NoSQL Database Systems and their Security Challenges

Morteza Amini
amini@sharif.edu

Data & Network Security Lab (DNSL)
Department of Computer Engineering
Sharif University of Technology

September 2015

Talk Outline

- Introduction
- NoSQL vs. Relational Databases
- Types of NoSQL Databases
- NoSQL Security Challenges

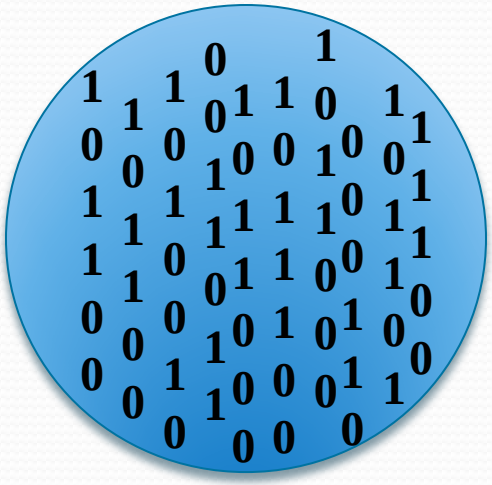
Introduction

Current Trends

- The new generation of applications like cloud or Grid apps, Business Intelligence, Web 2.0, Social networking requires storing and processing of terabytes and even petabytes of data

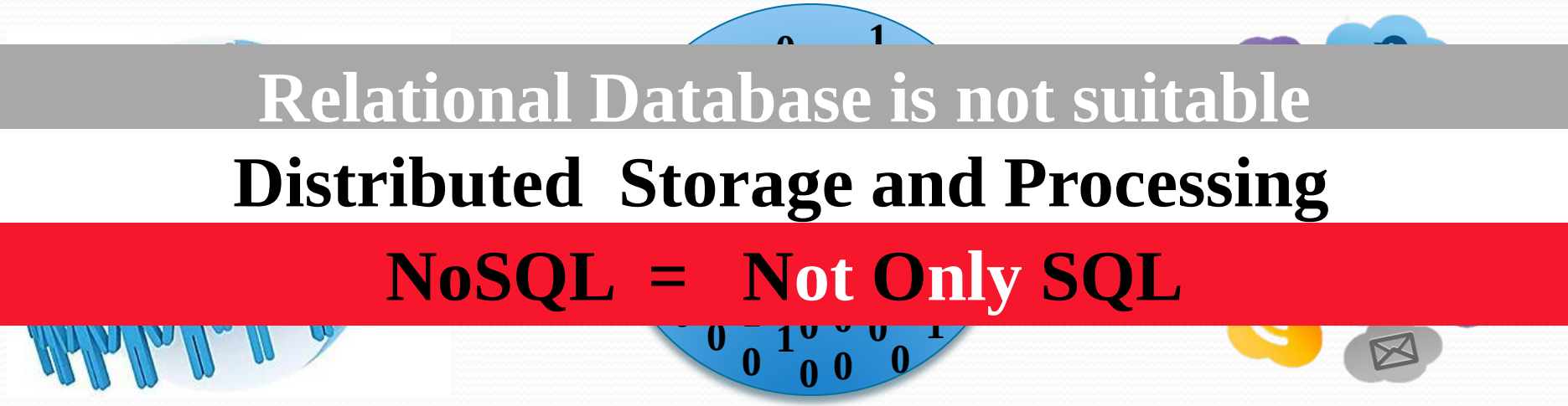
Today

- We have…
  - More users
  - More data
  - Interactive apps



Today

- The requirements of storage database systems is changed



Relational Database is not suitable

Distributed Storage and Processing

NoSQL = Not Only SQL

NoSQL vs. Relational Databases

Why relational database is not suitable ?

- A relational database is a data structure that allows you to link information from different ‘tables’

Car

CarKey	MakeKey	ModelKey	ColorKey	Year
1	1	1	2	2003
2	2	1	3	2005
3	2	1	2	2005

Color

ColorKey	Color
1	Red
2	Green
3	Blue

MakeModel

ModelKey	MakeKey	Model
1	1	Pathfinder
1	2	Bluebird
2	1	Civic

Make

MakeKey	Make
1	Nissan
2	Honda

Why relational database is not suitable ?

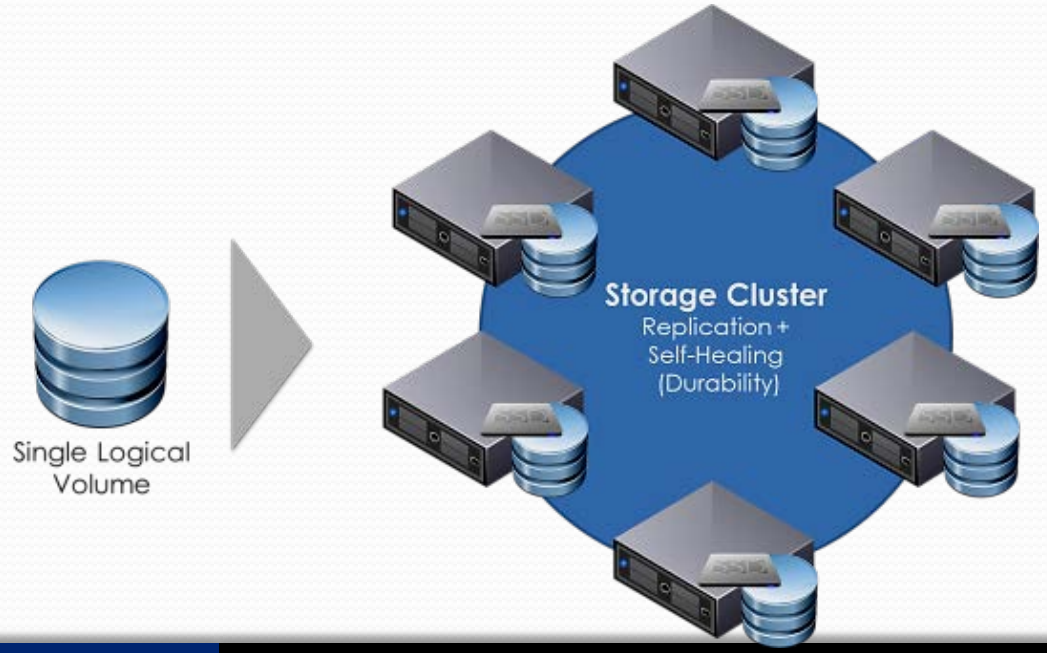
- Pros
  - Have been well-developed to meet confidentiality, availability and integrity
  - Work best with structured data
  - Use standard query language
  - ACID
  - **Very good for small dataset**

Why relational database is not suitable ?

- Cons
  - Scaling
    - Relied on *scale up* rather than *scale out*
  - Large feature set
  - Non-linear query execution time
  - Static schema

Reasons for Distributed Storage and Processing

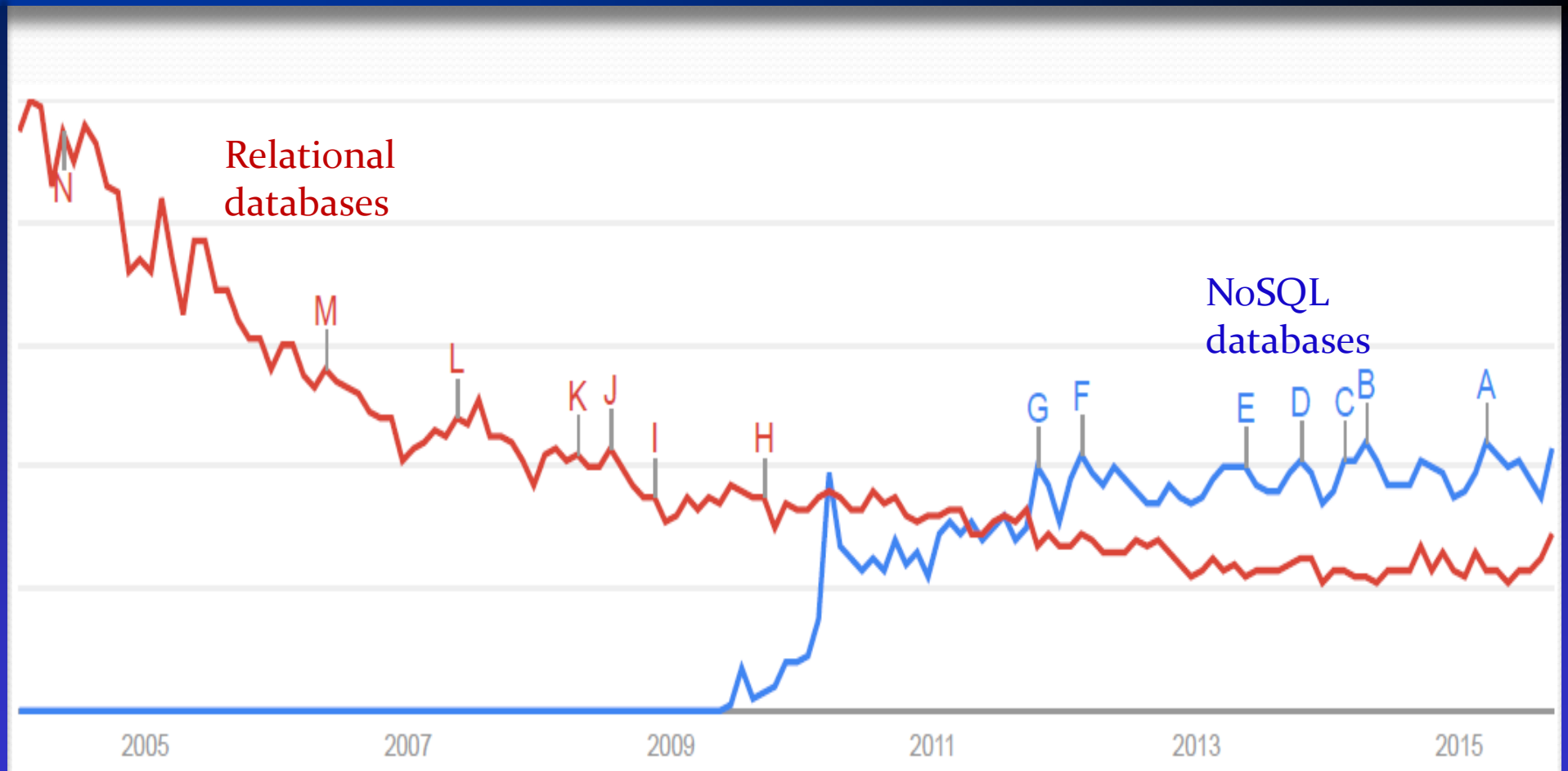
- Take advantage of multiple systems as well as multi-core CPU architectures
- Servers have to be globally distributed for low latency and failover



Characteristics of NoSQL Databases

- NoSQL databases have been designed for solving the Big Data issue by utilizing distributed, collaborating hosts to achieve satisfactory performance in data storage and retrieval.
- Mostly being non-relational
  - No join / Unstructured data
- Provide great performance, availability, scalability and flexibility
- Distribution, Replication, Failover

NoSQL Trend



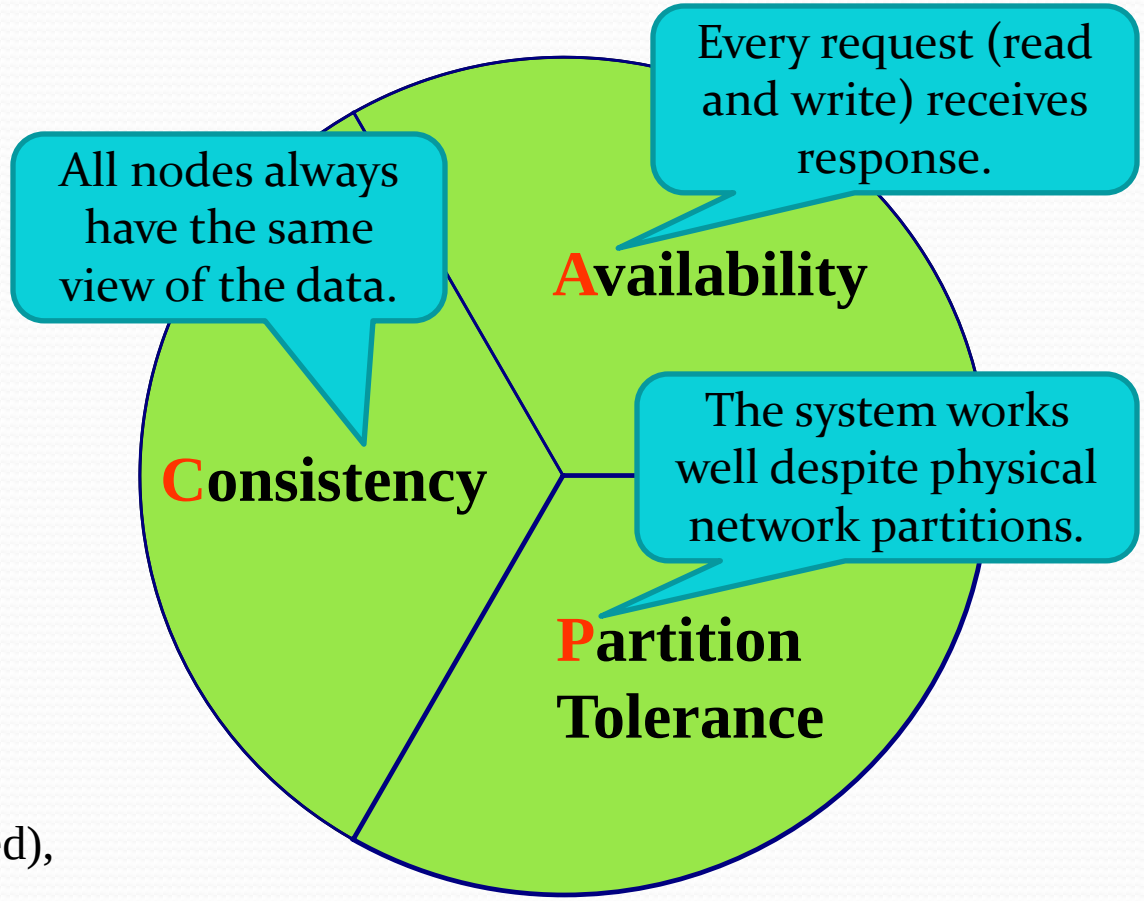
Google Trend

Characteristics of NoSQL Databases

- Provide BASE (Basically Available, Soft state, Eventual consistent) system, but not ACID as a Relational Database Management System.
- Schema-free
- Easy replication support and running well on clusters
- Simple API

CAP Theorem

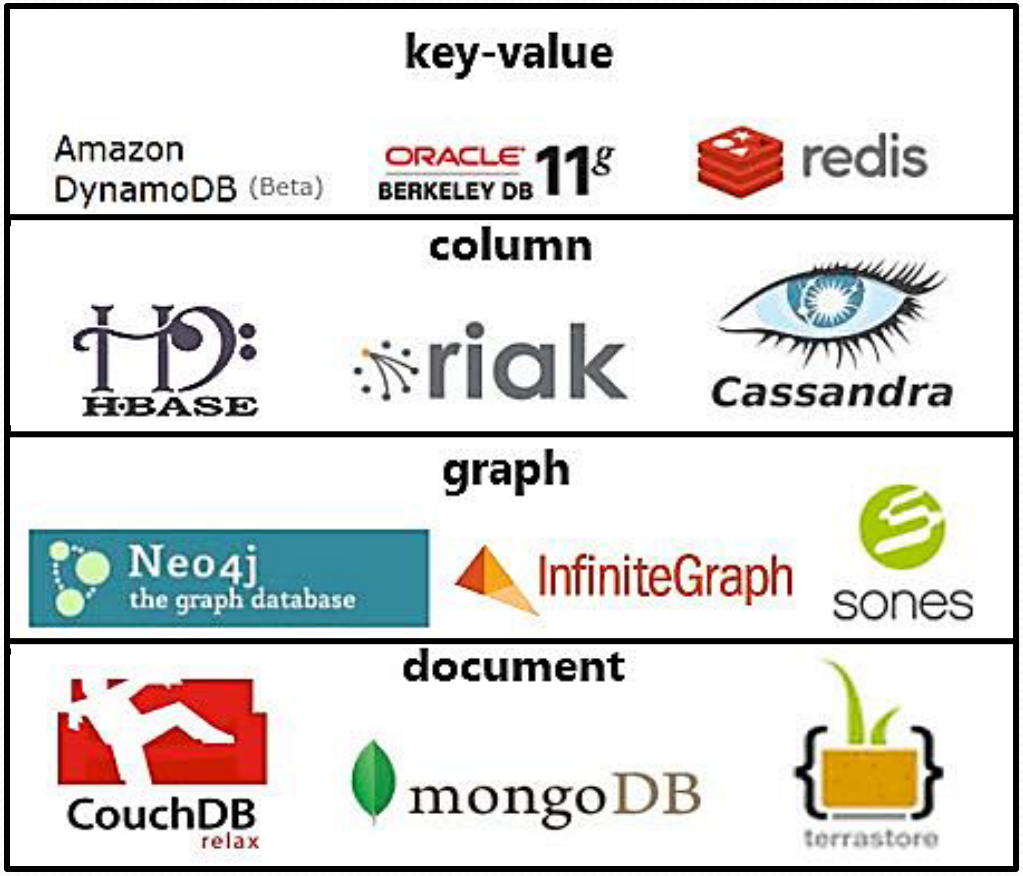
- Any shared-data system can have **at most two** of these properties
  - **AP**
    - Voldemart (Key-value)
    - CouchDB (Document),
    - Riak(Document)
  - **CA**
    - Relational databases
    - Vertica (column-oriented)
    - GreenPlum (Relational)
  - **CP**
    - BigTable (Column Oriented),
    - MongoDB(Document)



Types of NoSQL Databases

NoSQL Data Models

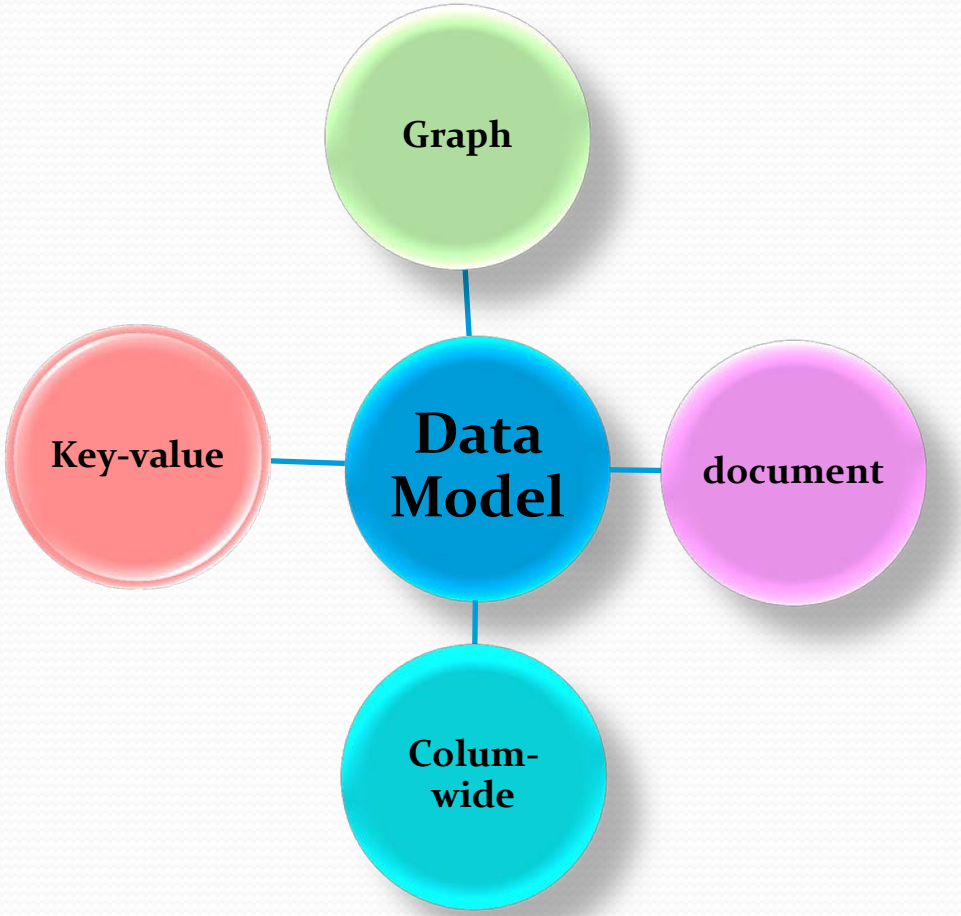
- There are **more than 150** NoSQL databases



Major Companies using NoSQL Databases

Company Name	NoSQL Name	NoSQL Storage Type
Adobe	HBase	Column
Amazon	Dynamo | SimpleDB	Key-Value | Document
BestBuy	Riak	Key-Value
eBay	Cassandra | MongoDB	Column | Document
Facebook	Cassandra | Neo4j	Column | Graph
Google	BigTable	Column
LinkedIn	Voldemort	Key-Value
LotsOfWords	CouchDB	Document
MongoHQ	MongoDB	Document
Mozilla	HBase | Riak	Column | Key-Value
Netflix	SimpleDB | HBase | Cassandra	Document | Column | Column
Twitter	Cassandra	Column

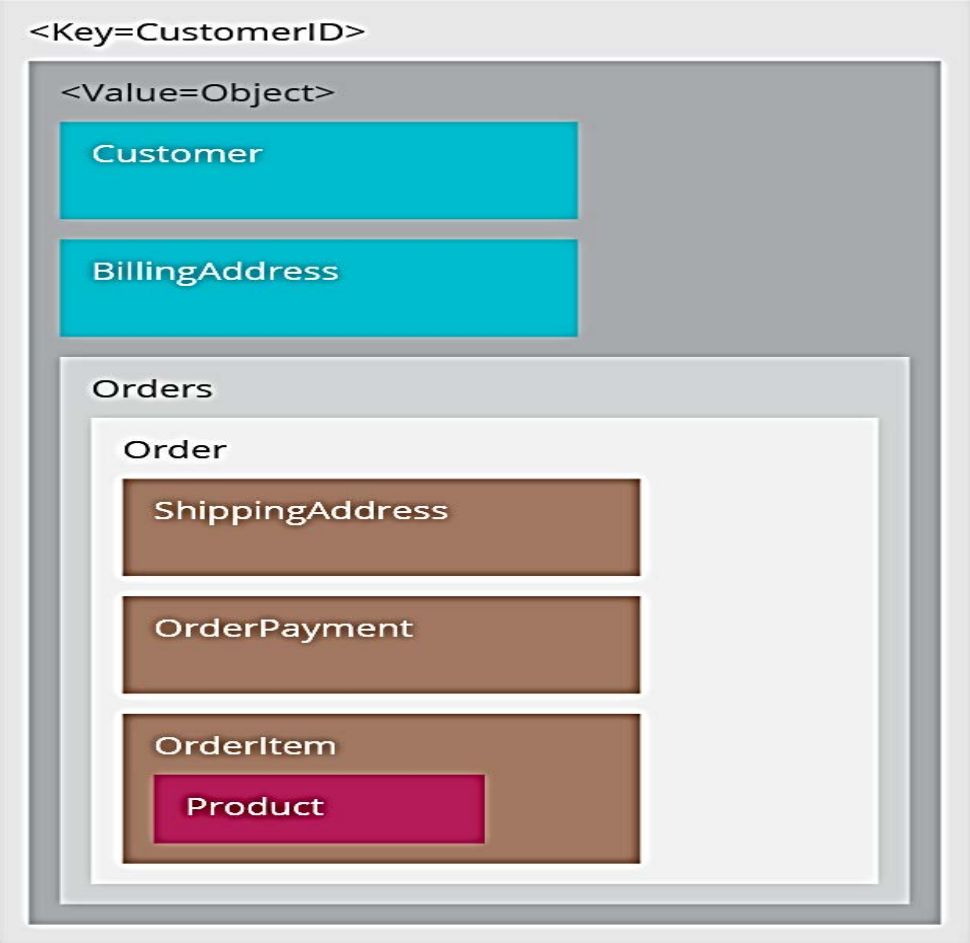
NoSQL Data Models



Key-value Stores

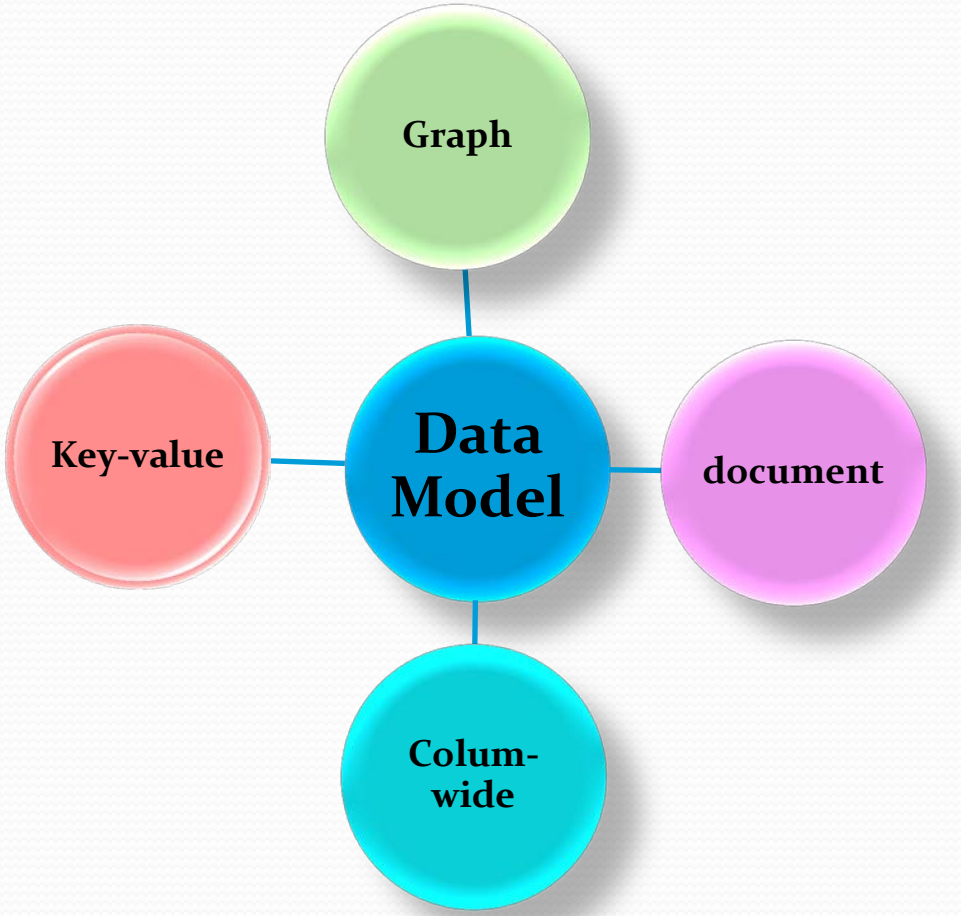
- Work by matching keys with values, similar to a dictionary
  - very fast
  - very scalable
  - simple model
  - able to distribute horizontally
- **Cons:** many data structures (objects) can't be easily modeled key value pairs

Key-value Stores

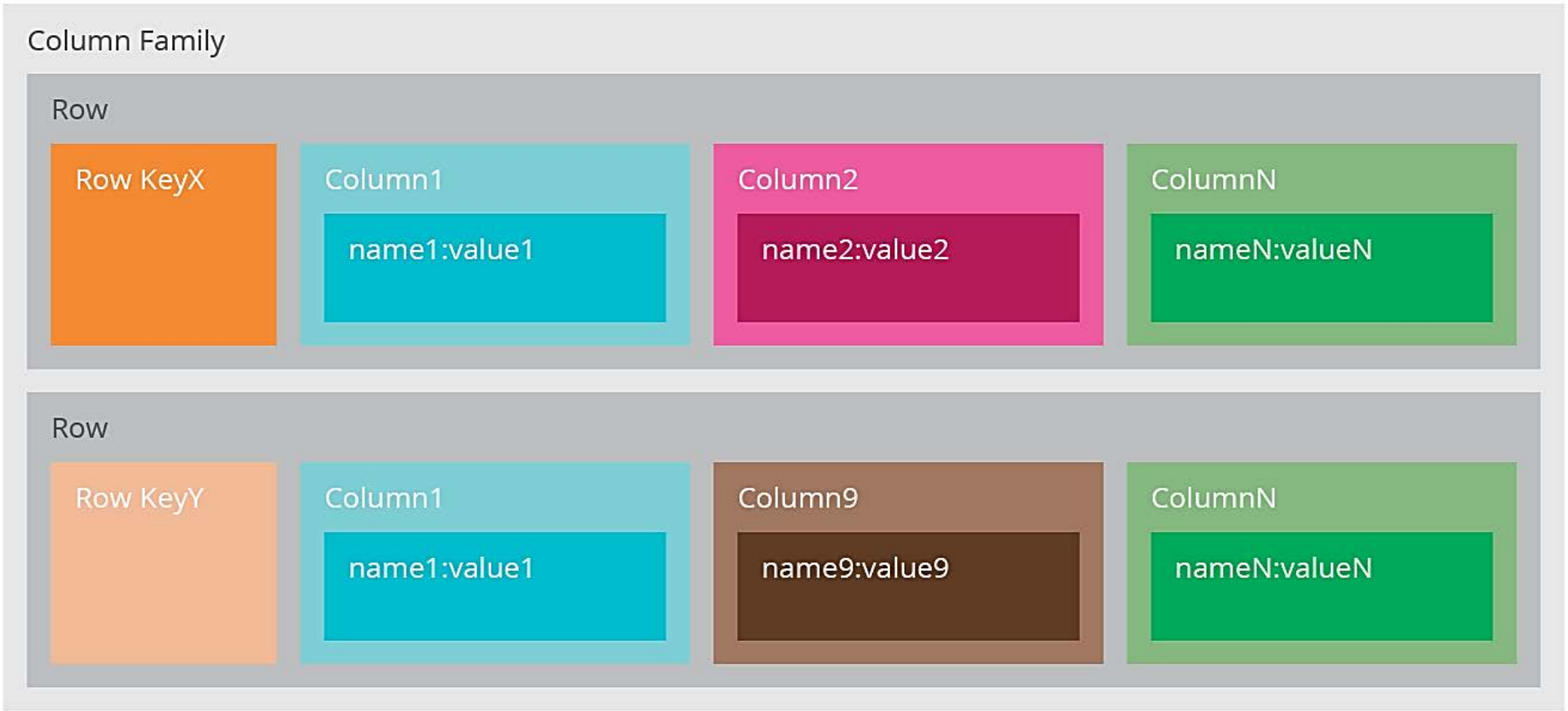


© http://www.thoughtworks.com/insights/blog/nosql-databases-overview

NoSQL Data Models

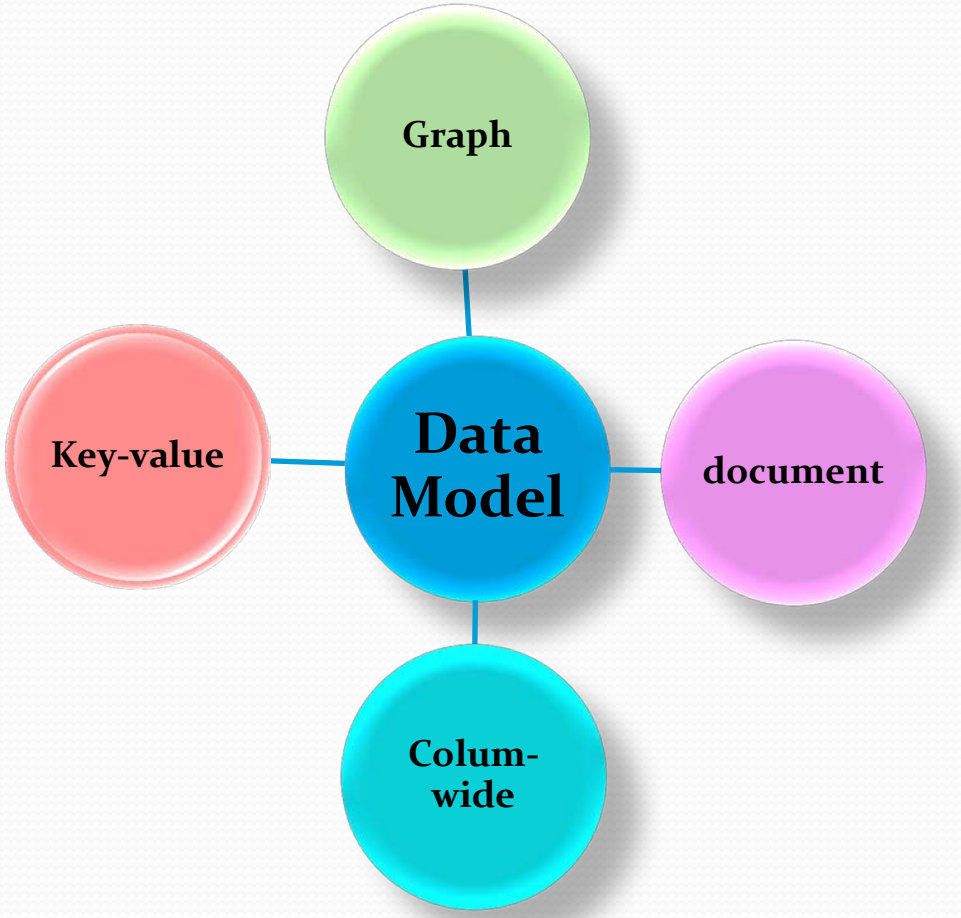


Column-Wide Stores



© http://www.thoughtworks.com/insights/blog/nosql-databases-overview

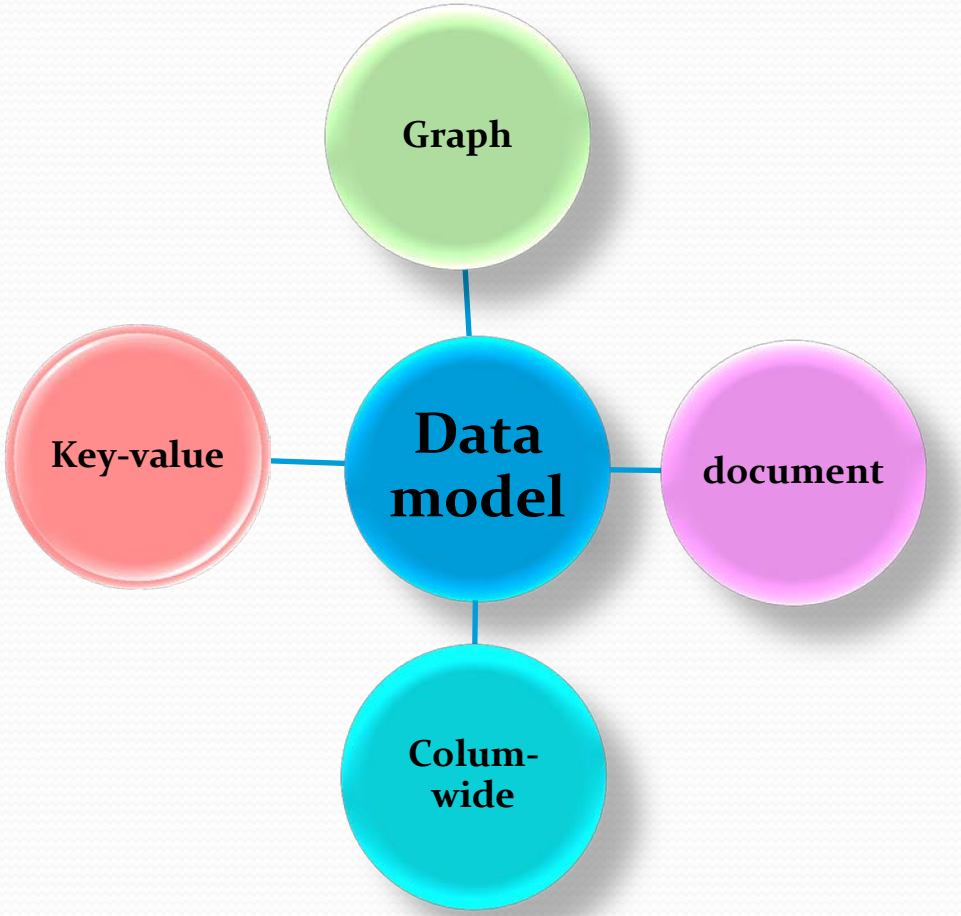
NoSQL Data Models



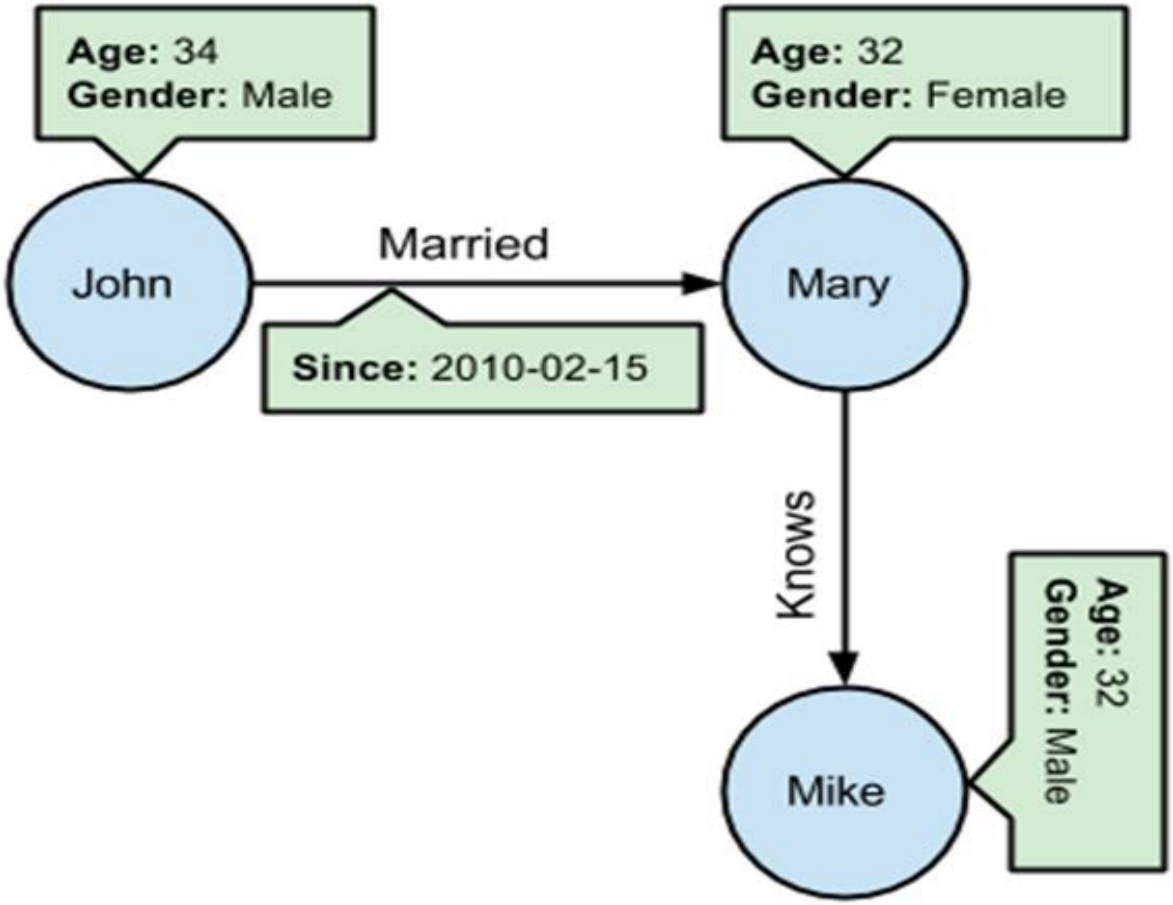
Document Stores

http://www.thoughtworks.com/insights/blog/nosql-databases-overview

NoSQL Data Models



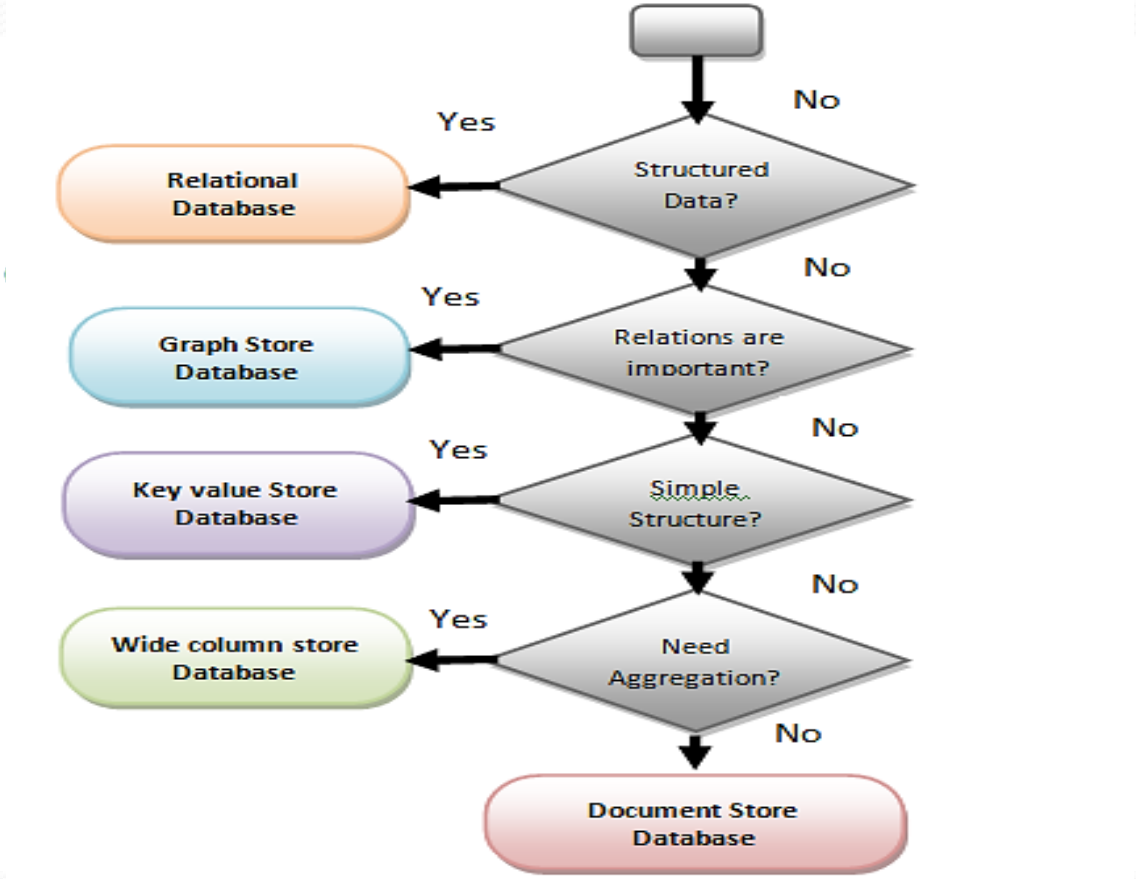
Graph Stores



© http://scraping.pro/where-nosql-practically-used/

Which one is the best?

- It depends on the application requirements
  - Size of data
  - Complexity
  - CAP theory
  - Format of data



NoSQL Security Challenges

NoSQL Security

- Most of NoSQL databases do not provide any feature of embedding security in the database itself.
  - Developers need to impose security in the middleware.
- Security issues that affected RDBMSs were also inherited in the NoSQL databases as well as new ones imposed by their new features.

NoSQL Security

- Security may be difficult
  - Owing to the unstructured (dynamic) nature of the data stored in these databases
  - Distributed environment
  - Cost of security in contrast to prformance
  - No strong consistency

NoSQL Major Security Challenges

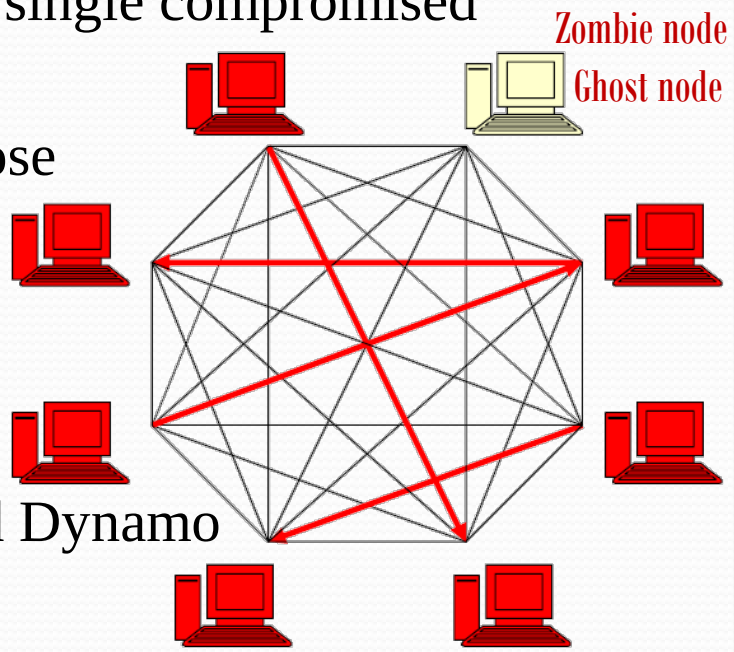
- Threats Posed By Distributed Environments
- Authorization and Access Control
- Safeguarding Integrity
- Protection of Data at Rest
- User Data Privacy

NoSQL Major Security Challenges

- **Threats Posed By Distributed Environments**
- Authorization and Access Control
- Safeguarding Integrity
- Protection of Data at Rest
- User Data Privacy

Threats Posed By Distributed Environments

- Distributed Environments increase attack surface across several distributed nodes
- Compromised Clients
  - Malicious data gets propagated from a single compromised location
  - Protecting nodes, name servers and those clients becomes difficult especially when there is no central management security point.
  - Vulnerabilities of **Gossip** based membership protocol in Cassandra and Dynamo [Aniello, et al. 2013]



NoSQL Major Security Challenges

- Threats Posed By Distributed Environments
- **Authorization and Access Control**
- Safeguarding Integrity
- Protection of Data at Rest
- User Data Privacy

Authorization and Access Control

- Two important challenges:
  - **Possibility:** how to define security policies for schema-less or dynamic-schema databases?
  - **Performance:** availability vs. access control overhead: how to manage cost of access control?

Authorization and Access Control

- **Fine-grained (row or column level) access control:**
  - heterogeneous data is stored together in one database as opposed to relational models which conform to defined schemas and tables that store only related data.
  - Schema-less nature of NoSQL DBs does not allow fine-grained access control.
  - Most of them allow Column Family level authorization.

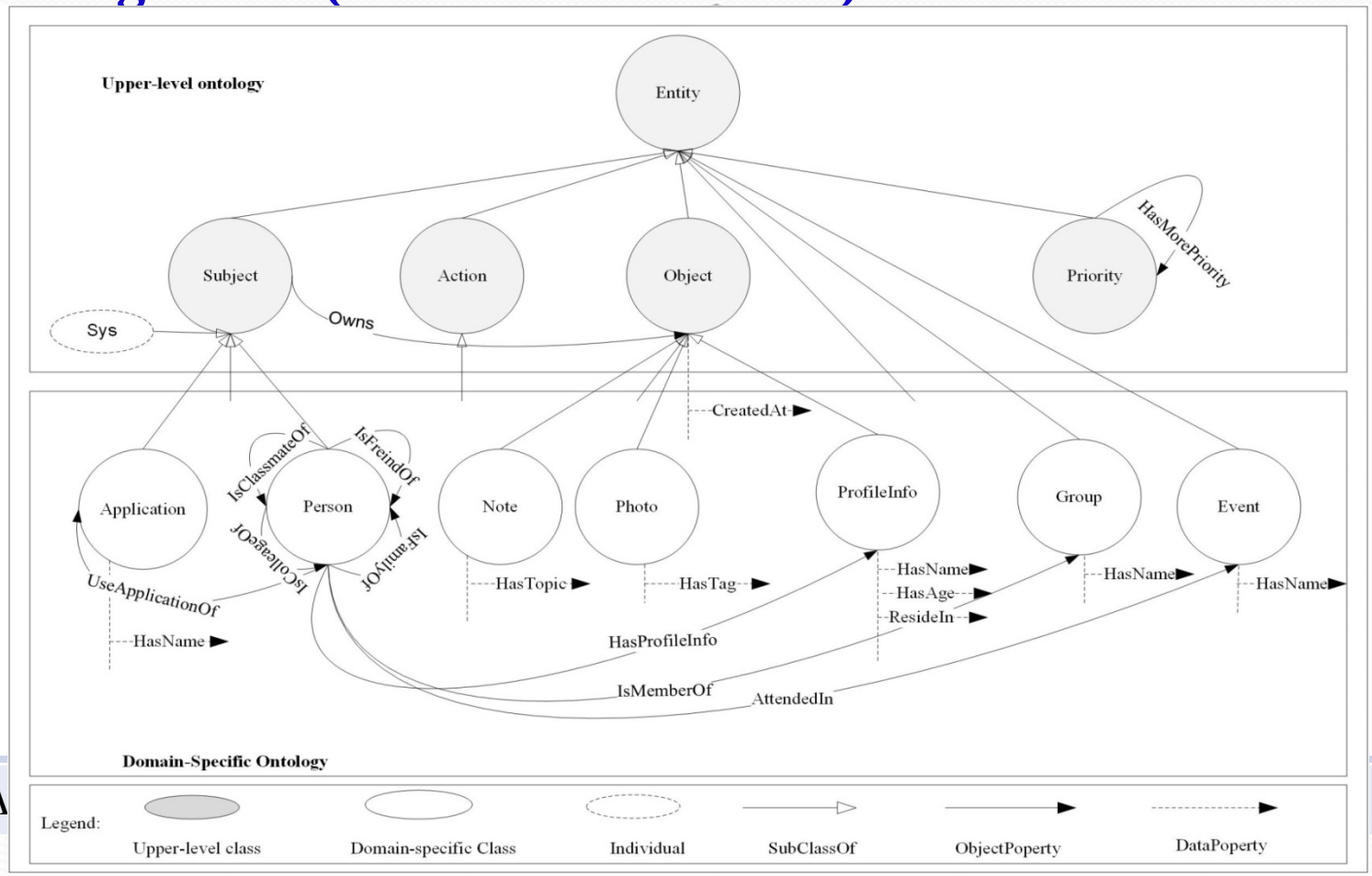
We need Looking Forward Security

NoSQL DBMS	Granularity	Explanation
BigTable	Column Family	Using ACL
Cassandra	Column Family	Using IAuthorizer API
HBase	Column Family / Cell	Group-based authorization
Accumulo	Cell	Using Visibility field

Authorization and Access Control

- **Fine-grained (row or column level) access control:**

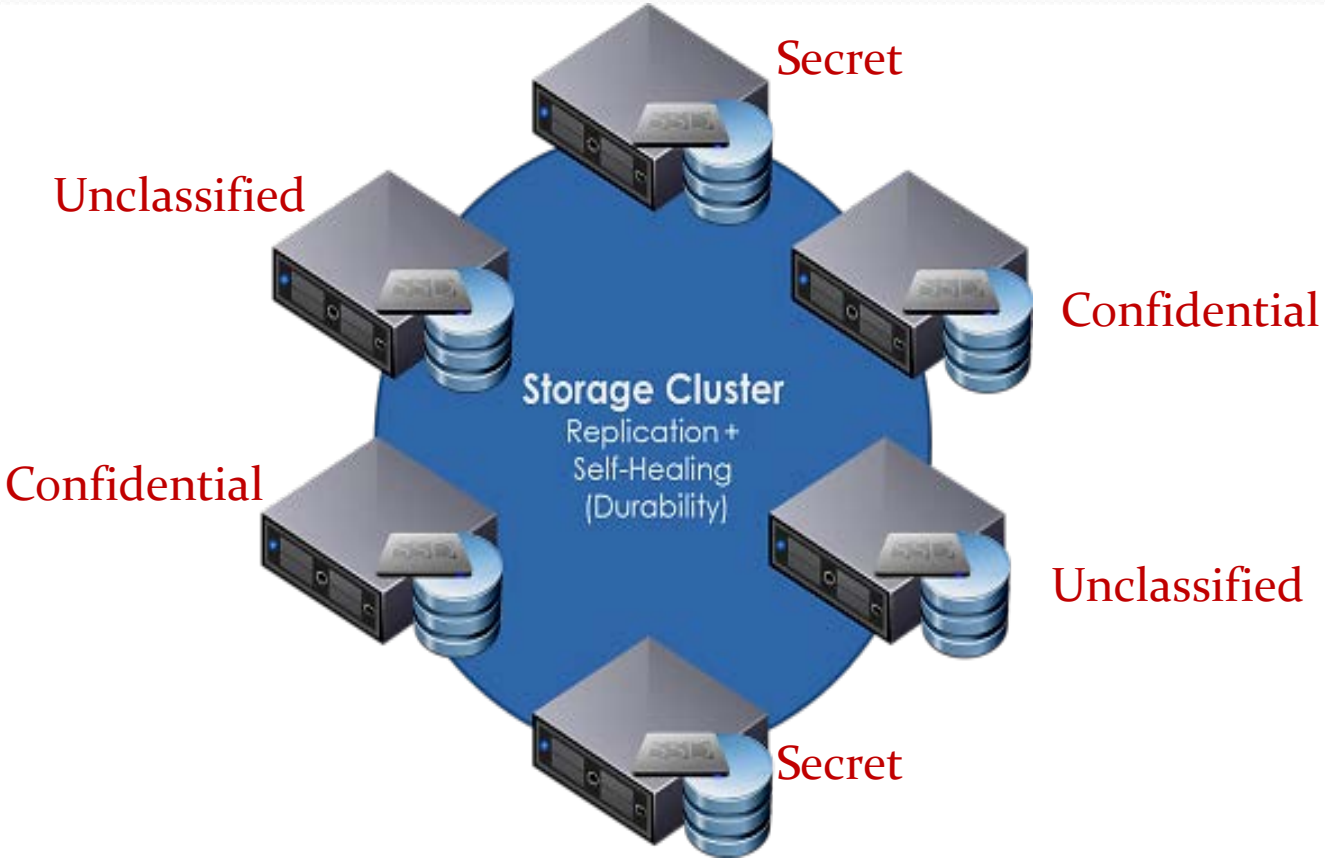
Using top-level ontology for access policy specification



Authorization and Access Control

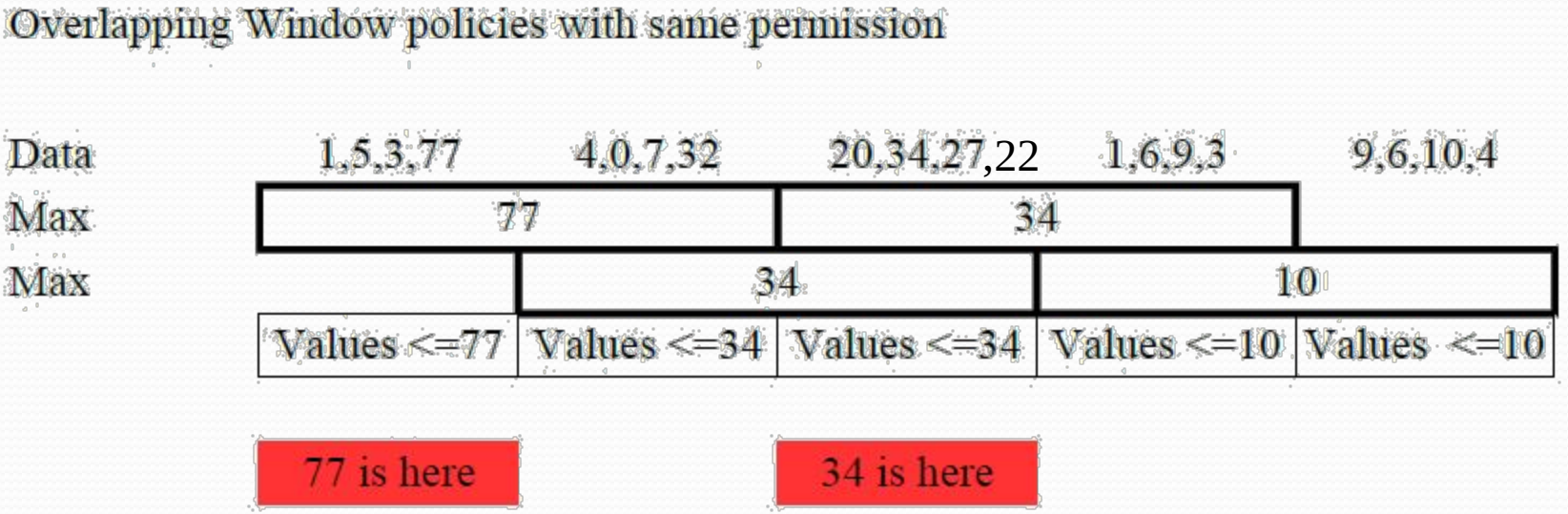
- **Fine-grained (row or column level) access control:**

Grouping data with the same security level (node level).



Authorization and Access Control

- **Inference Control:** Access control on aggregated data, especially in Column-Wide databases and Time-Series databases.



Authorization and Access Control

- **Administration / Access Control Management**: how and where to grant database accesses
  - **Local** vs. **Global** access policies and their possible conflicts.
  - **Centralized approach:** single-point-of-failure, availability issues
  - **Distributed approach:** consistency of distributed access rules
  - **Semidistributed approach:**

Authorization and Access Control

- By default, there is no authorization.
- Privileged admins can grant the privileges on resources to a selected user.

- By default, there is no authorization.
- Provisions authorization on a per--database level by using a role--based approach.

NoSQL Major Security Challenges

- Threats Posed By Distributed Environments
- Authorization and Access Control
- **Safeguarding Integrity**
- Protection of Data at Rest
- User Data Privacy

Safeguarding Integrity

- Enforcing integrity constraints is much harder in NoSQL database system
  - Consistency is in contrast with availability and performance
  - **Transactional integrity** is in contrast with its soft nature
- ***How to define integrity constraints?*** [its schema-less nature]
- ***Which types of integrity constraints can be defined?***
- ***How to control?*** [there is absence of central control/ performance and availability issues]

NoSQL Major Security Challenges

- Threats Posed By Distributed Environments
- Authorization and Access Control
- Safeguarding Integrity
- **Protection of Data at Rest**
- User Data Privacy

Protection of Data at Rest

- **Encryption** is widely regarded as the defacto standard for safeguarding data in storage.
- Most industry solutions offering encryption services lack horizontal scaling and transparency required in the NoSQL environment.

- Only a few categories of NoSQL databases provide mechanisms to protect data at rest by employing **encryption techniques**.

We need Light Weight Cryptography!

Protection of Data at Rest

- Use Transparent Data Encryption (TDE) to protect data that is written to disk.
- The commit log is not encrypted at all.

- Data files in MongoDB are never encrypted.

NoSQL Major Security Challenges

- Threats Posed By Distributed Environments
- Authorization and Access Control
- Safeguarding Integrity
- Protection of Data at Rest
- **User Data Privacy**

Users Data Privacy

- **Privacy**, main challenge of Web 2.0 and Virtual Social Networks.
- Large amounts of user- related sensitive information in NoSQL databases.
- ***Which kinds of methods is applicable in practice for NoSQL databases?***
  - Access Control
  - Encryption
  - Anonymization
  - ...

NoSQL Minor Security Challenges

- **Authentication (Users and Clients)**
- **Audit And Logging**
- **Protection of Data at Motion**
- **API Security**

Authentication

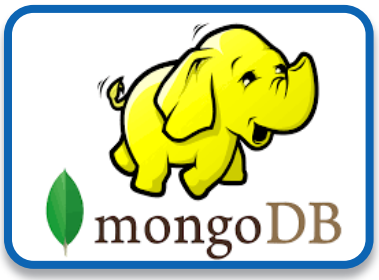
- By default, there is no authentication.
- Has Password Authenticator.
- Can further provide Kerberos authentication.

- By default, there is no authentication.
- Support for authentication on a per--database level.

Audit and Logging

- NoSQL databases has poor logging and log analysis methods

- Auditing is available in Enterprise Cassandra.
- Filters are available for logging



- MongoDB is far behind in implementing the desired security logging and monitoring.

Protection of Data in Motion

- Communication between clients and nodes (traditional issue)
- Communication between nodes
  - RPC over TCP/IP

Protection of Data in Motion

- **Client-Node Communications:** By default, is not encrypted. SSL can be configured.
- **Inter-Node Communications:** By default, is not encrypted. SSL can be configured**.**

- **Client-Node Communications:** it is required to either recompile MongoDB with the "--ssl" option.
- **Inter-Node Communications:** is not supported.

API Security

- APIs can be subjected to several attacks such as **Code injection**, **buffer over flows**, **command injection** as they access the NoSQL databases.
- Server Side JavaScript Injection (SSJS)
  - Schema injection / Query injection / JSON injection

➢ **In PHP:**

```
$query = 'function() {var search_year = \'' .$_GET['year'] . '\';' .
'return this.publicationYear == search_year || ' . ' this.filmingYear
== search_year || ' . ' this.recordingYear == search_year;}';
$cursor = $collection->find(array('$where' => $query));
```

- DoS Attacks

```
http://server/app.php?year=1995';while(1);var%20foo='bar
```

API Security

- Is vulnerable to injection

- MongoDB, $where operator can be used for injection.

Summary

- NoSQL Database Systems for unstructured and big data
  - Main features: Performance, Availability, Scalability
- **NoSQL Security Challenges:**
  - Threats posed by their distributed nature
  - Fine-grained authorization and inference control
  - Integrity constraint definition and control
  - Light weight transparent encryption of data in rest
  - Users' privacy
  - ...

Some References

- [Aniello, et al. 2013] L. Aniello, S. Bonomi, M. Breno, R. Baldoni, “Assessing Data Availability of Cassandra in the Presence of non-accurate Membership”, *The 2nd International Workshop on Dependability Issues in Cloud Computing*, 2013.
- [Kadebu , et al. 2014] P. Kadebu, I. Mapanga, A Security Requirements Perspective towards a Secured NOSQL Database Environment, *International Conference of Advance Research and Innovation*, 2014.
- [Noiumkar, et al. 2014] P. Noiumkar, and T. Chomsiri, A Comparison the Level of Security on Top 5 Open Source NoSQL Databases, *The 9th International Conference on Information Technology and Applications*, 2014.
- [Fidelis Cybersecurity 2014] Current Data Security Issues of NoSQL Databases, Fidelis Cybersecurity, 2014.
- [Okman, et al. 2011] L. Okman, N. Gal-Oz, Y. Gonen, E. Gudes, and J. Abramov, Security Issues in NoSQL Databases, *International Joint Conference of IEEE TrustCom-11/IEEE ICESS-11/FCST-11*, 2011.
- [Shermin 2013] M. Shermin, An Access Control Model for NoSQL Databases, *M.Sc. thesis*, The University of Western Ontario, 2013.
- [Ron, et al. 2015] A. Ron, A. Shulman-Peleg, E. Bronshtein, No SQL, No Injection? Examining NoSQL Security, *The 9th Workshop on Web 2.0 Security and Privacy*, 2015.
- [Rong, et al. 2013] C. Rong, Z. Quan, A. Chakravorty, On Access Control Schemes for Hadoop Data Storage, *International Conference on Cloud Computing and Big Data*, 2013.

Thanks for your attention …

Any Question?

amini@sharif.edu

Thank Ms Dolatnezhad for helping in preparing this presentation.