A classification of NoSQL data stores based on key design characteristics

Cloud Futures: From Distributed to Complete Computing, CF2016, 18-20 October 2016, Madrid, Spain

Antonios Makris
PhD Candidate – Department of Informatics and Telematics
Harokopio University of Athens
amakris@hua.gr

HAROKOPIO
UNIVERSITY

Cloud and DBMS

- As a response to modern application requirements…
  - support large numbers of concurrent users
  - deliver highly responsive experiences to a globally distributed base of users
  - be always available
  - handle semi- and unstructured data
  - rapidly adapt to changing requirements with frequent updates
  - handle huge data volumes
- …clouds facilitated the shift from RDBMS to NoSQL

The shift…

- Meeting the application requirement by employing RDBMS would imply high costs associated with scaling

NoSQL

NoSQL Databases

NoSQL key features:

- “Shared nothing” horizontal scaling across servers
- Supports semi- and unstructured data
- Dynamic schema
- Weaker concurrency than ACID
- No declarative query language
- Higher guarantees
- Open-source (configurable)

BASE

Guaranteeing ACID properties in a distributed transaction across a distributed database where no single node is responsible for all data affecting the transaction, presents complications.

Instead of ACID, NoSQL databases provide BASE semantics:

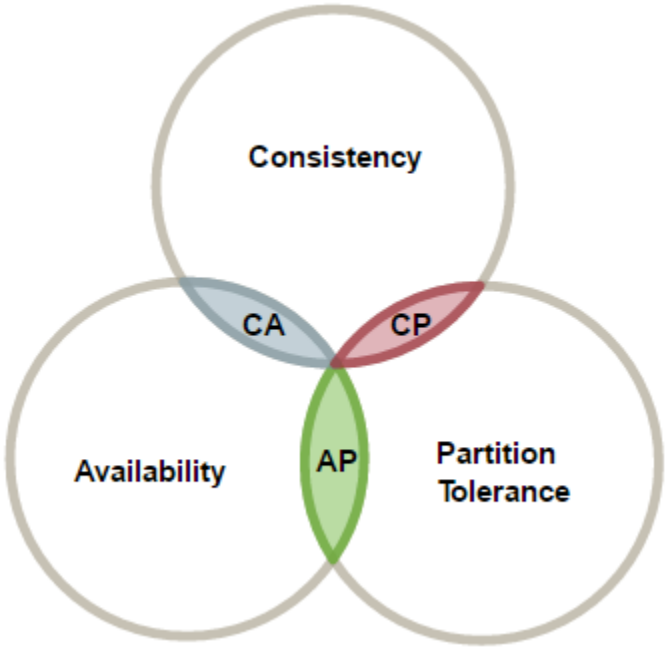
- **Basically available**: Systems seems to work always
- **Soft state**: May not be consistent always
- **Eventual consistency**: Will become consistent at some later time

CAP Theorem

Trade-off between consistency and availability.

CAP theorem: it is impossible for a distributed system to simultaneously provide all three of the guarantees in terms of Consistency, Availability and Partition Tolerance.

Rather, it is realistic for a distributed system to aim for two out of the three guarantees.

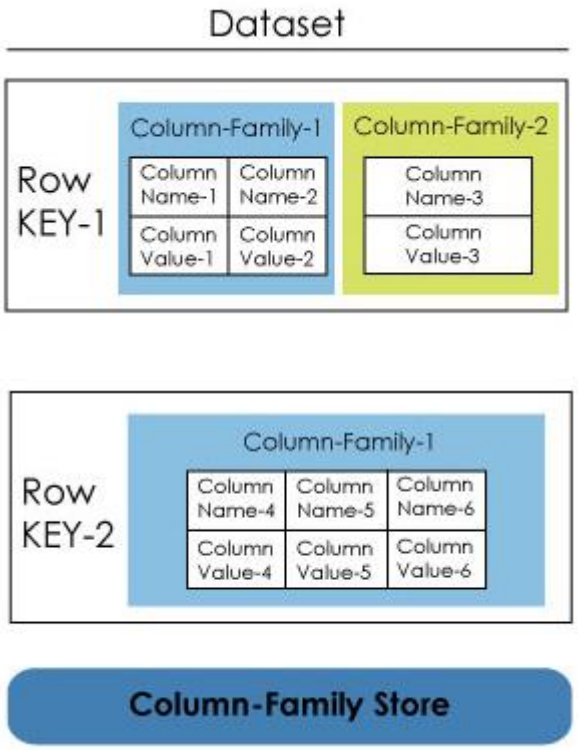


High-level taxonomy of the NoSQL data stores – Data Model

- **Key-value stores***:* every item in the database is stored as a key-value pair. The key value model uses a hash table which is comprised of a unique key and a pointer to a particular item of data. The keys are the only way to retrieve stored data.

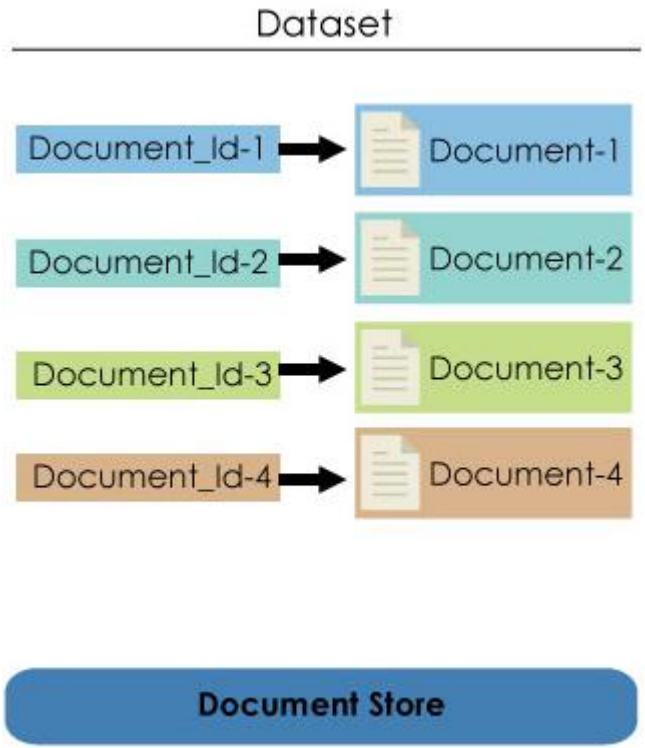
High-level taxonomy of the NoSQL data stores – Data Model

- **Column family stores***:* data are stored in a column-oriented way. The dataset consists of several rows, each of which is addressed by a unique row key.



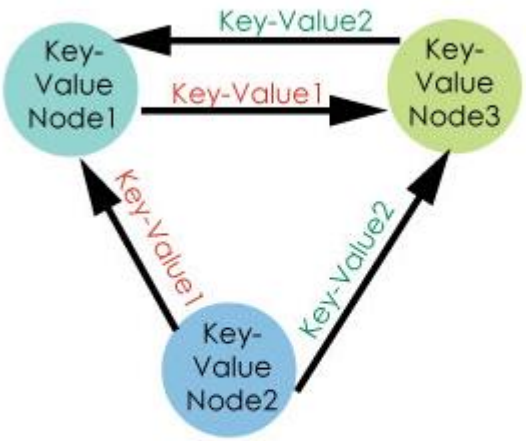
High-level taxonomy of the NoSQL data stores – Data Model

- **Document stores***:* data are a collection of key value pairs, but they are compressed as a document.



High-level taxonomy of the NoSQL data stores – Data Model

- **Graph stores***:* nodes and edges consist of objects with embedded key value pairs. Store information about networks.



Data Model - Systems

Data Model	Systems
Key-value Store	  
Column Store	 
Document Store	 
Graph Store	 

Other key characteristics

- Data placement
- Partition
- Replication
- Fault tolerance
- Consistency

Data placement

Data placement relates to the problem of optimal distribution of the data objects among the existing servers.

Data placement policies can be separated into two categories:

- geographical placement policies: data moved between geographically sparse data-centers
- data partitioning policies: data moved between nodes

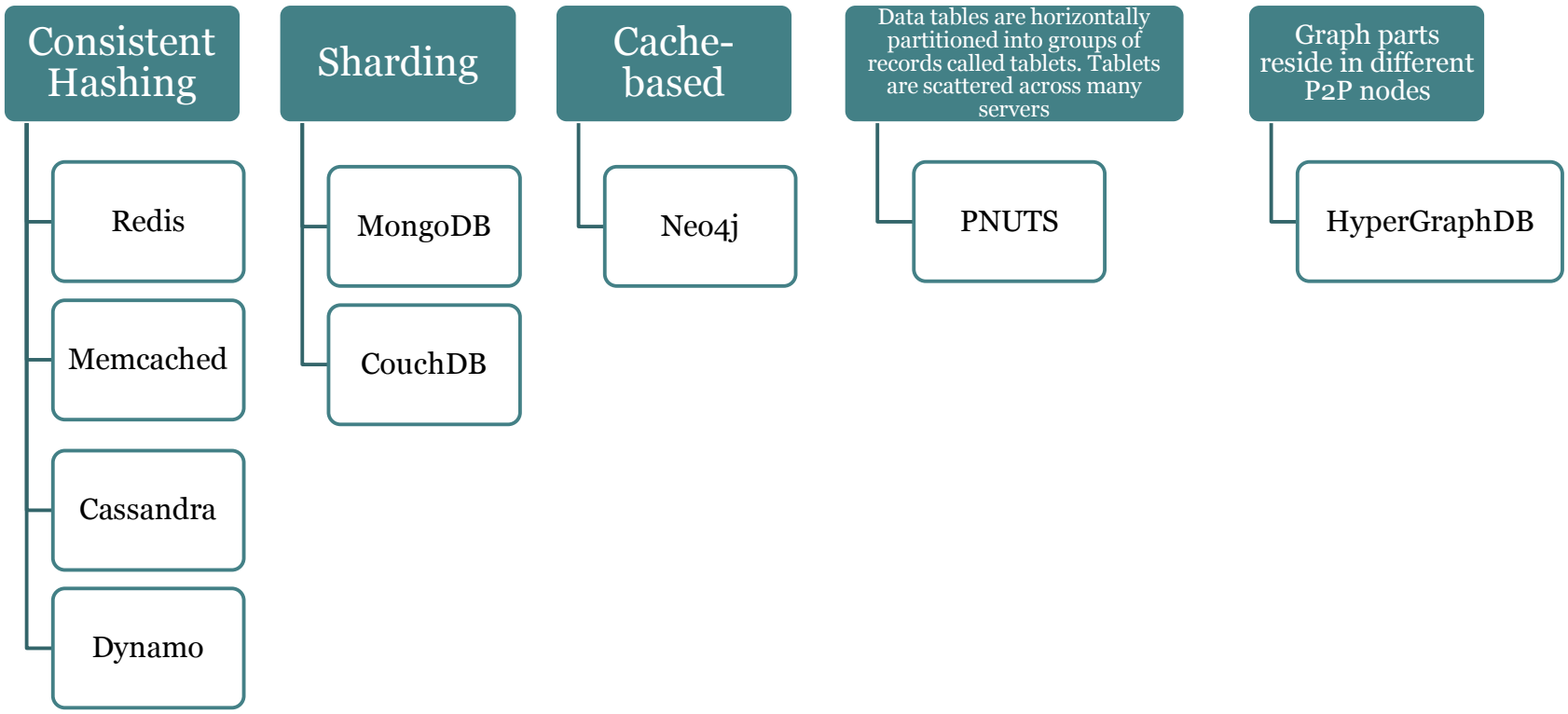
Main techniques that are used in distributed databases for data placement:

- Consistent hashing
- Multi-Attribute Sharding
- Bloom filters
- Probabilistic Associative Array
- Dynamic data placement

Partitioning

In order to store and process massive datasets, a commonly employed strategy is to partition the data and store the partitions across different server nodes.

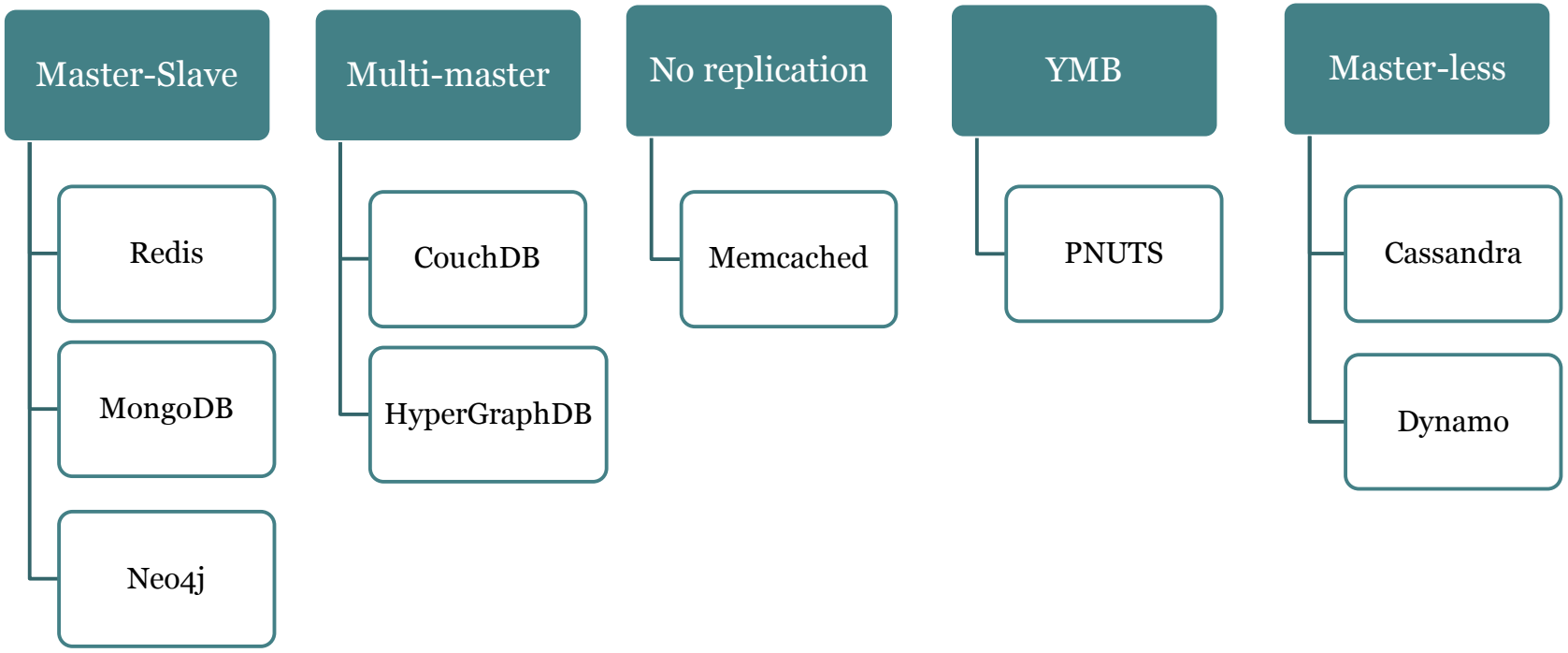
Partitioning



Replication

Replication is the process by which stored the same data on multiple servers so that read and write operations can be distributed over them.

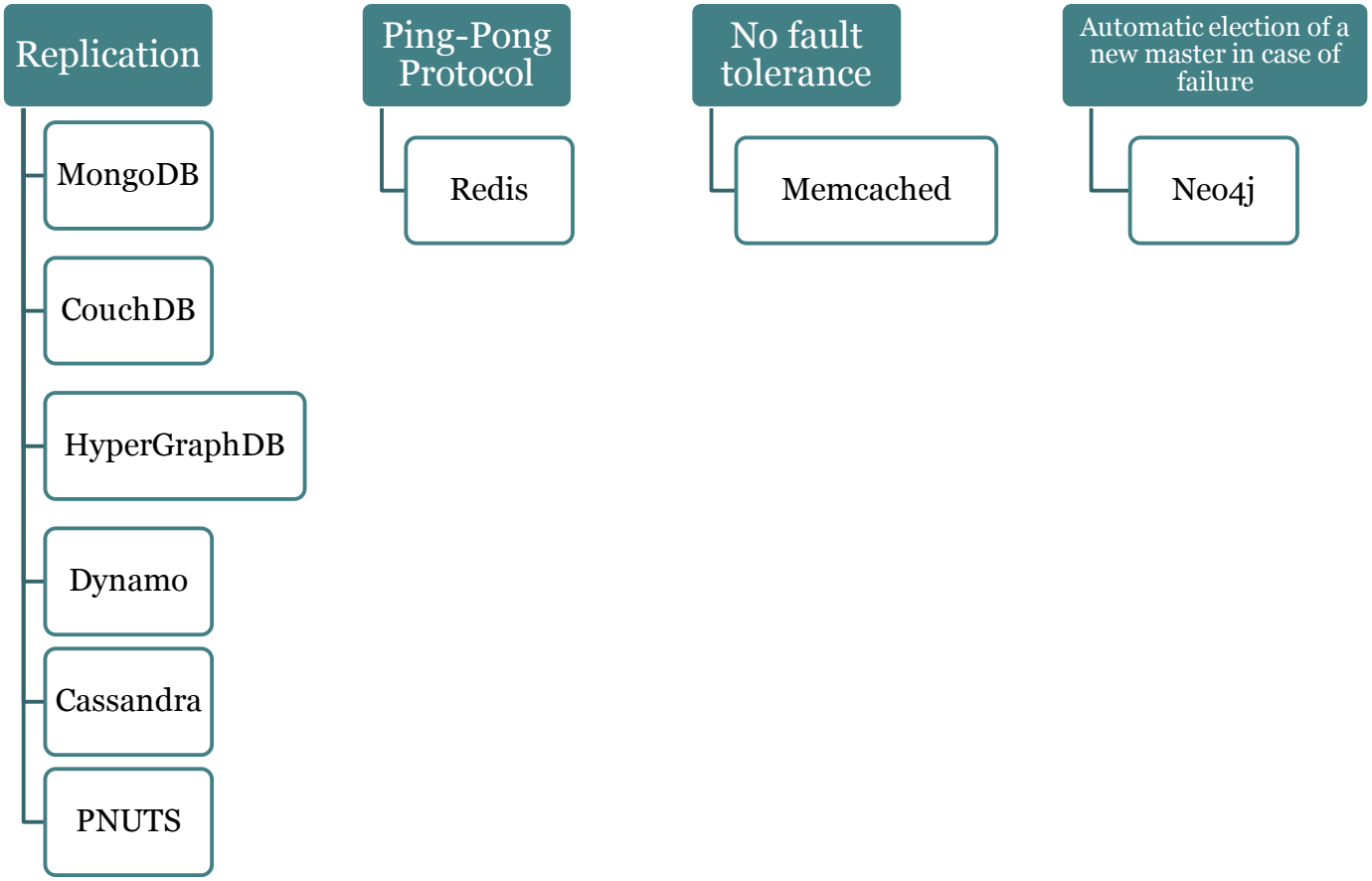
Replication



Fault tolerance

NoSQL database environments are no exception when it comes to hardware failures. However due to their distributed architecture there is no single point of failure and there is built-in redundancy of both function and data.

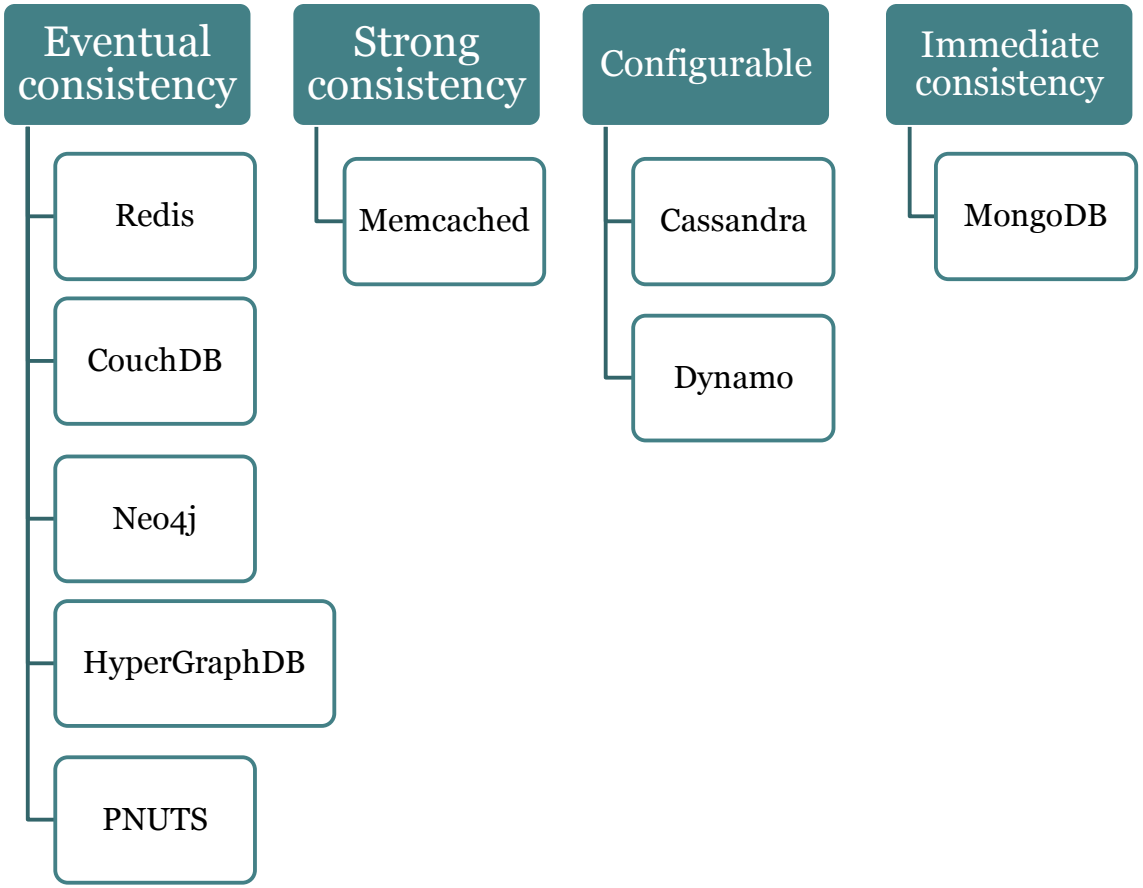
Fault tolerance



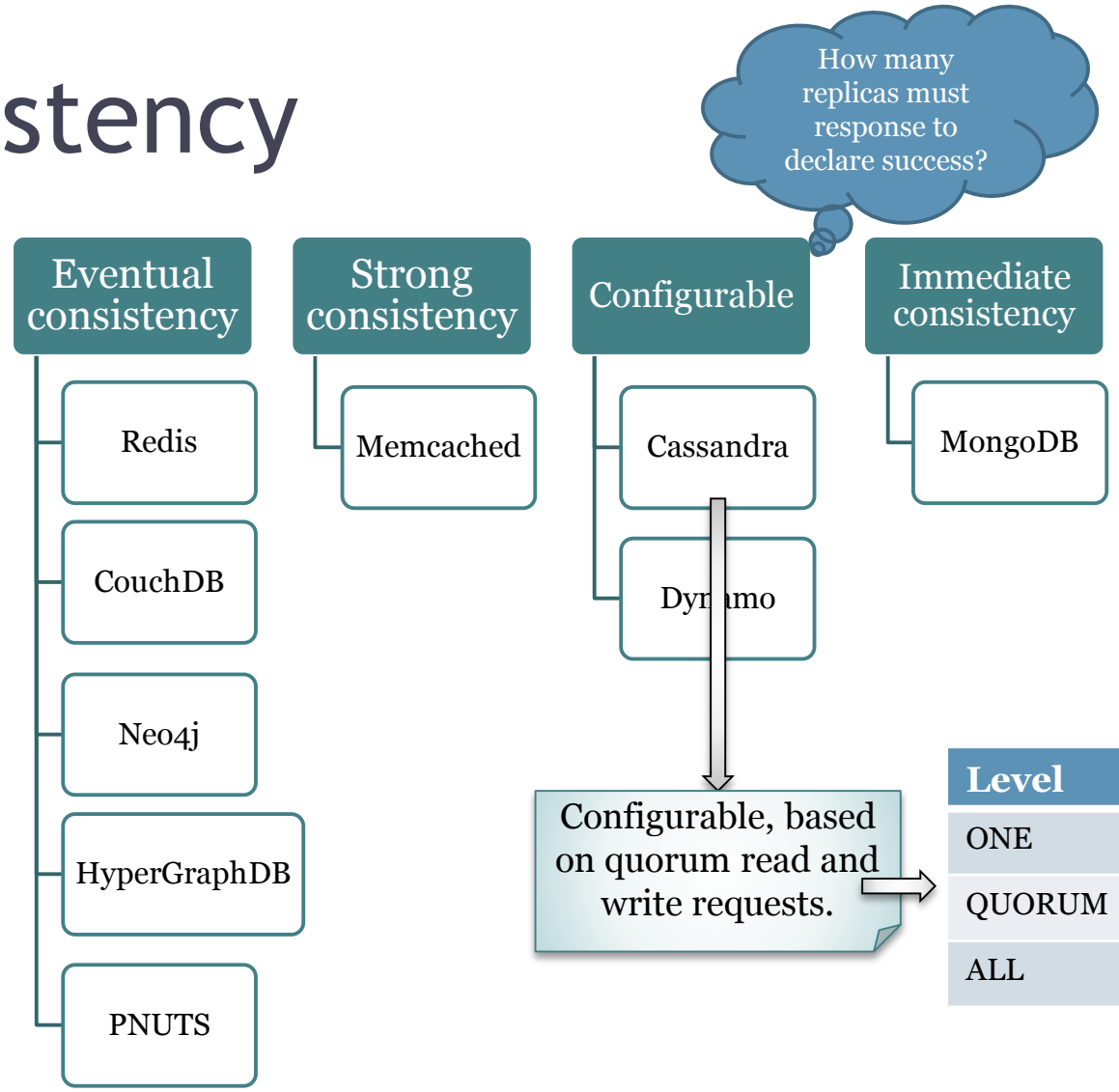
Consistency

Consistency is a system property that ensures that a transaction brings the database from one valid state to another.

Consistency

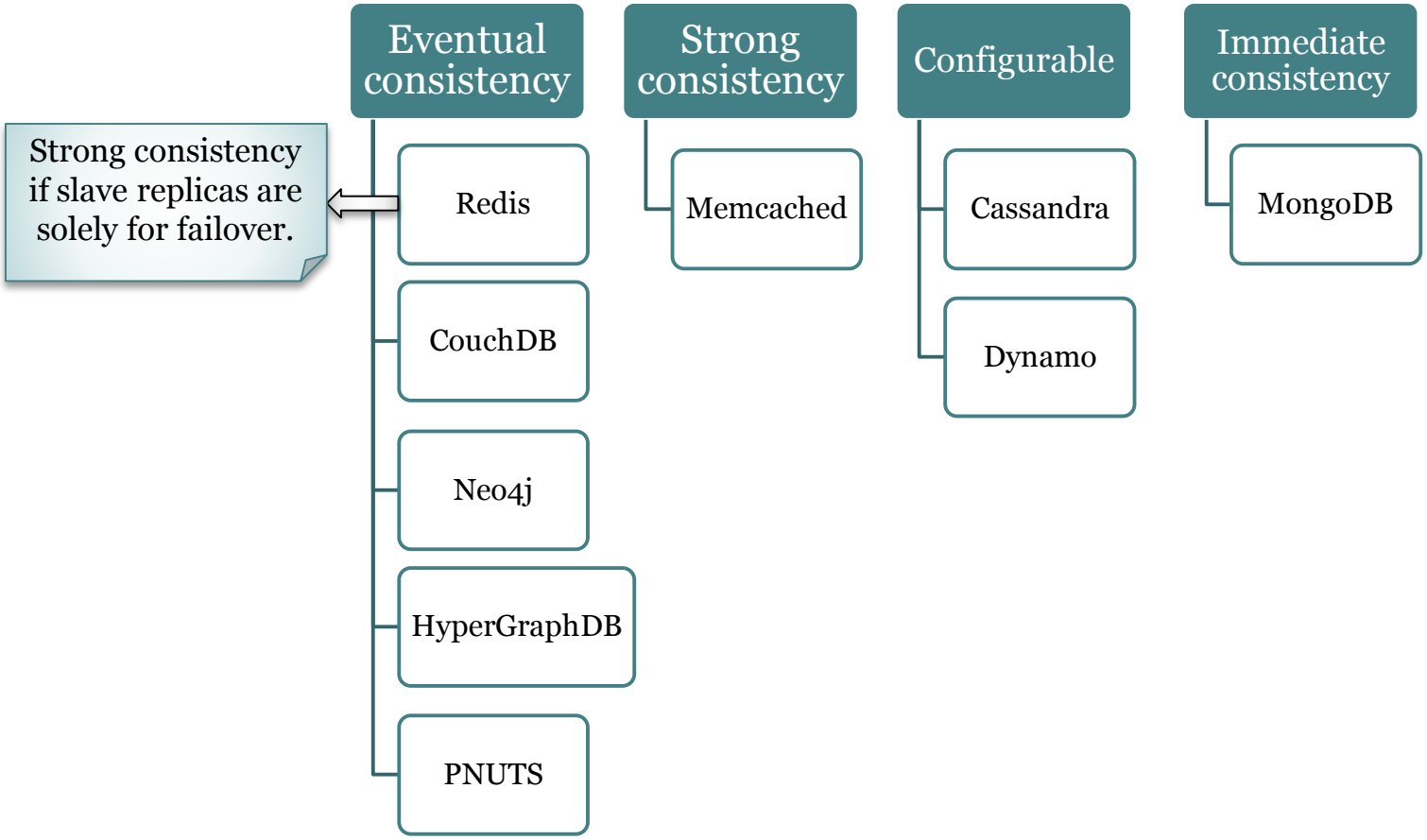


Consistency

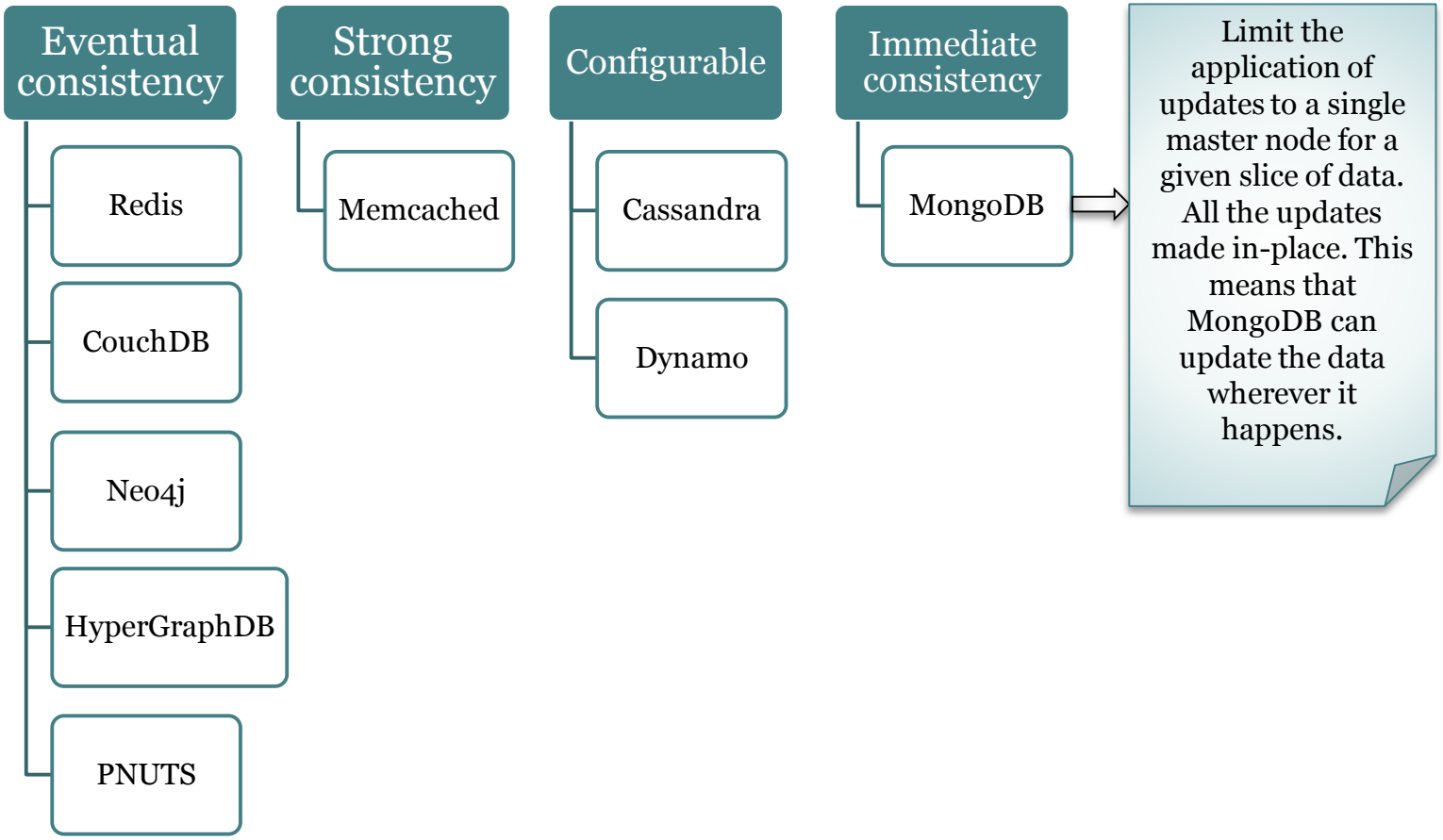


Level	Description
ONE	1st Response
QUORUM	N/2 +1 replicas
ALL	All replicas

Consistency



Consistency



Conclusion

- NoSQL comprise an alternative to traditional relational databases, capable of handling huge volumes of data leveraging on the capabilities of cloud environments. Each NoSQL database should be used in a way that it meets its claims and the overall system requirements.

- Also it is presented how the different data stores were designed to achieve high availability and scalability at the expense of strong consistency.

- The ‘one size fit’s it all’ notion does not work for the current application scenarios and it is a better to build systems based on the nature of the application and its work/data load.

- The different data stores use different techniques to achieve company or users requirements.

Thank you

- amakris@hua.gr
- http://www.dit.hua.gr