NoSQL

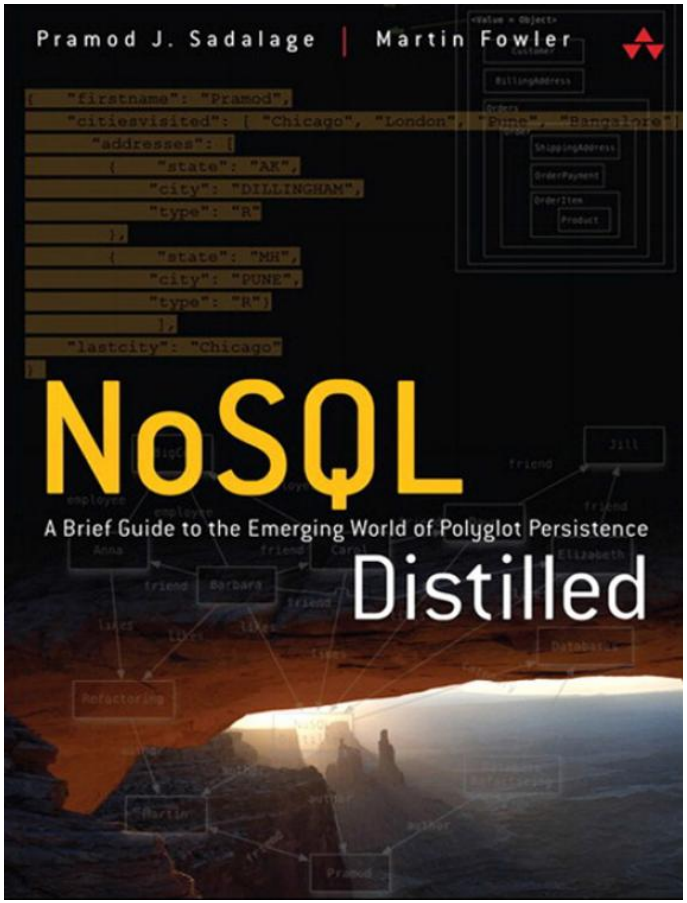
Stiati ca….

90% din datele existente pe Internet au fost produse in ultimii 2 ani ?

Referinte

- P.J.Sadalge, M.Fowler – *NoSQL Distilled*
- *http://nosql.mypopescu.com/*
- *http://leopard.in.ua/2013/11/08/nosql-world/*
- *https://www.youtube.com/watch?v=*

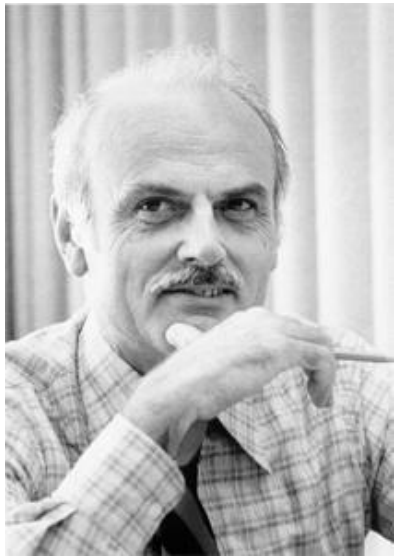
  nVWieRqU6HE, XPqrY7YEs0A, ql_g07C_Q5I, Eay2Ylhuf0k, HFplUBeBhcM, I_jZzT5gSEc



Baze de date – scurta istorie

- 1980 – 1990 – bazelor de date relationale (probleme legate de persistenta, consistenta, creare de rapoarte, translatarea *paper to DB* etc.) *Impedance mistmatch problem*

- Fiecare element al tabelei este contine informatie
- Toate celulele unei coloane sunt omogene
- Fiecare coloana are un nume unic
- Nu exista linii identice in tabel
- Ordinea randurilor si coloanelor este arbitrara

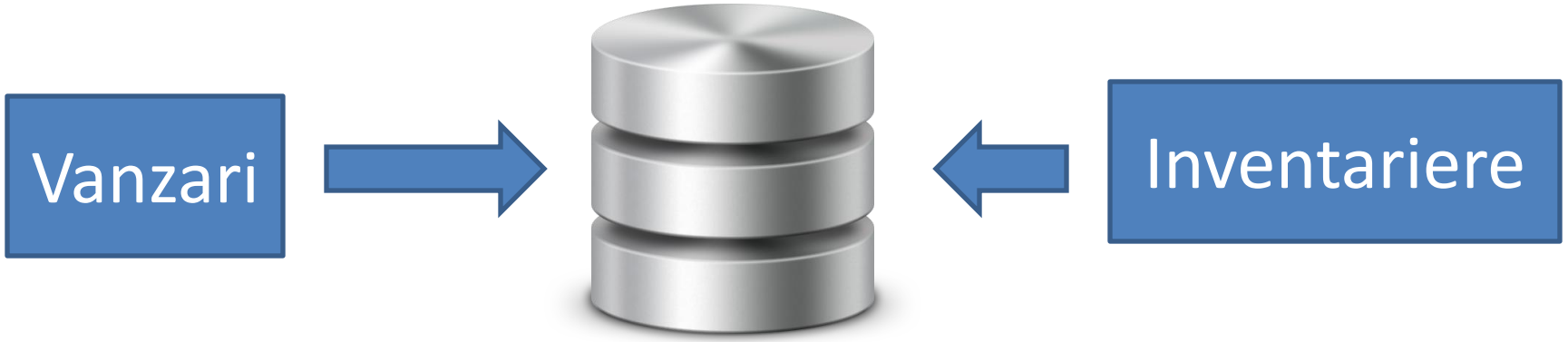


MySQL, PostgreSQL, DB2, Oracle

Edgar Codd

Baze de date – scurta istorie

- 1980 – 1990 –bazelor de date relationale (probleme legate de persistenta, consistenta, creare de rapoarte, translatarea info in baze de date etc.) *Impedance mistmatch problem*
- 1990 – 2000 – baze de date obiectuale *Relational dominance problem*



Baze de date – scurta istorie

- 1980 – 1990 –bazelor de date relationale (probleme legate de persistenta, consistenta, creare de rapoarte, translatarea info in baze de date etc.) *Impedance mistmatch problem*
- 1990 – 2000 – baze de date obiectuale *Relational dominance problem*
- 2000 – 2010 – baze de date relationale (din nou) de ce sunt insuficiente ?

Doua posibilitati de a scala:

- "*Buy bigger boxes*" – utilizarea de calculatoare mai performante [bun pt RDBMS]: Scale UP

  costuri mari, limite impuse de hardware

- "*Buy many boxes*" – paralelizarea datelor si a procesarii acestora [necesita alt model de stocare a datelor]: Scale OUT

  nu conteaza cate si cat de performante

Paralelizarea RDBMS ?!

- Au existat cativa “jucatori” care au incercat sa faca chestiunea asta. Concluzia: “*Very hard to do*”. Era nevoie de o noua abordare:

Bigtable

Dynamo

“NoSQL”

NoSQL – definitie ?!

- Nu pare prea ok sa definesti ceva prin ceea ce nu este…
- NoSQL (not only SQL)
- Aparut in “*late* 90’*s*”.
- Initial a fost numele unel baze de date open-source creata de *Carlo Strozzi* in care datele erau stocate ca fisiere ASCII si erau utilizate scripturi de tip shell in locul interogarilor SQL.

Carlo Strozzi

NoSQL – cine a facut termenul popular

- Origine: Johan Oskarsson, a participat la o conferinta pe aceste tehnologii in Iunie 2009 la San Francisco.
- S-au discutat conceptele intr-o sedinta de brainstorming [plecand de la Bigtable si de la Dynamo] si, ca orice adunare care se respecta, aveau nevoie de un twitter-hashtag:
- #nosql a fost sugerat de Eric Evans[*RackSpace*]

NOSQL meetup

Thursday, June 11, 2009 from 10:00 AM to 5:00 PM (PT)
San Francisco, CA

Ticket Information

TYPE	REMAINING	END		QUANTITY
Free ticket	**Sold Out**	Ended	Free	**Sold Out**

Share this! Share Tweet Like Be the first of your friends to like this.

Event Details

Introduction

This meetup is about "open source, distributed, non relational databases".

Have you run into limitations with traditional relational databases? Don't mind trading a query language for scalability? Or perhaps you just like shiny new things to try out? Either way this meetup is for you.

Join us in figuring out why these newfangled Dynamo clones and BigTables have become so popular lately. We have gathered presenters from the most interesting projects around to give us all an introduction to the field.

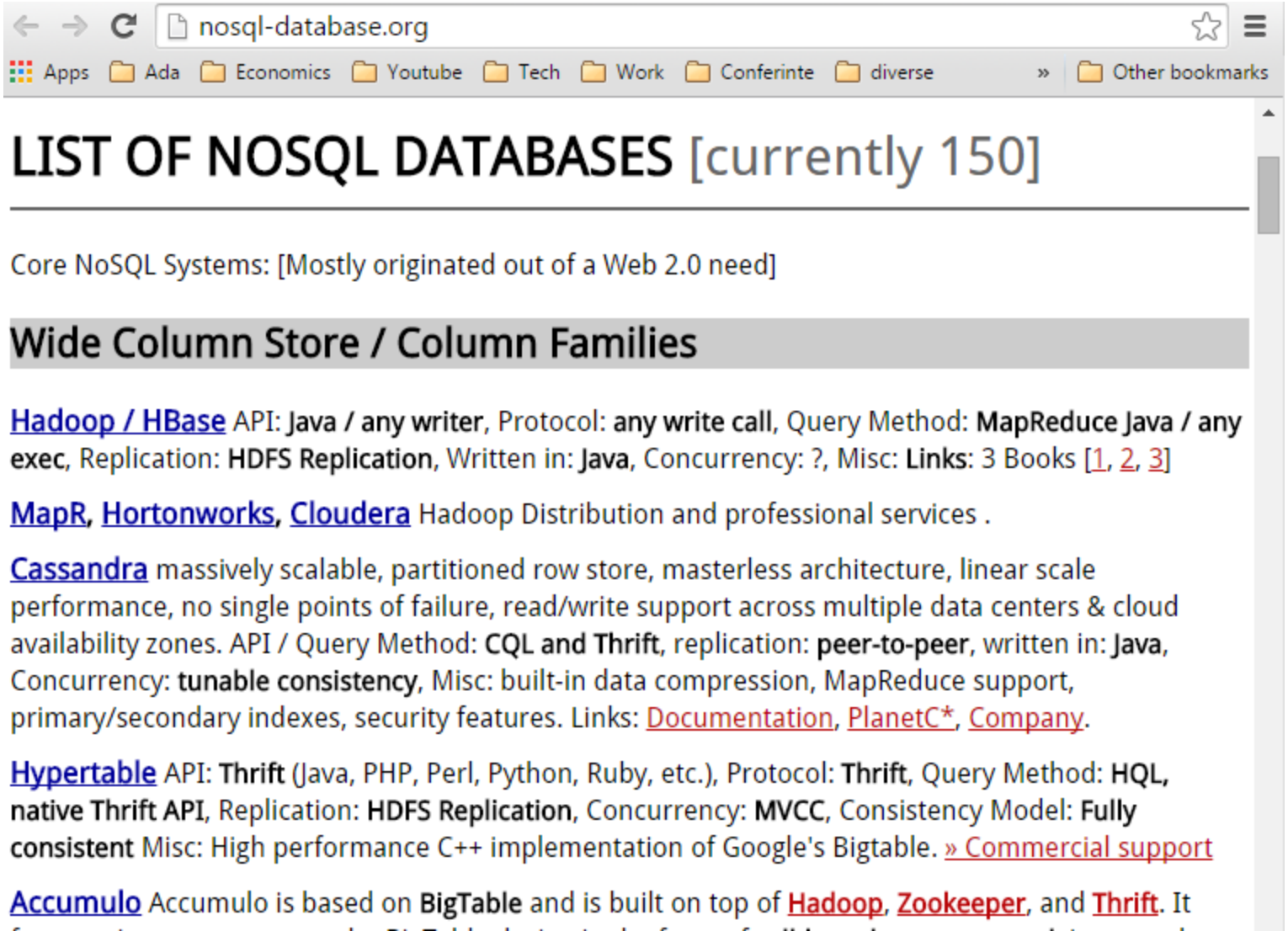
Preliminary schedule

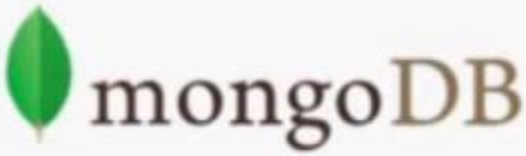
When & Where

CBS interactive, Magma room
235 Second Street
San Francisco, CA 94105

Thursday, June 11, 2009 from 10:00 AM to 5:00 PM (PT)

In prezent sunt aproximativ 150 tipuri de baze de date NoSQL (http://nosql-database.org/)




mongoDB


CouchDB
relax


Project Voldemort
A distributed database.


HYPERTABLE


Cassandra

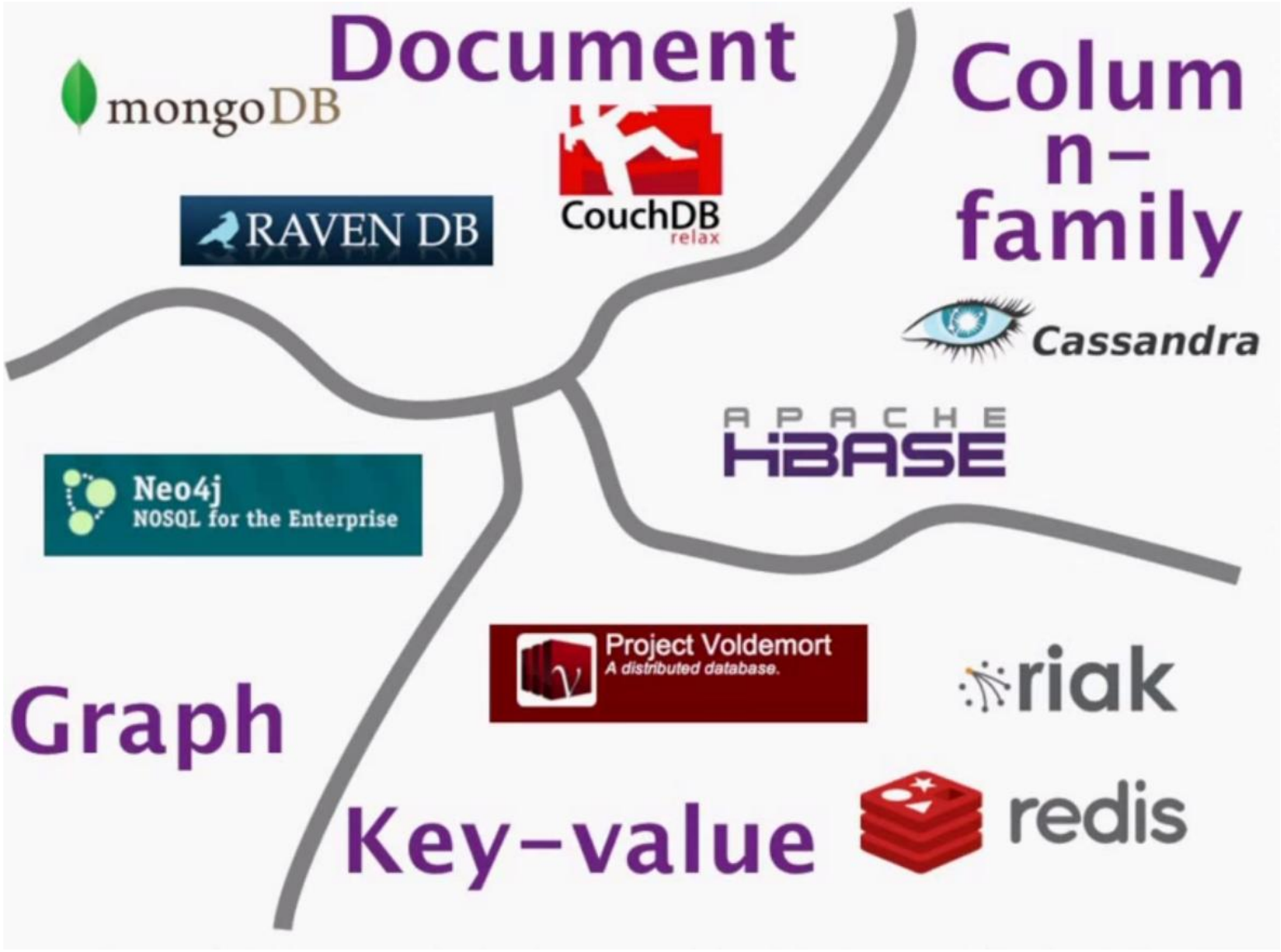

APACHE
HBASE

Dynomite

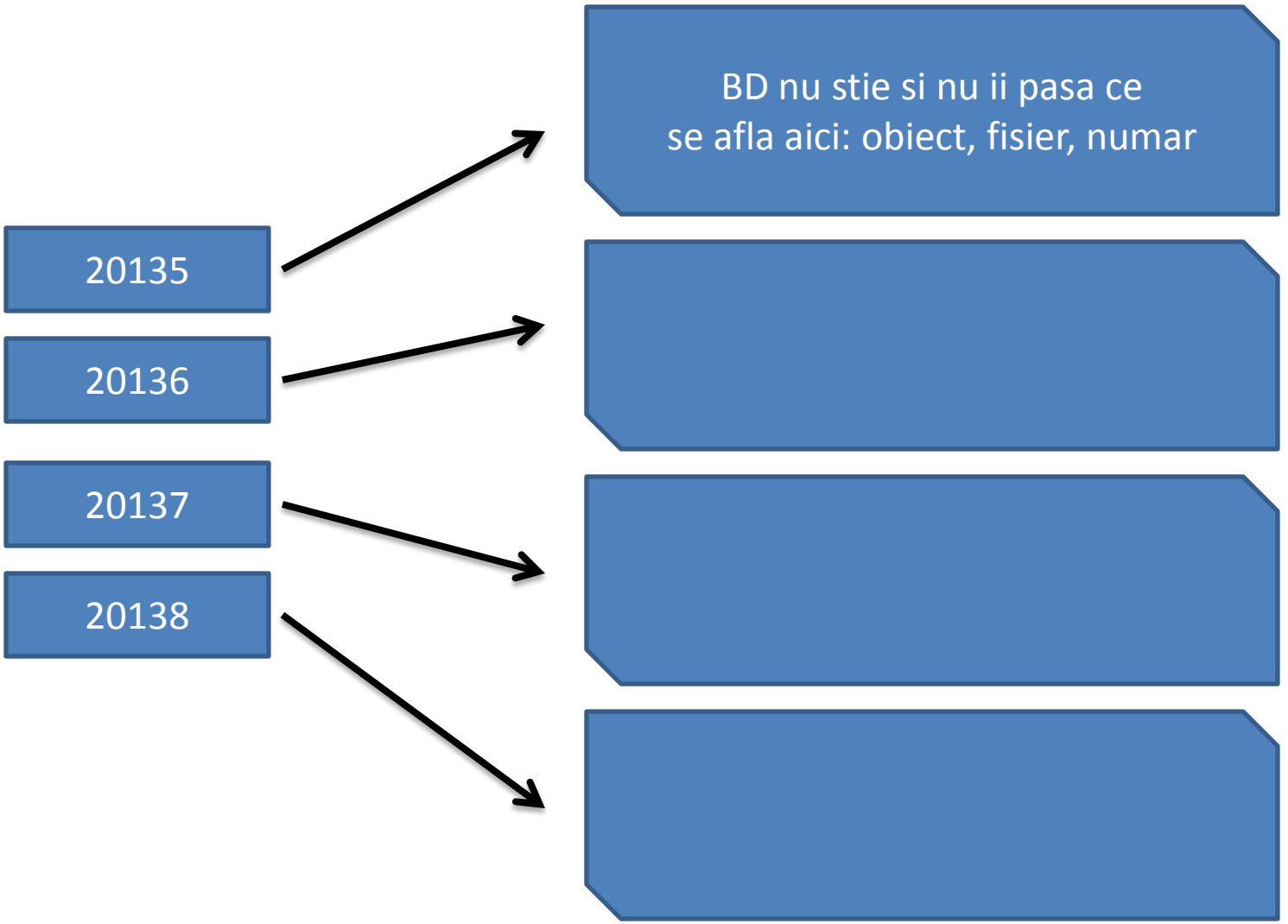
Ce este NoSQL atunci ?

- Martin Fowler – nu poate fi definit decat ca o serie de proprietati comune:
  - NON – RELATIONAL
  - OPEN SOURCE (desi exista *commercial tools*)
  - CLUSTER FRIENDLY (initial ideea) (~)
  - SCHEMA – LESS
  - 21ST CENTURY WEB
  - different DATA MODEL

Data Model


Document
mongoDB
CouchDB
relax
RAVEN DB
Colum
n-
family
Cassandra
APACHE
HBASE
Neo4j
NOSQL for the Enterprise
Project Voldemort
A distributed database.
riak
Graph
Key-value
redis

Cel mai simplu model: Key-Value



Modelul Document

[no schema]

```
{"id":20135,
 "titlu":'student',
 "an":2,
 "nume":'Becali',
 "prenume": 'George',
 "materii":["PSGBD":10, "TAP":9, "IP":10],
}
```

```
{"id":20136,
 "titlu":'student',
 "an":1,
 "nume":'Popescu',
 "prenume": 'Ionut',
 "prenume": 'Vasile',
 "materii":["FAI":7, "POO":9],
}
```

No Schema ?!?!

- Permite flexibilitate mai mare (putem adauga orice acolo)…
- Migrare de date mai usoara…

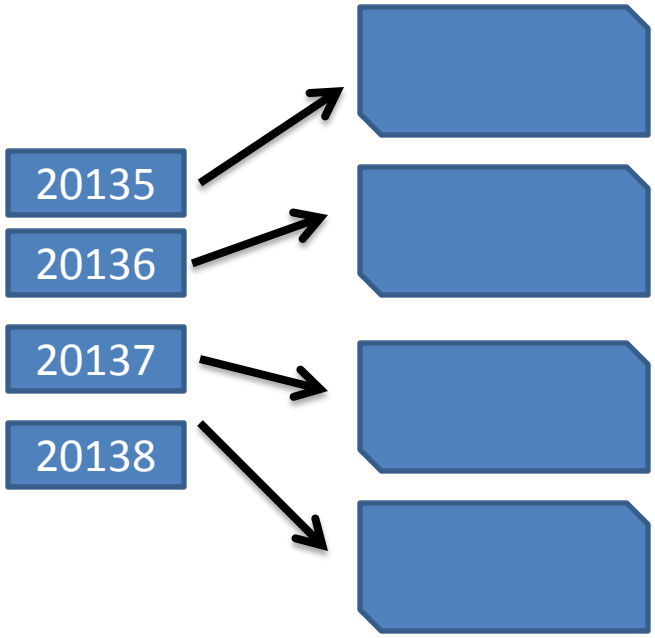
Toate bune si frumoase (de fapt mult mai bune si mult mai frumoase…). Pana ajungem sa interogam:

“vreau utilizatorii cu titlu=‘student’,din an=1”

Schema implicita

Schema implicita

Key – Value

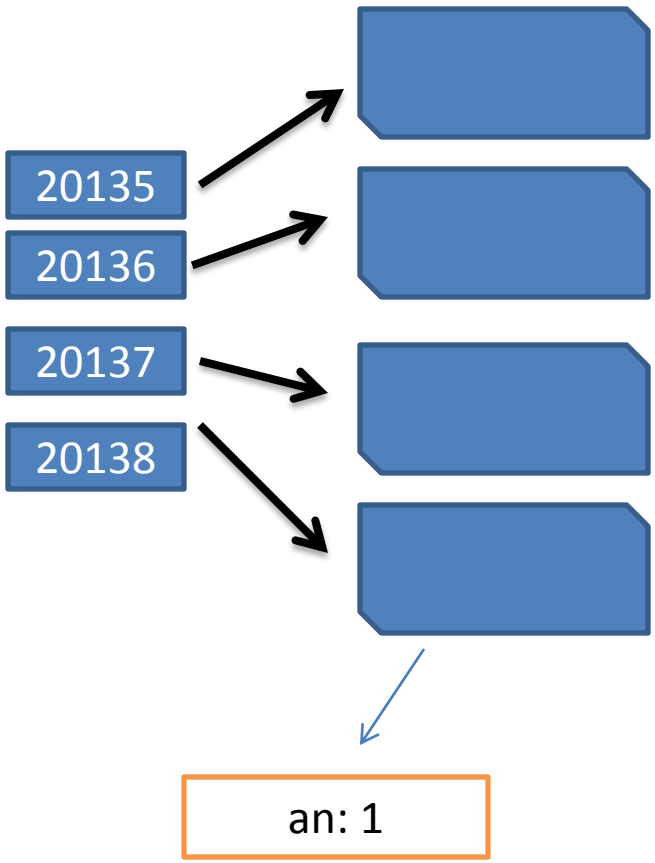


Document

```
{"id":20135,
 "titlu":'student',
 "an":2,
 "nume":'Becali',
```

```
{"id":20136,
 "titlu":'student',
 "an":1,
 "nume":'Popescu',
 "prenume": 'Ionut',
 "prenume": 'Vasile',
 "materii":["FAI":7, "POO":9],
}
```

Key – Value

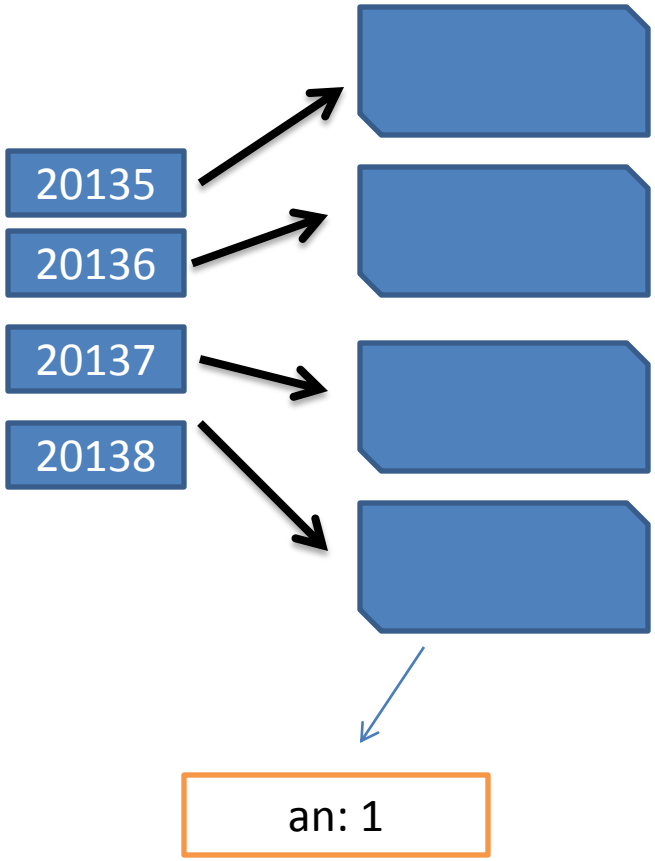


Document

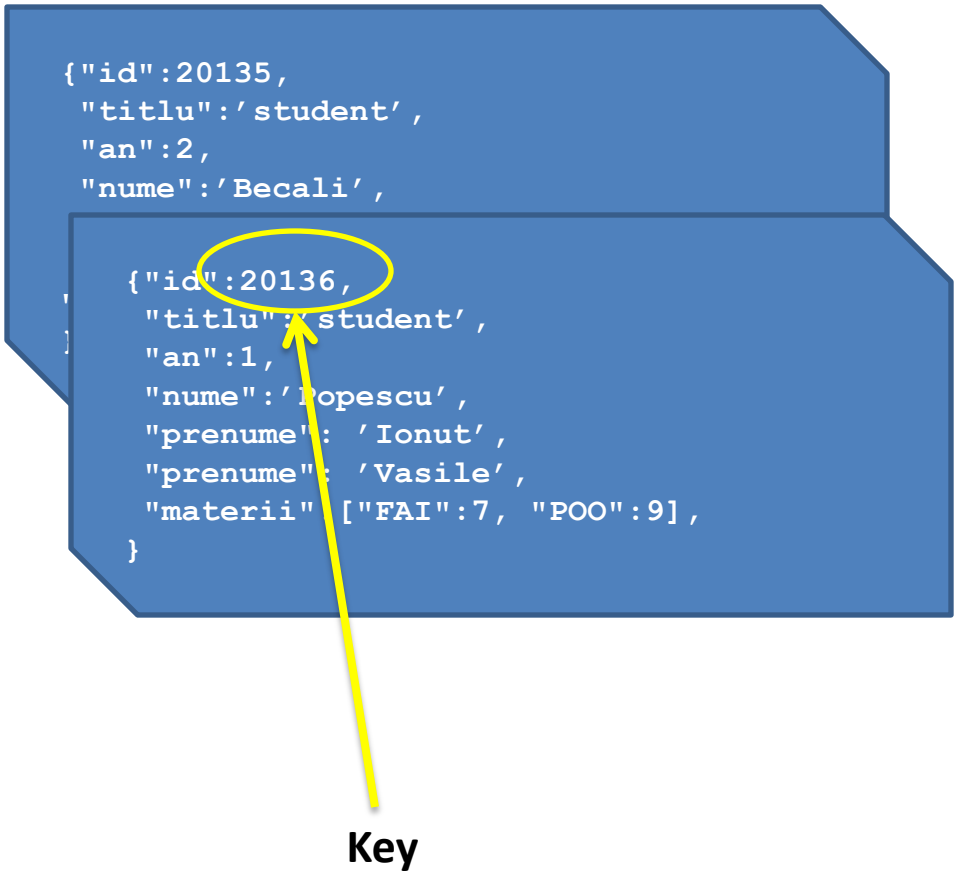
```
{"id":20135,
 "titlu":'student',
 "an":2,
 "nume":'Becali',

  {"id":20136,
   "titlu":'student',
   "an":1,
   "nume":'Popescu',
   "prenume": 'Ionut',
   "prenume": 'Vasile',
   "materii":["FAI":7, "POO":9],
  }
```

Pentru a indexa, as vrea sa stiu care dintre valori au an=1 (metadata)

Key – Value



Document



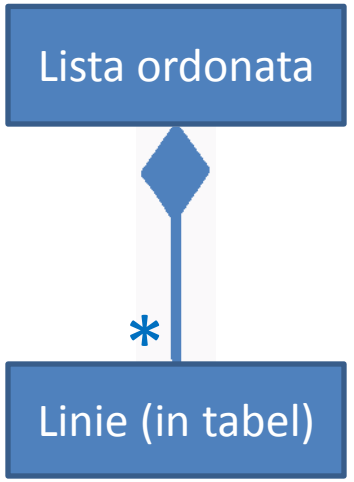
Pentru a indexa, as vrea sa stiu care dintre valori au an=1 (metadata)

Key – Value Document

- Nu conteaza ca sunt numite **key-value** sau sunt denumite **document**. Ceea ce e important este ca fac o agregare a unor date (obiecte: imagini, filme, pagini web).

- Din acest motiv ele se numesc si

Agregate-Oriented

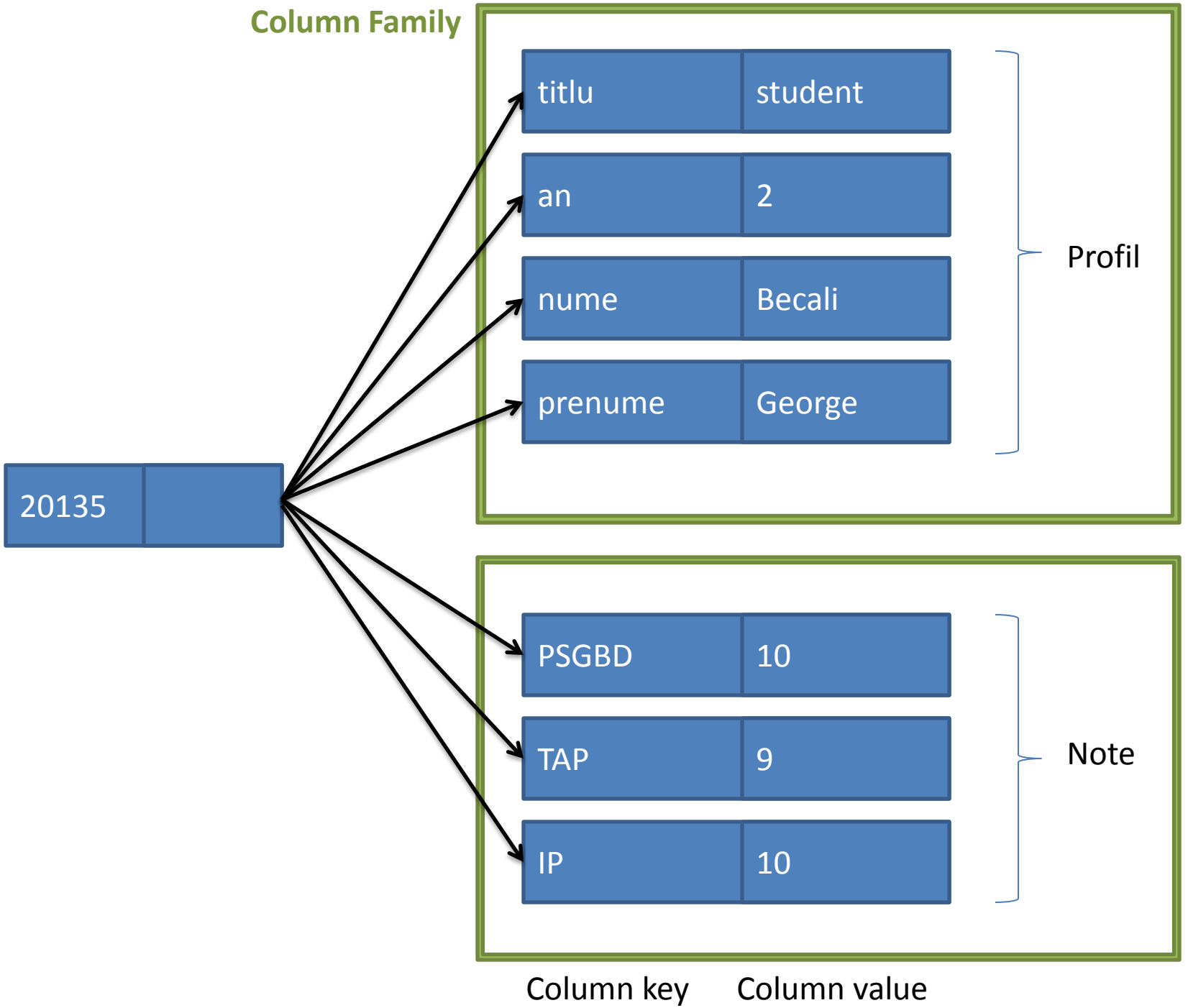


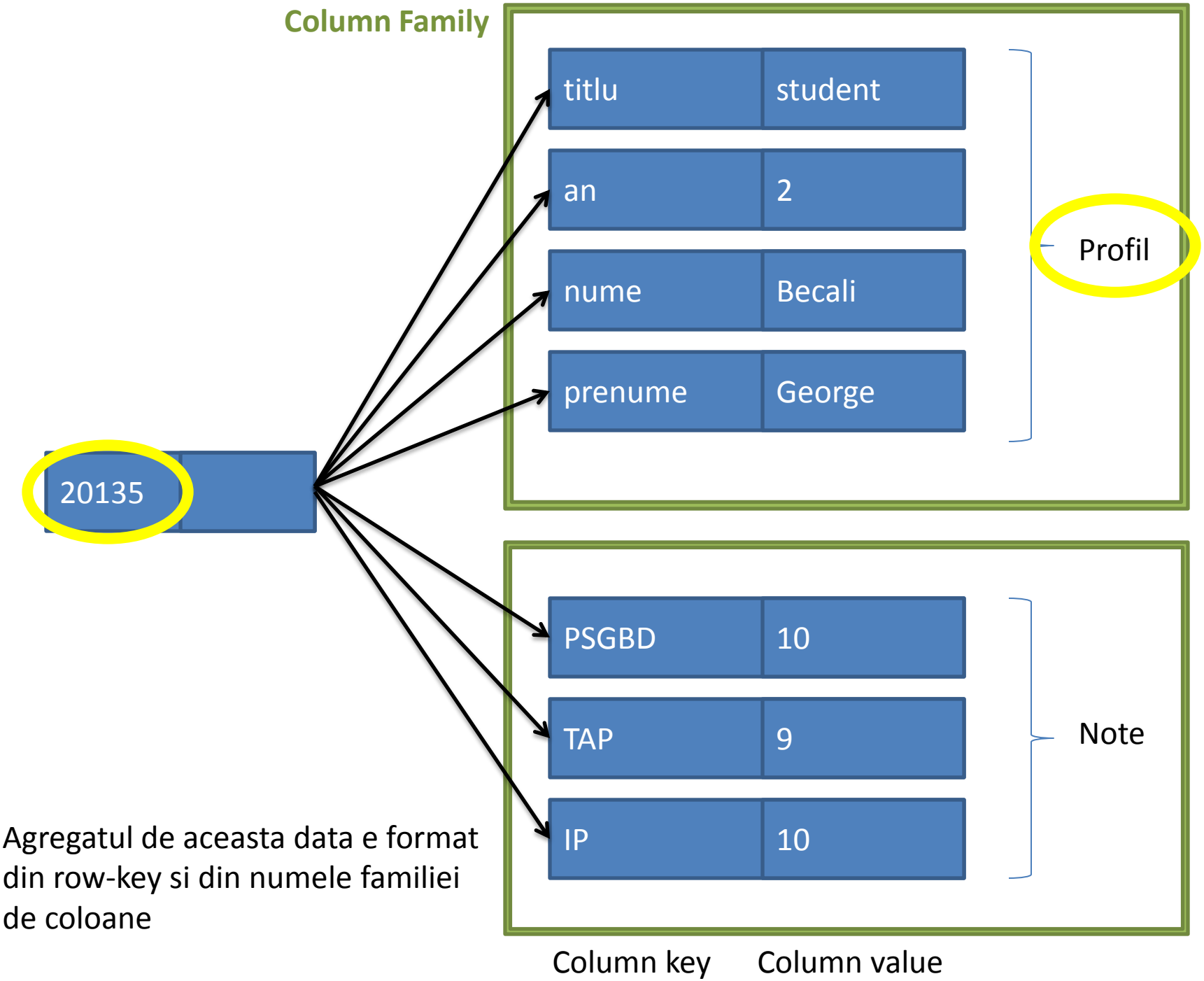
Agregate Oriented

- In RDBMS – obiectul agregat e imprastiat in mai multe tabele.
- In modelul Key-Value, obiectul agregat este secventa de key.
- In modelul document, obiectul agregat este chiar documentul.

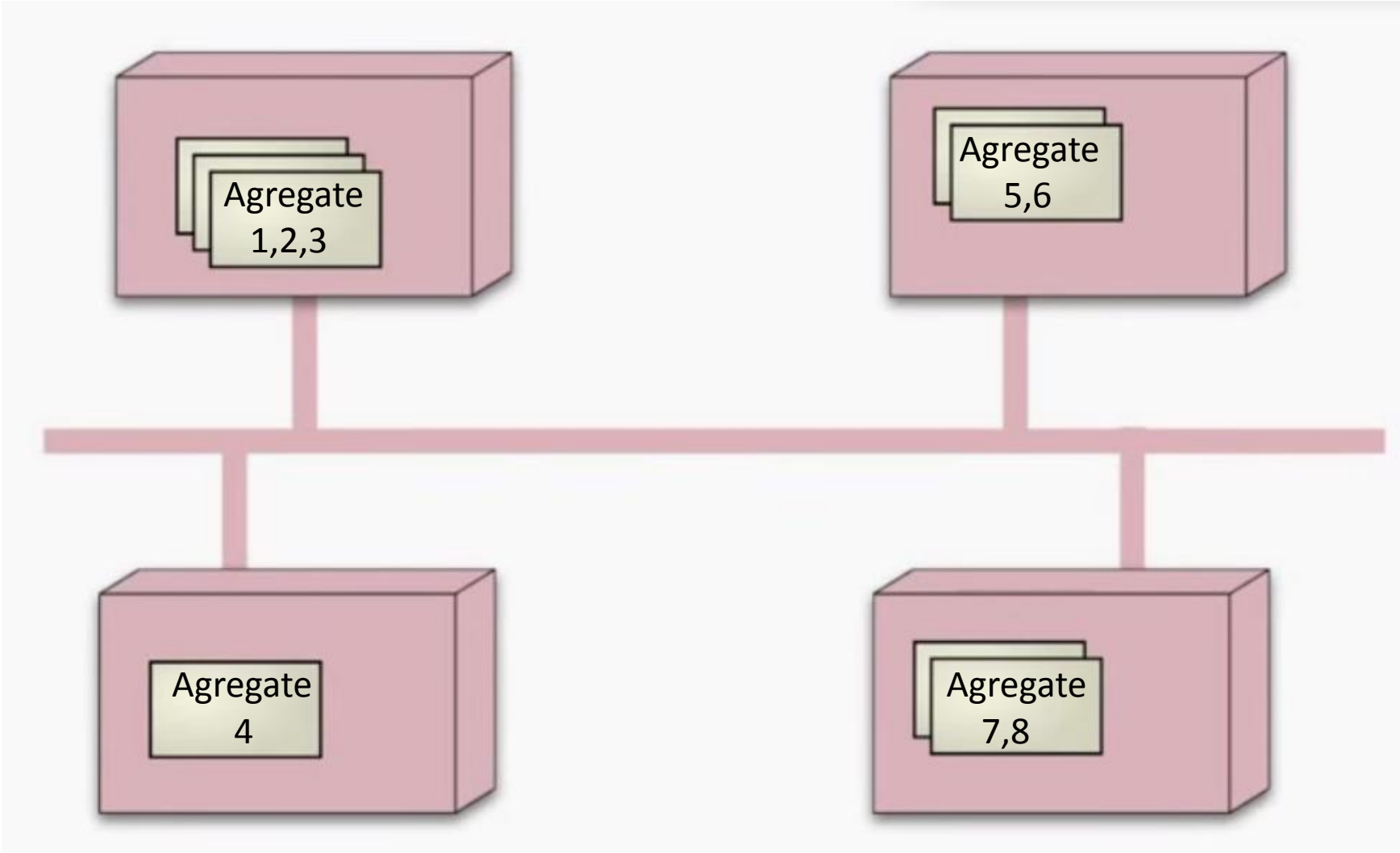
Column-family

- Memoreaza tabelele ca sectiuni de coloane si nu ca randuri.
- La nivel fizic, o tabela este o multime de coloane fiecare fiind de fapt o tabela avand un singur camp.

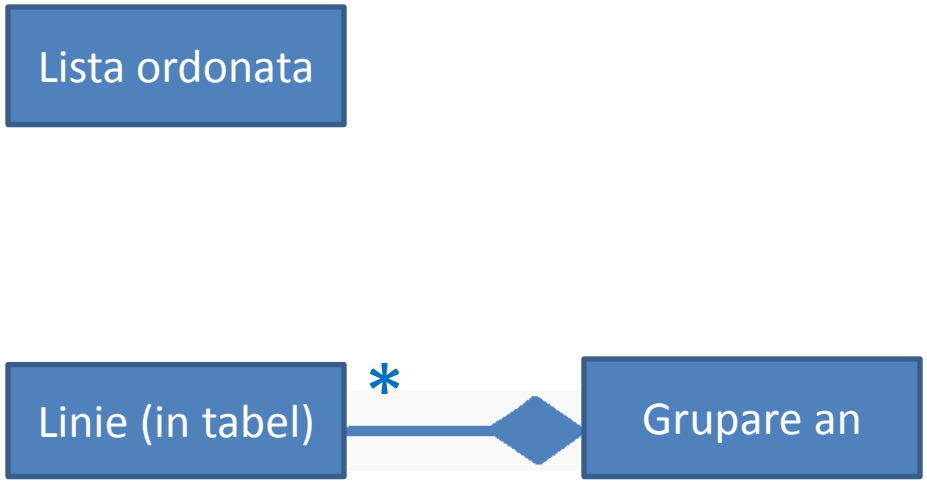

Column Family
titlu
student
an
2
nume
Becali
prenume
George
Profil
20135
PSGBD
10
TAP
9
IP
10
Note
Column key
Column value


Column Family
titlu
student
an
2
nume
Becali
prenume
George
Profil
20135
PSGBD
10
TAP
9
IP
10
Note
Agregatul de aceasta data e format din row-key si din numele familiei de coloane
Column key
Column value

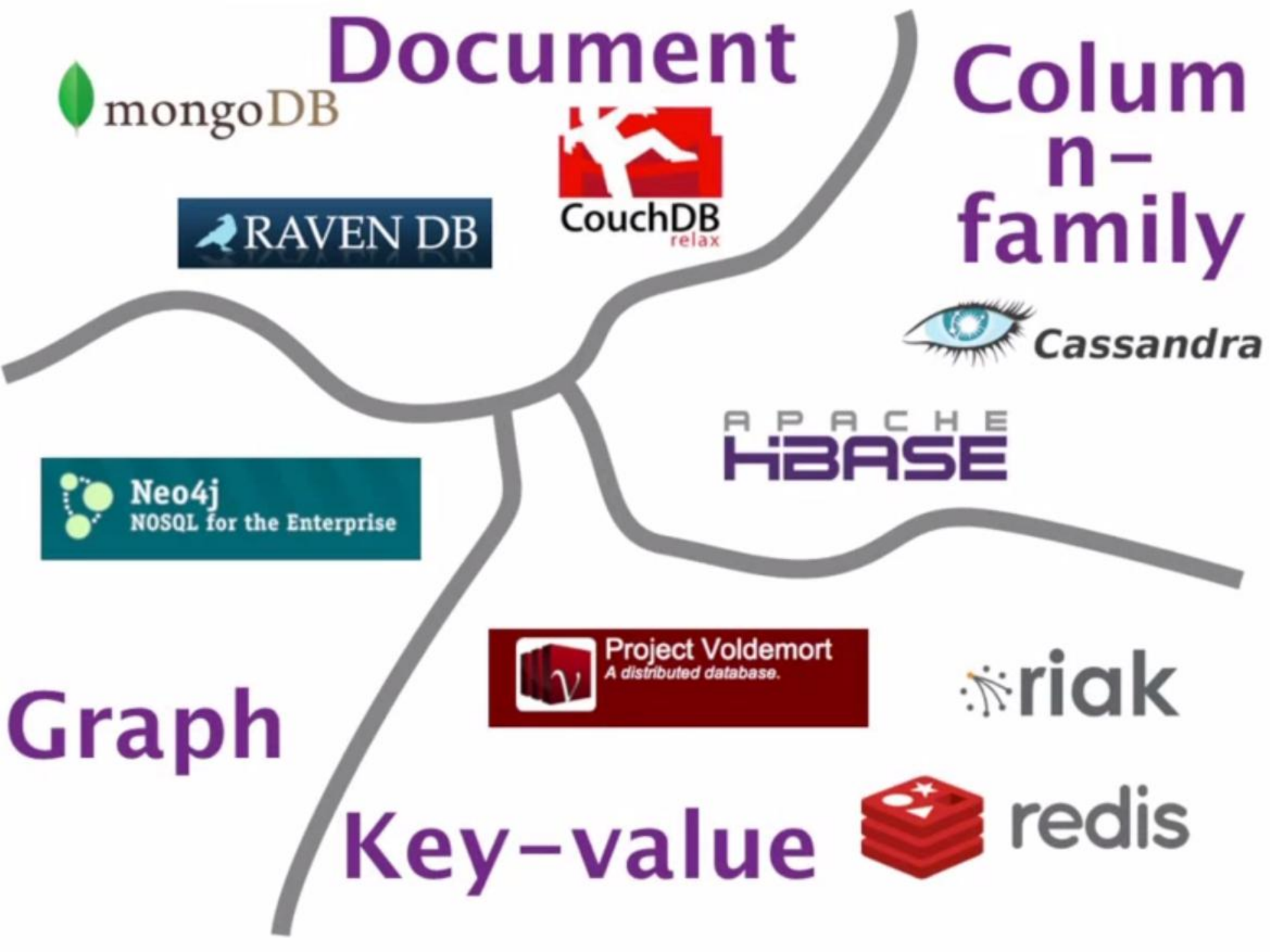
Ce se intampla in cluster ?

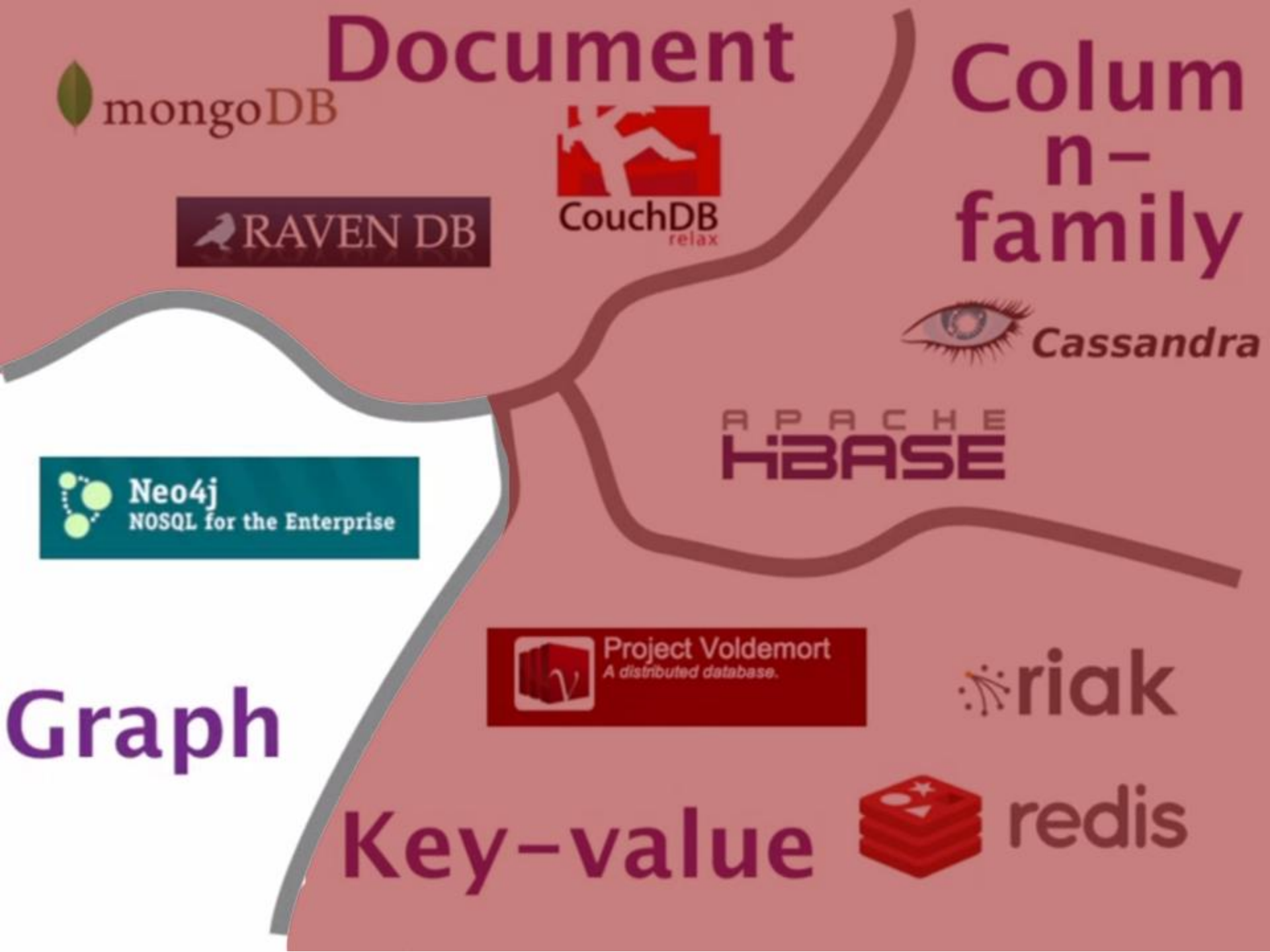


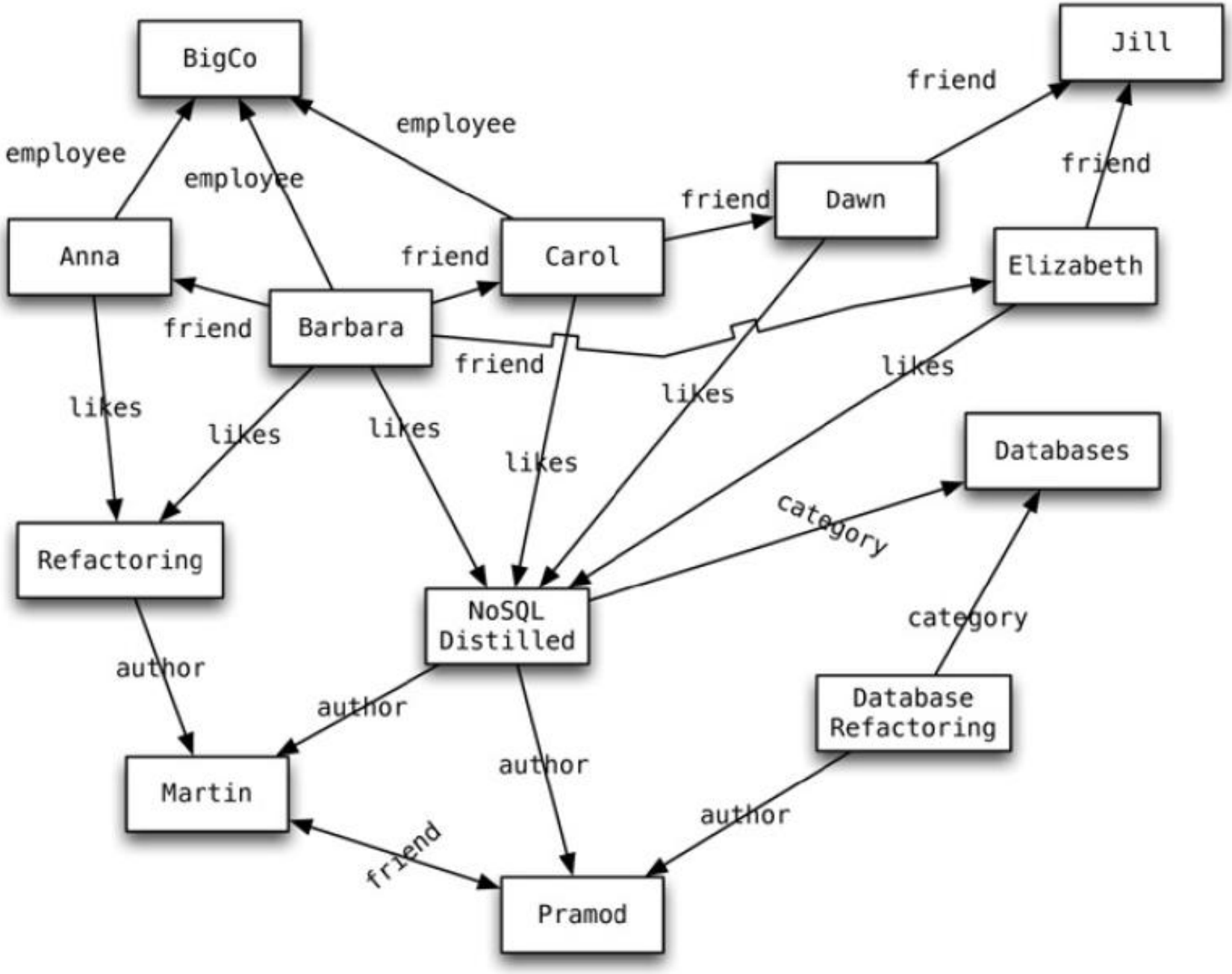
Daca vrem sa schimbam agregarea



Cu RDBMS este super simplu…. cu NoSQL in agregate-oriented nu chiar asa simplu.


Document
mongoDB
RAVEN DB
CouchDB
relax
Colum
n-
family
Cassandra
APACHE
HBASE
Neo4j
NOSQL for the Enterprise
Project Voldemort
A distributed database.
riak
Graph
Key-value
redis


Document
mongoDB
RAVEN DB
CouchDB
relax
Colum
n-
family
Cassandra
APACHE
HBASE
Neo4j
NOSQL for the Enterprise
Project Voldemort
A distributed database.
riak
Graph
Key-value
redis


BigCo
Jill
friend
employee
employee
employee
friend
friend
Dawn
Anna
friend
Carol
Elizabeth
friend
Barbara
friend
likes
likes
likes
likes
likes
likes
Databases
category
Refactoring
NoSQL
Distilled
category
author
author
Database
Refactoring
Martin
author
author
friend
Pramod

NoSQL – cum vei lucra cu datele ?!

- Agregate-Oriented incearca sa puna mai multe campuri laolalta.

- Graph-oriented sparge si mai mult modelul ajungand la componentele de baza.

NoSQL & Consistency

- RDBMS == ACID [ce insemna asta ?]
- NoSQL == BASE [**B**asically **A**vailable, **S**oft state **E**ventual consistency]

Graph este ACID (descompune atat de tare datele incat fiecare modificare este atomica...)

NoSQL & Consistency

- Ati putea sa va ganditi ca au nevoie de tranzactii…
- De fapt NoSQL-AO (Agregate-Oriented) nu (prea) au nevoie de tranzactii (si implementarea lor ar face complicata partea cu concurenta sistemului): *Keep the transactions inside a single agregate*. Problema apare doar daca vreau modificare multipla (dar sunt slabe sanse sa vreau sa modific mai multe agregate-uri in acelasi timp).

NoSQL & Consistency

- Consistency… amintim exemplul cu rezervarea locurilor de cinema/ camere hotel/locuri avion

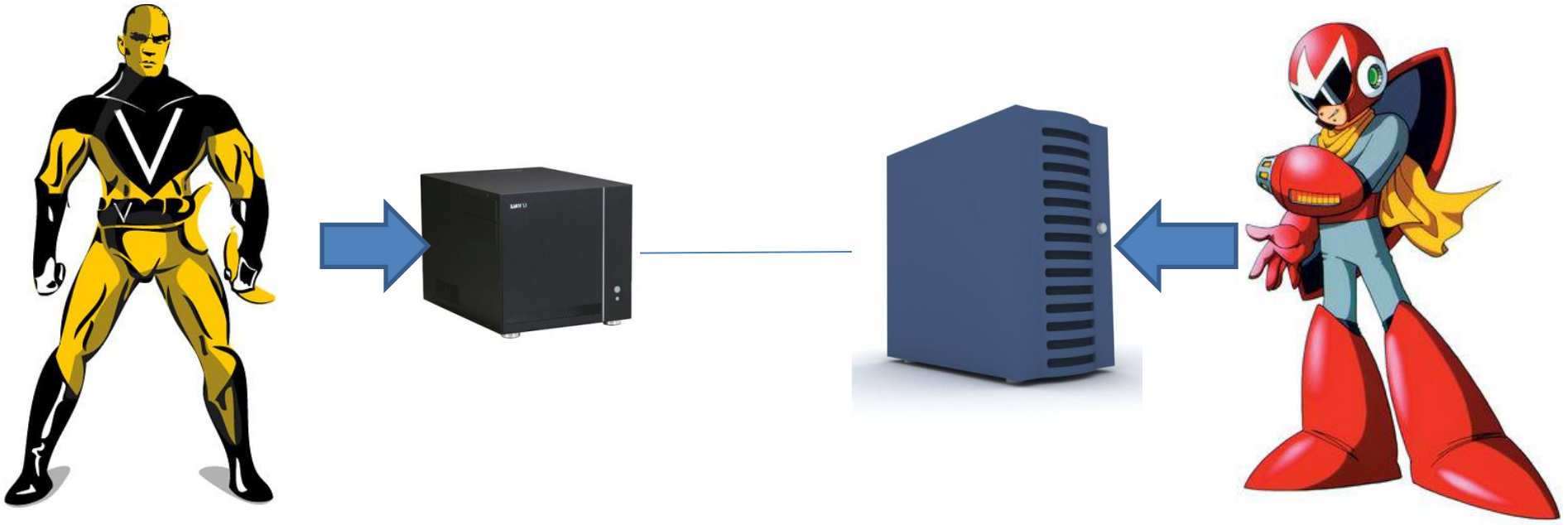
- Nu putem tine tranzactia deschisa cat timp utilizatorul tine pagina de rezervare deschisa.
- Am putea sa facem lock doar cand facem updateul (elimina coliziunea) dar unul dintre ei se va trezi ca trebuie sa refaca rezervarea.
- Utilizand timestamps si modificand cand scriu.

Consistency: Logical / Replication

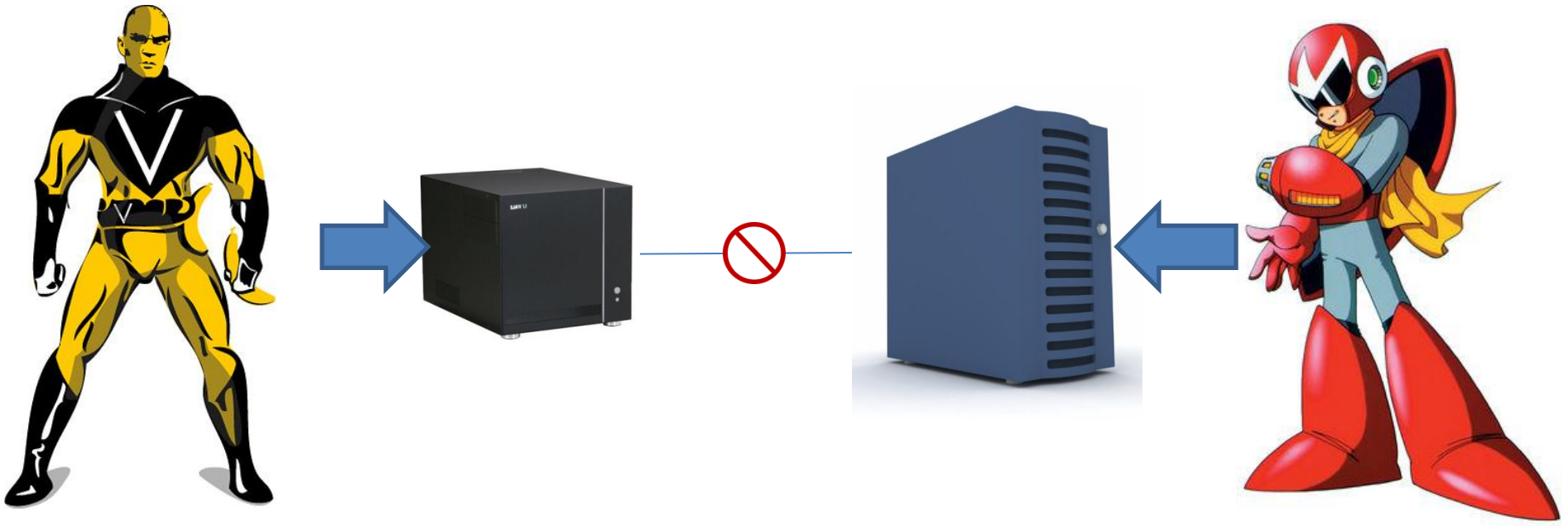
- Cele din exemplul anterior sunt de tipul Logical (care are loc si daca e pe o singura masina si daca e pe mai multe si se face replicare).
- Replication ?!

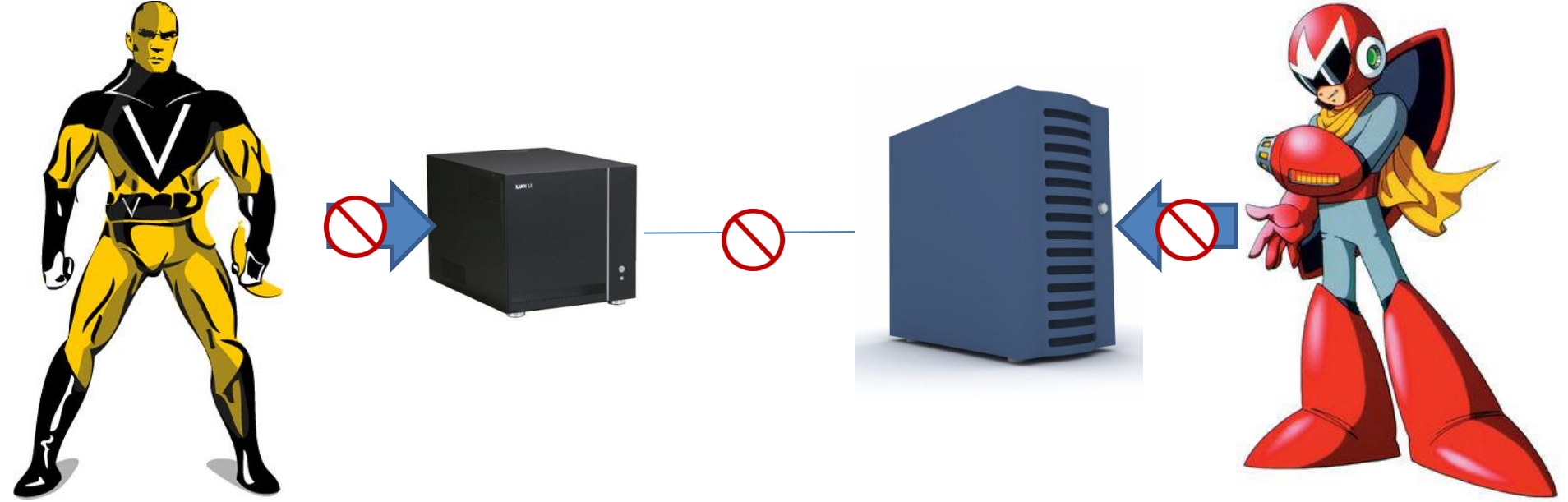
Consistency: replication – booking room

Consistency: replication – booking room



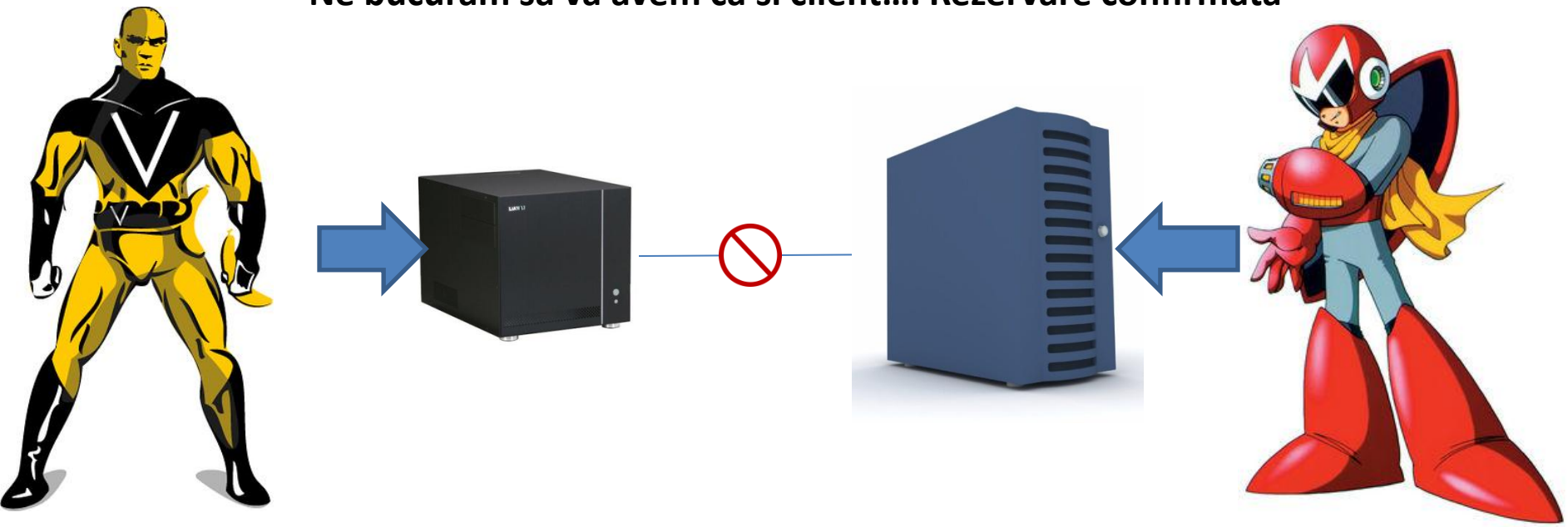
Consistency: replication – booking room

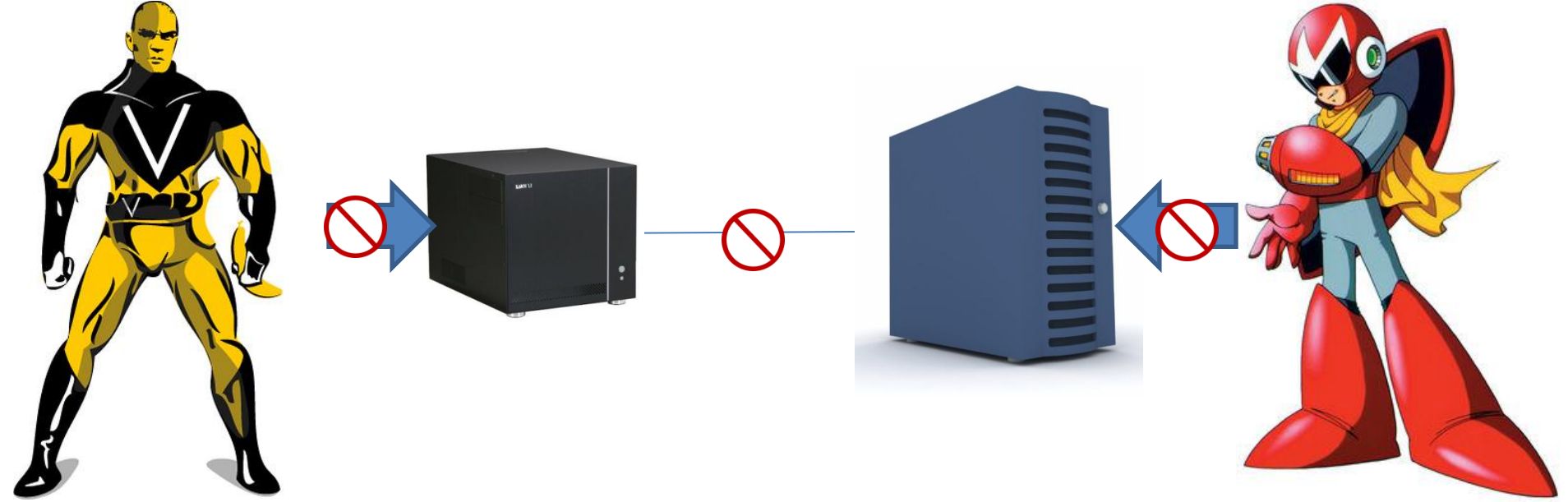




Ne pare rau… nu putem comunica cu serverul…

Ne bucuram sa va avem ca si client…. Rezervare confirmata





Consistency

Availability

NoSQL: Consistency vs Availability

- Este oare asa de rau faptul ca i-a cazat pe amandoi ?
- Ce este mai rau ? Sa faca double-booking la ultima camera sau sa nu dea voie nimanui sa faca booking ?
- Dynamo (Amazon): ei vroiau sa tina MEREU *shopping cart*-ul **available** ? Ce se intampla cand la sfarsit vedeai eroarea ? Corectezi…
- Chiar daca trimiteau mai multe obiecte si te supratAxau, iti permiteau sa le returnezi…

CAP Theorem… Pick two:

- Consistency
- Availability
- Partition Tolerance (e musai la NoSQL)

Nu e doar 0 sau 1, alb sau negru… pot exista diverse niveluri de consistenta/disponibilitate

Anumite operatii pot fi mai puternic consistente, unele vor fi mai mult disponibilitate.

Consistency vs Response time

- De fapt problema nu este la *availability* ci la timpul de raspuns.
- Adica in exemplul cu hotelul, daca conexiunea este foarte foarte slaba, se poate intampla ca hotelul sa prefere sa faca booking la amandoi decat sa piarda pe unul din clienti din cauza ca celalalt se misca prea incet.
- Amazon: WE ALWAYS WANT THE PPL

  SHOPPING FAST !

  (E cam la fel cu Safety vs Liveness)

Cand vrei sa folosesti NoSQL ?

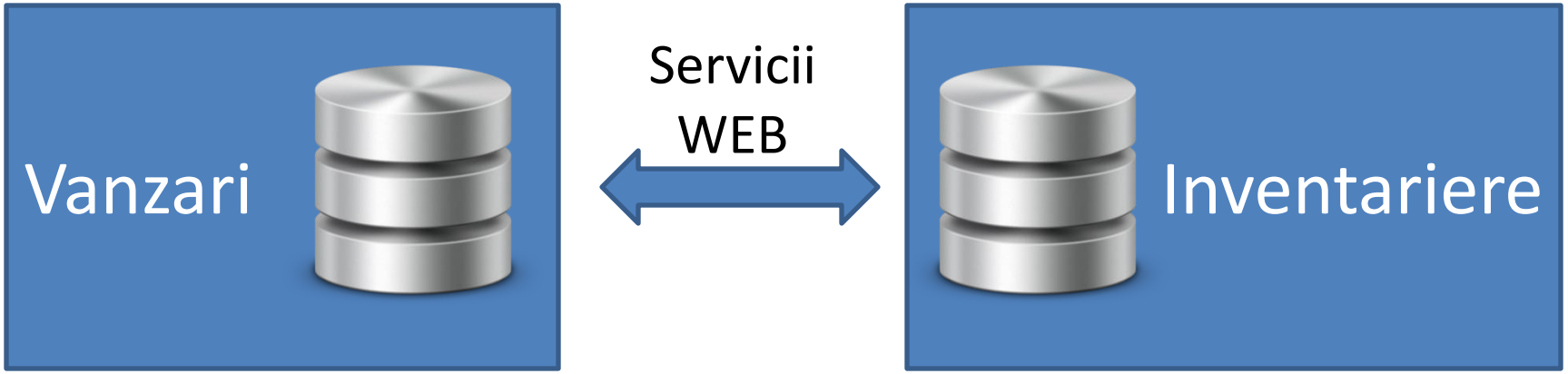
- Cand ai multe multe date [64k is enough for everybody].
- Este mai usor de programat (de exemplu pot agrega toata pagina web X sau Y intr-o singura inregistrare).

Cand vrei sa folosesti NoSQL ?

- De ce nu mai avem problema de la Object-DB ?

Cand vrei sa folosesti NoSQL ?

- Problema nu mai exista, pentru ca acum WEB-ul si aplicatiile sunt orientate pe servicii.



Inseamna asta ca NoSQL e viitorul ?

- Nicidecum….
- Probabil ca o combinatie dintre relational si NoSQL asta e viitorul.
- Trebuie sa va ganditi la Decizii / Organizational Change / Imaturity / Consitency issues

Joke(s)

- A bunch of RDBMS guys walk into a NoSQL bar. But they don't stay long. They could not find a table.

- Q: One day some of NoSQL guys walk into RDBMS bar. But they couldn't open the door! any one know why?

  A: The problem was they didn't have foreign keys !!

Joke(s)

- Q:Do you know why you can't join the NoSQL club?

  A: Joins are not supported.

- **you're data's so big...** it sat on a sql query and produced an unstructured database

- Date mai mari de 10TB
- Procesare statistica puternica.

Map Reduce:

- map: (K1,V1) -> list(K2, V2)
- reduce: (K2, list(V2)) -> (K3, V3)

- Pasi: Input -> Map -> Combine -> Shuffle and sort -> Reduce -> Output

- Exemplu cu o carte:

- input: cartea
- map: fiecare pagina este rupta si data unui server
- fiecare server face o lista cu fiecare cuvant de pe pagina, urmat de cifra 1: (cuvant,1)
- combine: functia de combinare aduna cuvintele identice si le da un numar egal cu suma lor
- shuffle and sort: ordonare cuvinte.
- reduce: preia rezultatele de la fiecare server si aduna sumele facute la punctul anterior
- output: este afisata distributia cuvintelor