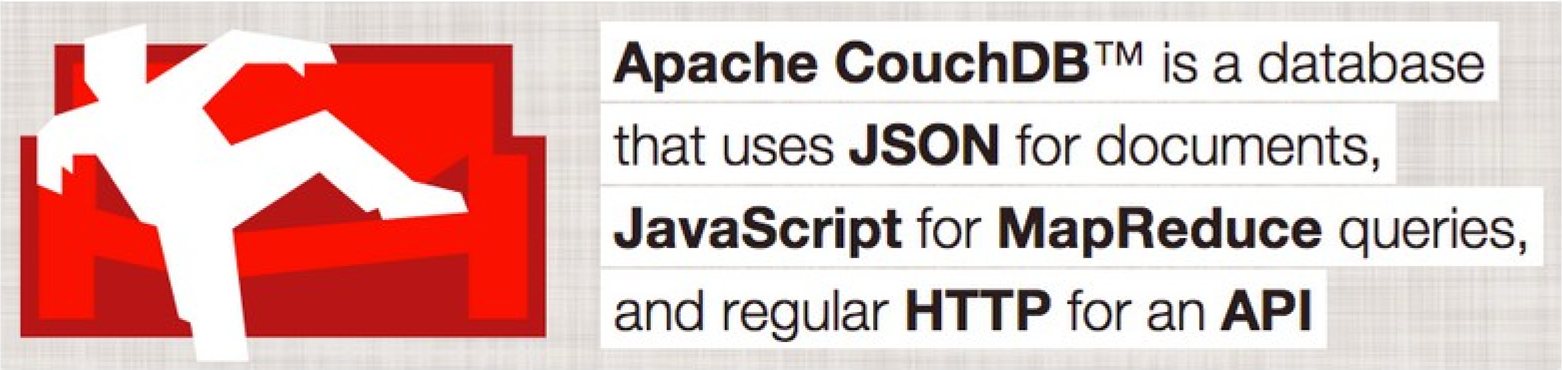


Apache CouchDB™ is a database
that uses **JSON** for documents,
JavaScript for **MapReduce** queries,
and regular **HTTP** for an **API**

CouchDB – overview

Sebastian Rothbucher @sebredhh
with lots of help from @awenkhh

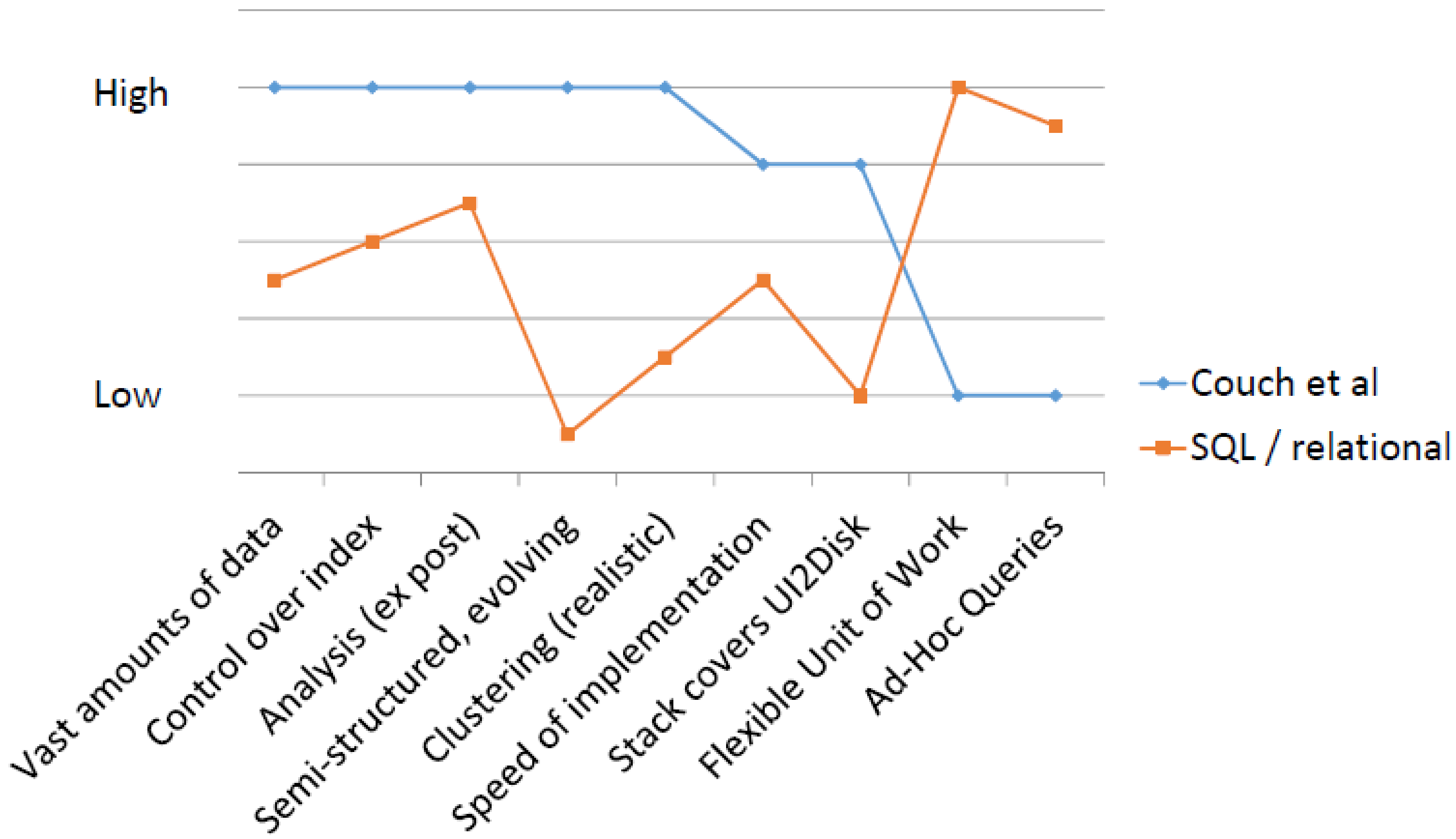
What is it - @all?

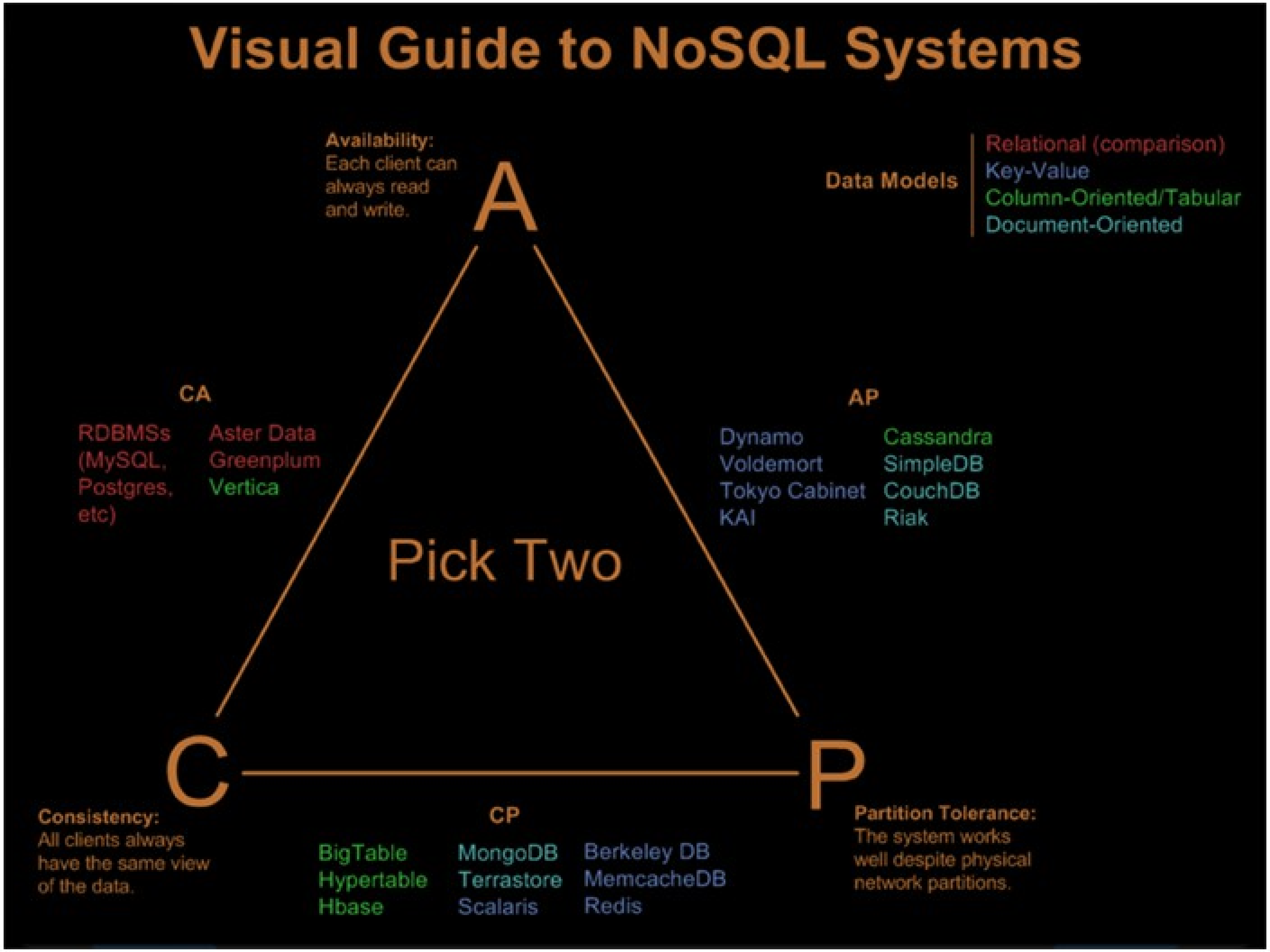
- Cluster of unreliable commodity hardware
- Map-Reduce & Document-oriented Database (no-SQL, no-schema)
- „Brainchild“ of a former senior IBM Lotus Domino Developer
- Web to the core (HTTP, JSON, JavaScript)
- Document = versioned JSON = unit of work = many roles
- Web-interface (and curl)
- Stable and scalable (Erlang-based foundation)

What you'll hear

- NoSQL database overview
- What is CouchDB?
- Querying CouchDB with http
- Fauxton
- Retrieving data with Map/Reduce
- MVCC
- _changes *magic*

How does it differ?





Visual Guide to NoSQL Systems

Availability:
Each client can always read and write.

A

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

Pick Two

C

P

Consistency:
All clients always have the same view of the data.

CP

BigTable
Hypertable
Hbase

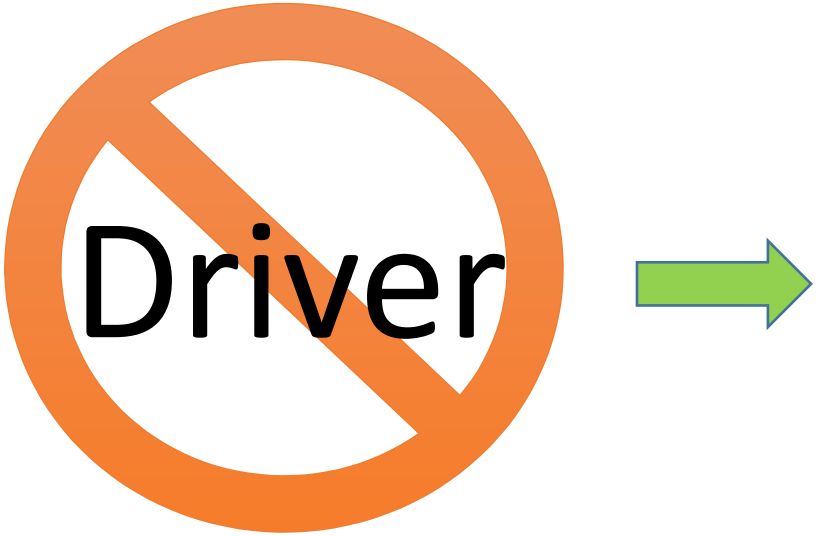
MongoDB
Terrastore
Scalaris

Berkeley DB
MemcacheDB
Redis

Partition Tolerance:
The system works well despite physical network partitions.

http://www.eecs.berkeley.edu/~brewer/cs262b-2004/PODC-keynote.pdf

Web, web, web



Driver

```
curl -X PUT -d …
```

Create DB & docs

Add attachment & serve infos

Futon > Fauxton & Docs

Map-Reduce („closures“)

Map (produce index)

- „Explode“ document base
- No-schema – hence check docs
- Produce keys/values (0-n)
- key, docid, value
- Sorted (B+Tree)
- Highly efficient & (somewhat) parallel
- (map-only possible)

Default: _all_docs (key = docid)

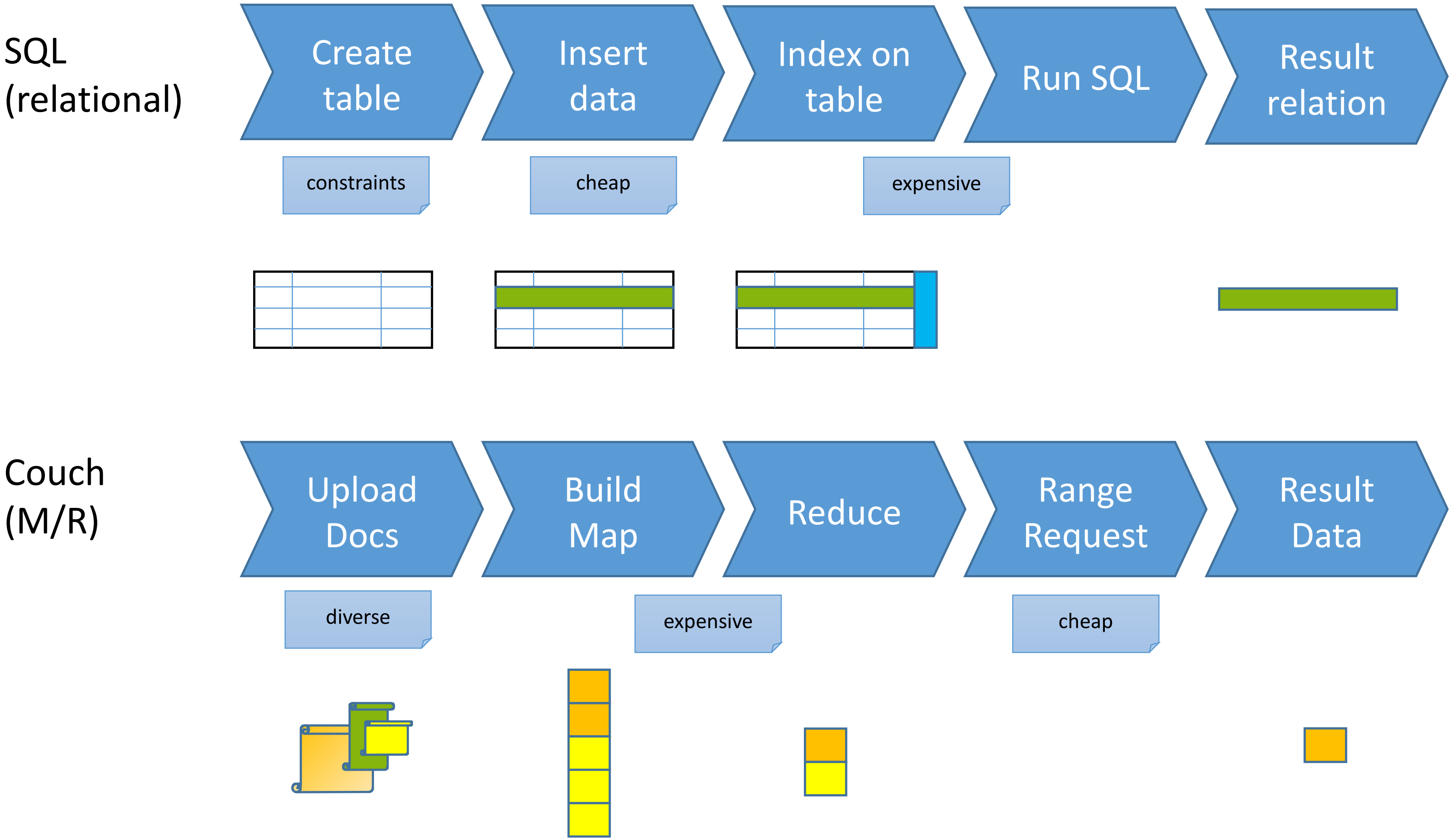
➔ Analyzed set is not base set
(query on index ONLY != SQL: on table)

Reduce (aggregate index)

- Collated set (one key)
- Produce one output from several
- „Collapse“ Info
- Sum, Count, Stats
- (Complex (Correlation, max. one „join“, …))

➔ Huge indices on one key

How does it fit together?



SQL (relational)

Create table — constraints

Insert data — cheap

Index on table — expensive

Run SQL

Result relation

Couch (M/R)

Upload Docs — diverse

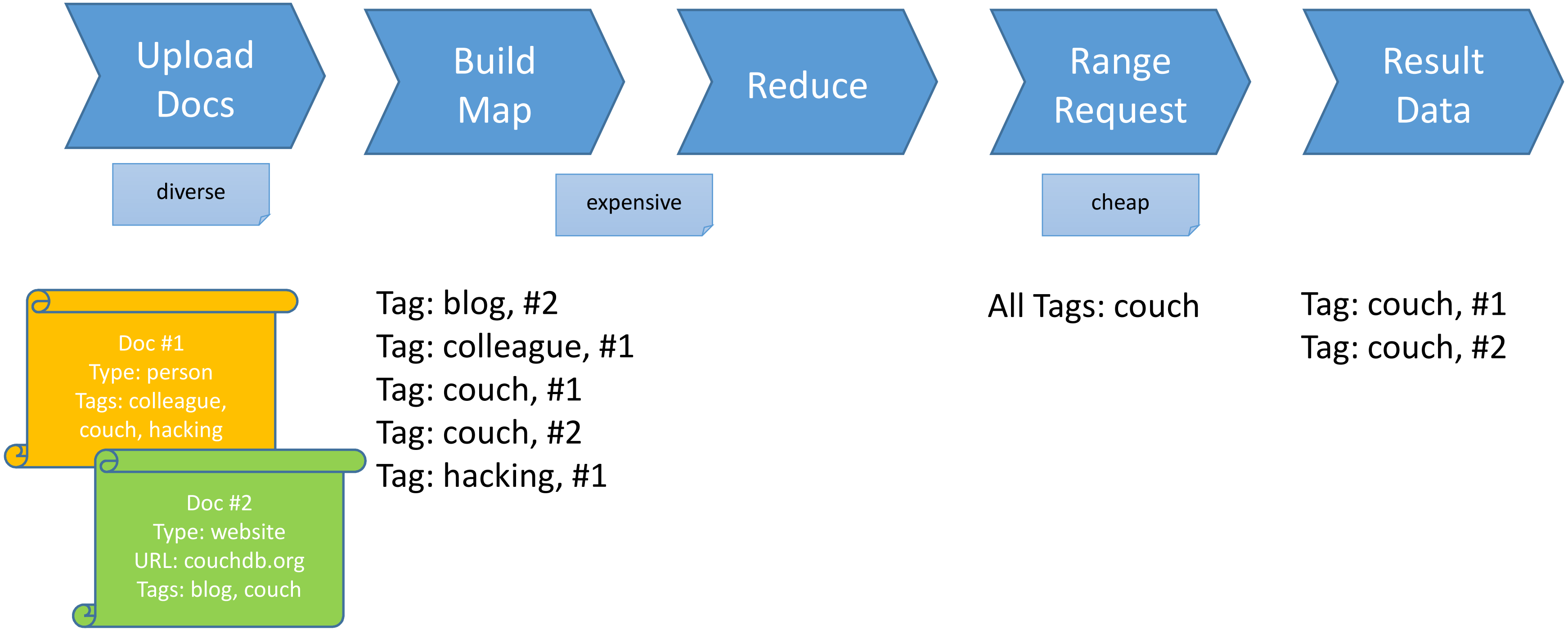
Build Map — expensive

Reduce

Range Request — cheap

Result Data

How does it fit together? (example)



Upload Docs

diverse

Build Map

expensive

Reduce

Range Request

cheap

Result Data

Doc #1
Type: person
Tags: colleague, couch, hacking

Doc #2
Type: website
URL: couchdb.org
Tags: blog, couch

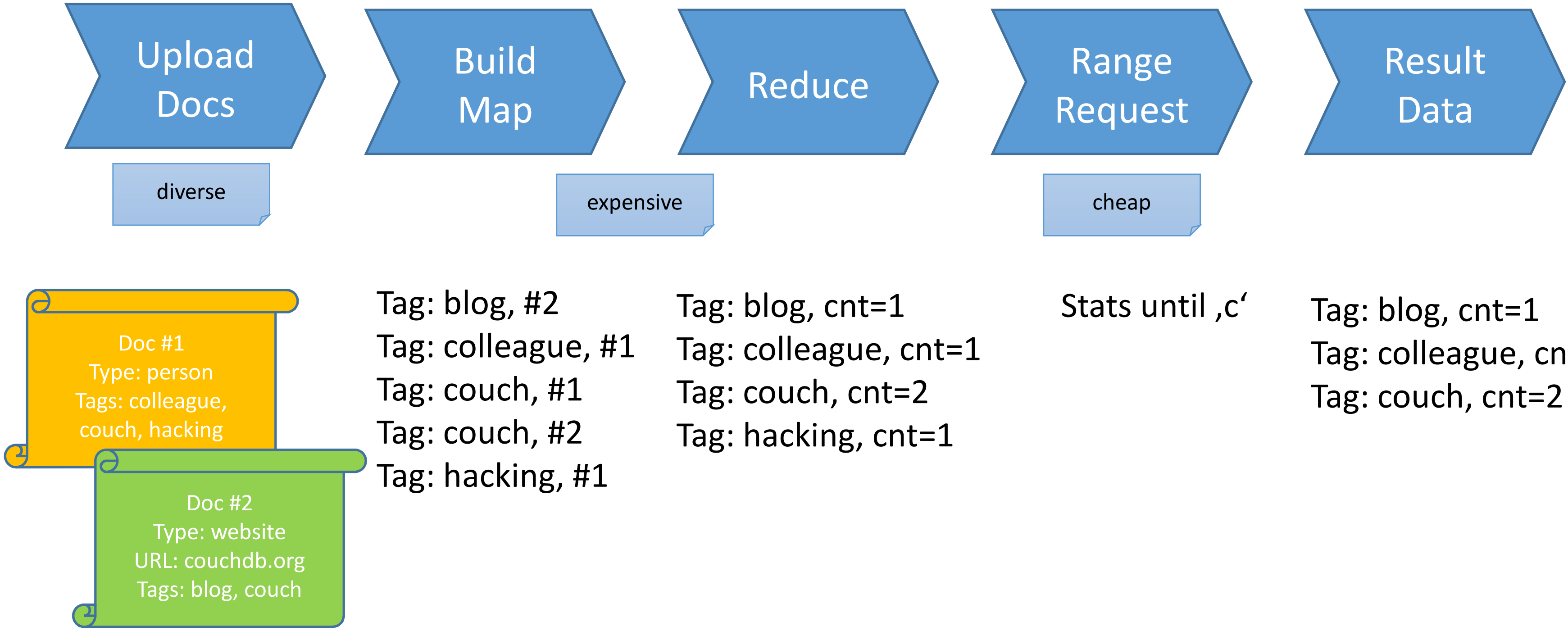
Tag: blog, #2
Tag: colleague, #1
Tag: couch, #1
Tag: couch, #2
Tag: hacking, #1

All Tags: couch

Tag: couch, #1
Tag: couch, #2

How does it fit together? (example cont.)

CouchDB
relax



Upload Docs

diverse

Doc #1
Type: person
Tags: colleague, couch, hacking

Doc #2
Type: website
URL: couchdb.org
Tags: blog, couch

Build Map

expensive

Tag: blog, #2
Tag: colleague, #1
Tag: couch, #1
Tag: couch, #2
Tag: hacking, #1

Reduce

Tag: blog, cnt=1
Tag: colleague, cnt=1
Tag: couch, cnt=2
Tag: hacking, cnt=1

Range Request

cheap

Stats until ‚c‘

Result Data

Tag: blog, cnt=1
Tag: colleague, cnt=1
Tag: couch, cnt=2

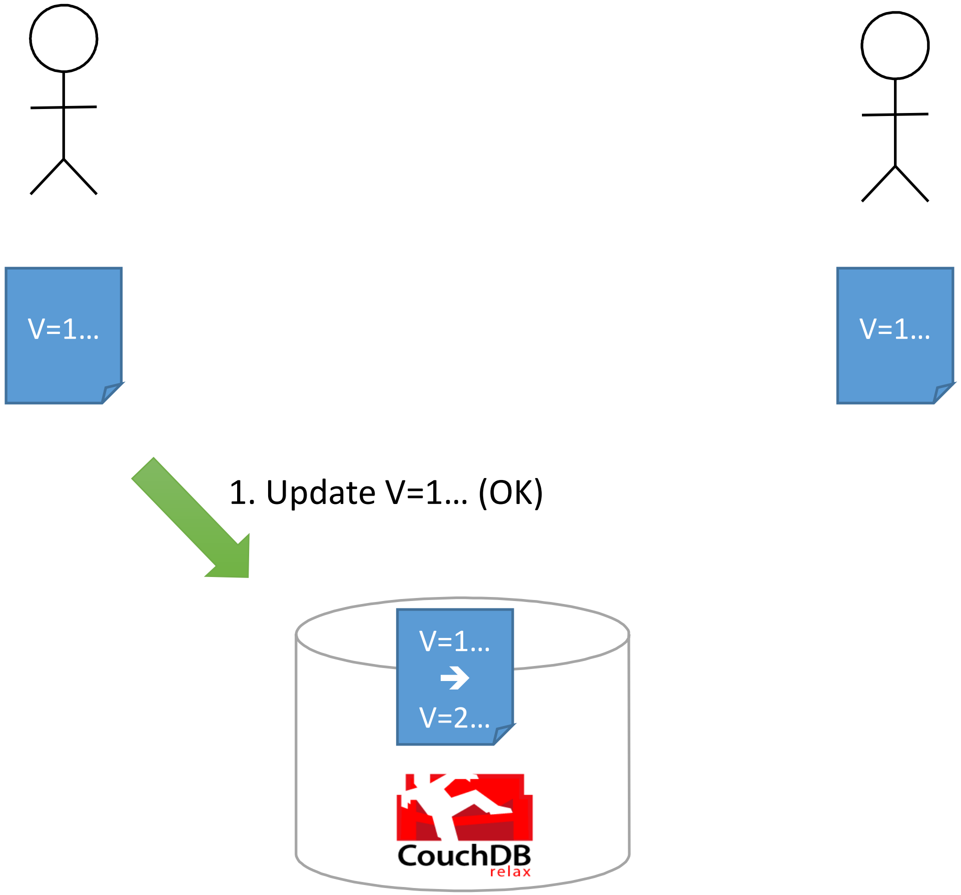
Creating a view

Querying a view & paging

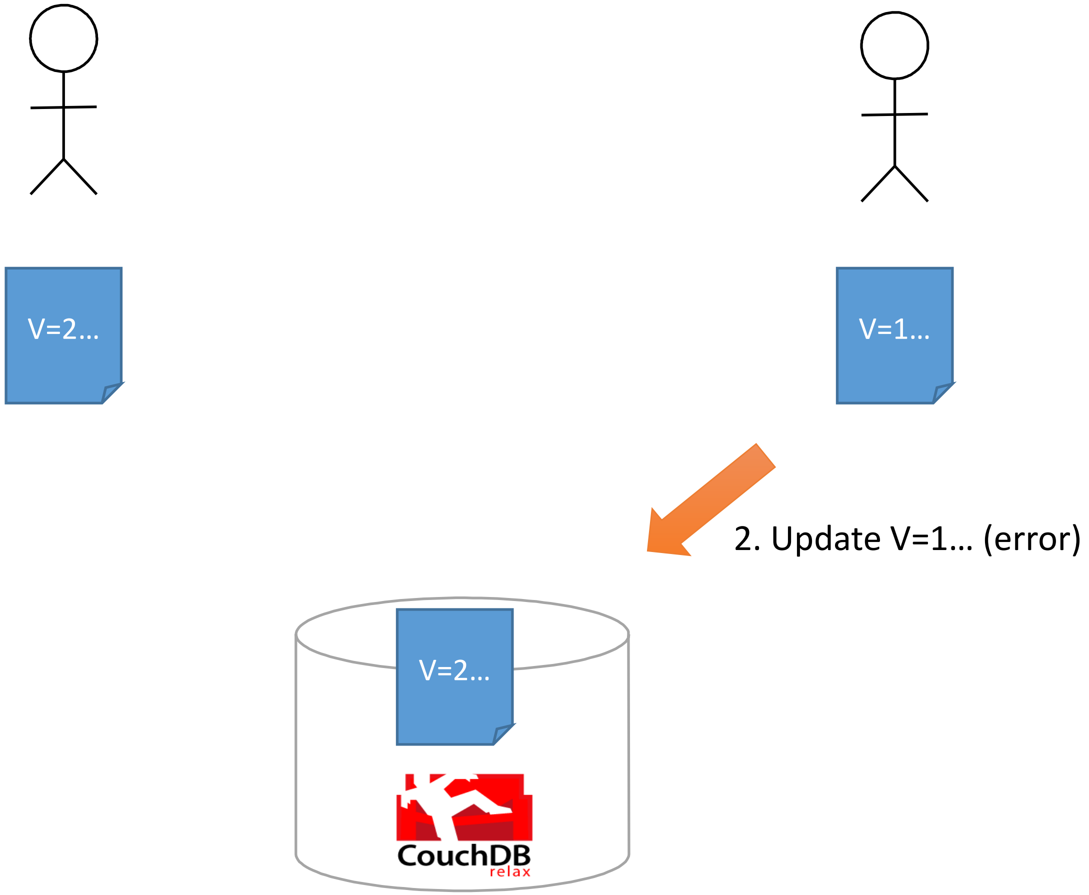
More curl magic (?)

Advanced: view collation (?)

MVCC



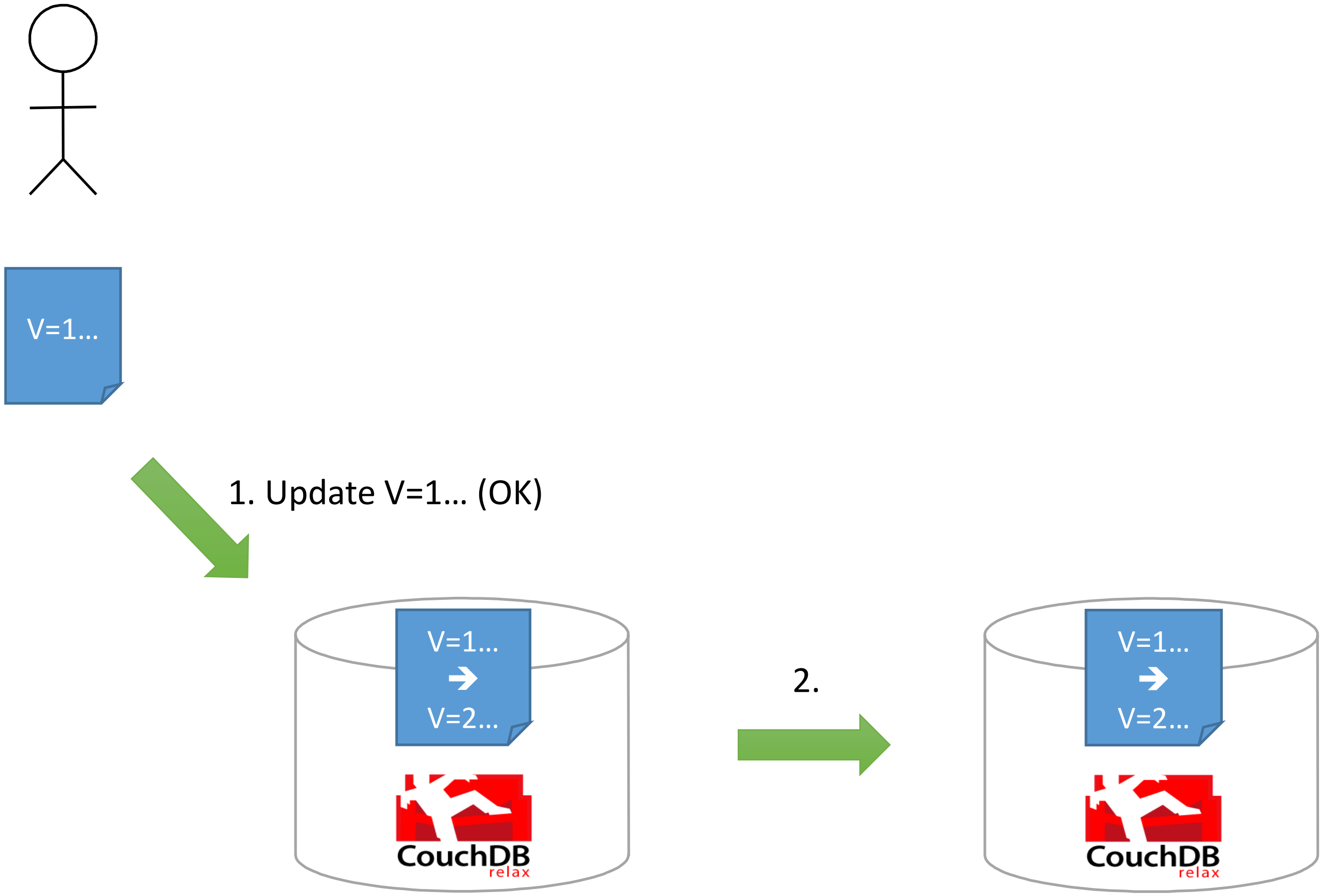
MVCC (cont.)



2. Update V=1… (error)

!= versioning!

Replication



V=1…

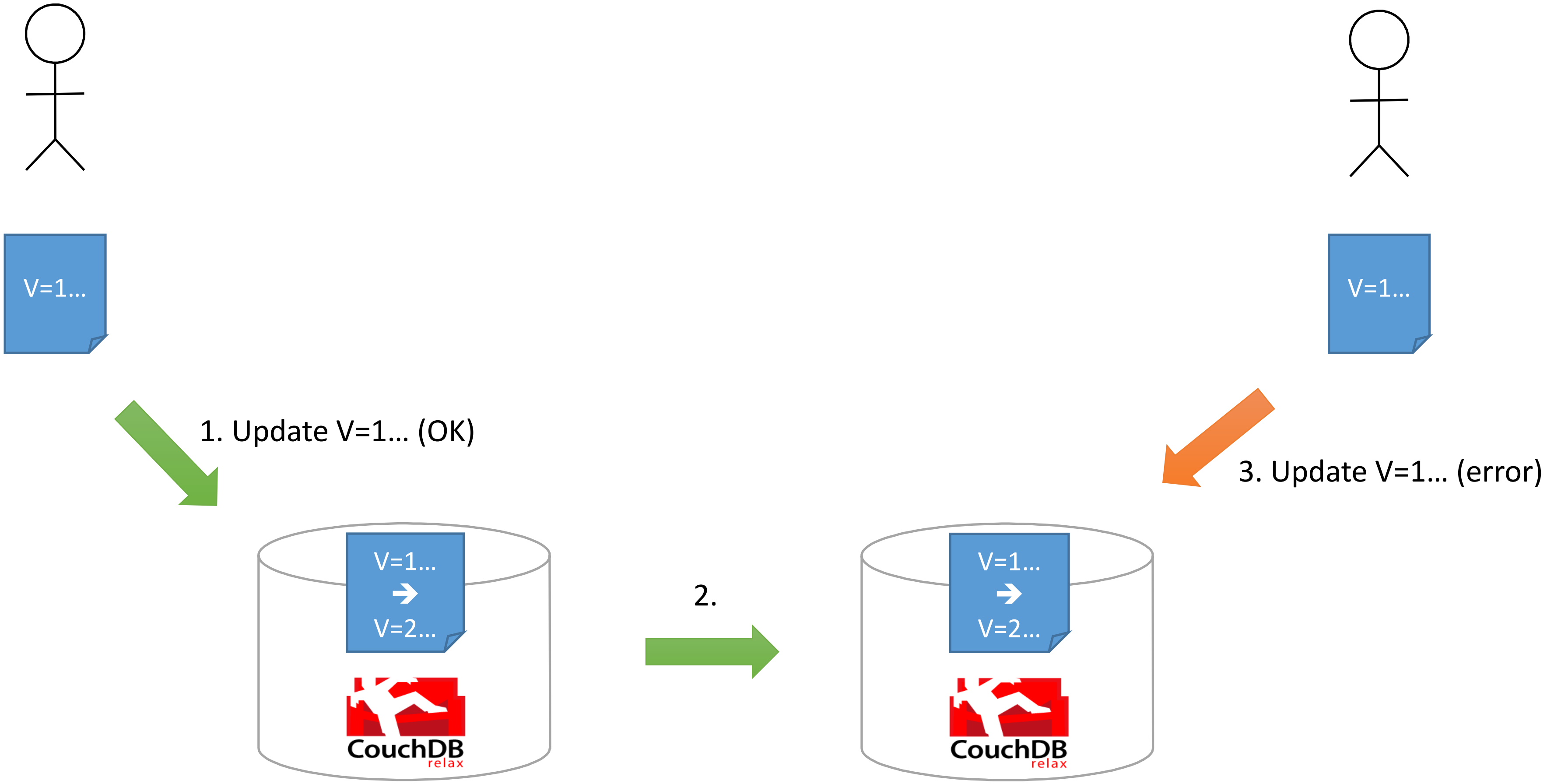
1. Update V=1… (OK)

V=1… ➔ V=2…

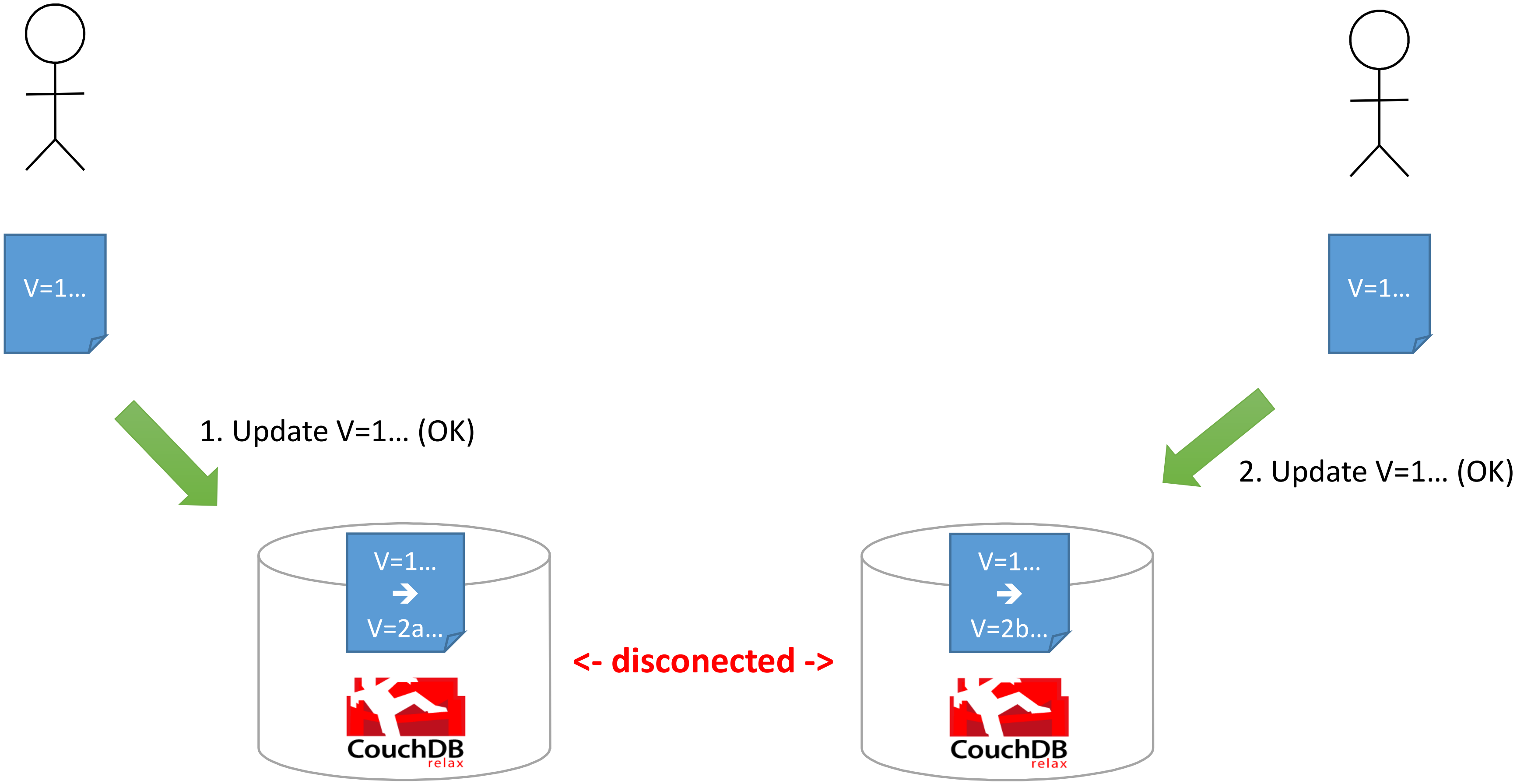
2.

V=1… ➔ V=2…

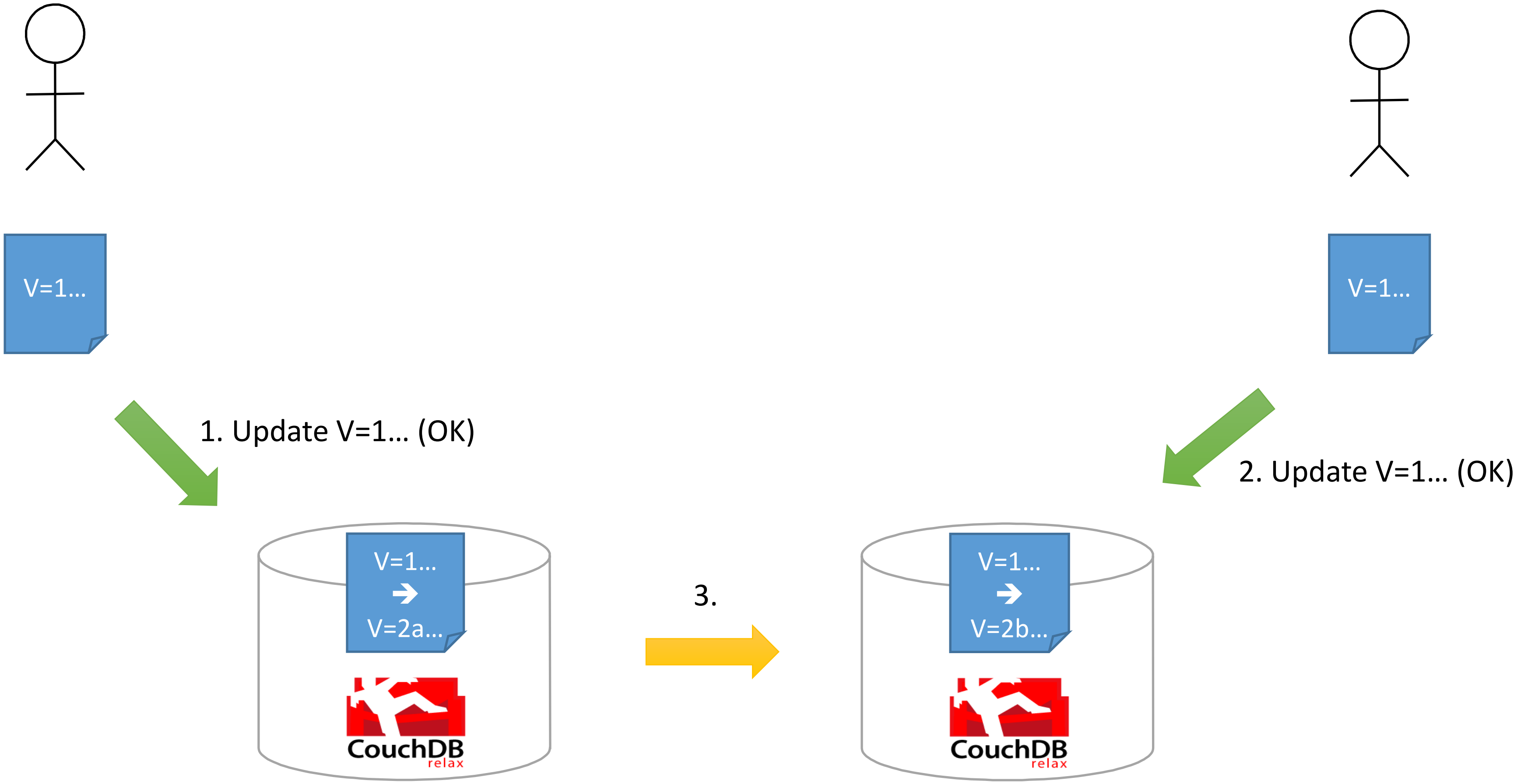
Replication (cont.)



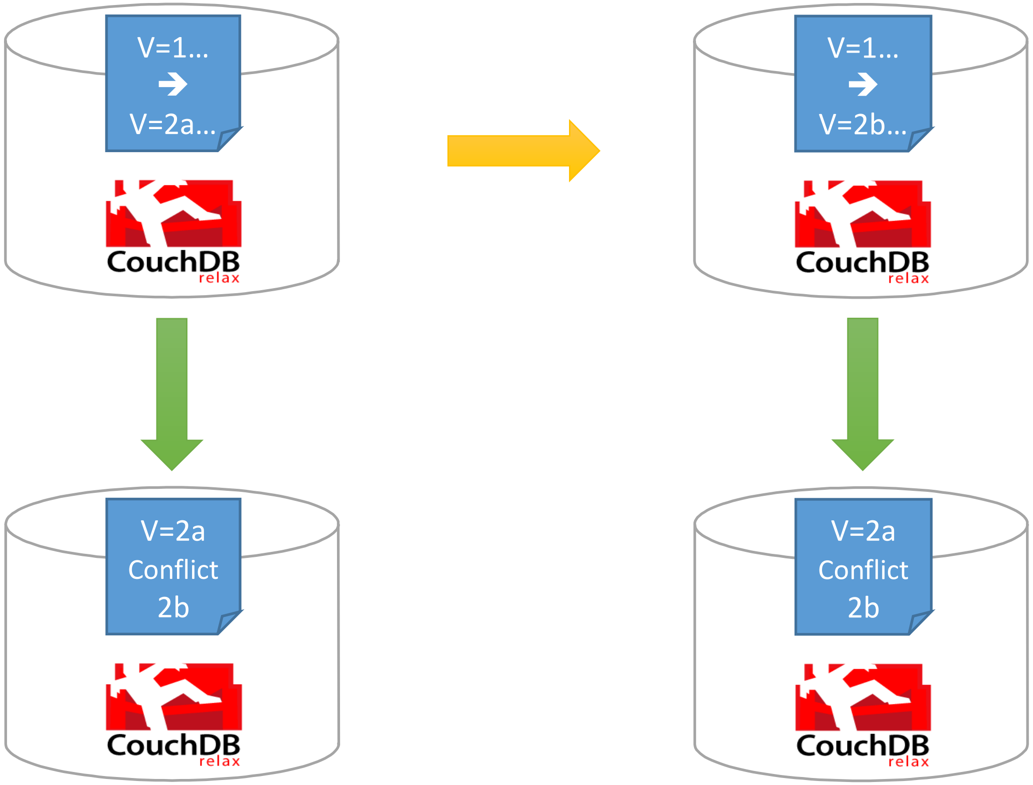
Replication (cont.)



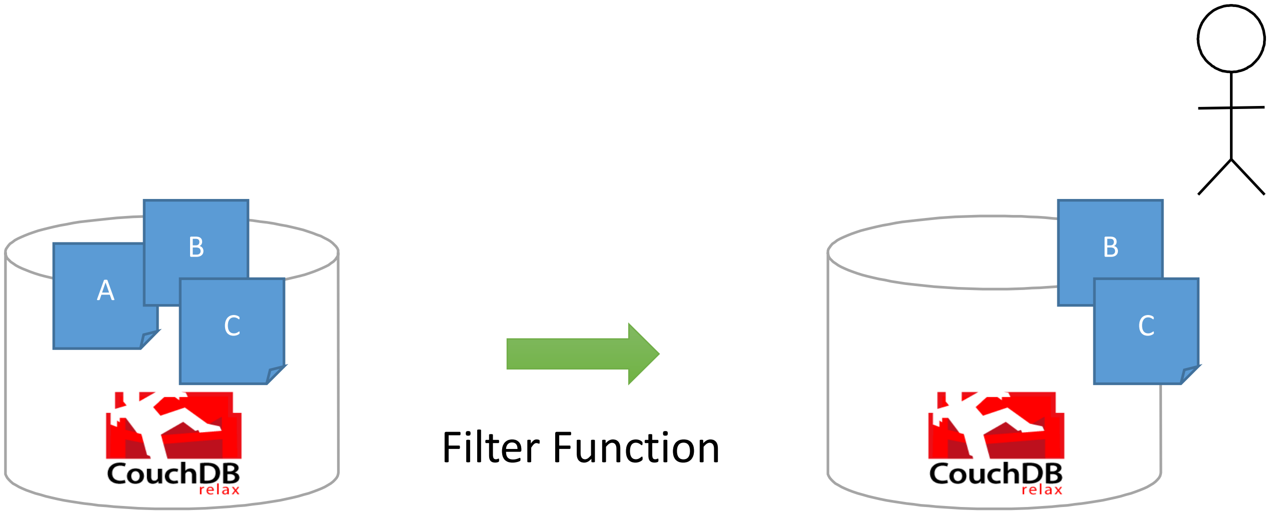
Replication (cont.)



Replication (cont.) – Conflict resolution



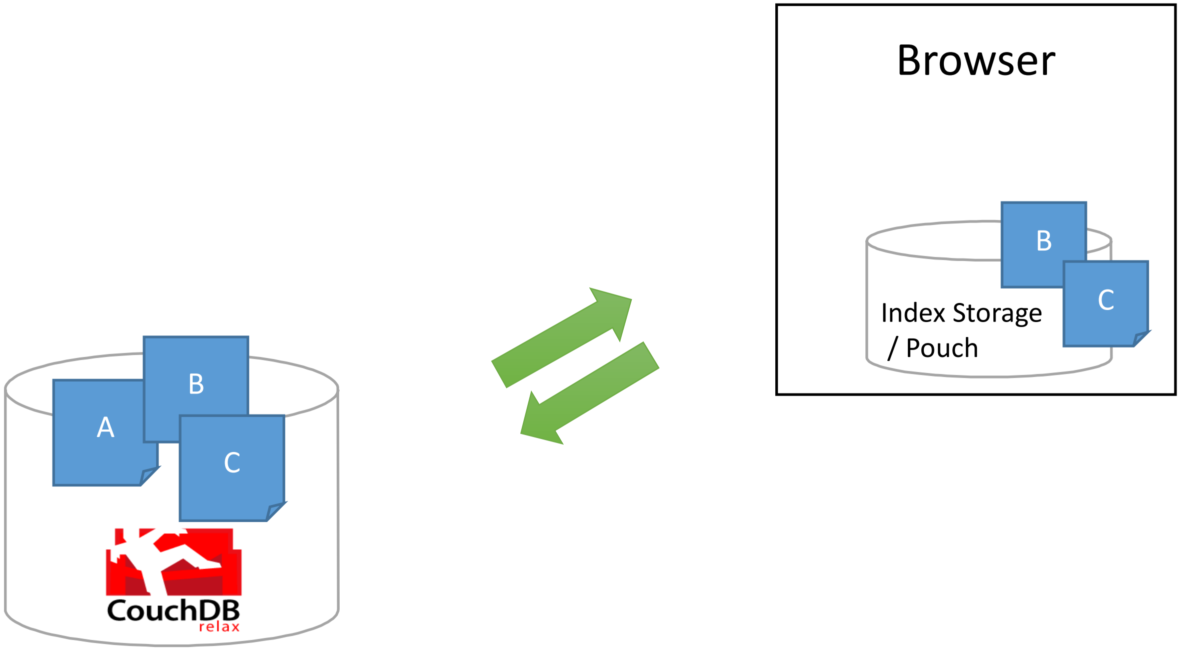
Replication - selective



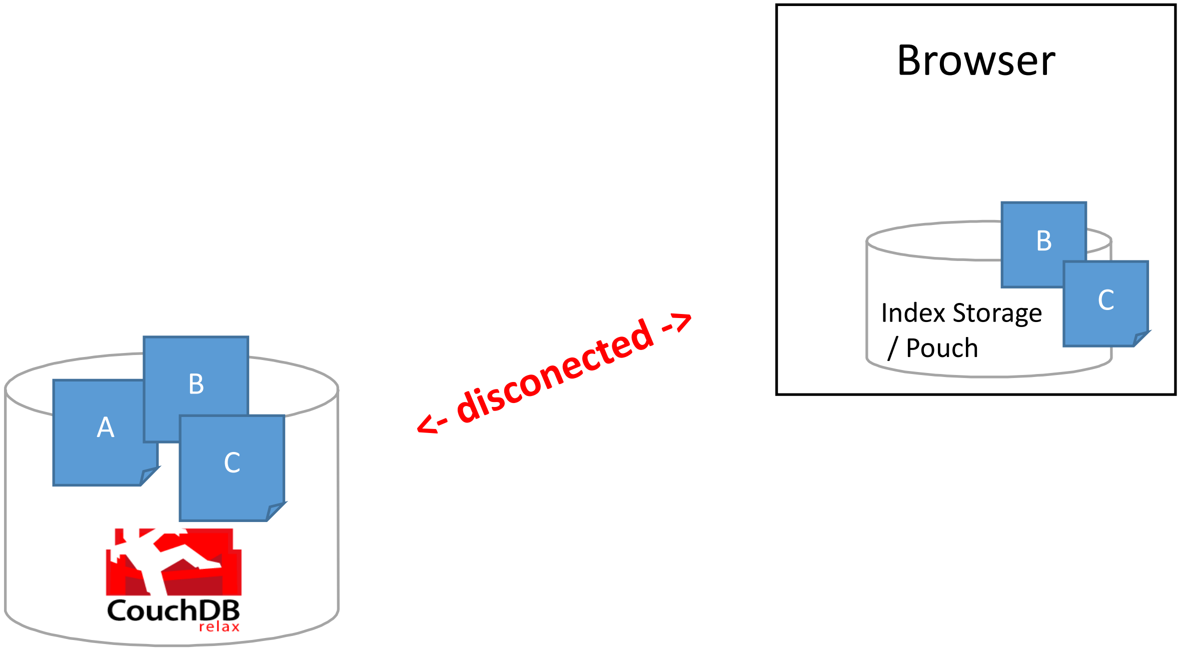
Filter Function

➔ Couch per user

Replication PouchDB



Replication PouchDB (cont.)



Doing some replication

The best is yet to come…

_changes

Links

- CouchDB Website: http://couchdb.apache.org/
- CouchDB day talks: http://blog.couchdb.org/2015/02/24/couchdb-day-2015-recap/
- CouchDB Manual: http://docs.couchdb.org/
- Joan‘s talk on MVCC et al: http://atypical.net/archive/2014/02/04/my-couchdb-conf-talk
- eHealth Africa – collected best practices: http://docs.ehealthafrica.org/couchdb-best-practices/
- Jan‘s intro into CouchDB: http://www.infoq.com/articles/apache-couchdb-the-definitive-introduction
- Cloudant Docs: https://docs.cloudant.com/

Summary & Wrap-Up

```
 http://
        function map(doc)

emit(doc.tags[i], null)

  ?startkey=couch&limit=10

<script src=“angular.js“

    _changes?feed=longpoll

{“couchdb“:“Welcome“
```

JUST DO IT