

JAOO October 5th, 2009



Apache
CouchDB

Jason Davies

About Jason

- Cambridge University (2002-2005)
- Director, Jason Davies Ltd (2005-present)
- Web Developer: Python, Django, JavaScript, jQuery, Erlang, &c.
- Apache CouchDB committer (2009)

A satellite view of the Earth showing continents and oceans. The text is overlaid on the image.

CouchDB

from

3,331km

CouchDB Features

- “Cluster of Unreliable Commodity Hardware”
- Schema-free (JSON)
- Document-oriented, distributed database
- RESTful HTTP API
- JavaScript-powered Map/Reduce
- N-Master Replication

“Of the Web”

<http://jacobian.org/writing/of-the-web/>

[▶ Run All](#) [↑ Reload](#)

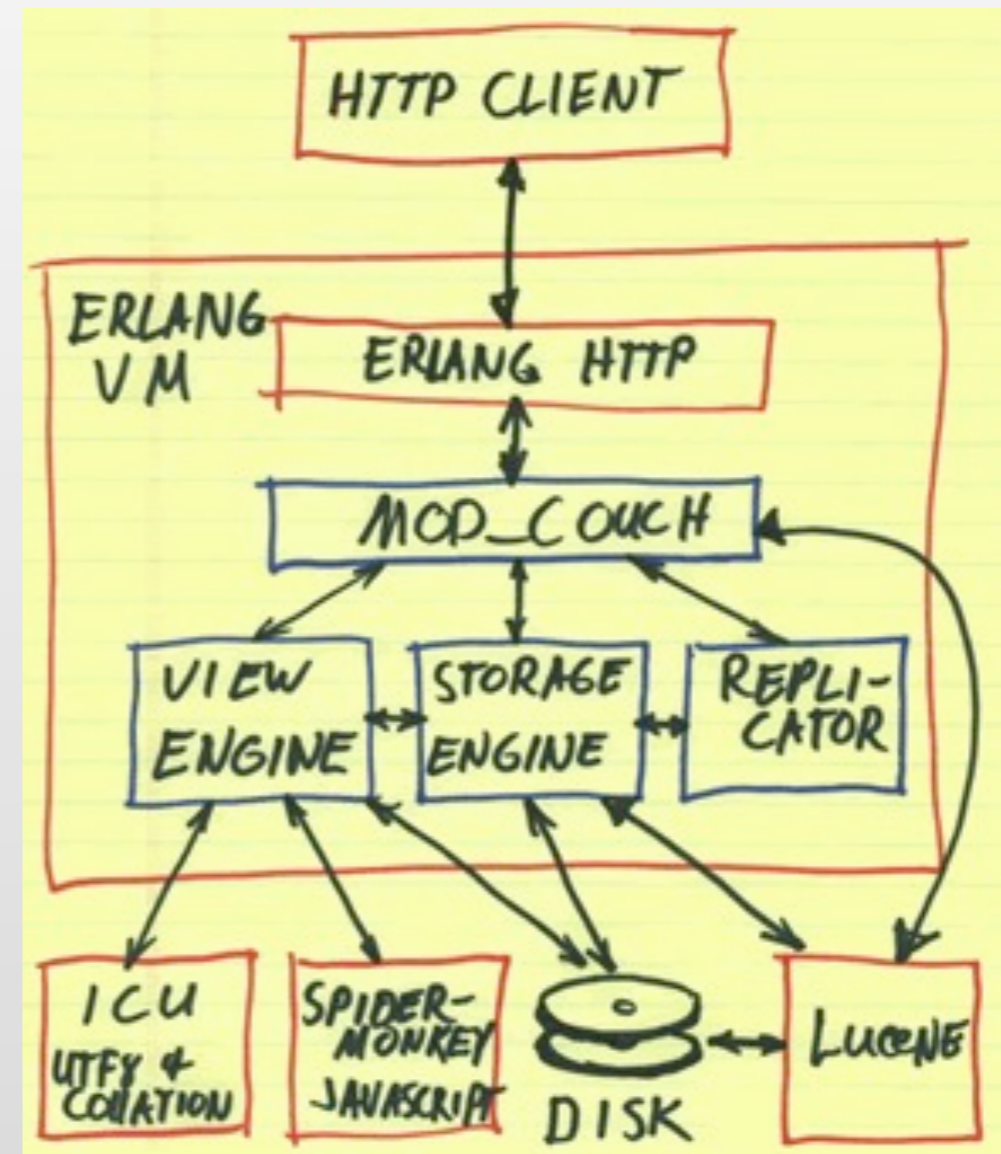
Note: Each of the tests will block the browser. If the connection to your CouchDB server is slow, running the tests will take some time, and you'll not be able to do much with your browser while a test is being executed.

Name	Status	Elapsed Time	Details
▶ basics	✓ success	318ms	
▶ conflicts	✓ success	130ms	
▶ recreate_doc	✓ success	817ms	
▶ bulk_docs	✓ success	112ms	
▶ lots_of_docs	✓ success	746ms	
▶ reduce	✓ success	3344ms	
▶ multiple_rows	✓ success	200ms	
▶ large_docs	✓ success	309ms	
▶ utf8	✓ success	153ms	
▶ attachments	✓ success	55ms	
▶ content_negotiation	✓ success	34ms	
▶ design_docs	⊛ running...		
▶ view_collation	not run		
▶ view_conflicts	not run		

**Tools**[Overview](#)[Replicator](#)[Test Suite](#)**Databases**[thoughts](#)

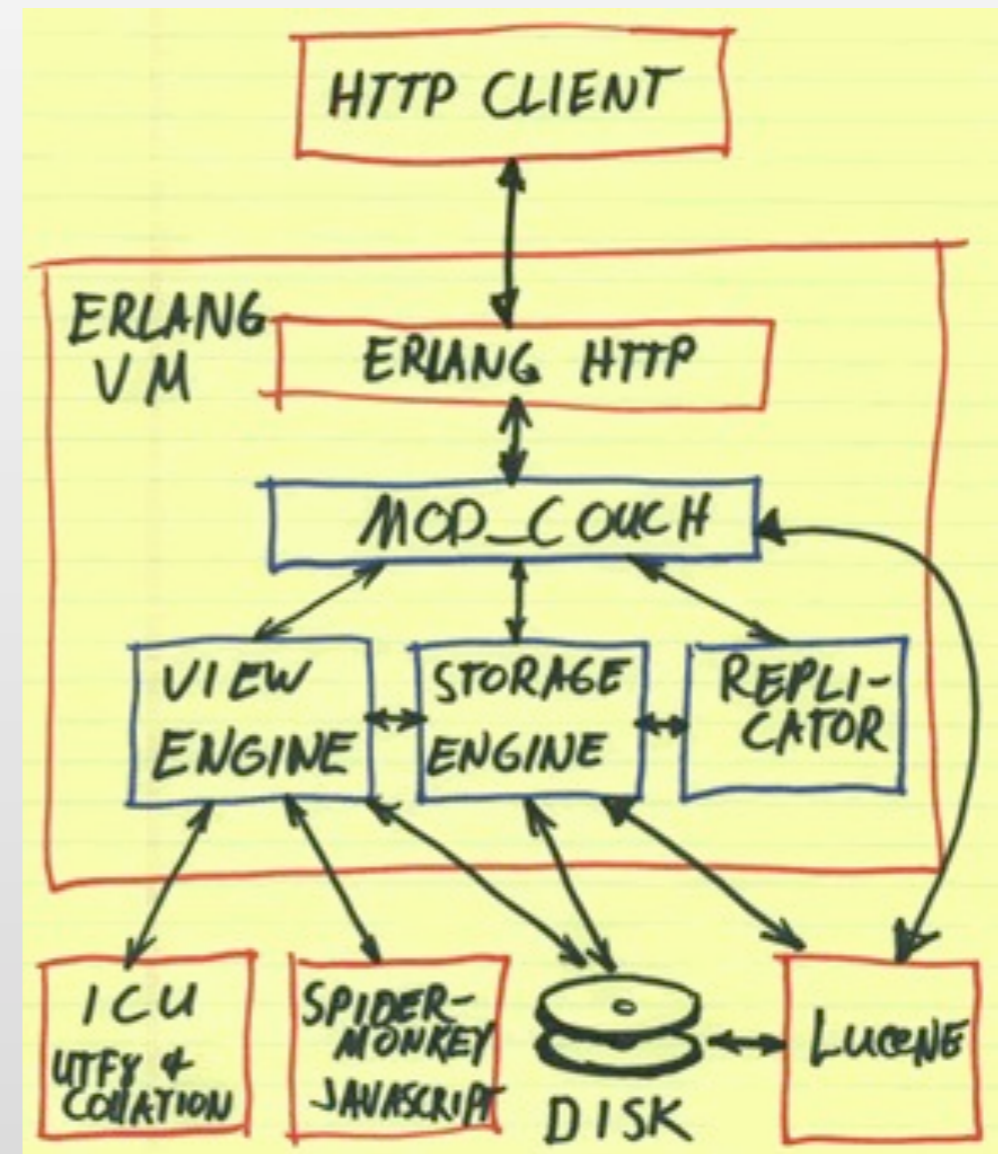
- Schema-Free (JSON)
- Document Oriented,
Not Relational
- Highly Concurrent
- RESTful HTTP API
- JavaScript-Powered
Map/Reduce
- N-Master Replication
- Robust Storage

Features



- **Schema-Free (JSON)**
- Document Oriented,
Not Relational
- Highly Concurrent
- RESTful HTTP API
- JavaScript-Powered
Map/Reduce
- N-Master Replication
- Robust Storage

Features





Documents

Schema-Free (JSON)

```
{  
  "_id": "BCCD12CBB",  
  "_rev": "AB764C",  
  
  "type": "person",  
  "name": "Darth Vader",  
  "age": 63,  
  "headware":  
    ["Helmet", "Sombrero"],  
  "dark_side": true  
}
```

Schema-Free (JSON)

```
{  
  "_id": "BCCD12CBB",  
  "_rev": "AB764C",  
  
  "type": "person",  
  "name": "Darth Vader",  
  "age": 63,  
  "headware":  
    ["Helmet", "Sombrero"],  
  "dark_side": true  
}
```

Schema-Free (JSON)

```
{
  "_id": "BCCD12CBB",
  "_rev": "AB764C",

  "type": "person",
  "name": "Darth Vader",
  "age": 63,
  "headware":
    ["Helmet", "Sombrero"],
  "dark_side": true
}
```

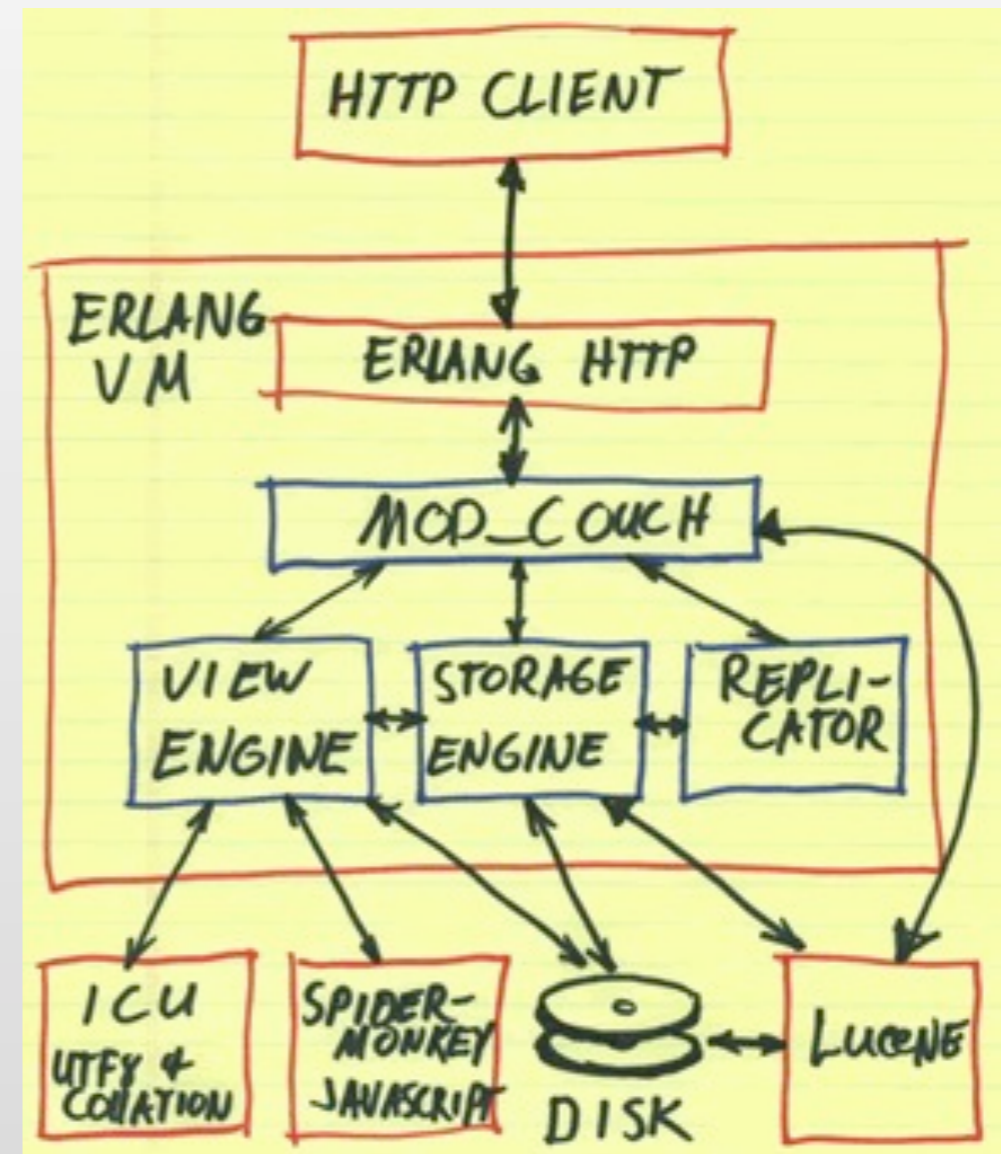
Schema-Free (JSON)

```
{
  "_id": "BCCD12CBB",
  "_rev": "AB764C",

  "type": "person",
  "name": "Darth Vader",
  "age": 63,
  "headware":
    ["Helmet", "Sombrero"],
  "dark_side": true
}
```

- Schema-Free (JSON)
- **Document-Oriented, Not Relational**
- Highly Concurrent
- RESTful HTTP API
- JavaScript-Powered Map/Reduce
- N-Master Replication
- Robust Storage

Features



Document-Oriented

Not Relational

- Documents in the Real World™
 - Bills, letters, tax forms...
 - Same type != same structure
 - Can be out of date (so what?)
 - No references

Document-Oriented

Not Relational

- Documents in the Real World™

- Bills, letters, tax forms...

Natural Data

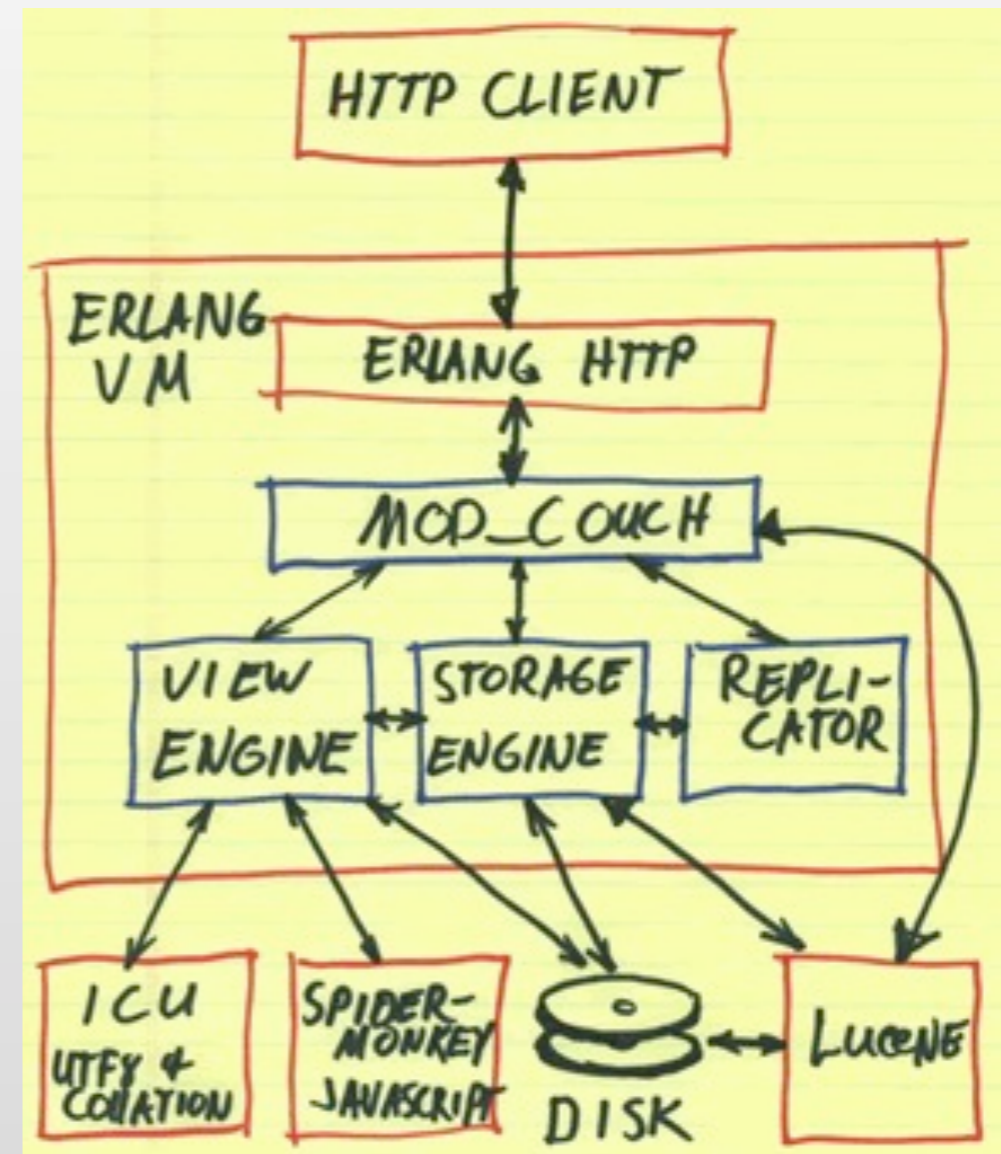
- Same type != same structure

Behaviour

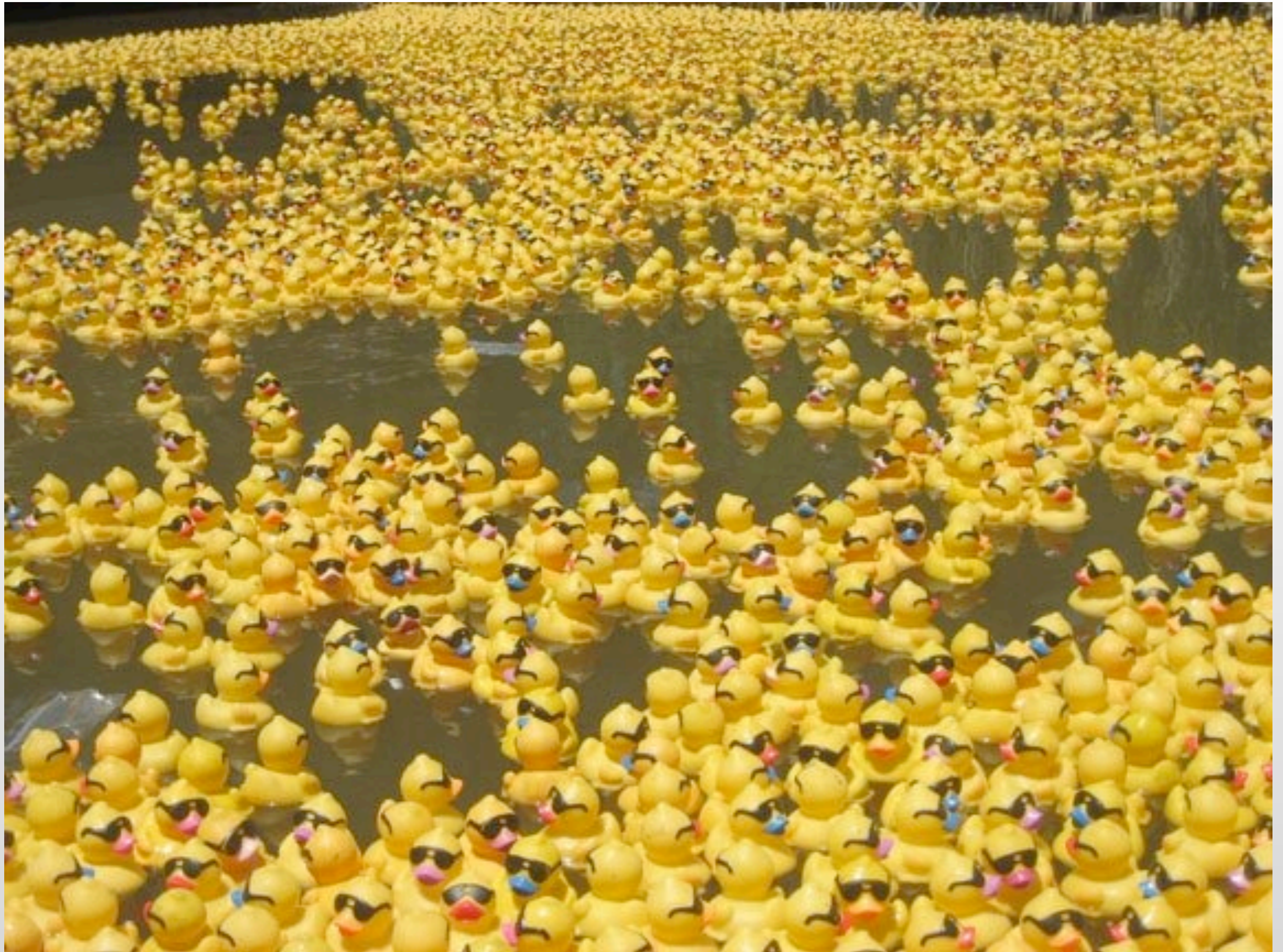
- Can be out of date (so what)
- No references

- Schema-Free (JSON)
- Document-Oriented,
Not Relational
- **Highly Concurrent**
- RESTful HTTP API
- JavaScript-Powered
Map/Reduce
- N-Master Replication
- Robust Storage

Features







Highly Concurrent

Erlang FTW

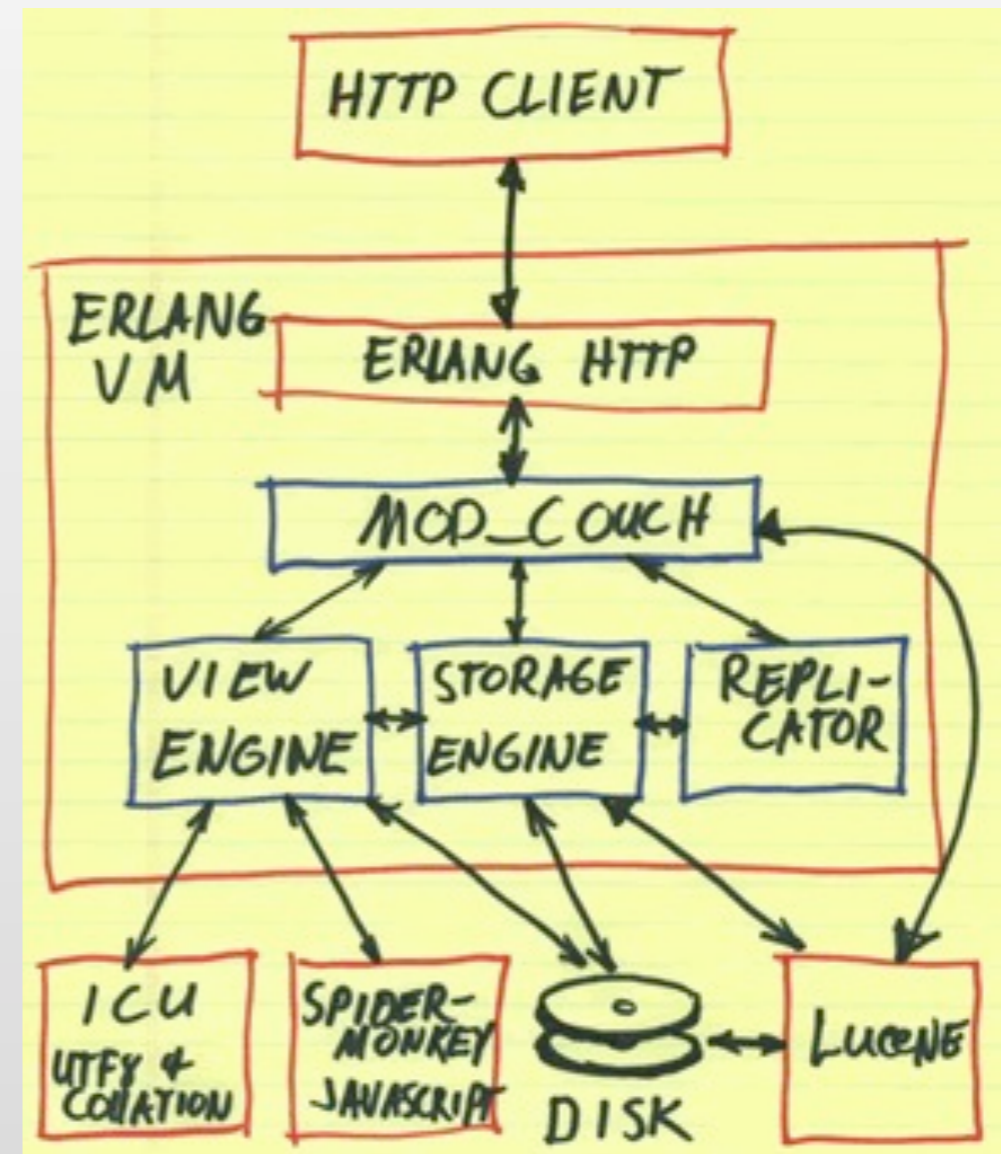
- Functional languages highly appropriate for parallelism
- Lightweight “processes” and message-passing; “shared-nothing”
- Easy to create fault-tolerant systems

MVCC

- Multiversion Concurrency Control
- Reads: lock-free; never block
 - Potential for massive horizontal scaling
- Writes: all-or-nothing
 - Success
 - Fail: conflict error, fetch and try again

- Schema-Free (JSON)
- Document-Oriented, *Not Relational*
- Highly Concurrent
- **RESTful HTTP API**
- JavaScript-Powered Map/Reduce
- N-Master Replication
- Robust Storage

Features



RESTful HTTP API CRUD

- **Create**
HTTP PUT /db/mydocid
- **Read**
HTTP GET /db/mydocid
- **Update**
HTTP PUT /db/mydocid
- **Delete**
HTTP DELETE /db/mydocid

RESTful HTTP API

Example

```
couch = CouchRest.database!("http://  
127.0.0.1:5984/tweets")
```

```
tweets_url = "http://twitter.com/statuses/  
user_timeline.json"
```

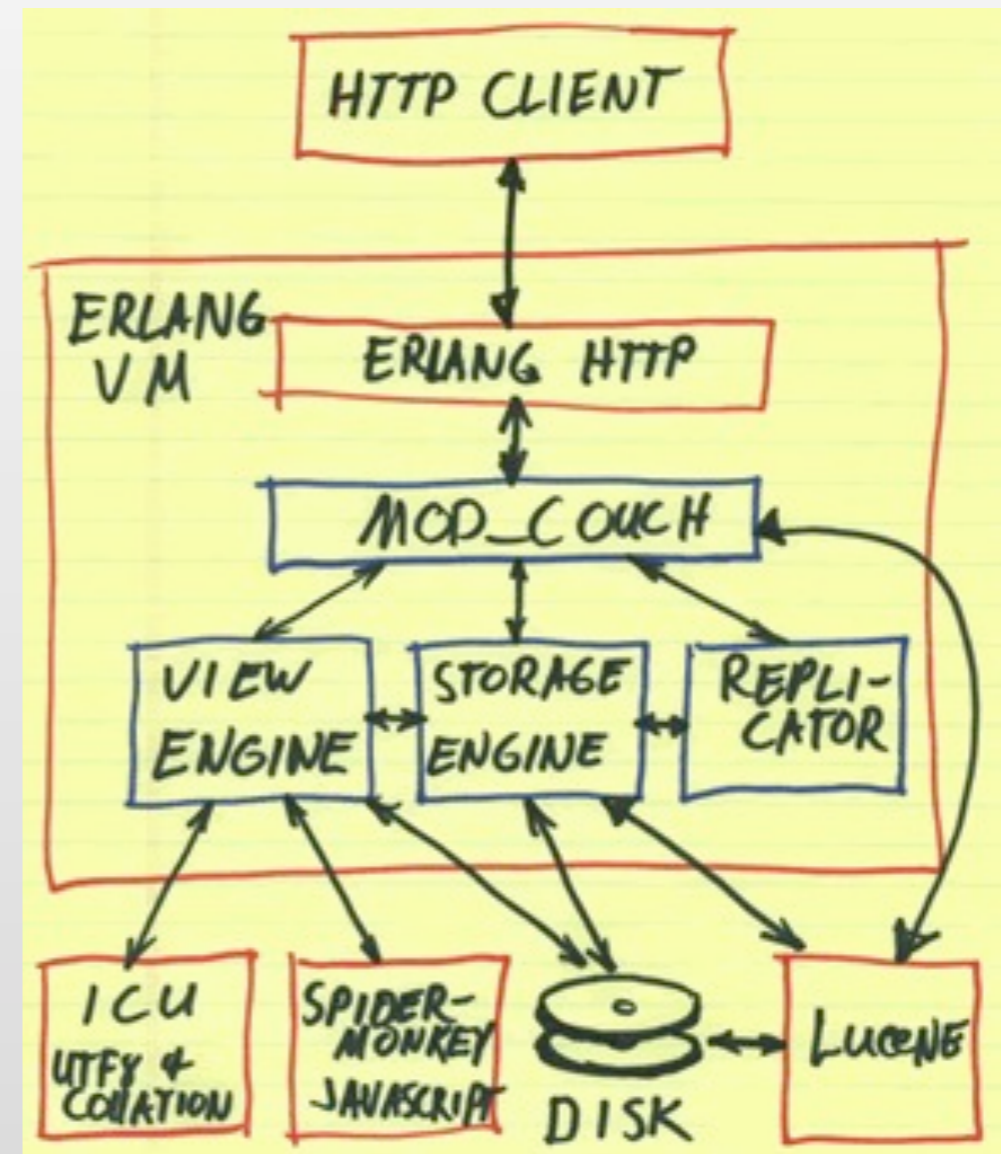
```
tweets = http.get(tweets_url)  
couch.bulk_save(tweets)
```


Cacheability

- Both documents and views return ETags
- Clients send If-None-Match
 - CouchDB responds with 304 Not Modified and bypasses potentially expensive lookup
- Can use Varnish/Squid as caching proxy
- Proxy- friendly

- Schema-Free (JSON)
- Document-Oriented, *Not Relational*
- Highly Concurrent
- RESTful HTTP API
- **JavaScript-Powered Map/Reduce**
- N-Master Replication
- Robust Storage

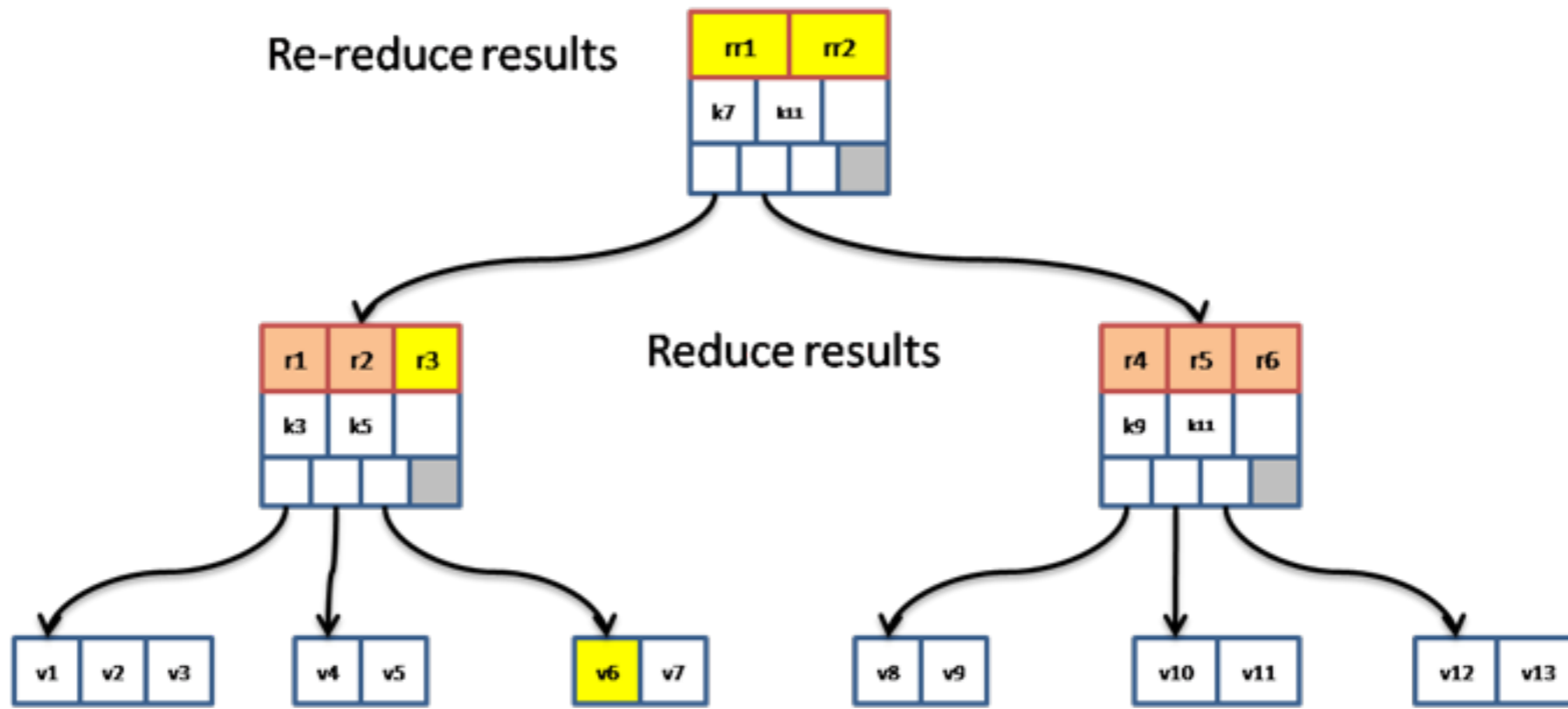
Features



JavaScript-Powered Map/Reduce

- Map functions extract data from your documents
- Reduce functions aggregate intermediate values
- The kicker: Incremental B-tree storage

View Index B+ Tree

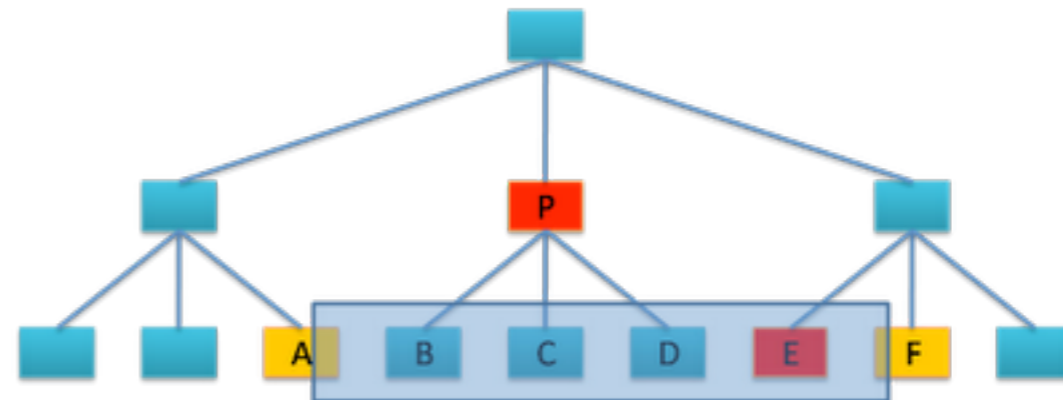


Multiple B+Trees
(one tree per view)

Each B+Tree holds the
results of the map()
execution

Key of B+Tree entry :
[emit_key, doc_id]

Header (B+Tree root)
Header Duplicate
B+Tree ViewX nodeA
nodeA's parent node
B+Tree ViewX root
B+Tree ViewY nodeB
nodeB's parent node
B+Tree ViewY root
B+Tree ViewX nodeP
nodeP's parent node
B+Tree ViewX root
B+Tree ViewZ nodeQ
nodeQ's parent node
B+Tree viewZ root



Map/Reduce Views

Docs

```
{ "user": "Chris",  
  "points": 3 }  
{ "user": "Joe",  
  "points": 10 }  
{ "user": "Alice",  
  "points": 5 }  
{ "user": "Mary",  
  "points": 9 }  
{ "user": "Bob",  
  "points": 7 }
```

Map

```
function(doc) {  
  if (doc.user && doc.points) {  
    emit(doc.user, doc.points);  
  }  
}
```

```
{ "key": "Alice", "value": 5 }  
{ "key": "Bob", "value": 7 }  
{ "key": "Chris", "value": 3 }  
{ "key": "Joe", "value": 10 }  
{ "key": "Mary", "value": 9 }
```

Reduce

```
function(keys, values, rereduce) {  
  return sum(values);  
}
```

```
Alice ... Chris: 15  
Everyone: 34
```

Map/Reduce Views

Docs

```
{ "user": "Chris",  
  "points": 3 }  
{ "user": "Joe",  
  "points": 10 }  
{ "user": "Alice",  
  "points": 5 }  
{ "user": "Mary",  
  "points": 9 }  
{ "user": "Bob",  
  "points": 7 }
```

Map

```
function(doc) {  
  if (doc.user && doc.points) {  
    emit(doc.user, doc.points);  
  }  
}
```

```
{ "key": "Alice", "value": 5 }  
{ "key": "Bob", "value": 7 }  
{ "key": "Chris", "value": 3 }  
{ "key": "Joe", "value": 10 }  
{ "key": "Mary", "value": 9 }
```

Reduce

```
function(keys, values, rereduce) {  
  return sum(values);  
}
```

```
Alice ... Chris: 15  
Everyone: 34
```

Map/Reduce Views

Docs

```
{ "user": "Chris",  
  "points": 3 }  
{ "user": "Joe",  
  "points": 10 }  
{ "user": "Alice",  
  "points": 5 }  
{ "user": "Mary",  
  "points": 9 }  
{ "user": "Bob",  
  "points": 7 }
```

Map

```
function(doc) {  
  if (doc.user && doc.points) {  
    emit(doc.user, doc.points);  
  }  
}
```

```
{ "key": "Alice", "value": 5 }  
{ "key": "Bob", "value": 7 }  
{ "key": "Chris", "value": 3 }  
{ "key": "Joe", "value": 10 }  
{ "key": "Mary", "value": 9 }
```

Reduce

```
function(keys, values, rereduce) {  
  return sum(values);  
}
```

```
Alice ... Chris: 15  
Everyone: 34
```

Render Views as HTML

lists/index.js

/drl/_list/sofa/index/recent-posts?descending=true&limit=8

```
1 function(head, row, req) {
2   // ijson templates.index
3   // ijson blog
4   // lcode helpers.couchapp
5   // lcode helpers.template
6   // log(req.headers.Accept);
7   var indexPath = listPath('index', 'recent-posts', {descending:true, limit:8});
8   var feedPath = listPath('index', 'recent-posts', {descending:true, limit:8, format:'atom'});
9   return respondWith(req, {
10    html : function() {
11      if (head) {
12        return template(templates.index.head, {
13          title : blog.title,
14          newPostPath : showPath('edit'),
15          index : indexPath,
16          assets : assetPath()
17        });
18      } else if (row) {
19        var post = row.value;
20        return template(templates.index.row, {
21          title : post.title,
22          summary : post.summary,
23          date : post.created_at,
24          link : showPath('post', row.id),
25          assets : assetPath()
26        });
27      } else {
28        return template(templates.index.tail, {
29          assets : assetPath()
30        });
31      }
32    },
33    atom : function() {
34      // with first row in head you can do updated.
35      if (head) {
36        var f = <feed xmlns="http://www.w3.org/2005/Atom"/>;
37        f.title = blog.title;
38        f.id = makeAbsolute(req, indexPath);
39        f.link.href = makeAbsolute(req, feedPath);
40        f.link.rel = "self";
41        f.generator = "Sofa on CouchDB";
42        f.updated = new Date().rfc3339();
43        return (body:f.toString().replace(/<\s\/feed\s\/, ""));
44      } else if (row) {
45        var entry = <entry/>;
46        entry.id = makeAbsolute(req, '/' + encodeURIComponent(req.info.db_name) + '/' + encodeURIComponent(row.value.id));
47        entry.title = row.value.title;
48        entry.content = row.value.summary;
49        entry.content.type = "html";
50        entry.updated = new Date(row.value.created_at).rfc3339();
51        entry.author = <author name="[row.value.author]"/></author>;
52        entry.link.href = makeAbsolute(req, showPath('post', row.id));
53        entry.link.rel = "alternate";
54        return (body:entry);
55      } else {
56        return (body : "</feed>");
57      }
58    }
59  });
60 }
```

Daytime Running Lights

web, vinyl, Scala, quote, photo,
music, lorem, life, hello world,
dev, couchdb, Couch, code couchapp,
code

Recently...

Peer Commit
7 days ago
I don't know the literature, but I have an idea for a shared data-space across peer nodes (not in a cluster). A replication helper, which tracks all known replicas, and their last successfully replicated sequence num. In this way, Node A can, for each update, provide to the user a list of remote nodes that have knowledge of the update. I think t...

Old News from a New Framework
12 days ago
Sling is a Scala app server for building Ajax CouchDB apps. The second half of this quote is Old News: By taking advantage of these characteristics pure-CouchDB applications can reach near parity with traditional server-heavy approaches. Notable exceptions include the ability to render dynamic text that isn't JSON, or return results that requi...

Deployed Sofa
16 days ago
For more information about this blog software, visit the Sofa source code on Github I plan to run my blog from this software, and I've got a few other projects in the pipeline. Also, don't forget about the Twitter client. Same great software, great new address....

Talk Announcements
4 weeks ago
My conference calendar is rapidly filling up with exciting CouchDB things. Here's the full list of them, but also: A new episode of the CouchDB podcast is out. Download the mp3 or subscribe to the RSS feed CouchDB Talks Feb 25 in Portland. Demoing CouchApps at PDX.js (7pm) The Portland JavaScript Admirers Group had its first meetin...

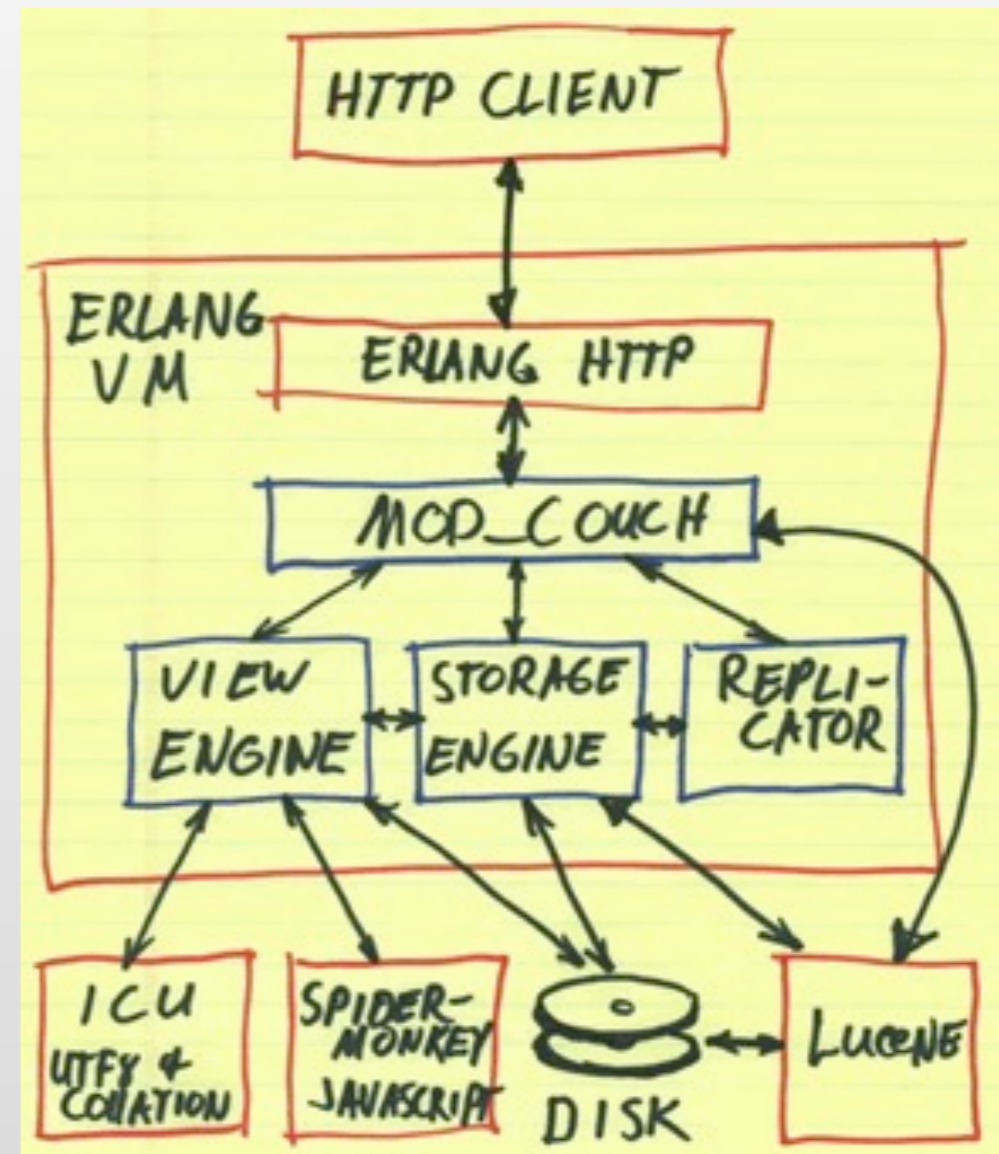
More Lorem
5 weeks ago
Lorem Lorem Lorem ...

Server-Side JavaScript

- `_show` for transforming documents
- `_list` for transforming views
- `_update` for transforming PUTs/POSTs
- Code-sharing between client and server
- Easy deployment

- Schema-Free (JSON)
- Document-Oriented,
Not Relational
- Highly Concurrent
- RESTful HTTP API
- JavaScript-Powered
Map/Reduce
- **N-Master Replication**
- Robust Storage

Features



Replication

- Incremental
- Near-real-time
 - Clustered mirrors
- Scheduled
- Ad-hoc

“Ground Computing”

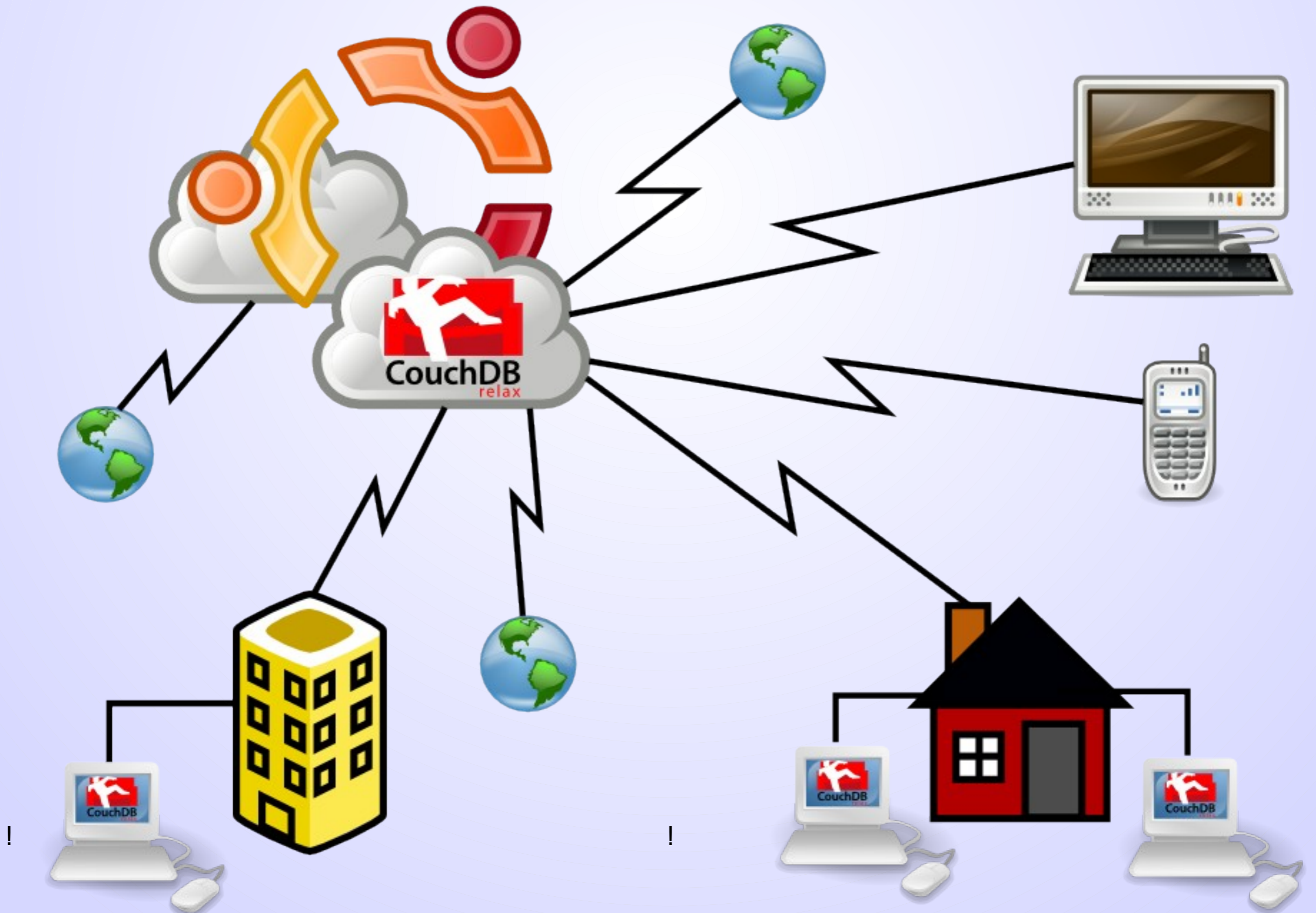
@jhuggins

FREMONT
5 CARS - BOARD CENTER
PLATFORM 2

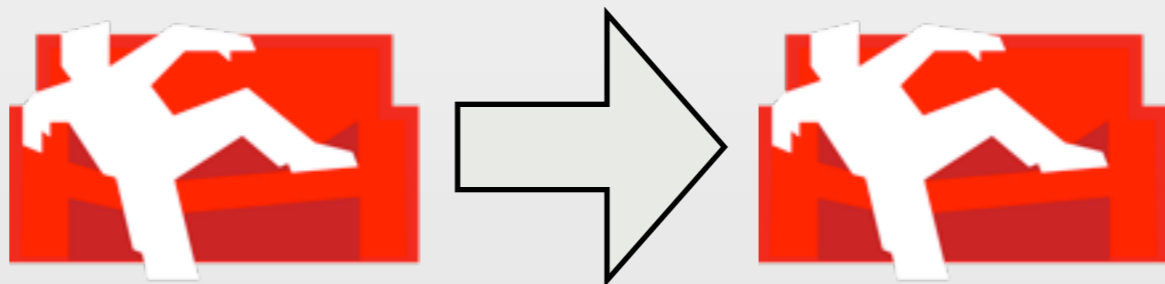


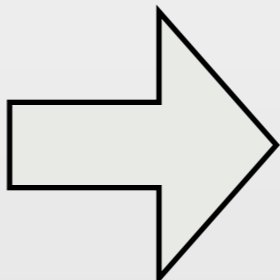
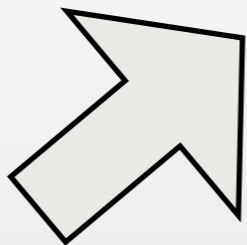
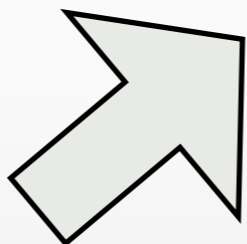
Latency Sucks

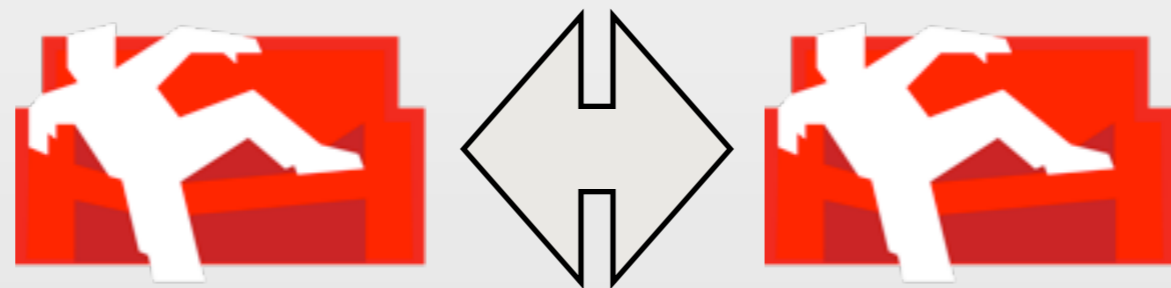
$$rt^b = rt^a + \sum_{i \in \text{Links}} (l_i^b - l_i^a) \cdot \vec{\nabla} \bar{rt}(l_i)$$

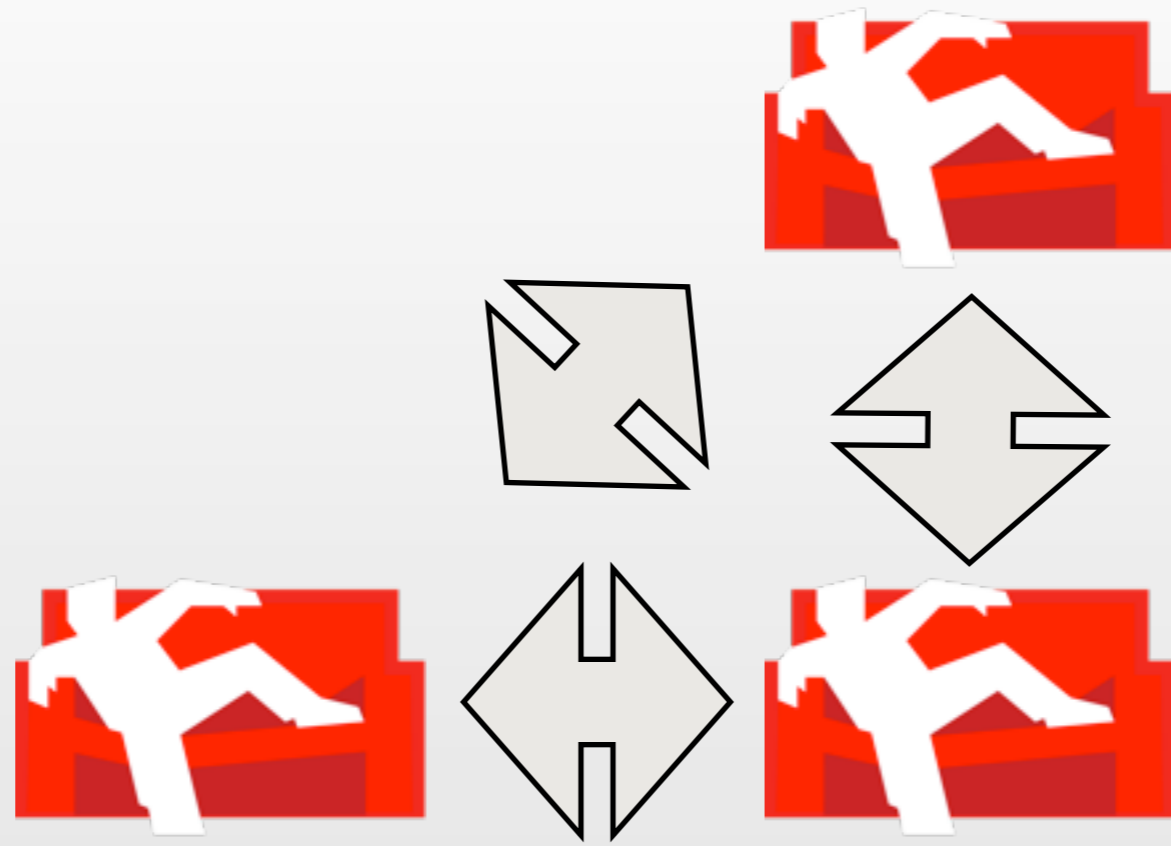


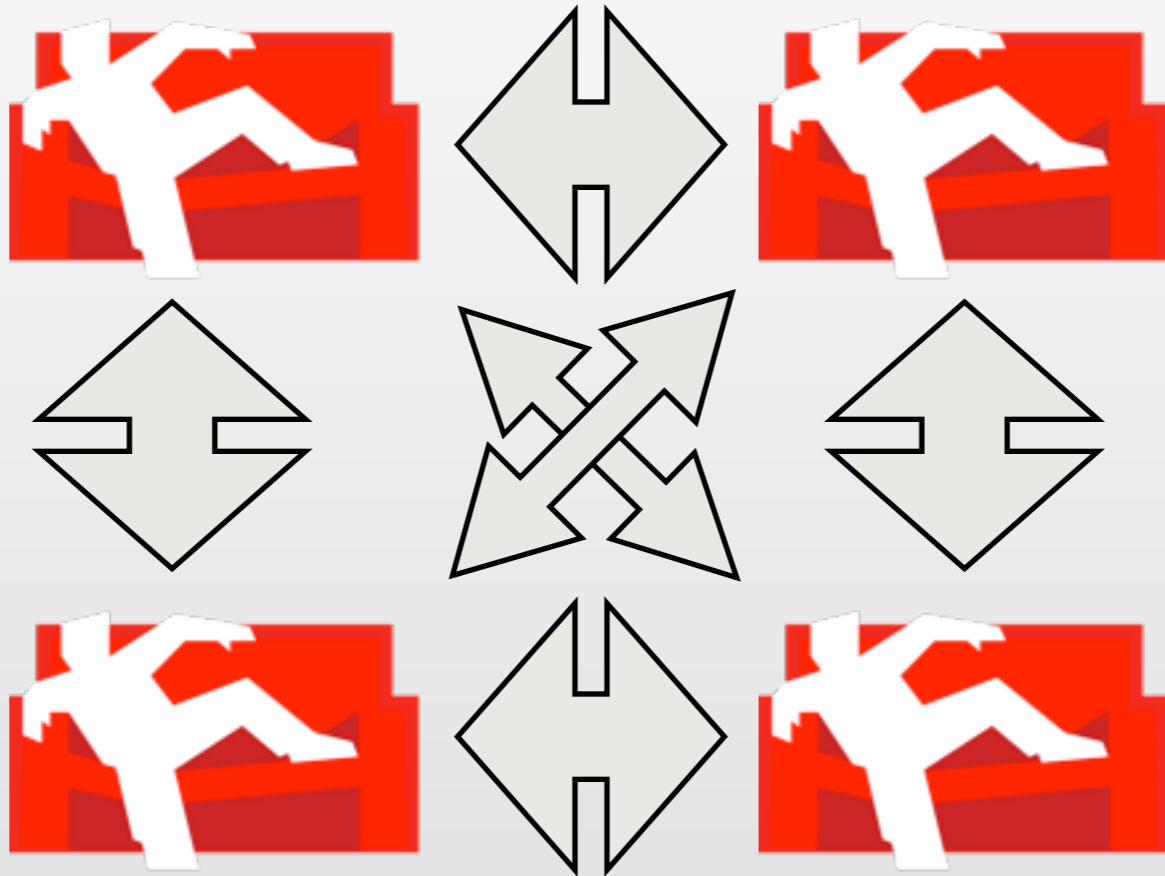


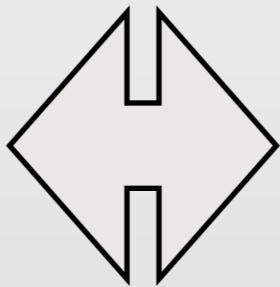
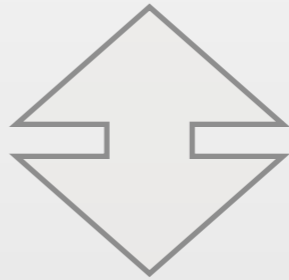
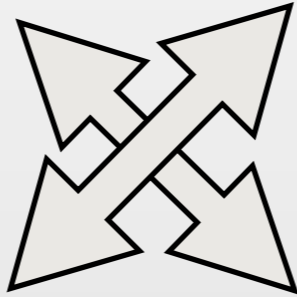
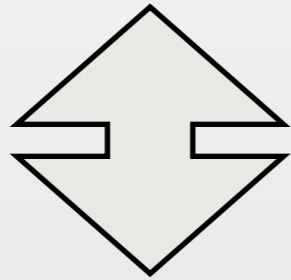


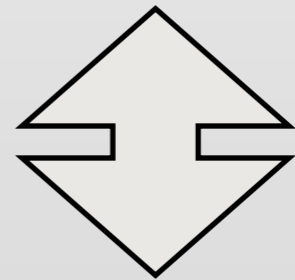
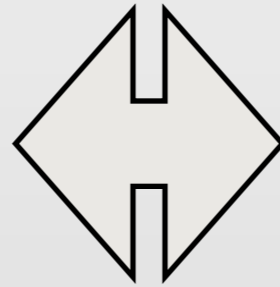
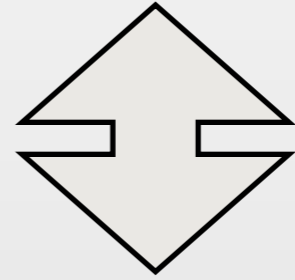
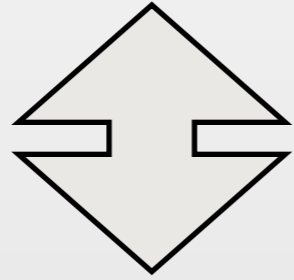
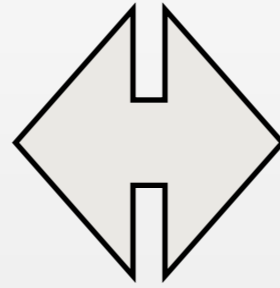
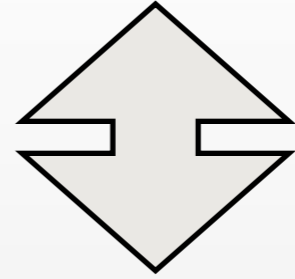


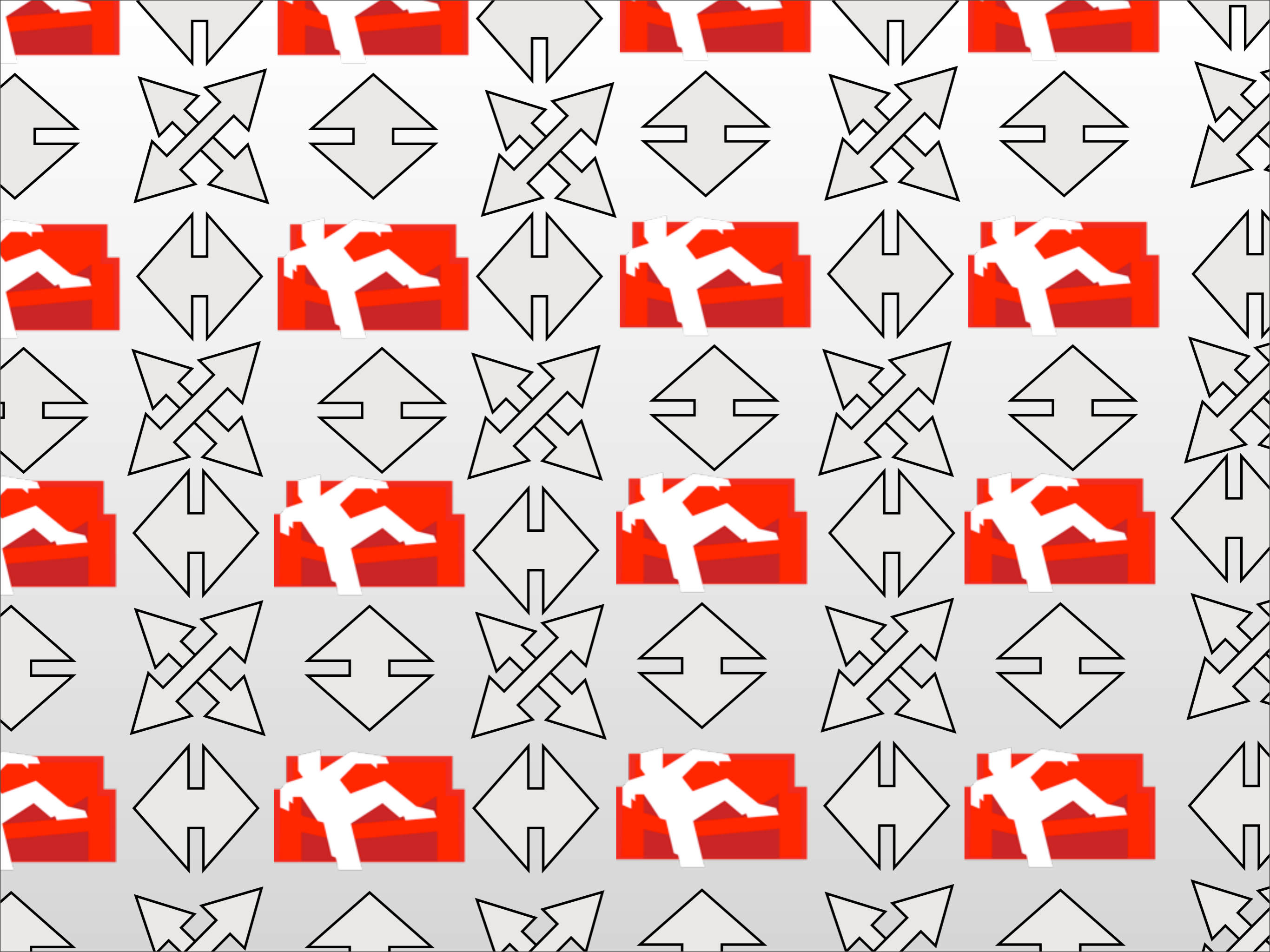










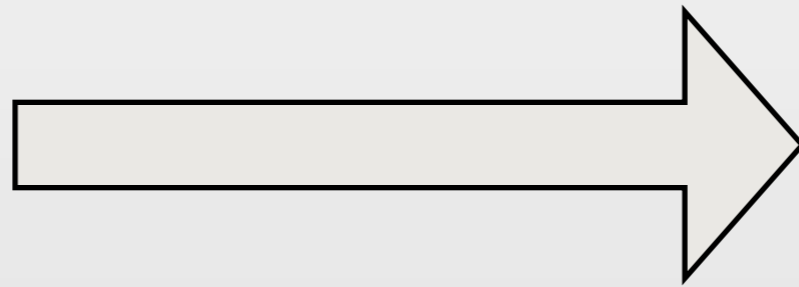


Conflicts

Conflict resolution by example



A



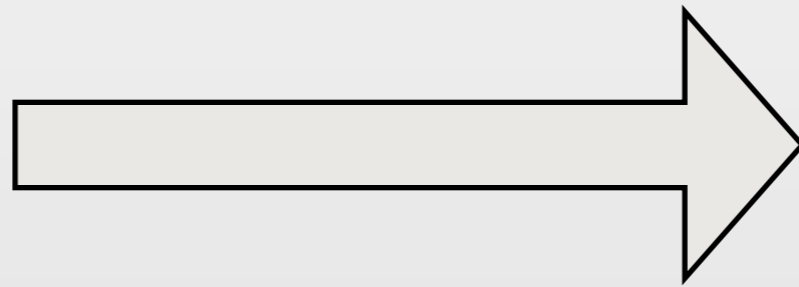
B



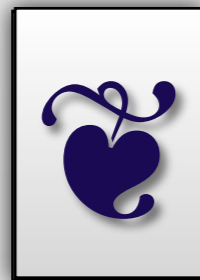
Conflict resolution by example



A



B



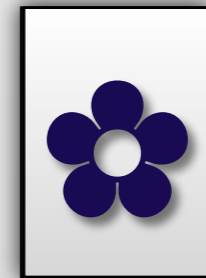
Conflict resolution by example



A



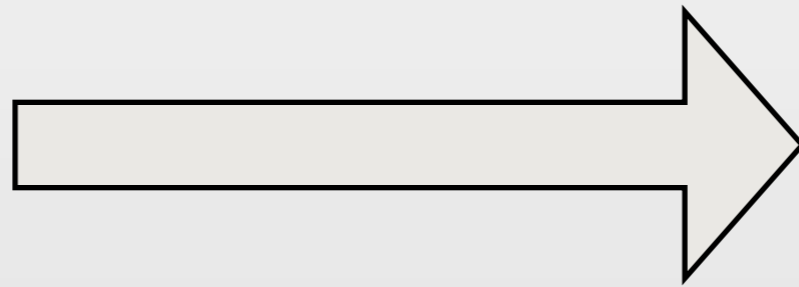
B



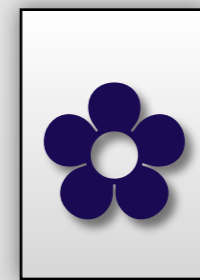
Conflict resolution by example



A



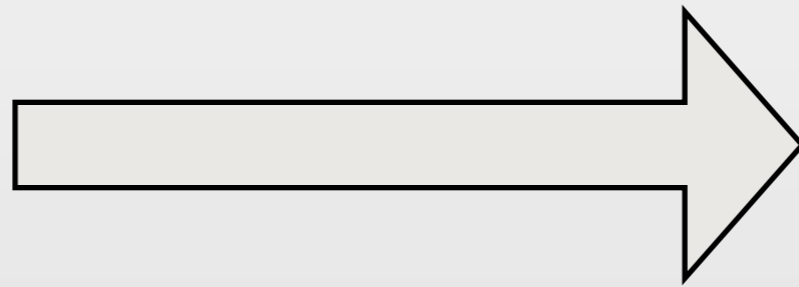
B



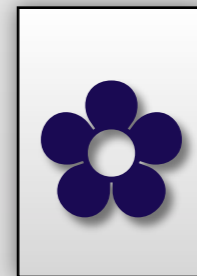
Conflict resolution by example



A

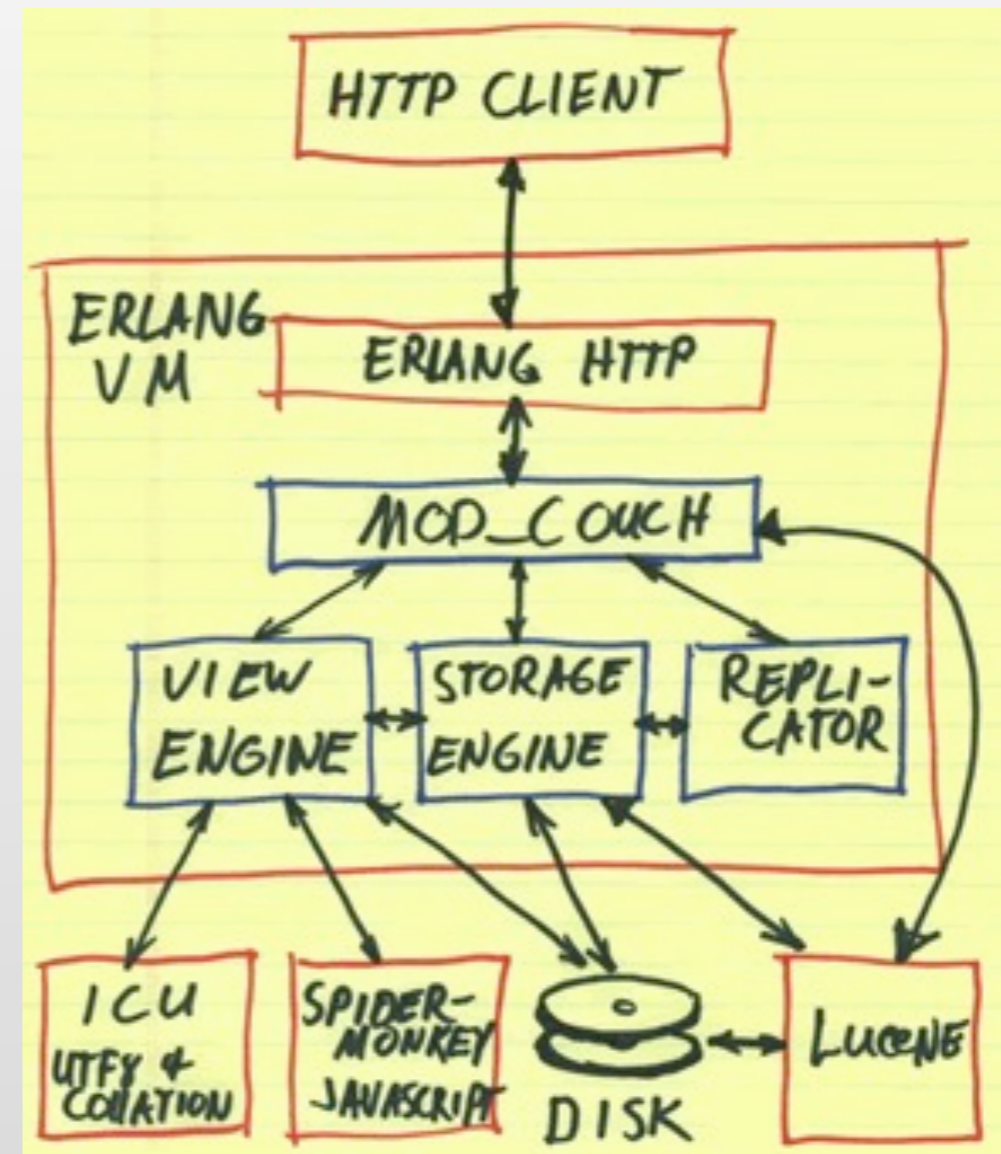


B



- Schema-Free (JSON)
- Document-Oriented,
Not Relational
- Highly Concurrent
- RESTful HTTP API
- JavaScript-Powered
Map/Reduce
- N-Master Replication
- **Robust Storage**

Features



Robust Storage

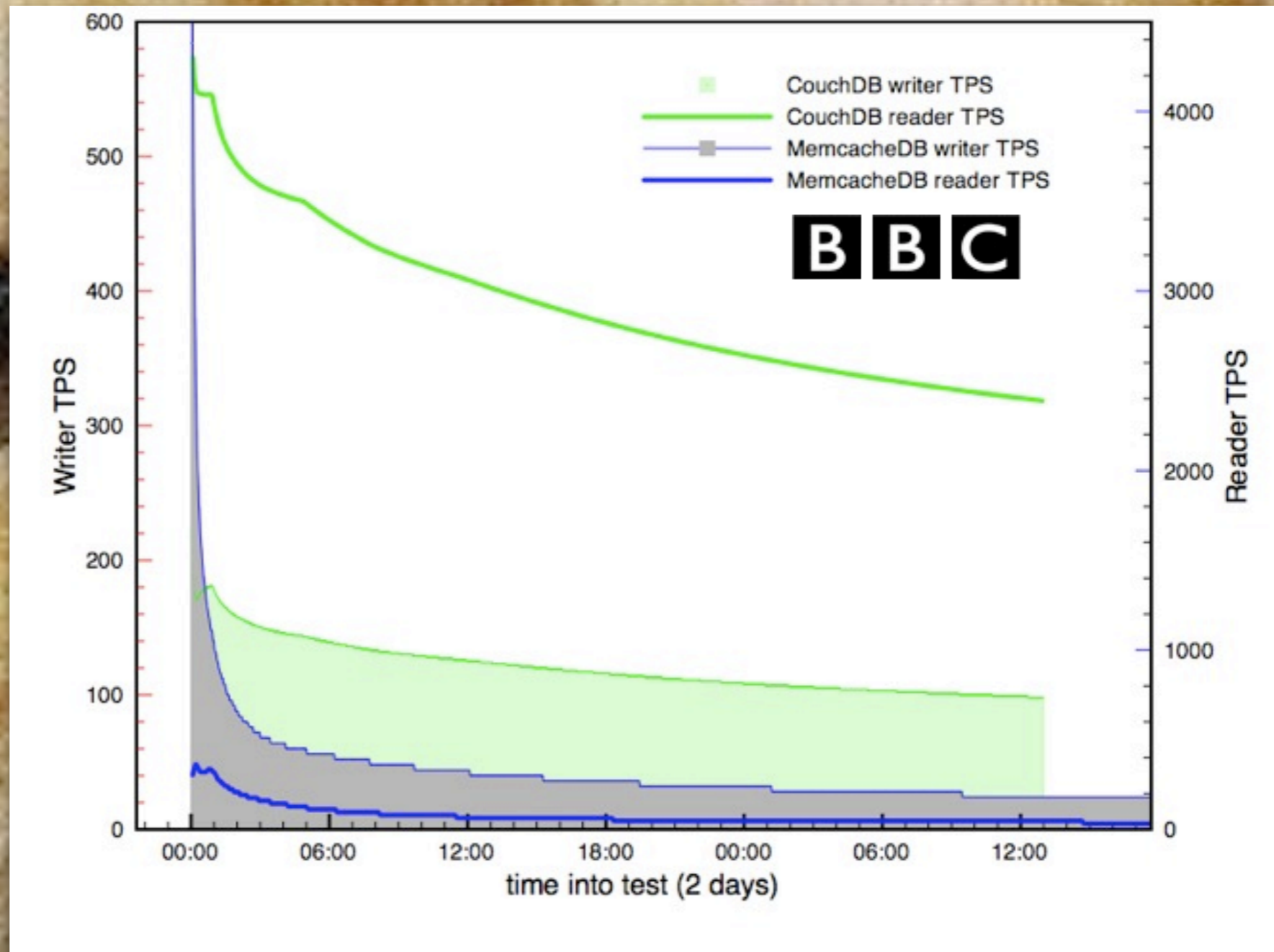
Append-Only
File Structure

Designed to
Crash

Instant-On



Robust



No Silver Bullet!

- Read-optimised; not so much for writes
 - Use `?batch=ok` for “don’t care” logging
- Views must be specified beforehand
 - Relational databases are better at highly dynamic queries
 - But: use externals e.g. Lucene, SQLite

Experiments!



`couchapp`

- Scripts written in Python to make developing pure CouchDB applications easier
- `sudo easy_install couchapp`
- `couchapp generate relax && cd relax`
- `couchapp push http://127.0.0.1:5984/mydb`

Directory Structure

```
elyservice/  
  _attachments/  
  lib/  
    helpers/  
      ejs.js  
  templates/  
    layout/  
      head.html  
      tail.html  
  shows/  
    contact.js  
    page.js
```

Resulting Design Doc

Field	Value
_attachments	<ul style="list-style-type: none">✕ index.html 413 bytes, text/html✕ style/main.css 21 bytes, text/css✕ vendor/couchapp/jquery.couchapp.js 6.8 KB, application/javascript
_id	"_design/relax"
_rev	"5-2484564001"
✕ couchapp	<ul style="list-style-type: none">+ signatures+ length+ objects+ manifest
✕ lib	<ul style="list-style-type: none">+ templates+ helpers
✕ lists	
✕ shows	
✕ vendor	+ couchapp
✕ views	

list

- Arbitrary JS transformation for views
- http://127.0.0.1:5984/mydb/_design/app/_list/myview?startkey=...&endkey=...
- JSON -> HTML, JSON -> XML, ...
- E4X nice for XML generation
- Iteratively call `getRow()` and use `send(...)`

```
function (head, req) {  
  var row;  
  send('<ul>');  
  while (row = getRow()) {  
    send('<li>' + row.value.name + '</li>');  
  }  
  return '</ul>';  
}
```

_show

- Arbitrary transformation for documents
- http://127.0.0.1:5984/mydb/_design/app/_show/mydoc
- `function (doc, req) { return "foo"; }`

JavaScript Templating

- EmbeddedJS (EJS)
 - `<% /* execute arbitrary JS */ %>`
 - `<%= /* execute and include result */ %>`
 - `new EJS({ text: mytemplate }).render(doc);`
- John Resig's Micro-Templating
 - `new template(mytemplate)(doc);`
 - Doesn't preserve whitespace or LaTeX backslashes

Push Helper Macros

- Simple macros to facilitate code re-use
- Insert code directly
 - `// !code path/to/code.js`
- Encode file as JSON: `path/to/test.html`
 - `// !json path.to.test`
 - `// !json _attachments/test.html`

Secure Cookie Authentication

- Reasonable performance/simplicity of JavaScript implementation
- Mutual authentication
- Resistance to off-line dictionary attacks based on passive eavesdropping
- Passwords stored in a form that is not plaintext-equivalent
- Limited resistance to replay attacks



EPIC BEARD

Safest place to keep your private key

Tamper-Proof Cookies

```
username + ':' + timestamp + ':' + HMAC(username + ':' + timestamp)
```

Timestamp + signature => limited forward-security
(outside of timestamp window)

Secure Remote Password Protocol (SRP)

- Zero-Knowledge Password Proof
- Simple to implement in Erlang using BigInteger and crypto libraries
- JavaScript too slow: over 5s for 1024 bits
- Vulnerable to active injection attacks
- There are simpler protocols that can be used to give equivalent security
- Just add SSL for protection from active attacks (or lobby for TLS-SRP/J-PAKE!)

#couchdb

Update

jan (521 lines total)

30 APR 2009

30 APR 2009

30 APR 2009

30 APR 2009

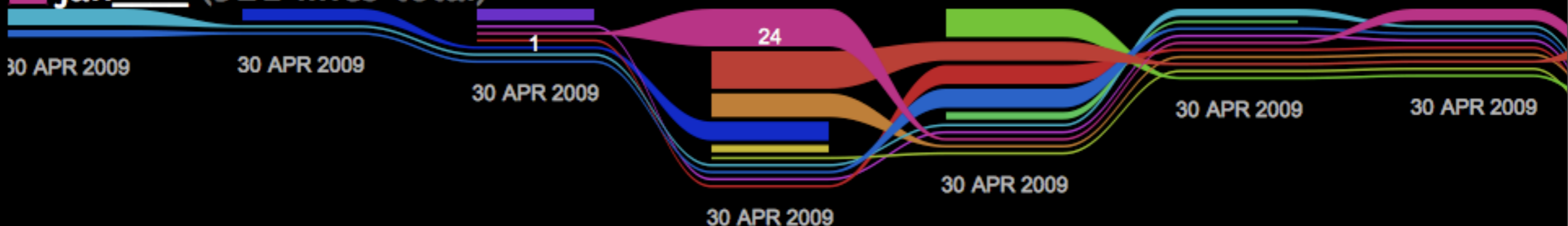
30 APR 2009

30 APR 2009

30 APR 2009

1

24





Ely Service

Specialists in Classic Rovers

We specialise in classic Rovers (P4, P5 and P6 models) as well as other classic cars, ranging from a 1937 Austin 10 Cambridge through the decades to modern-day classics from the '90s.

Classic Car Services

- Basic servicing
- M.O.T.
- complete restoration, including:
- mechanical work
- bodywork
- paintwork
- interior trim
- rechroming
- supply and fitting of correct tyres for your classic.



[Read more...](#)

Modern Car Services

For those of you unlucky enough ever to have anything but a modern 'tin box'—but we love them really!—we offer you a similar level of service to that which our classic-car customers enjoy.

[Read more...](#)



History

The business first opened sometime in 1930, but the important date is March 1st, 1931, when Mr Charles Hull purchased the business. The following years saw the Company go from strength to strength, enjoying a number of franchises, including those of the well-known Rootes



A Message from the Owner

I bought my first P6 in 1990 and they have gradually taken over. Over time I then progressed into P5 ownership as well. So when my chance came to take over at Ely, it seemed like manna from heaven!

[Read more...](#)



Shop by Category

[Anointing Oil](#)

[Books & Bibles](#)

[Candlesticks](#)

Edom Dead Sea

Products

[Flags](#)

[Hanukkah](#)

[Jewellery](#)

[Kiddush Cups](#)

[Kippahs](#)

[Menorahs](#)

[Mezuzahs](#)

[Music](#)

[Prayer Shawls \(Tallit\)](#)

[Shofars](#)

[T-shirts](#)

[Wall Hangings & Prints](#)

Edom Dead Sea Products

The remarkable benefits of the Dead Sea minerals on skin and body are now available to you with Edom's range of Dead Sea mineral-rich skin care products. The minerals in Edom's products stimulate the flow of oxygen to the skin cells, improving their metabolism and vitality, and thus improving the overall appearance of the skin.



Edom Dead Sea Body Lotion

In stock

£9.99

For all skin types. This Dead Sea mineral rich Body Lotion is mild, non-greasy and absorbed quickly by the skin. It ...

[More details...](#)



Edom Dead Sea Shower Gel

In stock

£8.99

Spoil yourself with a refreshing, invigorating and nourishing shower gel. Scented and mineral-rich, Edom shower Gel leaves your skin feeling ...

[More details...](#)

More Information

[Shipping & Returns](#)

[Privacy Policy](#)

[Conditions of Use](#)

[Links & Resources](#)

[How to Order](#)

[RSS Feeds](#)

Currency

Pound Sterling (£)

JASON DAVIES & RACHEL LOVELL

[HOME](#)[PHOTOS](#)[PROGRAMME](#)[DIRECTIONS](#)[GIFT LIST](#)[F.A.Q.](#)

Married on the 26th of July, 2008

Welcome! Please ensure you [reply to your invitation](#) by the first of July, in order for us to confirm numbers with [New Hall College](#). So far, 200 people have confirmed they are attending.

The proceedings will begin with the wedding ceremony at 2 o'clock in the afternoon, followed by the Drinks Reception and photographs in the Scholar's Garden. Please see the [Programme](#) for more information about the day.

If you are travelling from London or elsewhere outside Cambridge, please refer to the [Directions](#).

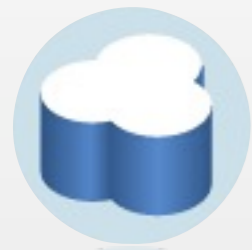
Those wishing to buy us gifts should refer to the [Gift List](#) for ideas.

If you have any other questions, please [get in touch](#).



The important date: Saturday, the twenty-sixth of July at 2 o'clock in the afternoon.

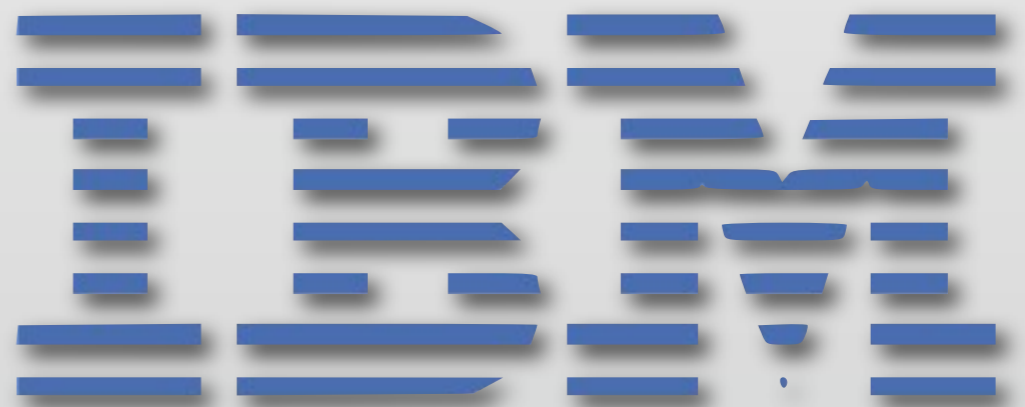
Time since Saturday, the twenty-sixth of July, MMVIII at two o'clock p.m.: 335 days, 11 minutes and 59 seconds.



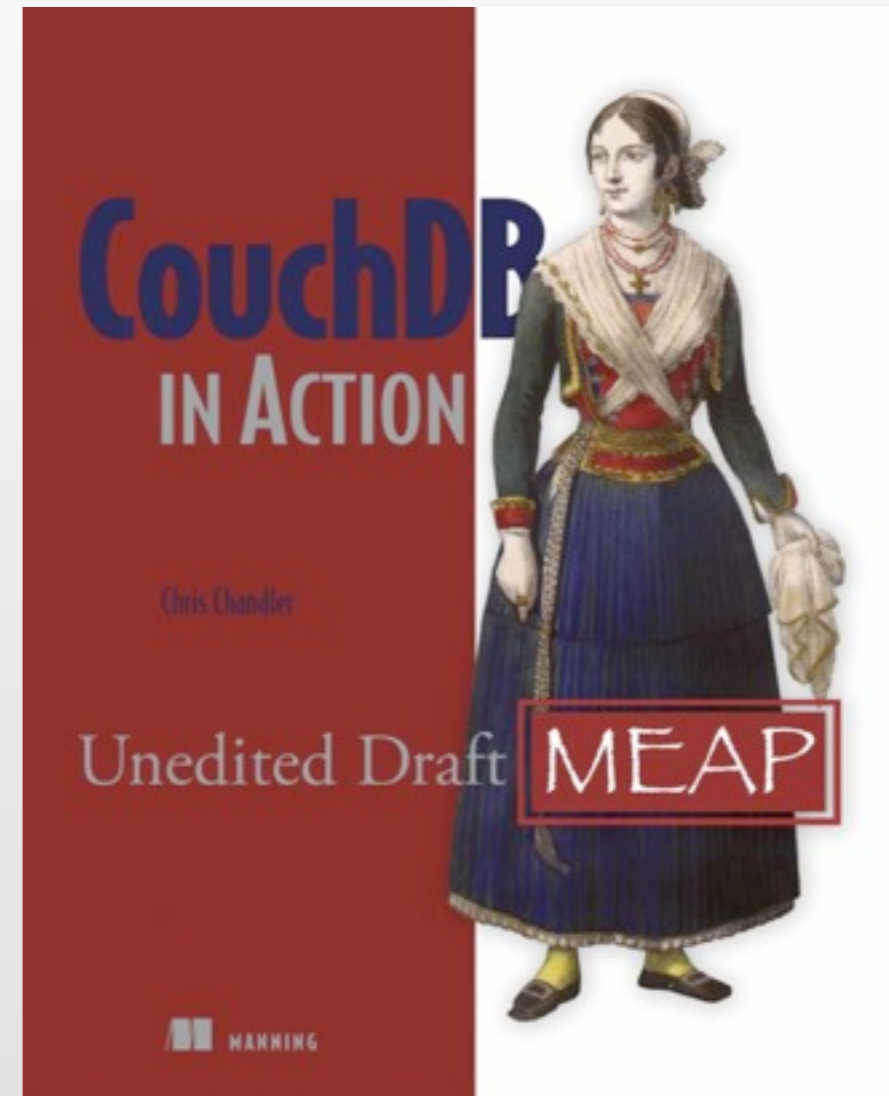
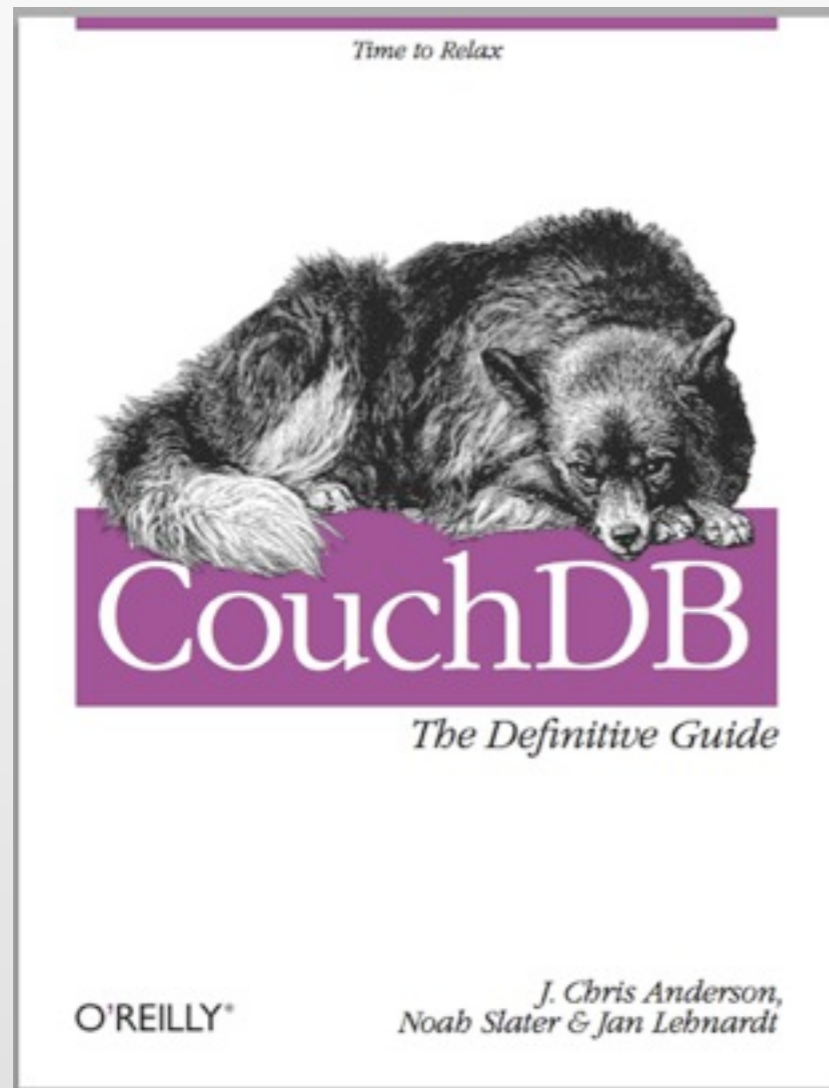
CLOUDANT

THE DATABASE DISTRIBUTION LEADER

THE DATABASE DISTRIBUTION LEADER



@couchdbinaction



<http://books.couchdb.org/relax>

Thank you
for listening!

www.jasondavies.com

@jasondavies