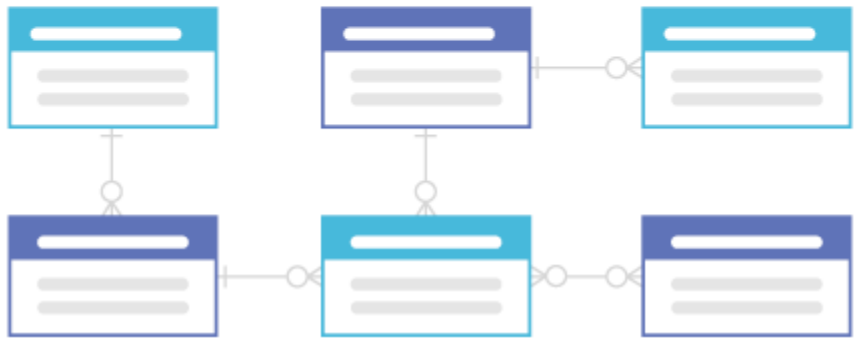
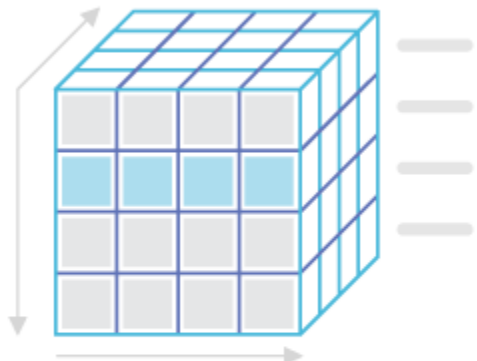

# CAP THEOREM IN DISTRIBUTED DATABASES

SOHRAB CHALISHHAFSHEJANI

## SQL

*Relational Database Management Systems (RDBMS)*

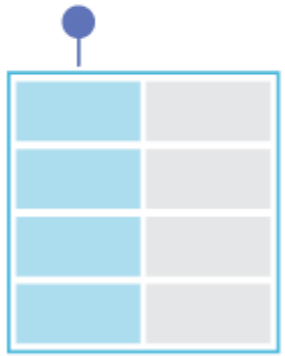




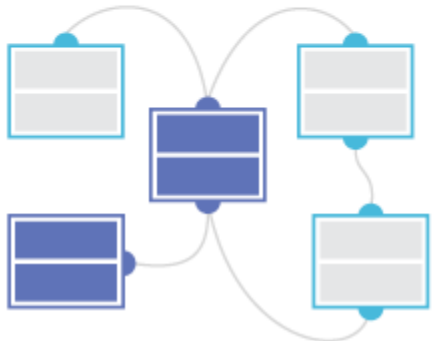
*Online Analytical Processing (OLAP) Cube*

## NoSQL

*Key-Value*



*Graph*



*Document*

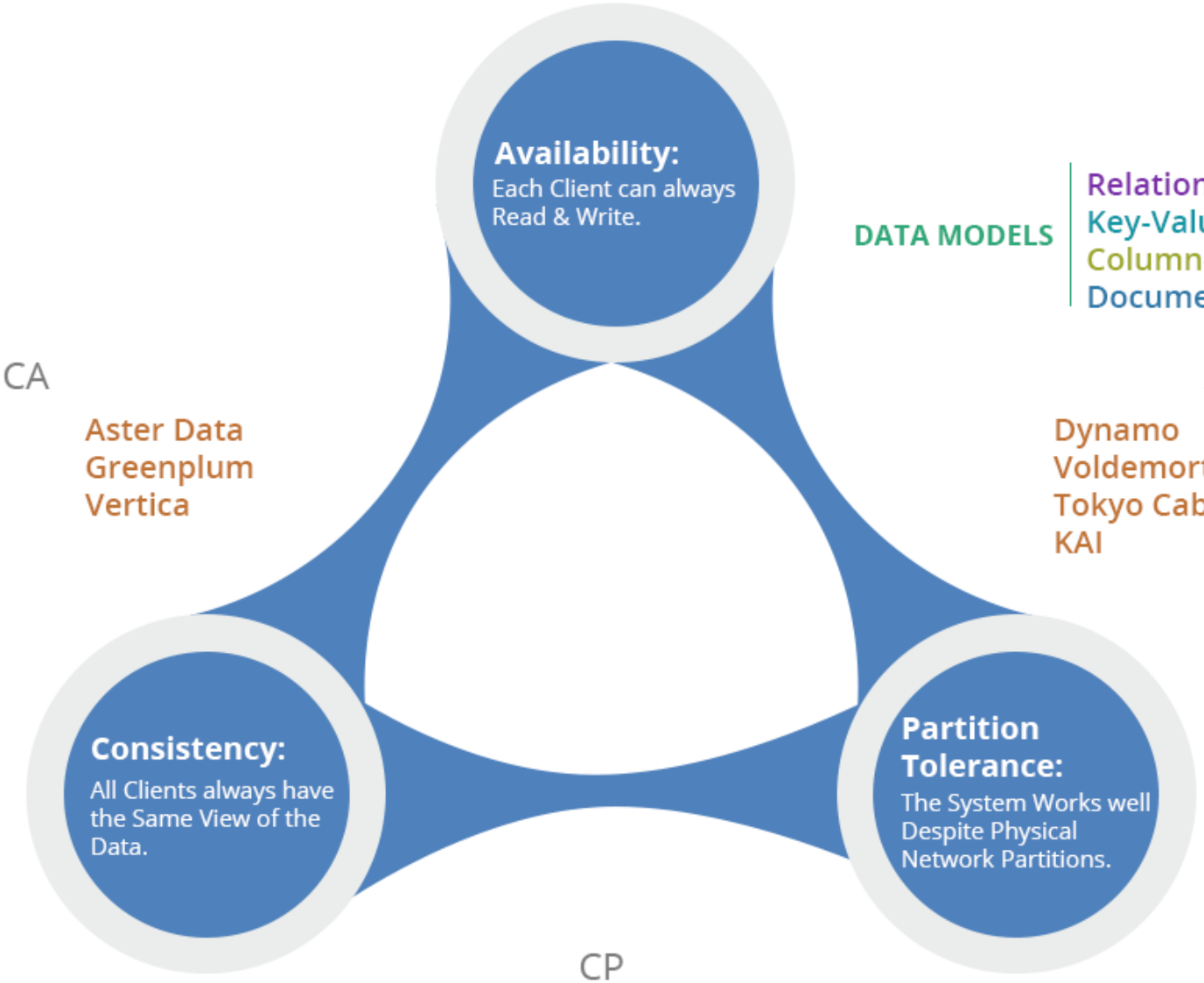
*Column store*

# Relational databases problems

- were not designed in such a way that they can run perfectly on clusters.
- The storage needs of an ERP application are very different than the data storage needs of Facebook and other such applications.
- There is much demand to achieve higher scalability, higher speed, and continuous availability.


NoSQL
Scalable
High Performance
Flexible

# CAP Theorem




DATA MODELS

- Relational (Comparison)
- Key-Value
- Column-Oriented/Tabular
- Document-Oriented

CA

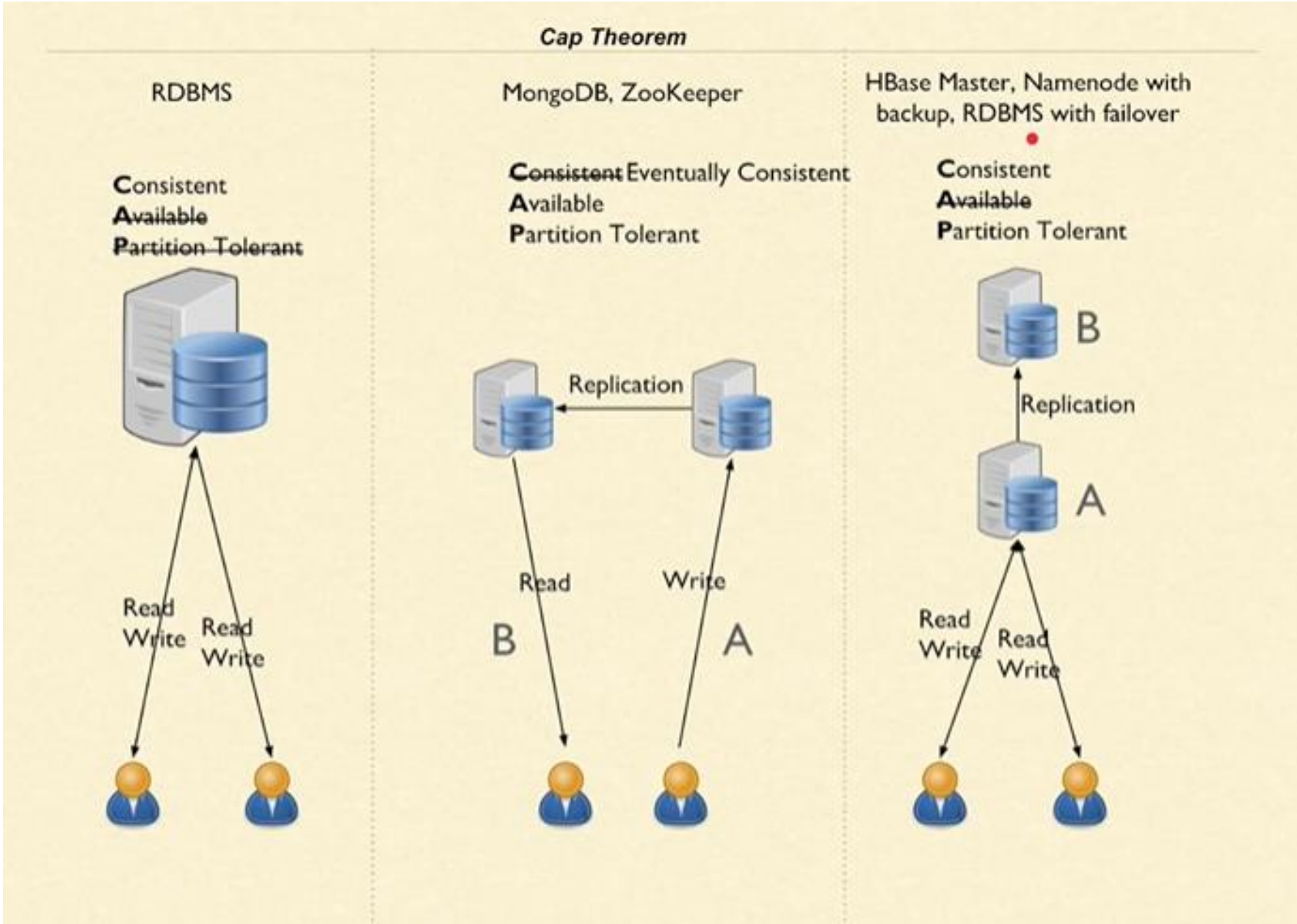
- RDBMSs (MySQL, Postgres, etc)
- Aster Data
- Greenplum
- Vertica

AP

- Dynamo
- Voldemort
- Tokyo Cabinet
- KAI
- Cassandra
- SimpleDB
- CouchDB
- Riak

CP

- BigTable
- Hypertable
- Hbase
- MongoDB
- Terrastore
- Scalaris
- Berkeley DB
- MemcacheDB
- Redis


Cap Theorem
RDBMS
Consistent
Available
Partition Tolerant
Read
Write
Read
Write
MongoDB, ZooKeeper
Consistent Eventually Consistent
Available
Partition Tolerant
Replication
Read
Write
B
A
HBase Master, Namenode with backup, RDBMS with failover
Consistent
Available
Partition Tolerant
B
Replication
A
Read
Write
Read
Write

# References :

“Choosing the right NoSQL database for the job: a quality attribute evaluation” by João Ricardo Lourenço, Bruno Cabral, Paulo Carreiro, Marco Vieira and Jorge Bernardino,  2015