NOSQL OVERVIEW

Yunghans Irawan (yirawan@nus.edu.sg)

Agenda

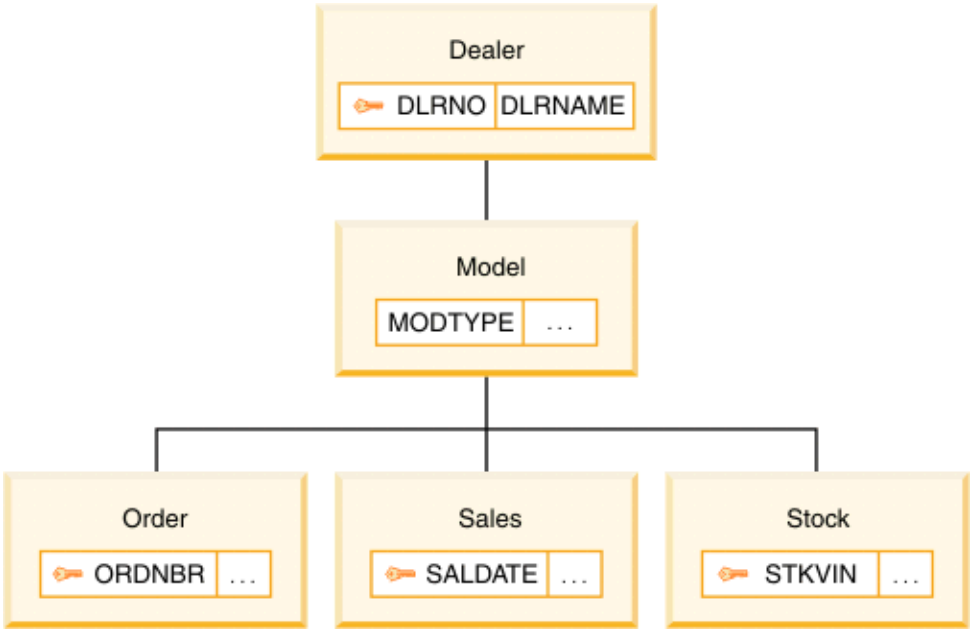
- Brief History of Database Models
  - Pre-relational
  - Relational
  - NoSQL
- NoSQL Classification
  - Key value
  - Document
  - Graph
  - Column Family

Navigational DBMS (1960s)

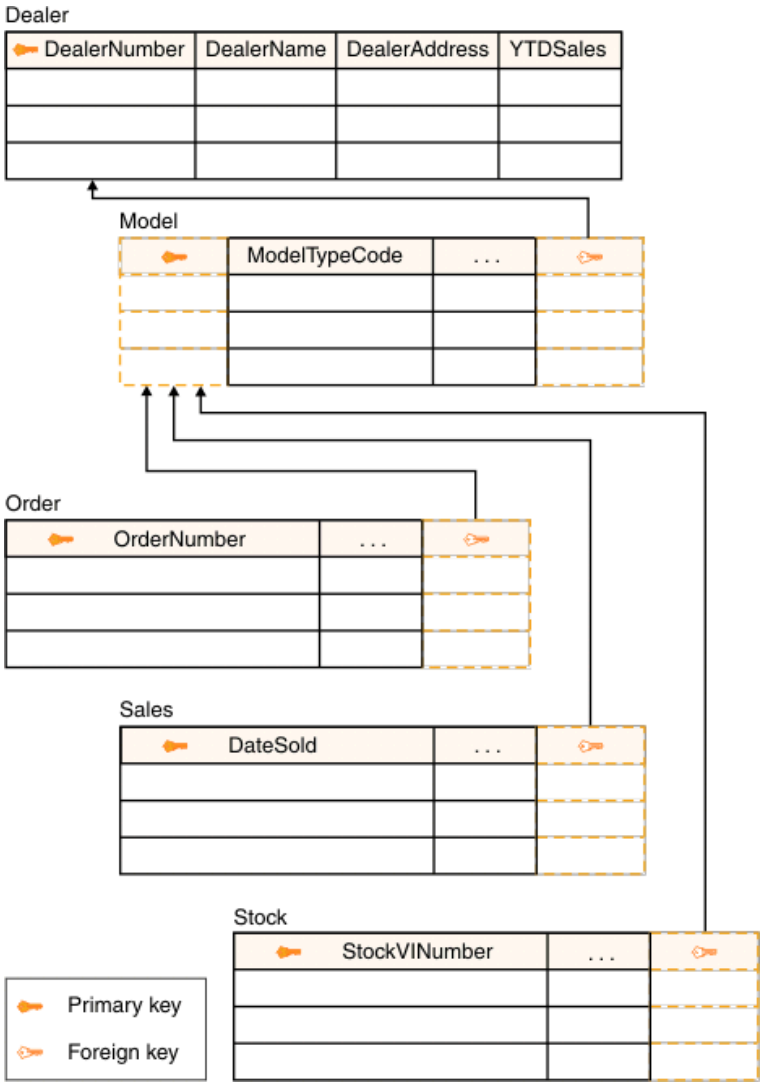
- Sometimes called as pre-relational database
- Hierarchical Model
  - data is organized into a tree-like structure
  - mandates that each child record has only one parent, whereas each parent record can have one or more child records. In order to retrieve data from a hierarchical database the whole tree needs to be traversed starting from the root node.
  - E.g.: IBM IMS (Information Management System)
- Network Model
  - schema, viewed as a graph in which object types are nodes and relationship types are arcs, is not restricted to being a hierarchy
  - allows each record to have multiple parent and child records
  - E.g.: IDMS (Integration Database Management System

Hierarchical Model

Hierarchical Model

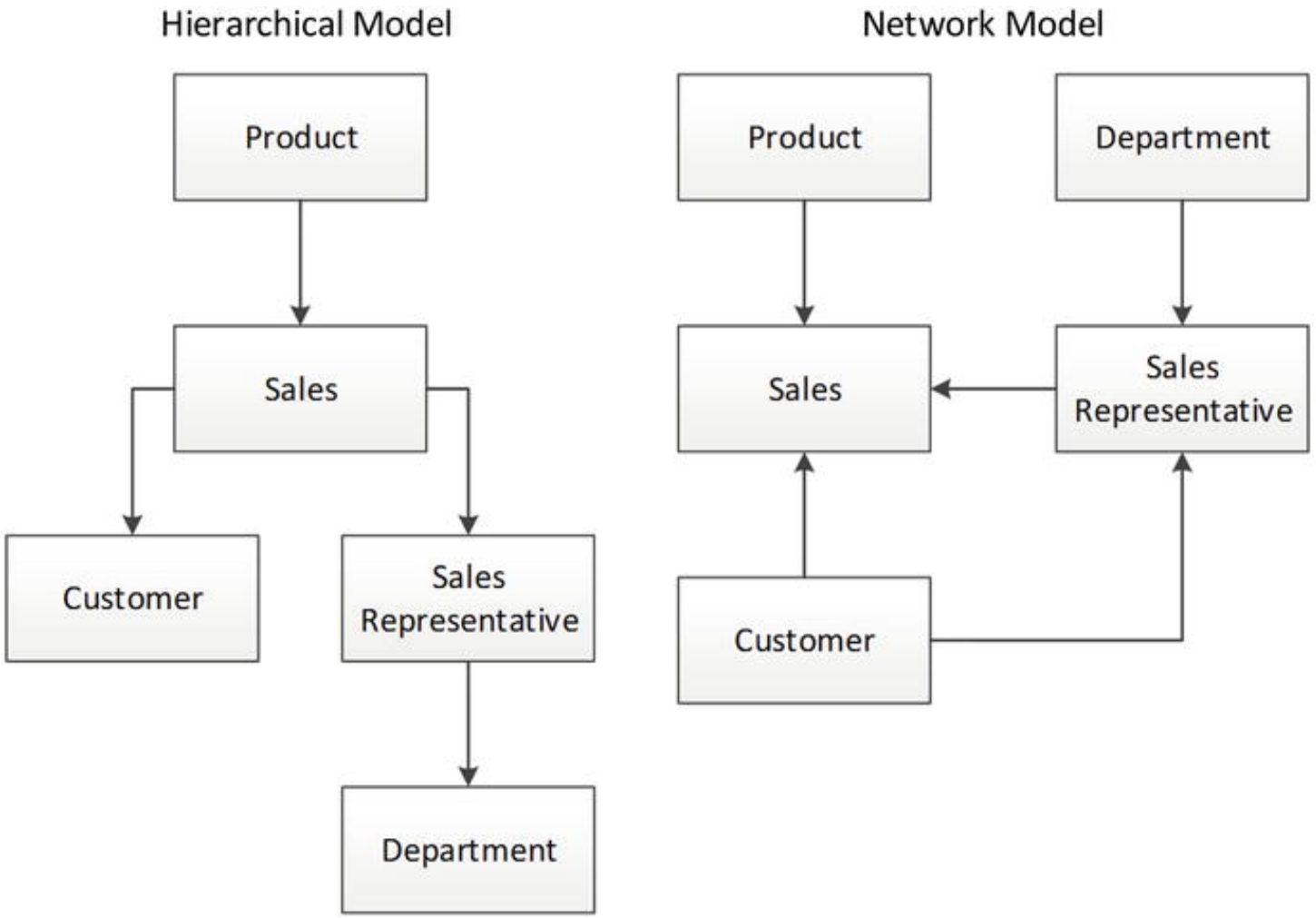


RDBMS Equivalent



Source:
http://www.ibm.com/support/knowledgecenter/SSEPH2_13.1.0/com.ibm.ims13.doc.apg/ims_comparehierandreldbs.htm

Network Model



Source: Next Generation Databases, NoSQL, NewSQL and Big Data

Relational DBMS (1970s)

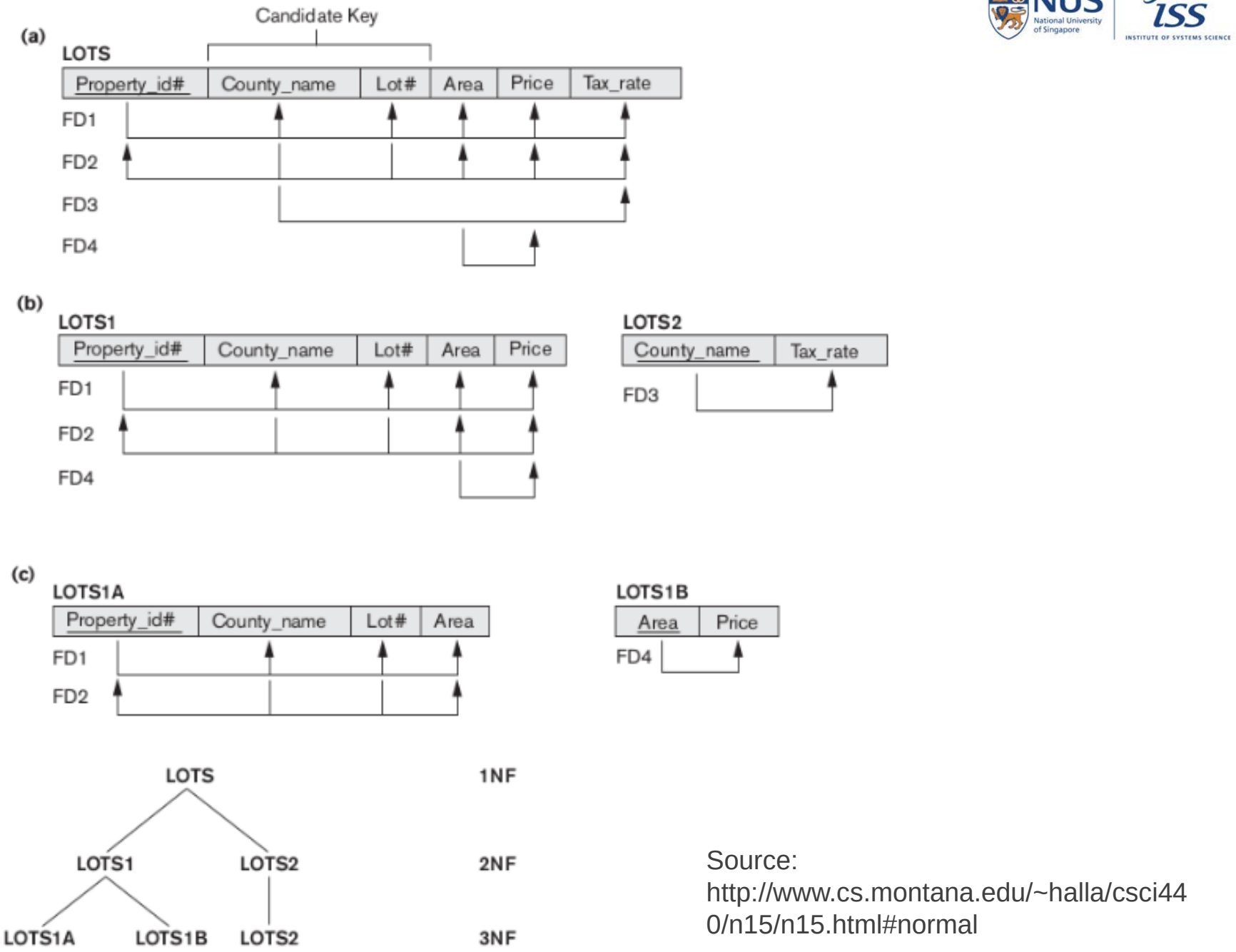
- Based on a paper from Edgar Codd: "*A Relational Model of Data for Large Shared Data Banks*"
- Essentially describes how a given set of data should be presented to the user, rather than how it should be stored on disk or in memory
- Levels of conformance to the relational model are described in the various "*normal forms*."
- Jim Gray: "A transaction is a transformation of state which has the properties of atomicity (all or nothing), durability (effects survive failures) and consistency (a correct transformation)."
- ACID transactions is strongly associated with relational databases

Normal Forms

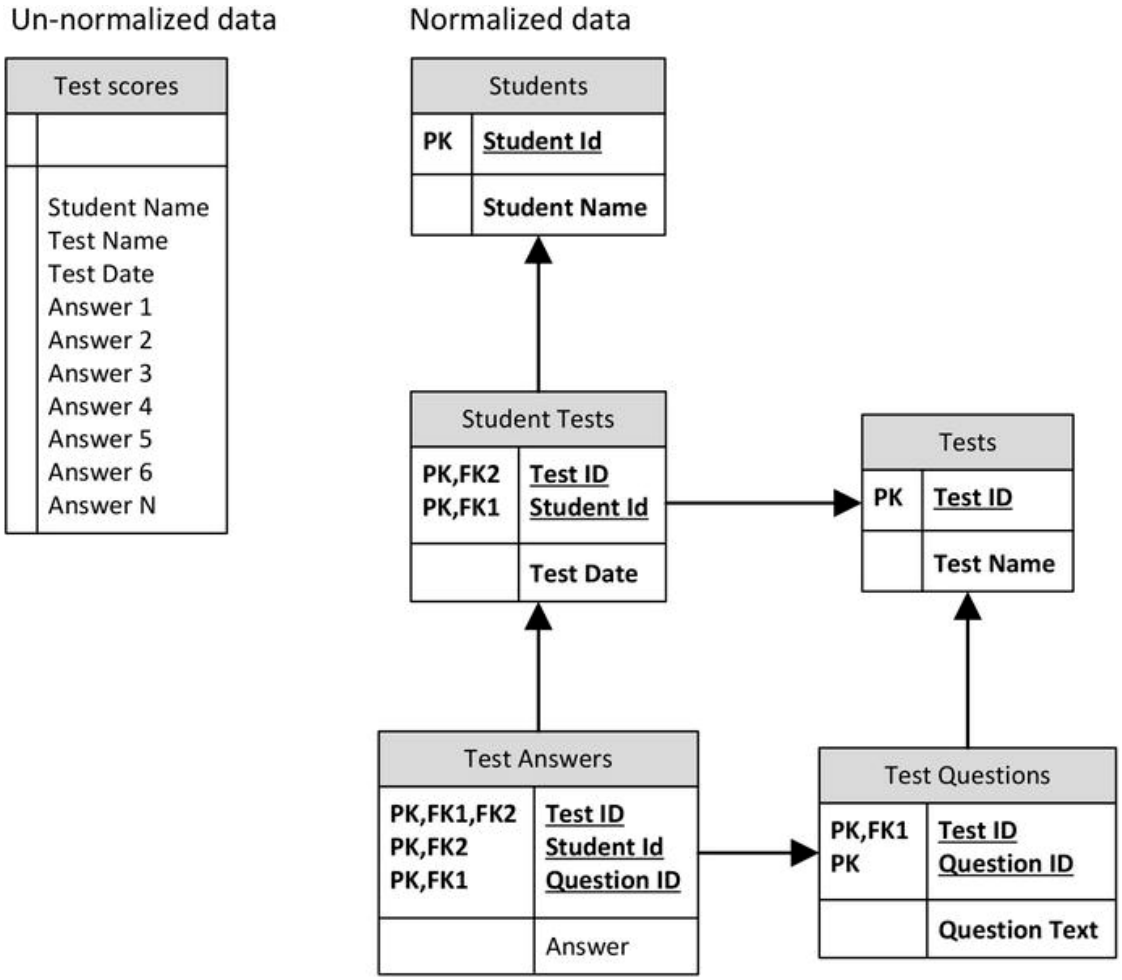
- 1NF
  - all attributes must be atomic and single-valued
- 2NF
  - every non-prime attribute of the table is dependent on the whole of every candidate key.
  - No partial dependency on the primary key or any of the candidate key(s)
- 3NF
  - Every non-key attribute must provide a fact about the key, the whole key, and nothing but the key

Example

- Suppose you want to store price, area and tax rate for different land lots in multiple counties
- Each county have a fixed tax rate
- The land area of the lots are standardized. There are few standard sizes with different prices



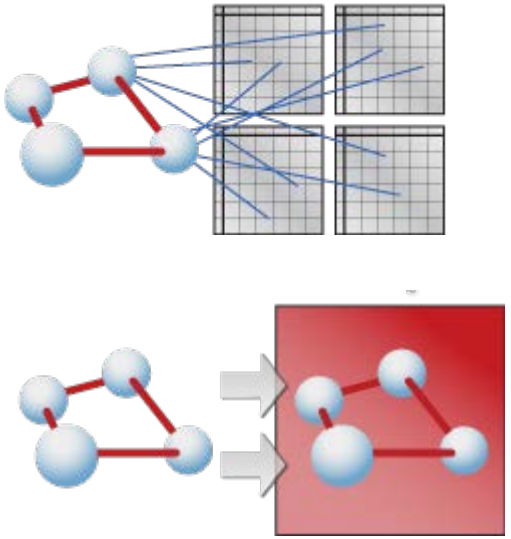
Source: http://www.cs.montana.edu/~halla/csci440/n15/n15.html#normal



Source: Next Generation Databases, NoSQL, NewSQL and Big Data

OODBMS -1990s

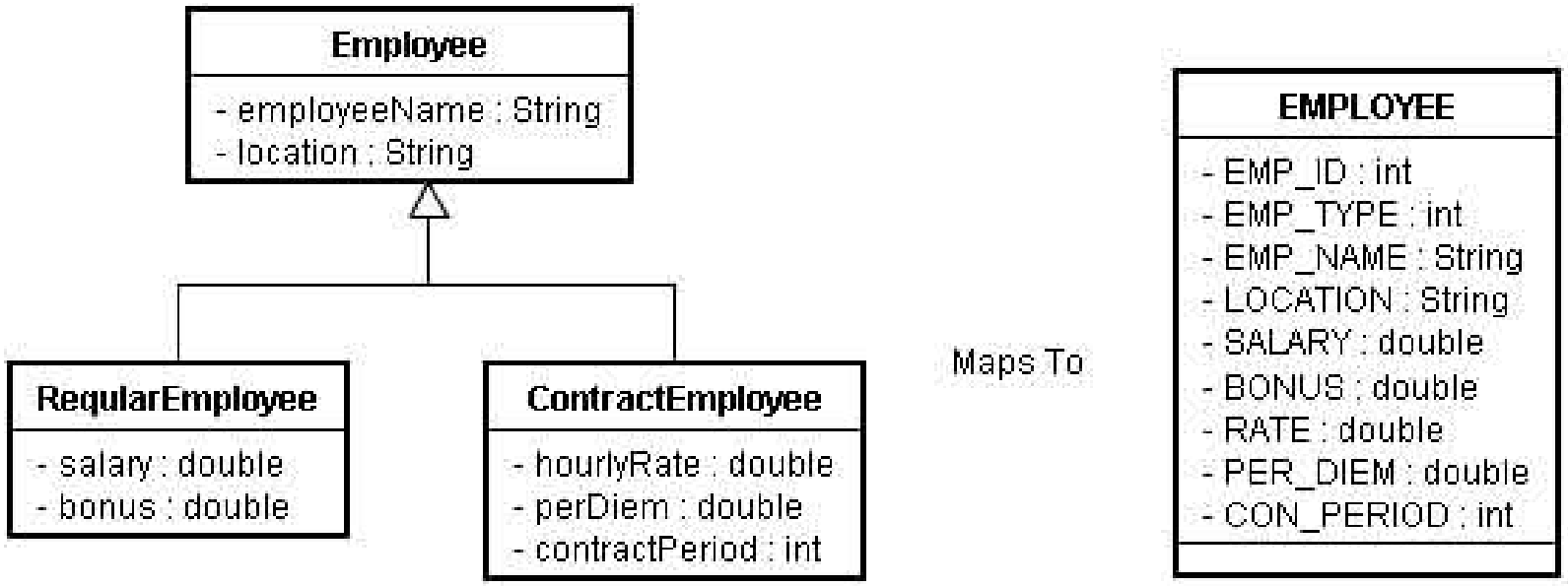
- **The OODBMS Manifesto (Atkinson/Bancilhon/DeWitt/Dittrich/Maier/ Zdonik, '90)**
- "A relational database is like a garage that forces you to take your car apart and store the pieces in little drawers“
  - Also SQL is ugly
- “A Object database is like a closet which requires that you hang up your suit with tie, underwear, belt socks and shoes all attached” (Dave Ensor)



OODBMS (1990s)

- Due to success of Object Oriented Programming
  - Storing object into RDBMS is not simple
- Store objects without normalization
- Support **complex objects**, object identity, **encapsulation**, types or classes, **inheritance**, overriding combined with late binding, extensibility and computational completeness.
- https://www.cs.cmu.edu/~clamen/OODBMS/Manifesto/htManifesto/Manifesto.html
- Failed to get market share
- Object-Relational Mapping (ORM) helps to solve part of the problems OODBMS tried to solve

Object vs RDBMS Table



Source: https://simsonlive.wordpress.com/2008/03/09/how-inheritance-works-in-hibernate/

NoSQL (2005 -)

- Inadequacy of existing products to cope with volumes and velocity of data needed by massive web-scale applications (e.g. Google)
- Trigger many innovations
  - Google: Google File System (2003) → MapReduce (2004) → BigTable (2006)
  - Yahoo: Google Map Reduce → Hadoop (2007)
  - Amazon: DynamoDB (2007)
  - Facebook: Sharding with MySQL → Cassandra (2008)
  - Any many many others

NoSQL (2005 -)

- Need a name:
  - Distributed Non-Relational Database Management System (DNRDBMS)
  - NoSQL – most popular
  - NewSQL
- Some desired characteristics
  - Availability
  - Tolerant to network partition
  - Low latency – fast response time
  - Run on commodity hardware
  - Incremental scalability with no downtime

THE THIRD PLATFORM

The Third Platform is described by IDC as the next-generation compute platform that is accessed from mobile devices, utilizes Big Data, and is cloud based.

BILLIONS OF USERS

3RD PLATFORM

Mobile Big Data Social

CLOUD

MILLIONS OF APPS

HUNDREDS OF MILLIONS OF USERS

2ND PLATFORM

LAN/Internet Client/Server

DISTRIBUTED

TENS OF THOUSANDS OF APPS

MILLIONS OF USERS

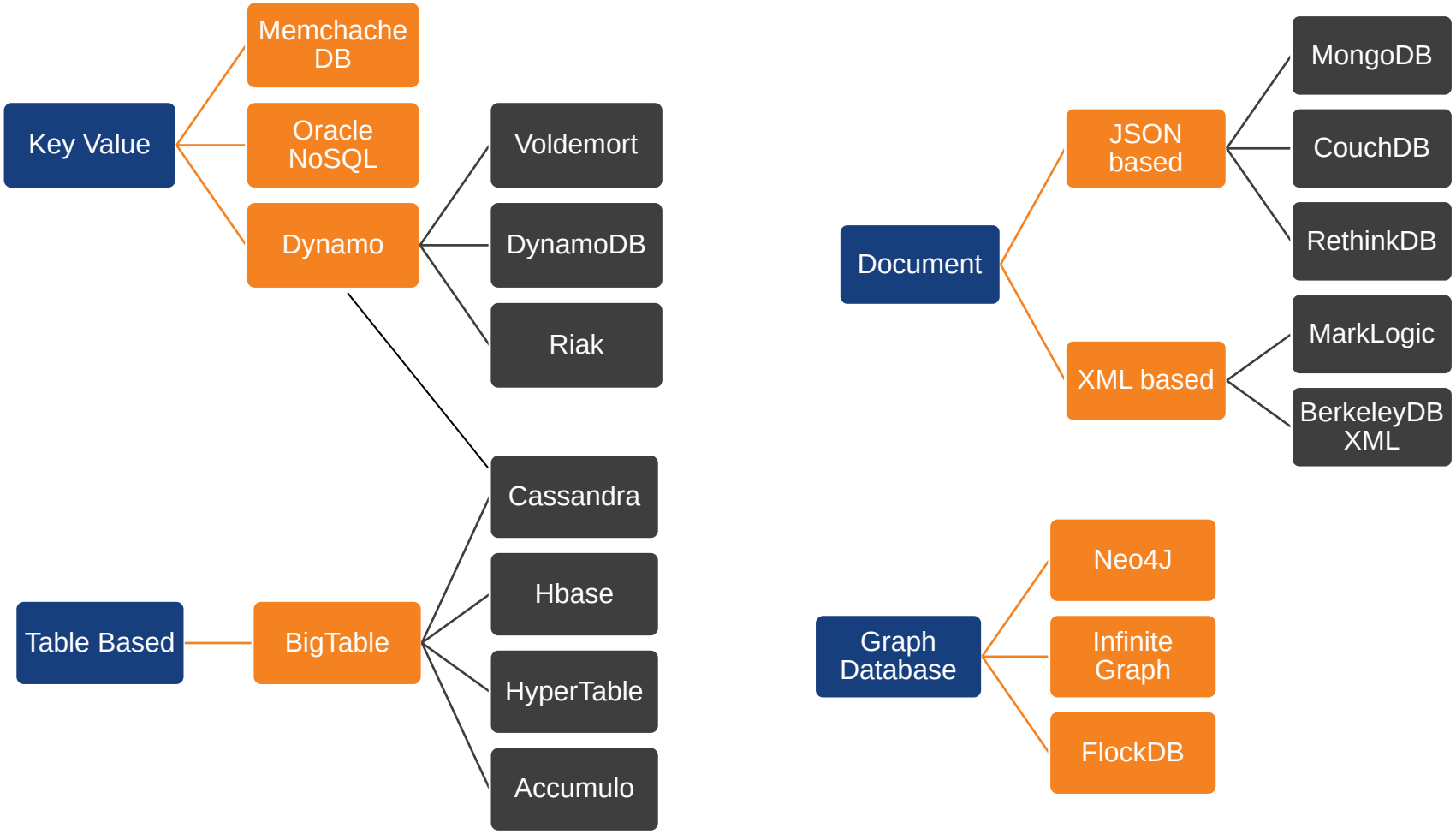
1ST PLATFORM

Mainframe, Mini Computer

MAINFRAMES

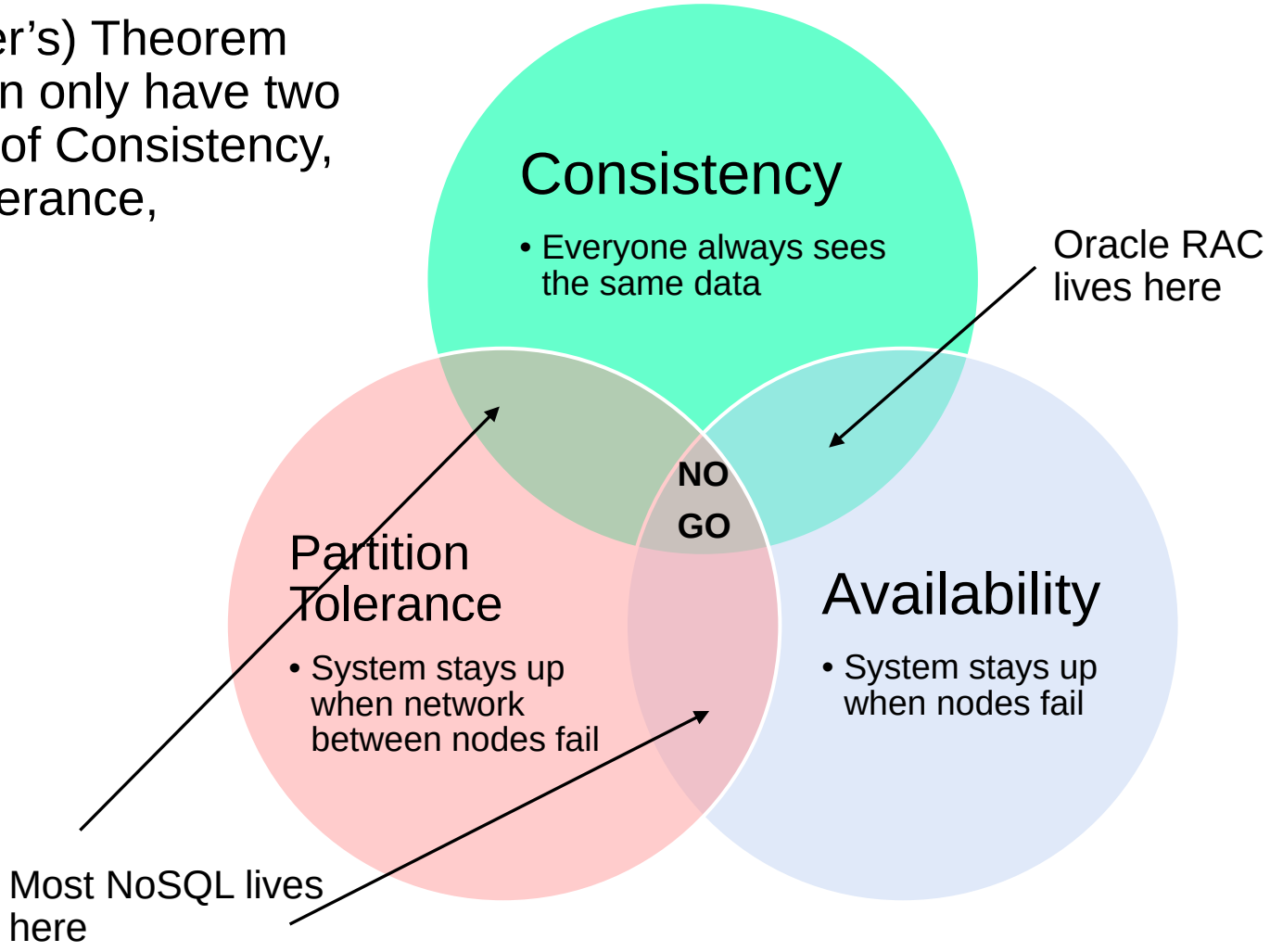
THOUSANDS OF APPS

NoSQL Classification



CAP Theorem

CAP (Brewer's) Theorem says you can only have two out of three of Consistency, Partition Tolerance, Availability



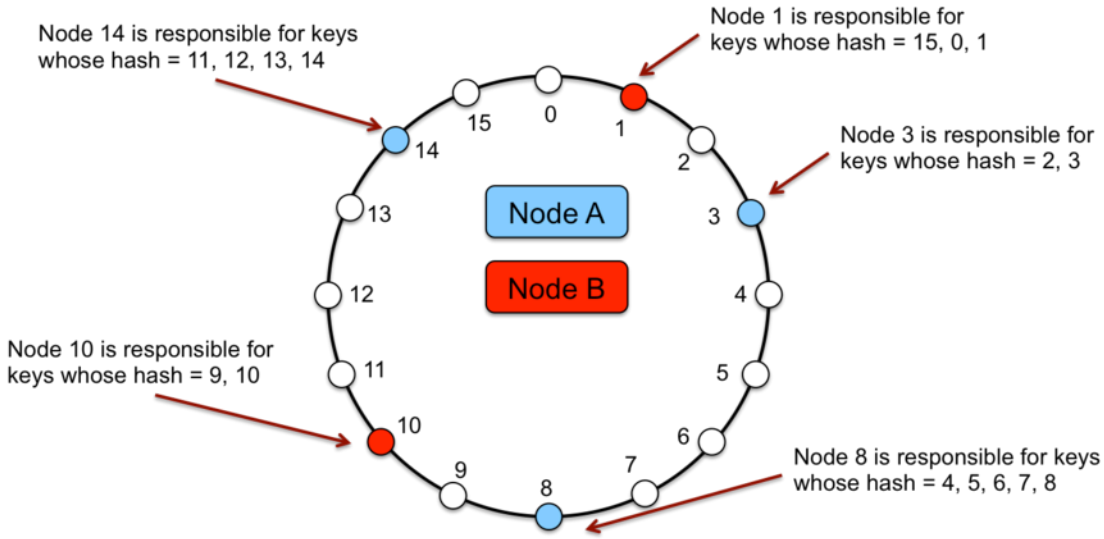
Amazon Dynamo Model

DOI:10.1145/1435417.1435432

Building reliable distributed systems at a worldwide scale demands trade-offs between consistency and availability.

BY WERNER VOGELS

Eventually Consistent

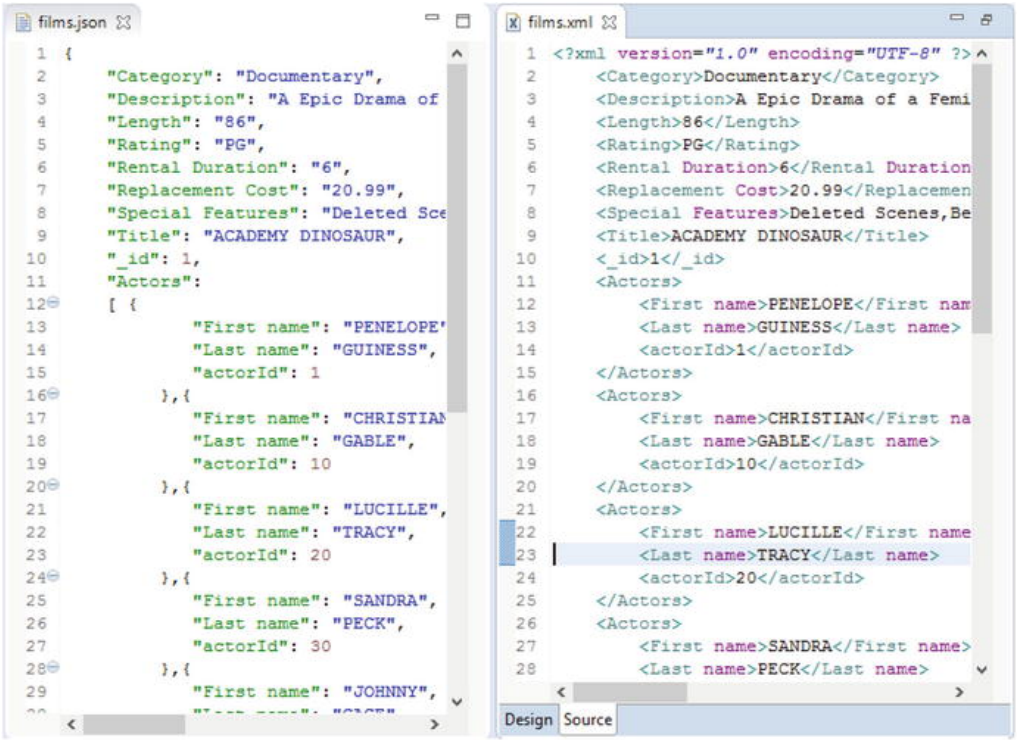


Key Value Database

- Support primary key based lookup
- Schemaless - no data model
- Some incorporate built-in data structure like sets and maps with their operations

Document Database

- Non relational database that stores data as structured document
- Usually in XML or JSON formats
- Usually schemaless
- Relatively easy to transform object into JSON or XML



Graph Database

- Relationship between things is the information that are of primary interest
- Graph can be modeled with RDBMS
  - Performance issue when dealing with large graph
  - SQL is not designed for graph data

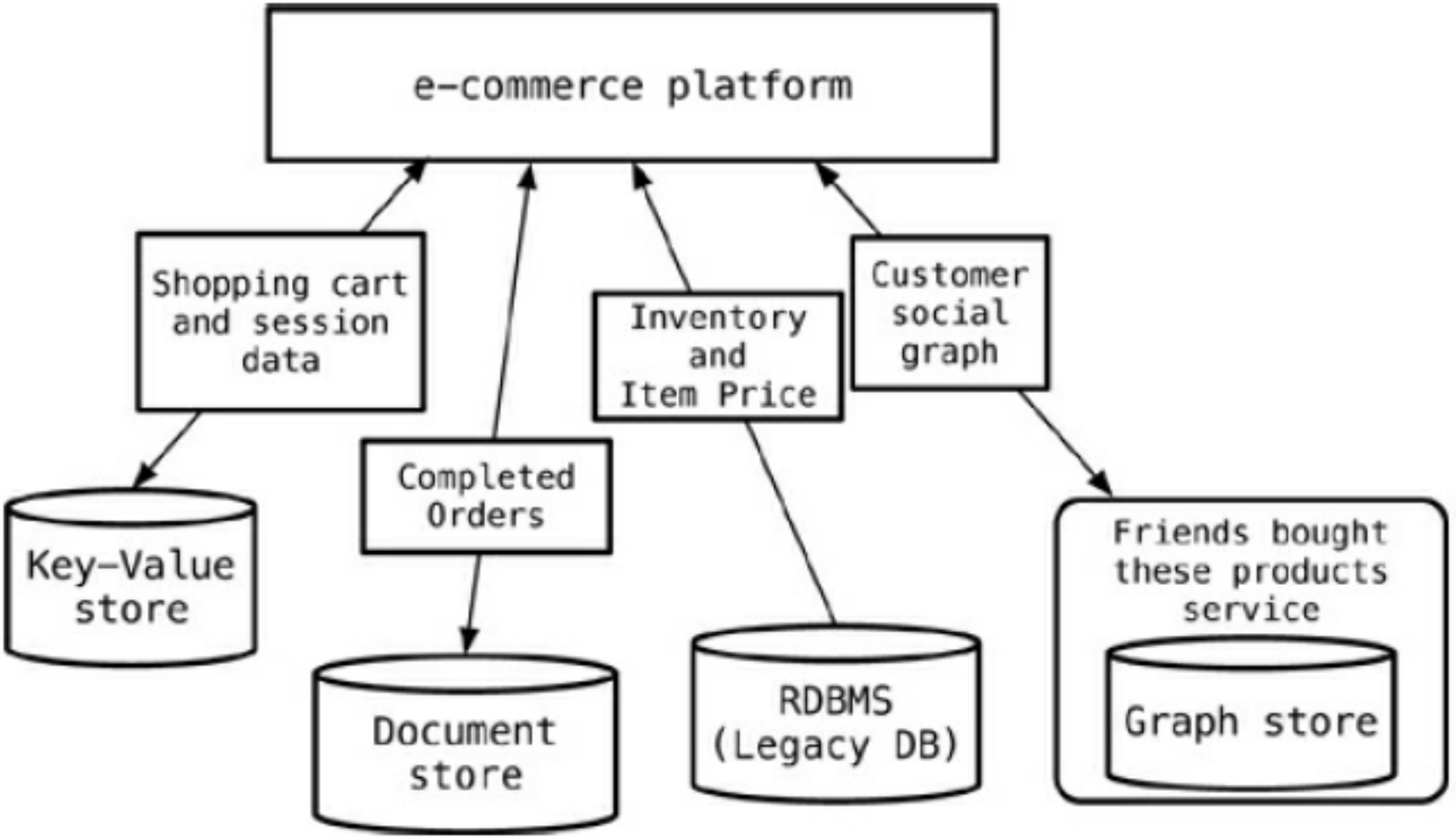
Column Family Database

- Store data in column families
  - Many column associated with a row key
- Column families are groups of related data that is often accessed together
  - Like a table in RDBMS
- Arguably more similar to RDBMS compared to other types of NoSQL databases
- Main motivation of the early products are high throughput and scalability using commodity hardware

Polyglot Persistence

- Polyglot persistence is about using different data storage technologies to handle varying data storage needs
- Polyglot persistence can apply across an enterprise or within a single application
- Encapsulating data access into services reduces the impact of data storage choices on other parts of a system
- Adding more data storage technologies increases complexity in programming and operations, so the advantages of a good data storage fit need to be weighed against this complexity

Polyglot Persistence



Summary

- We have discussed development of database concepts over many decades
  - Some similarity between ideas
  - Some good ideas but doesn't get any market traction
- We are now at a time where there are various database concepts with reasonable traction and potentially useful features and applications
  - Many of them are proven by companies with very challenging requirements

References

- CS4221 Lecture Notes – A Brief Introduction on Hierarchical and Network Data Models
  - https://www.comp.nus.edu.sg/~lingtw/hierarchical.network.models.pdf
- Next Generation Databases – NoSQL, NewSQL and Big Data
  - Guy Harrison, Apress 2016