto replacement of RDBMS

RDMS v.s. non-RDMS

- Data driven to Query driven data model
  - RDMS: Data integrity, relationship
  - non-RDBMS: starts from query, not from the data.

- Normalization to Denormalization
  - RDBMS: Start from normalization → query
  - non-RDBMS: Query based data

- ACID to BASE
  - Consistency vs. availability

ACID and BASE data model

Different consistency models

ACID : Relational Database

- Atomic
- Consistent
- Isolated
- Durable

BASE : Non-relational Database

- Basically Availability
- Soft State
- Eventual Consistent

ACID

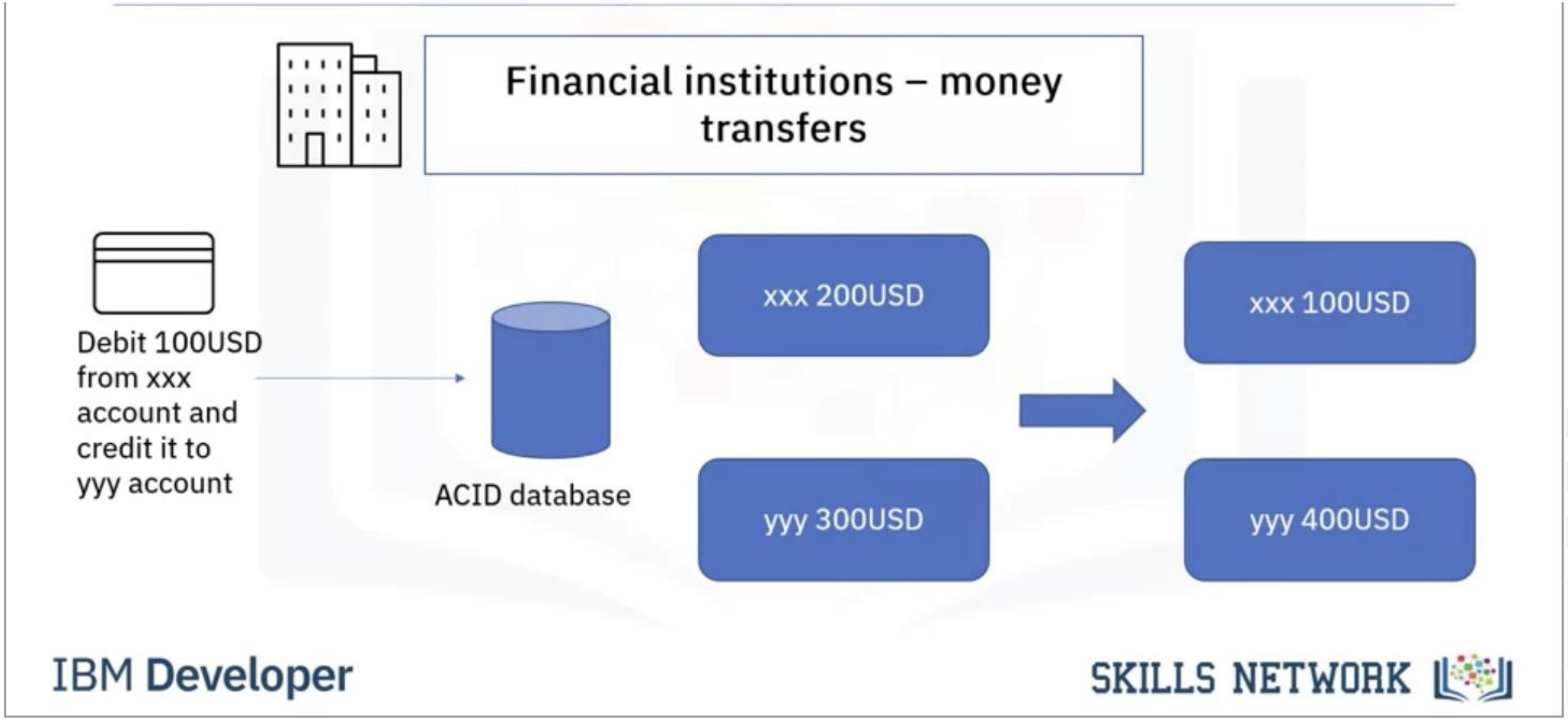
- Atomic
  - All operations must succeed
  - If not -> roll back
- Consistent
  - After each transaction all data must be consistent
- Isolated
  - Do not interfere while the other operation is in progress
- Durability
  - The data remain permanently in the database until the next update

ACID CONSISTENCY MODEL

- Ensures a performed transaction is always consistent
- Used by
  - Financial institutions
  - Business transaction data
  - Data warehousing
- Work well with many samll simultaneous transactions

- Fully consistent system

EXAMPLE



Financial institutions – money transfers

Debit 100USD from xxx account and credit it to yyy account

ACID database

xxx 200USD

yyy 300USD

xxx 100USD

yyy 400USD

IBM Developer

SKILLS NETWORK

Relational database

ID	Name	SurName	Faculty	Department
5913000	Johny	English	EG	CO
5913001	Abraham	Lincoln	EG	CE
5913002	Sunny	English	PT	PT

Relational database

ID	Name	SurName	Faculty	Department
5913000	Johny	English	EG	CO
5913001	Abraham	Lincoln	EG	CE
5913002	Sunny	English	PT	PT
5913000	Will	Smith	IC	CI

Duplication of the keys

Relational database

ID	Name	SurName	Faculty	Department
5913000	Johny	English	EG	CO
5913001	Abraham	Lincoln	EG	CE
5913002	Sunny	English	PT	PT
EG	Sunny	English	PT	PT

Incorrect Format

Mahidol University
Wisdom of the Land

Relational database

ID (has 7 digits)	Name	SurName	Faculty (has 2 char)	Department
5913000	Johny	English	EG	CO
5913001	Abraham	Lincoln	EG	CE
5913002	Sunny	English	PT	PT
591111	Sunny	English	Science	PT

Constraint restriction

Relational database “Relation”

ID	Grade
5913000	2.5
5913001	3.4
5913002	3.6
5911111	1.0

foreign Key

Relational database

ID	Hobby
5913000	Football, Reading
5913001	Sky Diving, Reading
5913002	Football
5911111	Gaming

Atomic property

Relational database “Relation”

ID	Grade
5913000	2.5
5913001	3.4
5913002	3.6
5911111	1.0

foreign Key

‘Join’ Operation

ID	Name	SurName	Faculty	Department	Grade
5913000	Johny	English	EG	CO	2.5
5913001	Abraham	Lincoln	EG	CE	3.4
5913002	Sunny	English	PT	PT	3.6
5911111	Sunny	English	SC	MA	1.0

Easy and fast join opearation

BASE

- Basically Availability
  - Always available even with multle failures
  - Use highly distributed database management

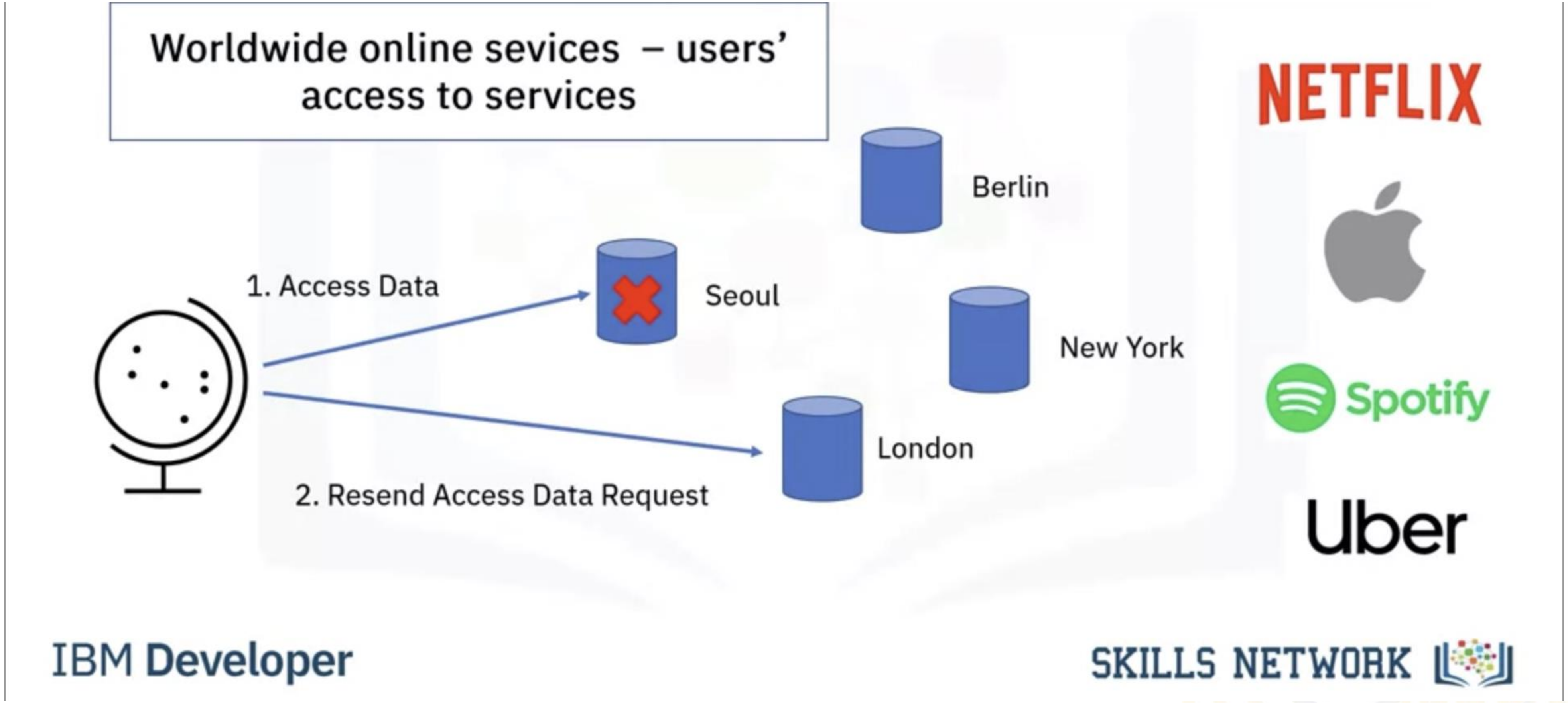
- Soft State
  - Abandom the consistency requirement
  - It can change the overtime
  - Different replica can be inconsistent at some point
- Eventual Consistent
  - Ensure at some point, the data will converge to a consistent state
  - ’No garuntee are made’ DATA READ can be inconsistent at some point

Mahidol University
Wisdom of the Land

BASE CONSISTENCY MODEL

- Few requirements for immediate consistency, data freshness and accuracy
- Benefit: availability, Scale and resilience
- Favour availability over consistency
- Used by:
  - Online services
  - Social Media
  - Marketing and customer wervice company

EXAMPLE



Worldwide online sevices – users' access to services

Non relational databases

- Open-Source
- Scale horizontally
- Share data more easily
- more use-case specific
- more developer friendly

No SQL

- Not only SQL
- Non-relational
- Provide new way of storing and query data
- Design to handle a different breed of scale – ‘Big Data’
- Simpler to develop functionality for than RDBMS

Characteristics

- Mostly open-source
- Build to horizontally scaling
- Share data more easilty than RDBMS
- More case specific than RDBMS
- Flexible scheme

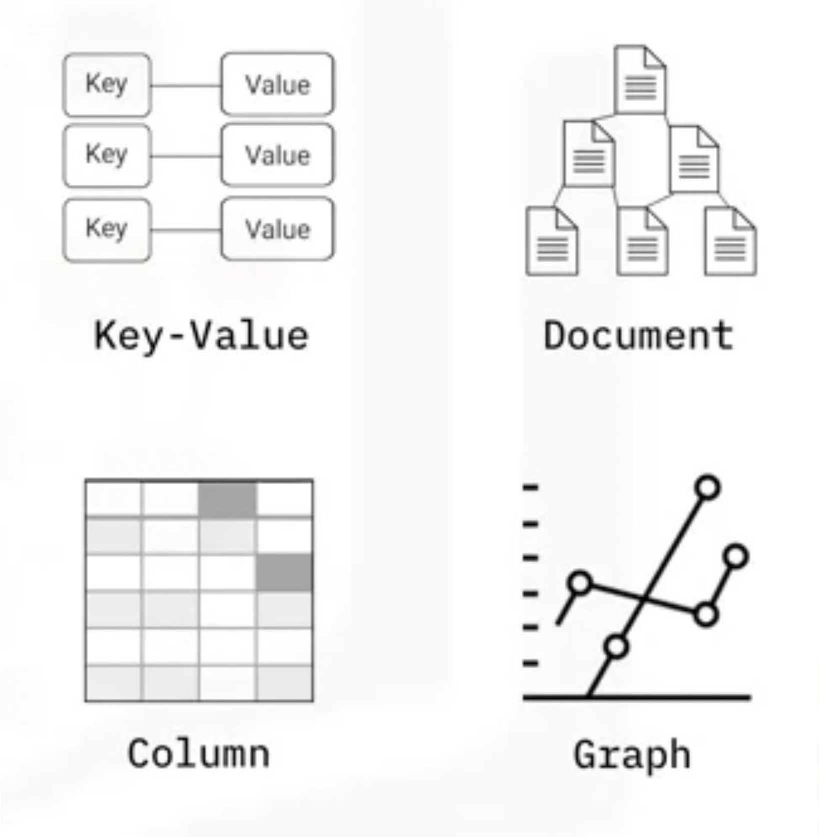
Benefits

- Easy Scalability
- High Performance
- High Availablity
- Easily support cloud Architecture
- Cut cost
- Flexible Schema (No locking)
- Varied Data structures

Mahidol University
Wisdom of the Land

Non relational databases

- Key-value types
- Document type
- Column Type
- Graph style type



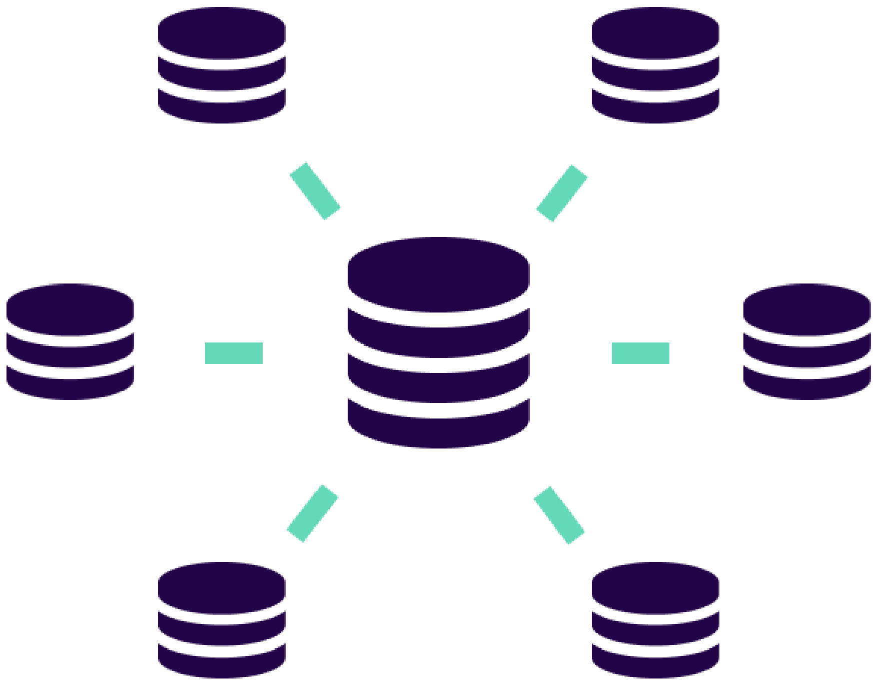
Summary

RDBMS

  - Consistency
  - Structured data(Fixed Schema)
  - Transaction
  - Join operations

- Non- RDBMS
  - High Performance
  - Unstructured data(Flexible Schema)
  - Availability
  - Easy Scalibility

Distributed File System



A collection of multiple interconnected database

Spread physically across various location

Use **Fragmentation** and **replication**

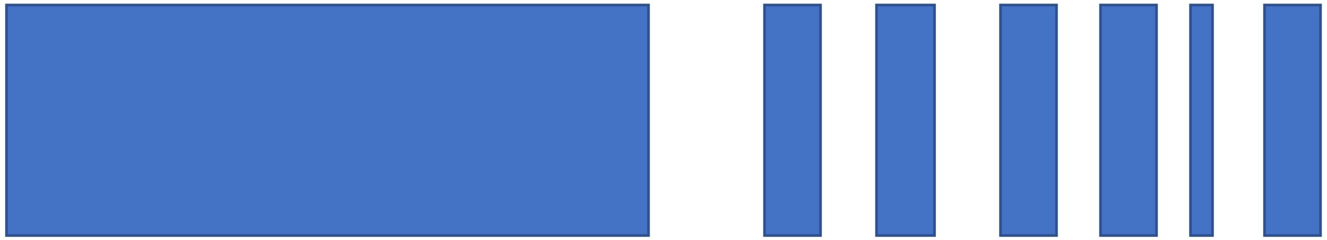
Concept

- Distributed in a different location (physically)
- Each database is an intereconected database
- It uses fragmentation and replication

Fragmentation

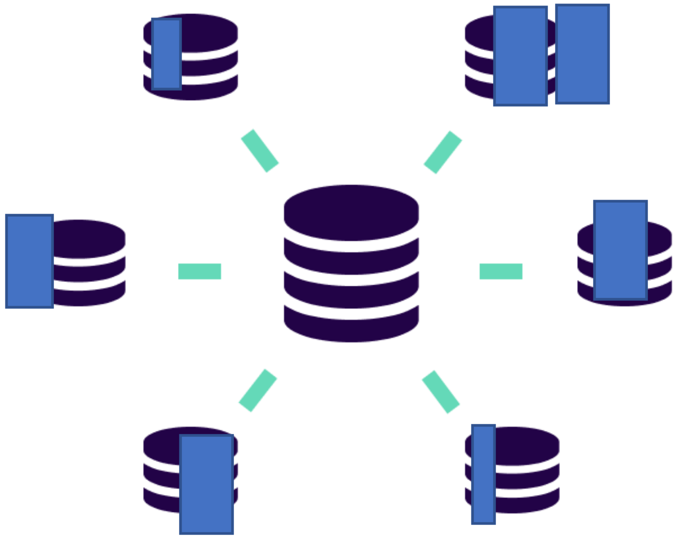
Input Data

Fragmented data



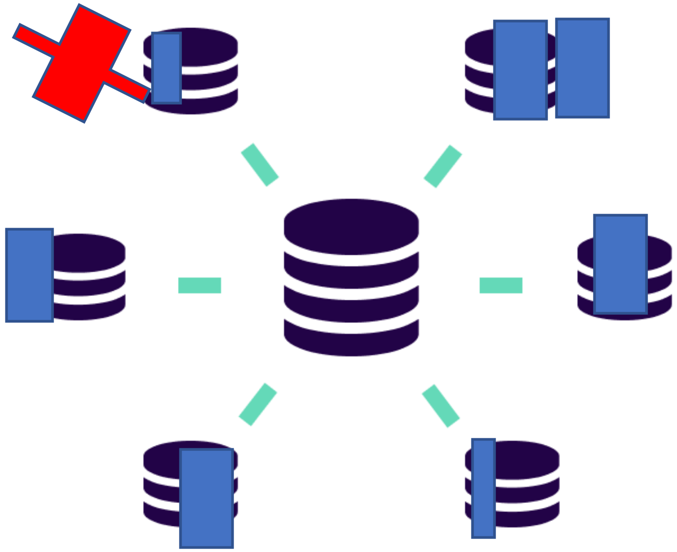
Fragmenting data(partitioning, sharding)

- Fragment lexically
- Fragment by key



Replication

- Protection of node failure
- All data fragments are stored redundantly in 2 or more sites
- Increase availabiltiy
- **require highly synchrnozation
  → inconsistency



Advantages of distributed system

- Reliability and availability
- Improved performance
- Query processing time reduced
- Ease of growth/scale
- Continuing reliability

Challenges

- Concurrency control → consistency of data
  - WRITE/READ to a single nodes →synchornized

  or

  - WRITE to all , READS from some nodes

  or

  - Devloper-driven consistency

BASE system → Eventually consistency

Non relational databases

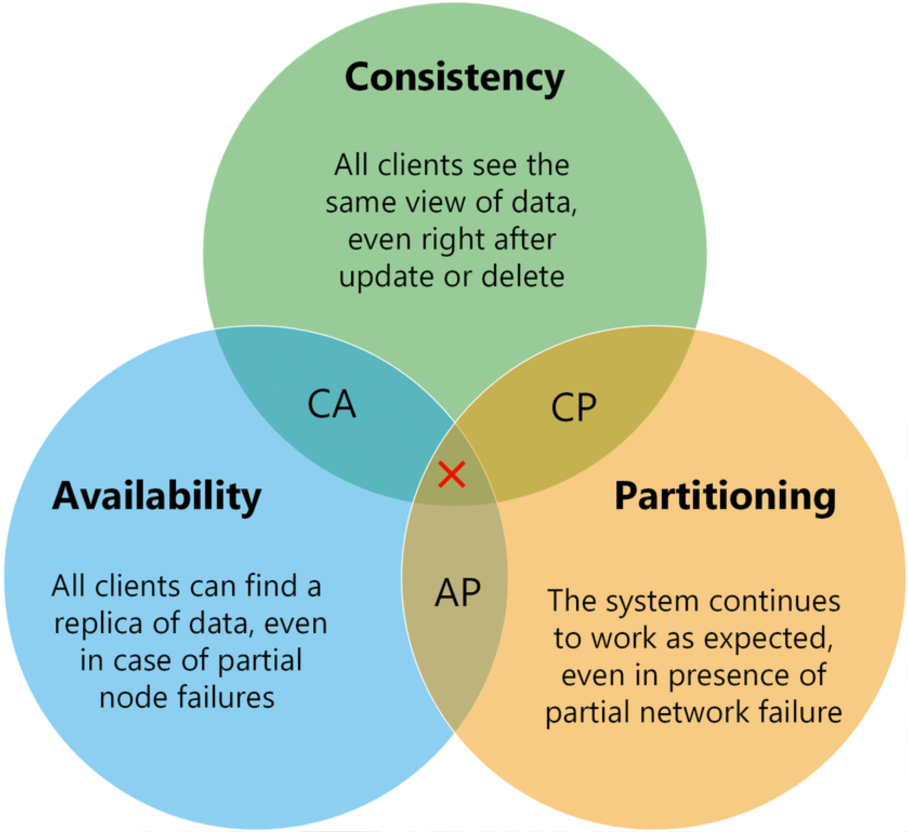
- Open-Source
- Scale horizontally
- Share data more easily
- more use-case specific
- more developer friendly

Non relational databases

- Key-value types
- Document type
- Column Type
- Graph type

CAP Theorems

- CAP thorem (Brewer’s Theorem)
- Evolved by MIT professors Seth Gilbert and Nancy Lynch
- The design can garuntee 2 out of 3 from this theorem


Consistency
All clients see the same view of data, even right after update or delete
CA
CP
Availability
All clients can find a replica of data, even in case of partial node failures
AP
Partitioning
The system continues to work as expected, even in presence of partial network failure

Consistency

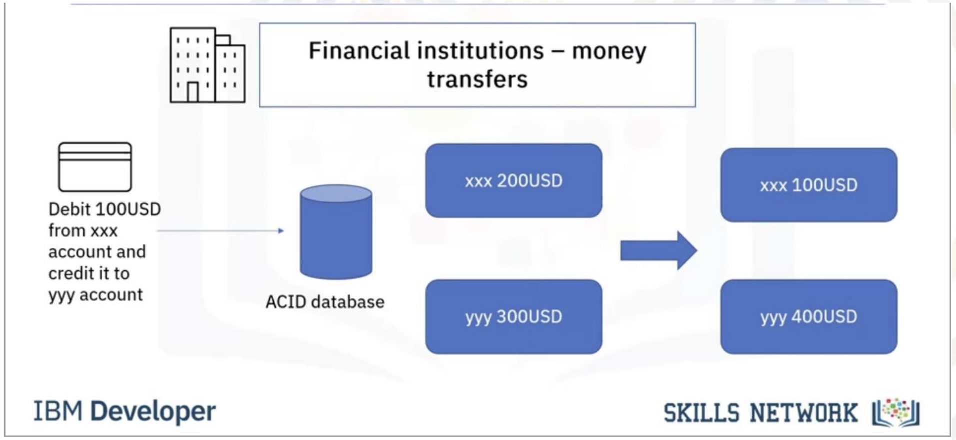
After a write, all subsequent reads will return that updated data

Advantage

- Every node see the same things → Consistency
- Atomic operation

Challenges

- The transaction may be blocked (locked tem
- May slow down the operation



Financial institutions – money transfers

Debit 100USD from xxx account and credit it to yyy account

ACID database

xxx 200USD

yyy 300USD

xxx 100USD

yyy 400USD

IBM Developer

SKILLS NETWORK

Availablility

The system is **always responsive.**

Advantage

- Every node see the same things → Consistency
- Every request receives a response.

Challenges

- The response may be outdated (not consistent)
- May slow down the operation

Partition Tolerance

- The cluster must work despite network issues

Advantage

- The system can **survive failures of links, nodes, or regions** without complete shutdown.
- Distributed systems mostly **<u>require</u>** partition tolerance

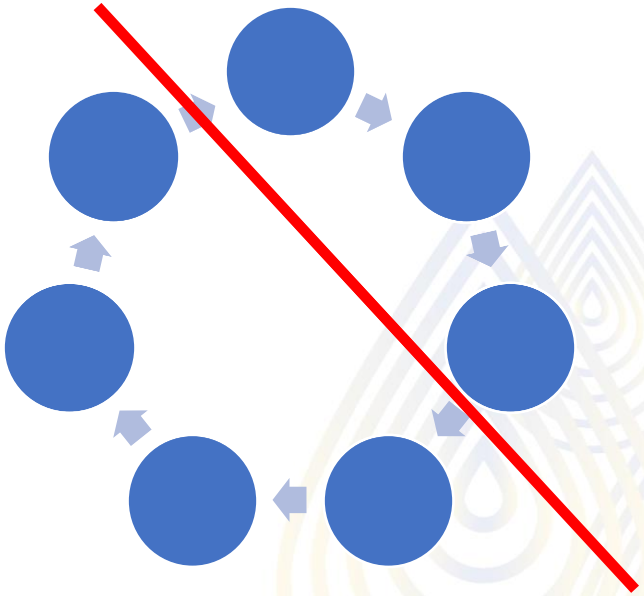
Challenges

- Work with either **availability** (always allow reads/writes) or **consistency** (block until the partition heals).

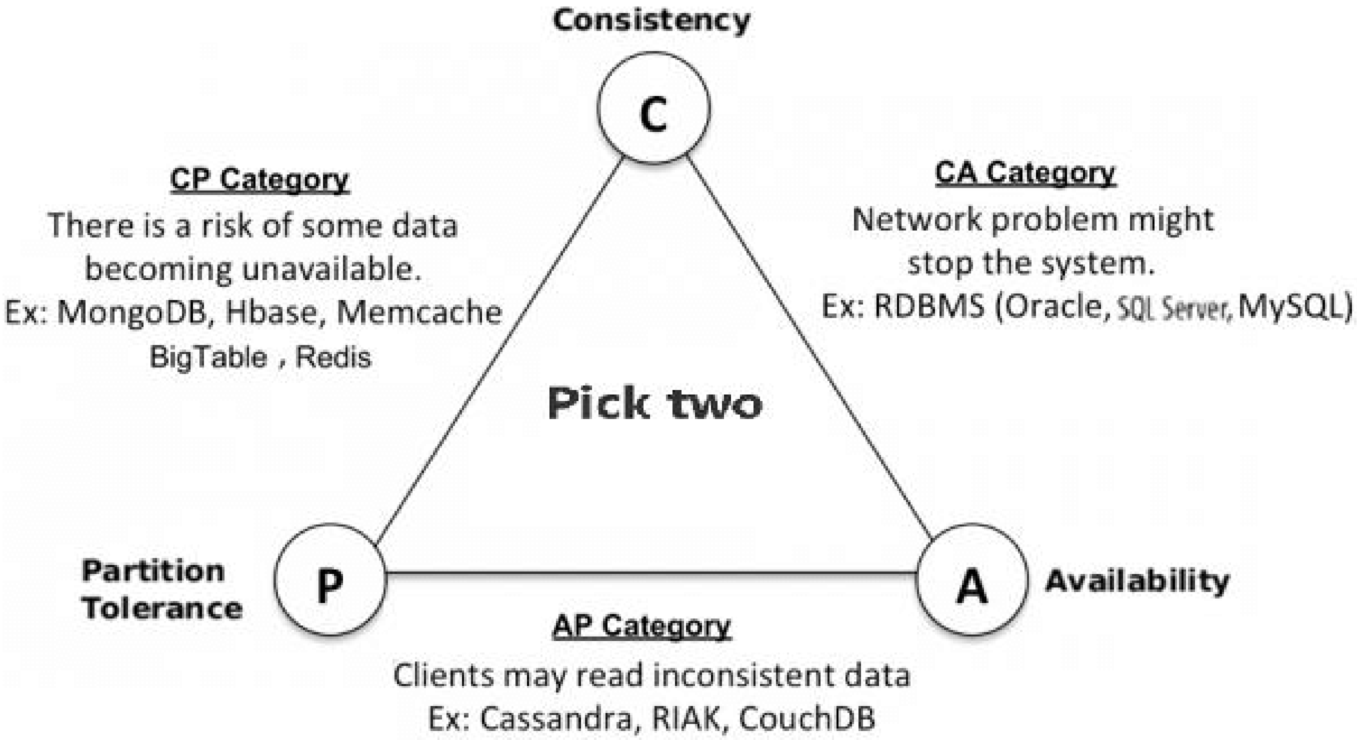
Partition Tolerance

Partition = Network split — like half of your servers can't talk to the other half.

- Increase number of replica



CP or AP



CP Category

There is a risk of some data becoming unavailable.
Ex: MongoDB, Hbase, Memcache
BigTable , Redis

CA Category

Network problem might stop the system.
Ex: RDBMS (Oracle, SQL Server, MySQL)

AP Category

Clients may read inconsistent data
Ex: Cassandra, RIAK, CouchDB

Summary

Database	Choices Made
Traditional RDBMS	Consistency + Availability (CA) (only when no partition)
MongoDB	Availability + Partition Tolerance (AP)
HBase	Consistency + Partition Tolerance (CP)
Cassandra	Availability + Partition Tolerance (AP)