

Apache
CouchDB

Relax

Who's Talking?

- J Chris Anderson / jchris@apache.org / @jchris
 - PHP -> Rails -> JSON -> CouchDB
- Jan Lehnardt
 - CouchDB early contributor
- Directors, couch.io

And You?

- Scalable Key Value Storage
- Good Fit For Web Development
- Triggered Replication
- High-Throughput Applications



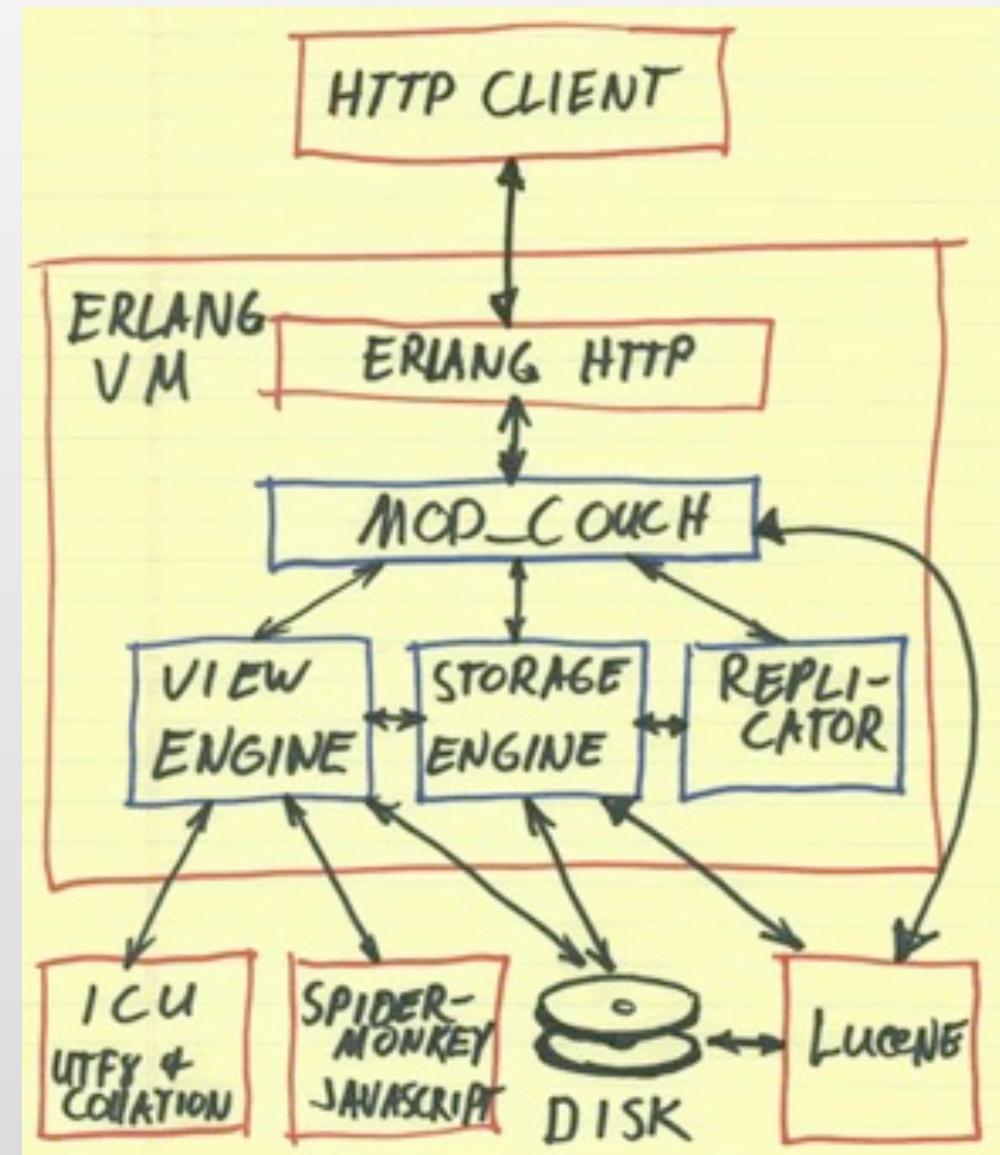
Damien Katz

Wassat?

CouchDB 0.9

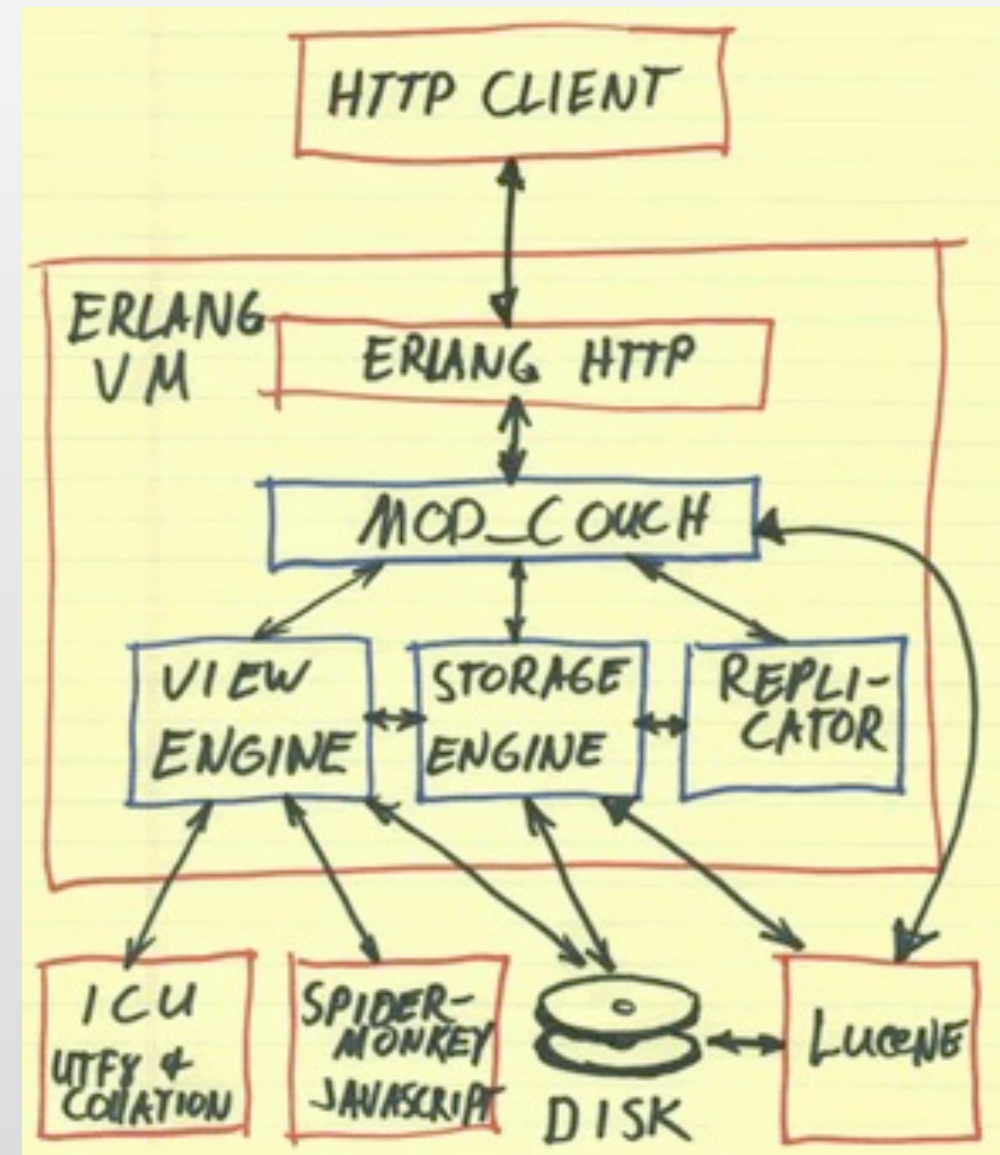
- Schema Free (JSON)
- Document Oriented,
Not Relational
- Highly Concurrent
- RESTful HTTP API
- JavaScript Powered
Map/Reduce
- N-Master Replication
- Robust Storage

Features



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

Features



Schema Free (JSON)

```
{  
  "_id": "BCCD12CBB",  
  "_rev": "1-AB764C",  
  
  "type": "person",  
  "name": "Darth Vader",  
  "age": 63,  
  "headware":  
    ["Helmet", "Sombrero"],  
  "dark_side": true  
}
```

Schema Free (JSON)

```
{  
  "_id": "BCCD12CBB",  
  "_rev": "2-AB764C",  
  
  "type": "person",  
  "name": "Darth Vader",  
  "age": 63,  
  "headware":  
    ["Helmet", "Sombrero"],  
  "dark_side": true  
}
```

Schema Free (JSON)

```
{  
  "_id": "BCCD12CBB",  
  "_rev": "3-AB764C",  
  
  "type": "person",  
  "name": "Darth Vader",  
  "age": 63,  
  "headware":  
    ["Helmet", "Sombrero"],  
  "dark_side": true  
}
```

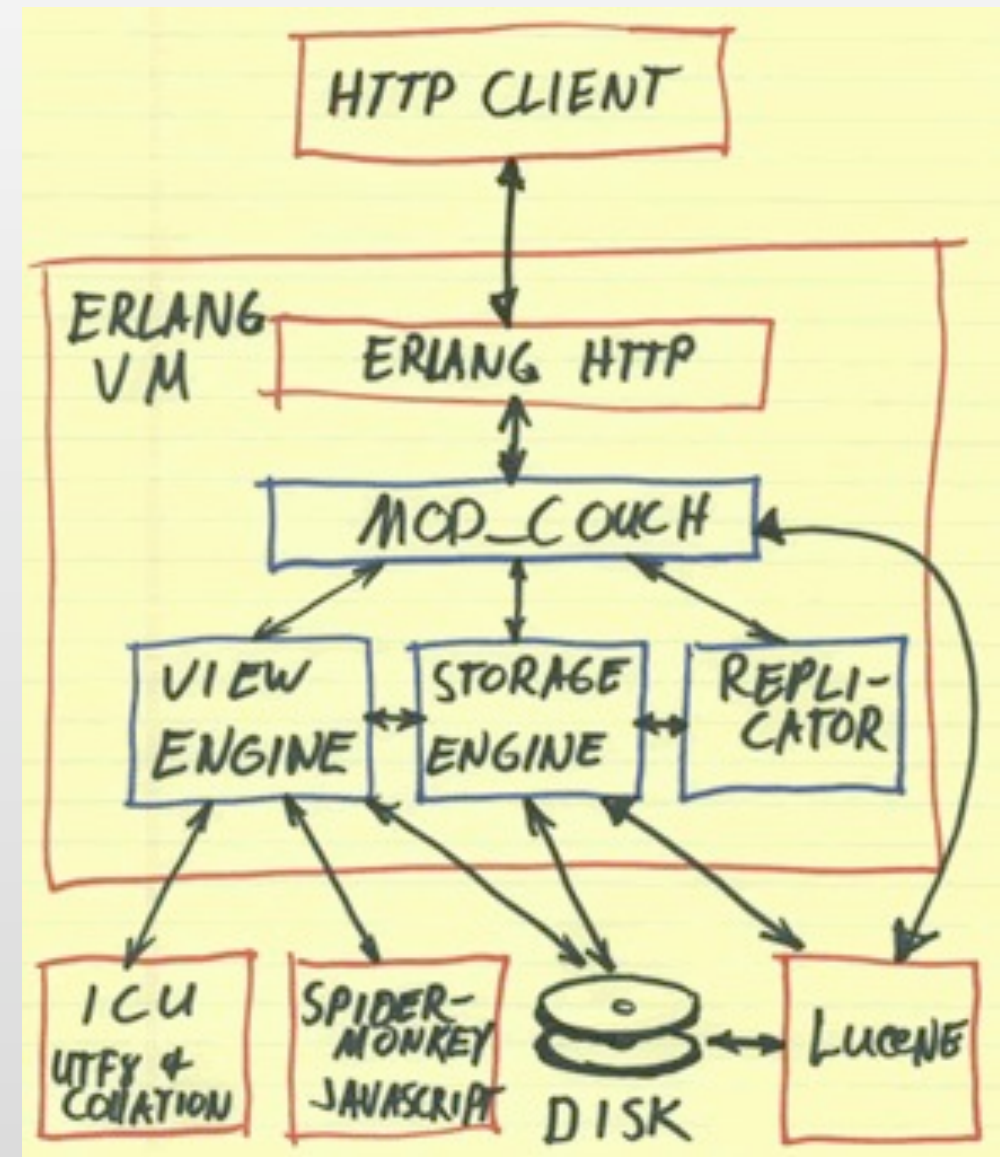
Schema Free (JSON)

```
{
  "_id": "BCCD12CBB",
  "_rev": "3-AB764C",

  "type": "person",
  "name": "Darth Vader",
  "age": 63,
  "headware":
    ["Helmet", "Sombbrero"],
  "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 - Can be out of date (so what?)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

Document Oriented

Not Relational

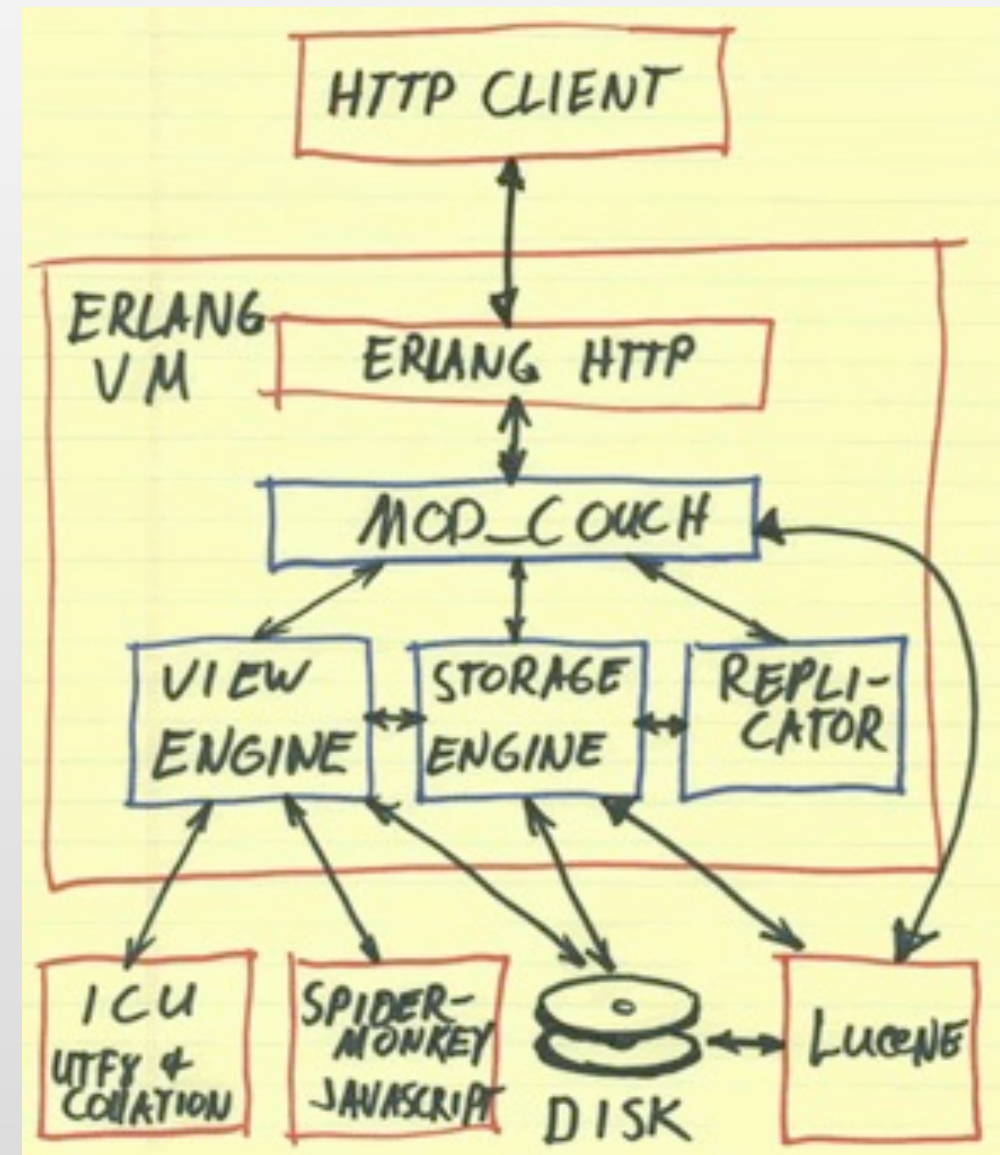
```
couch = CouchRest.database! ("http://  
127.0.0.1:5984/tweets")
```

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

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

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

Features



Highly Concurrent

Number Bragging

Highly Concurrent

Number Bragging

- Silly read-only benchmark with memory saturation

Highly Concurrent

Number Bragging

- Silly read-only benchmark with memory saturation
- 2,500 req/s sustained on a 2Ghz dual core 64-bit Athlon

Highly Concurrent

Number Bragging

- Silly read-only benchmark with memory saturation
- 2,500 req/s sustained on a 2Ghz dual core 64-bit Athlon
- **Using 9.8 MB RAM**

Highly Concurrent

Erlang Praising

Highly Concurrent

Erlang Praising

- Erlang is worth its own talk

Highly Concurrent

Erlang Praising

- Erlang is worth its own talk
- Executive Summary

Highly Concurrent

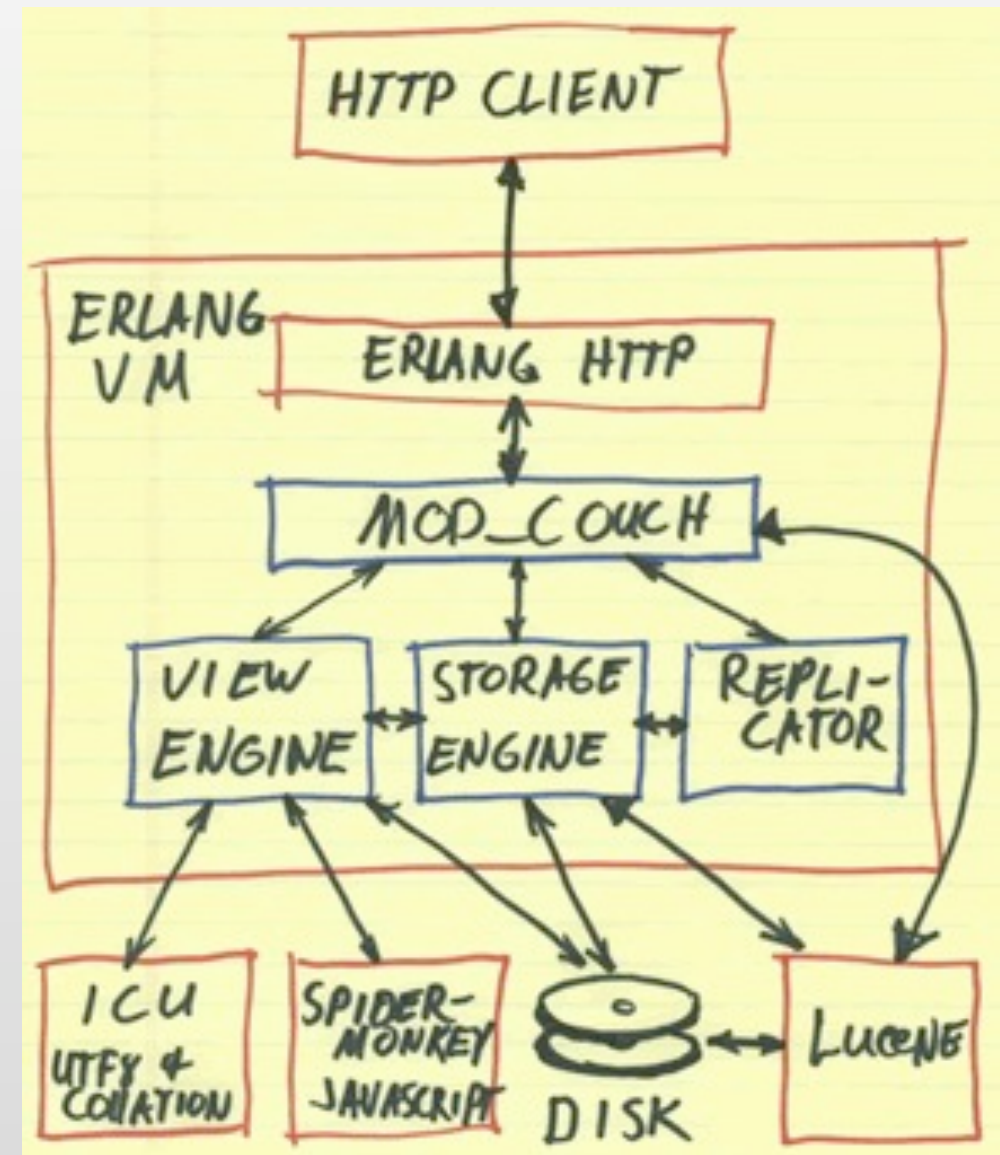
Erlang Praising

- Erlang is worth its own talk
- Executive Summary

*It's fucking
awesome!*

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

Features

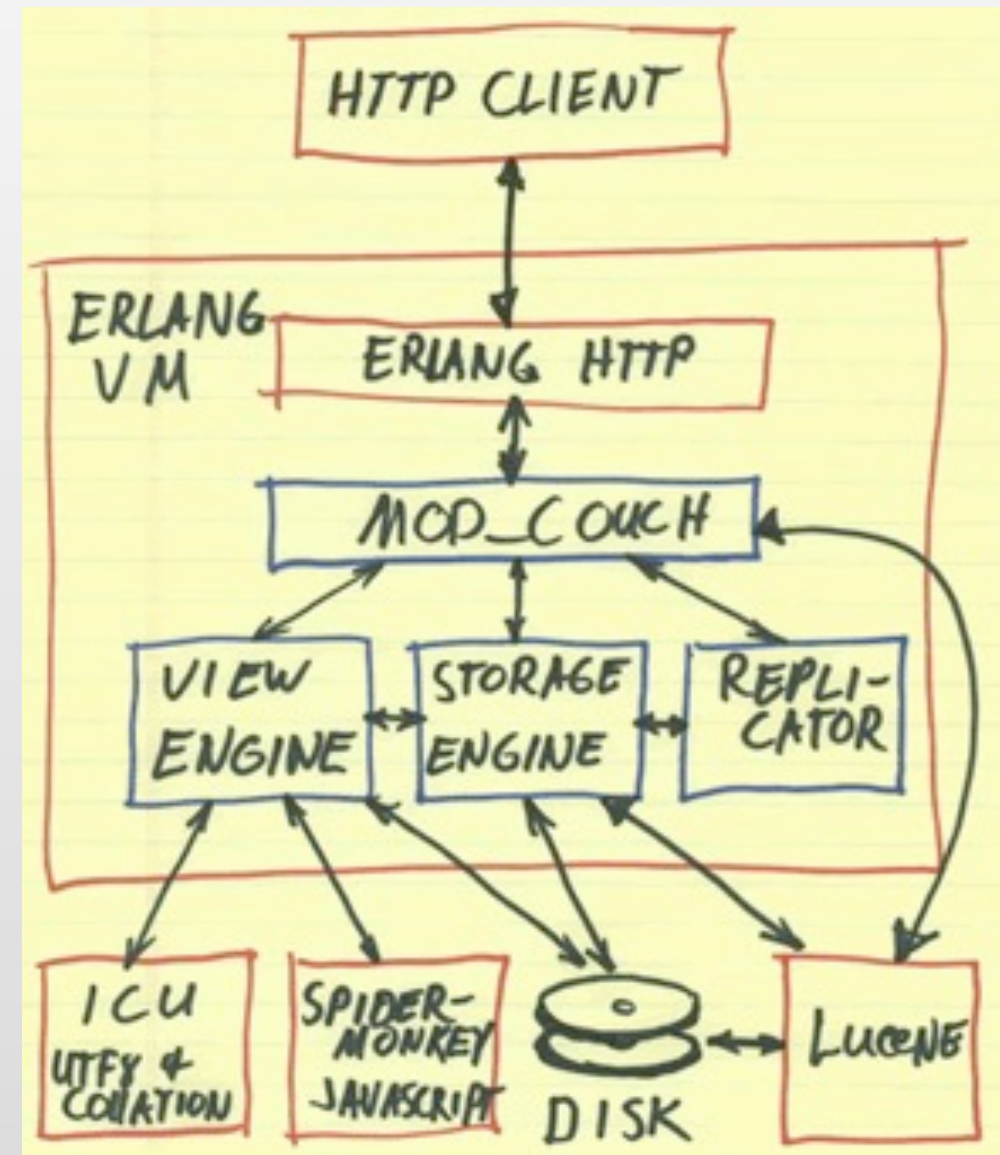


RESTful HTTP API CRUD

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

- 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 **yawn**.
- The kicker: Incremental b-tree storage.

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 }
```

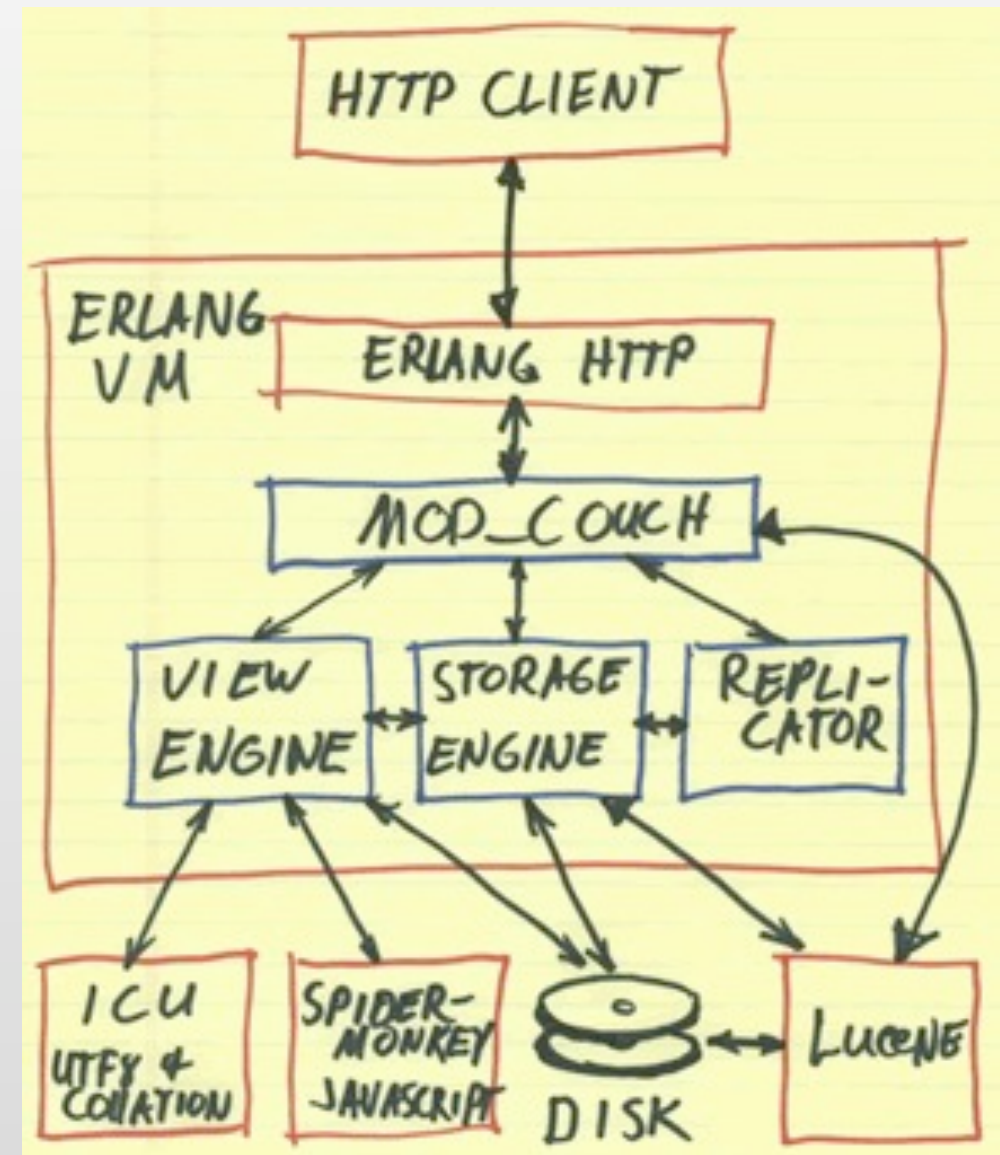
Reduce

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

Alice ... Chris: 15
Everyone: 34

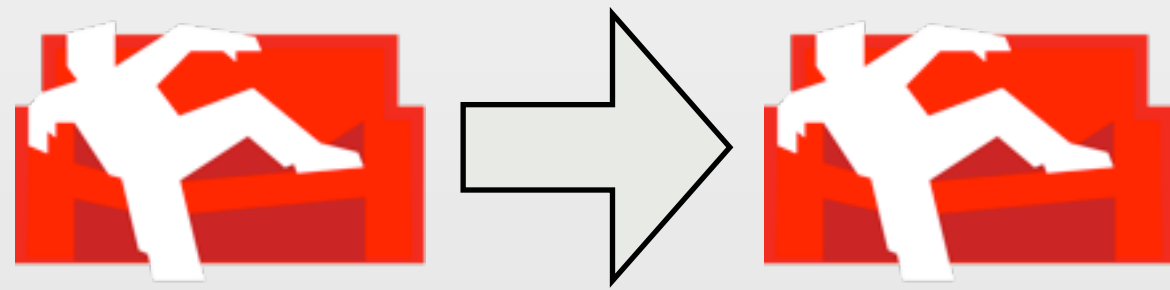
- Schema Free (JSON)
- Document Oriented,
Not Relational
- Highly Concurrent
- RESTful HTTP API
- JavaScript Powered
Map/Reduce
- **N-Master Replication**
- Robust Storage

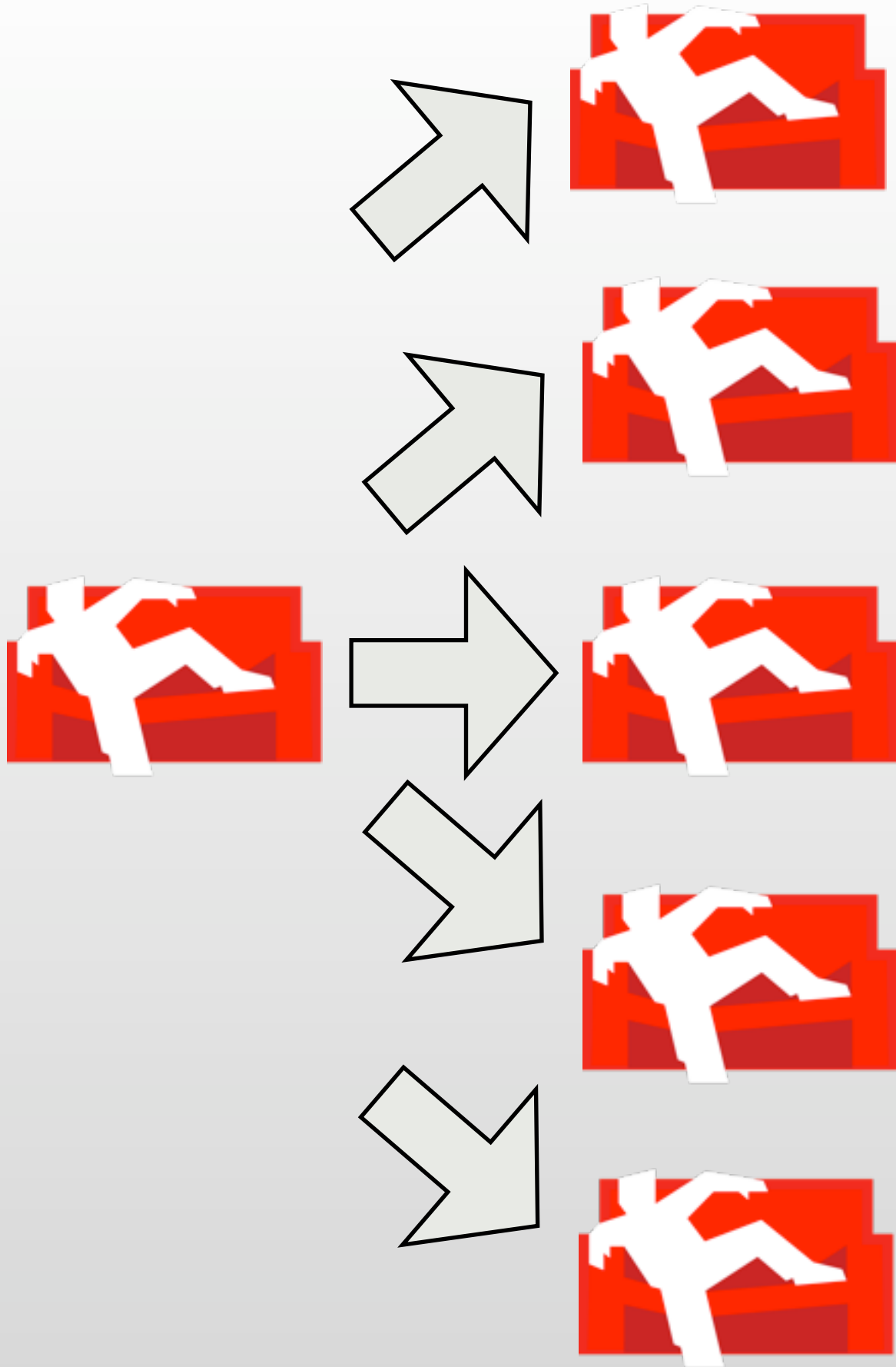
Features

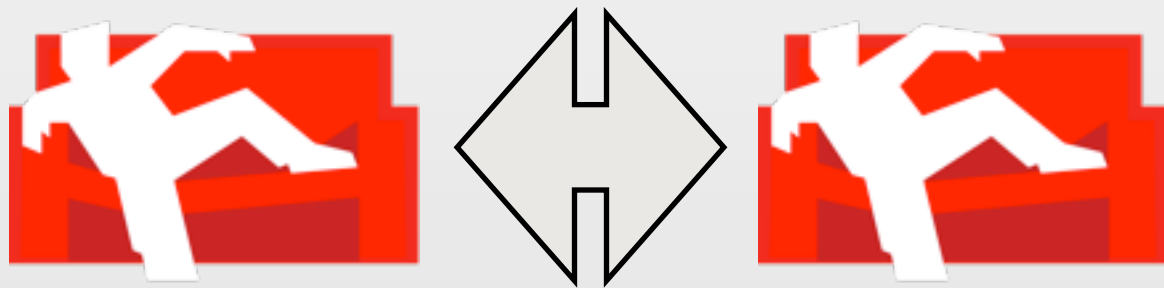


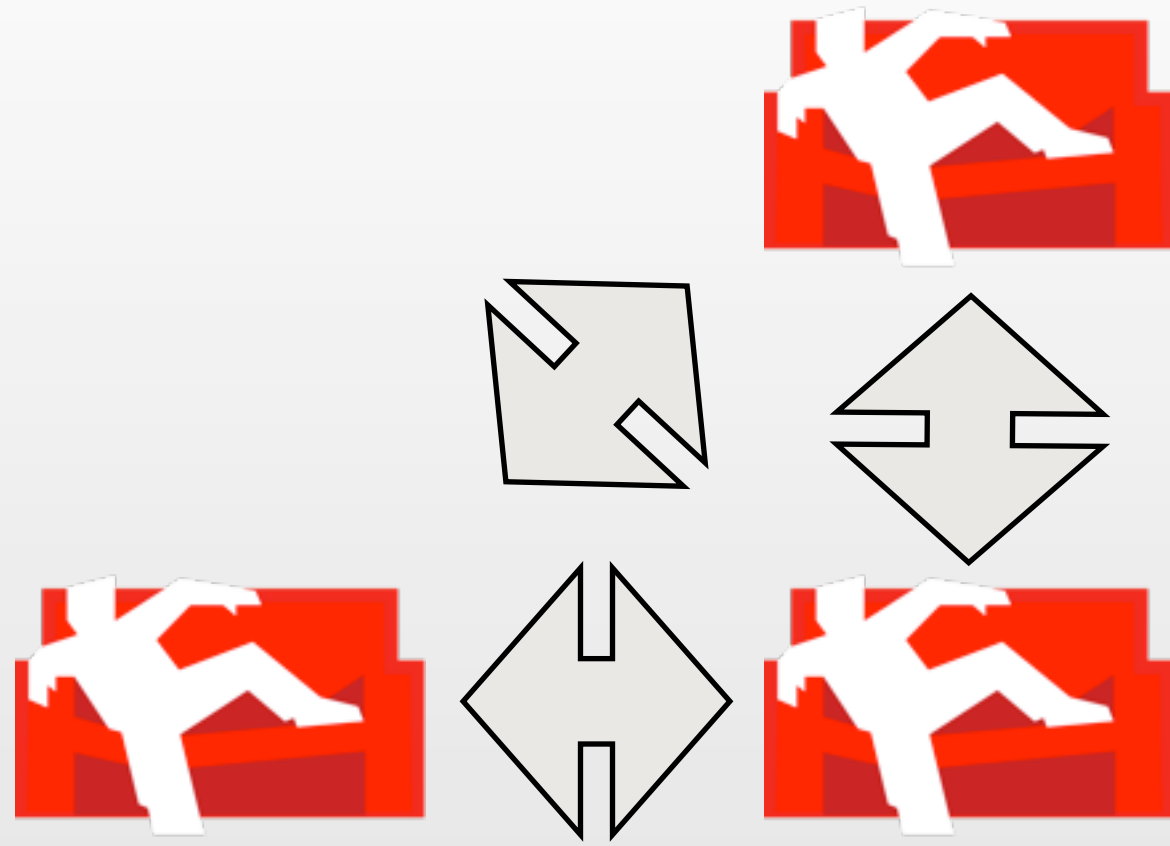


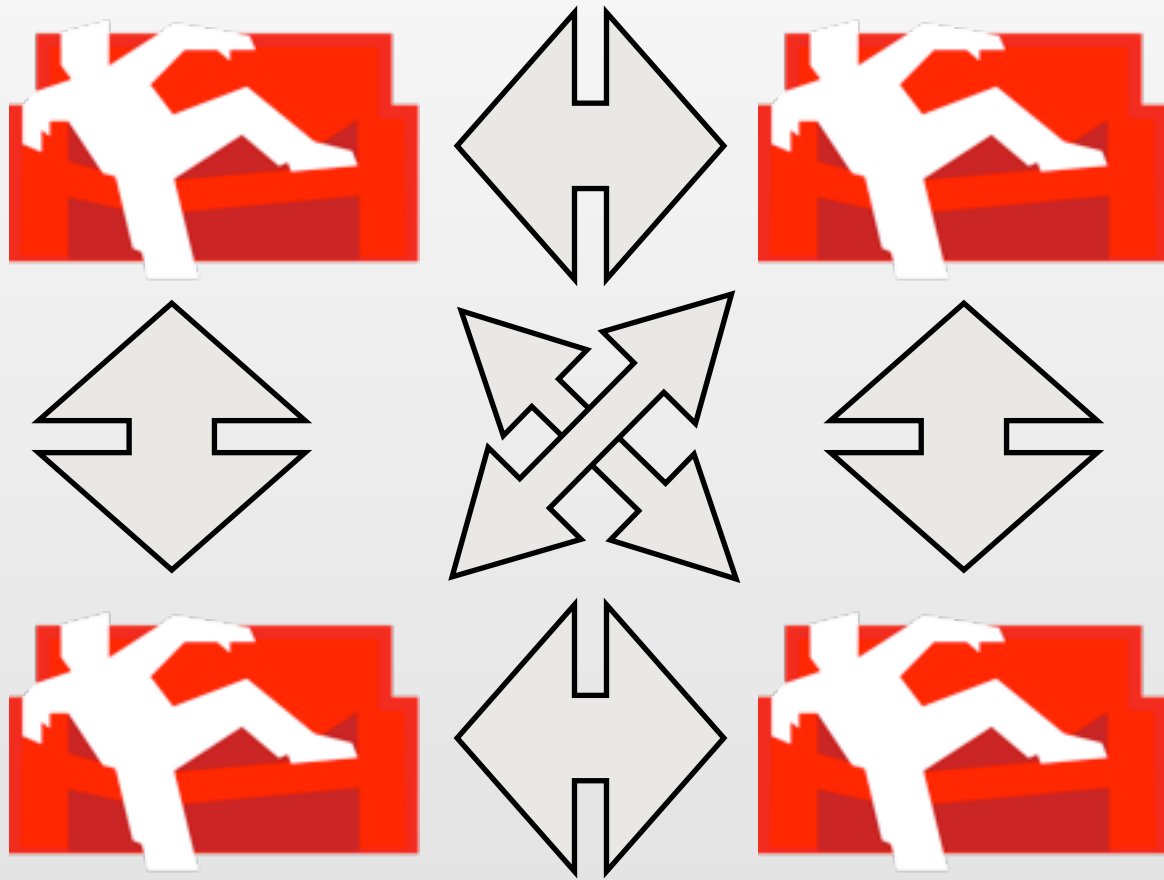


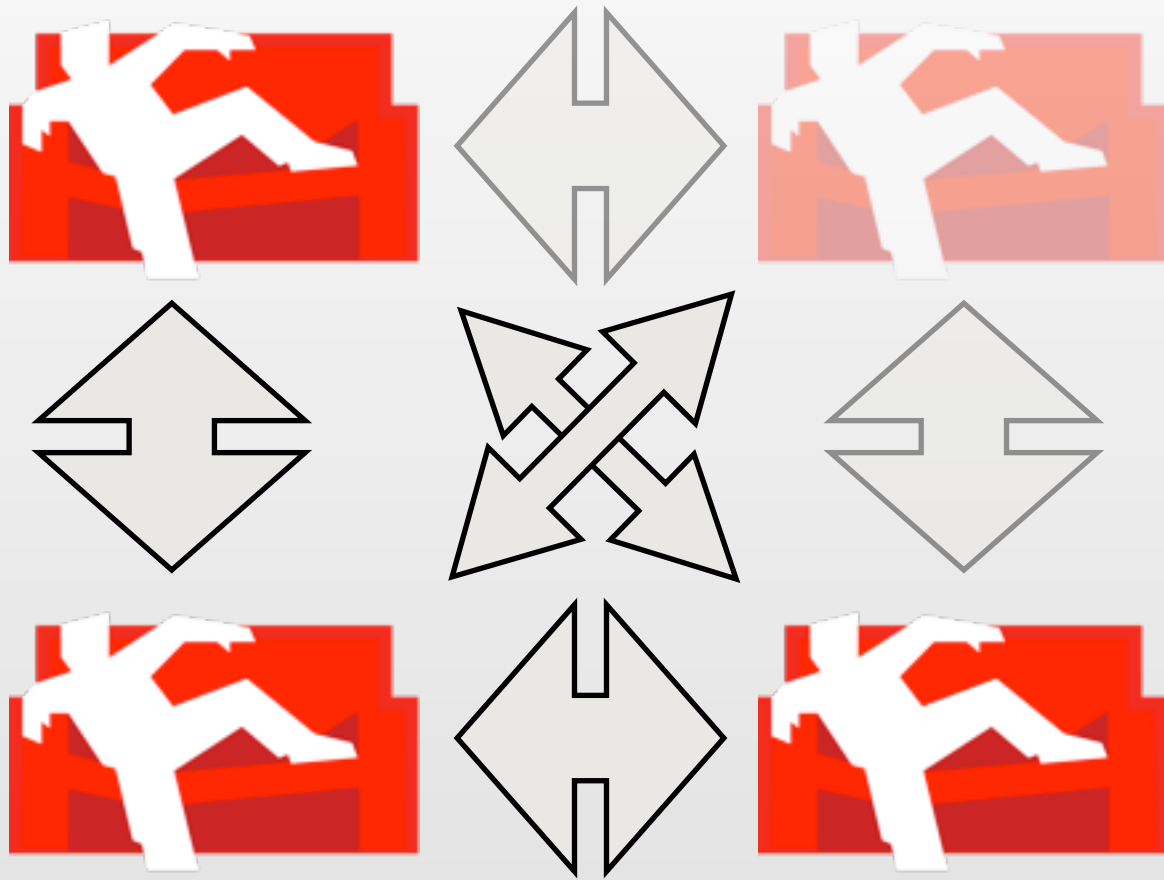


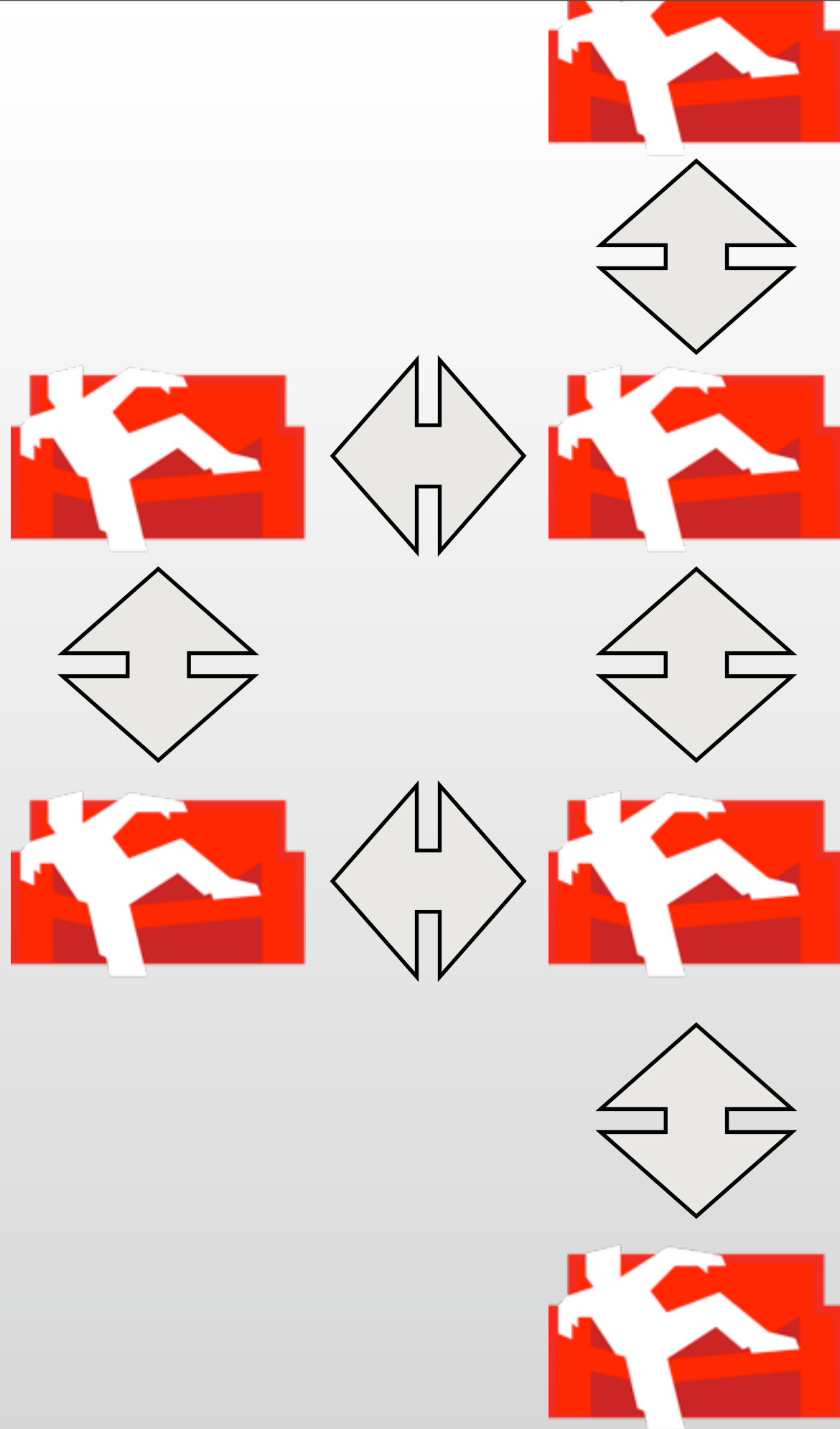


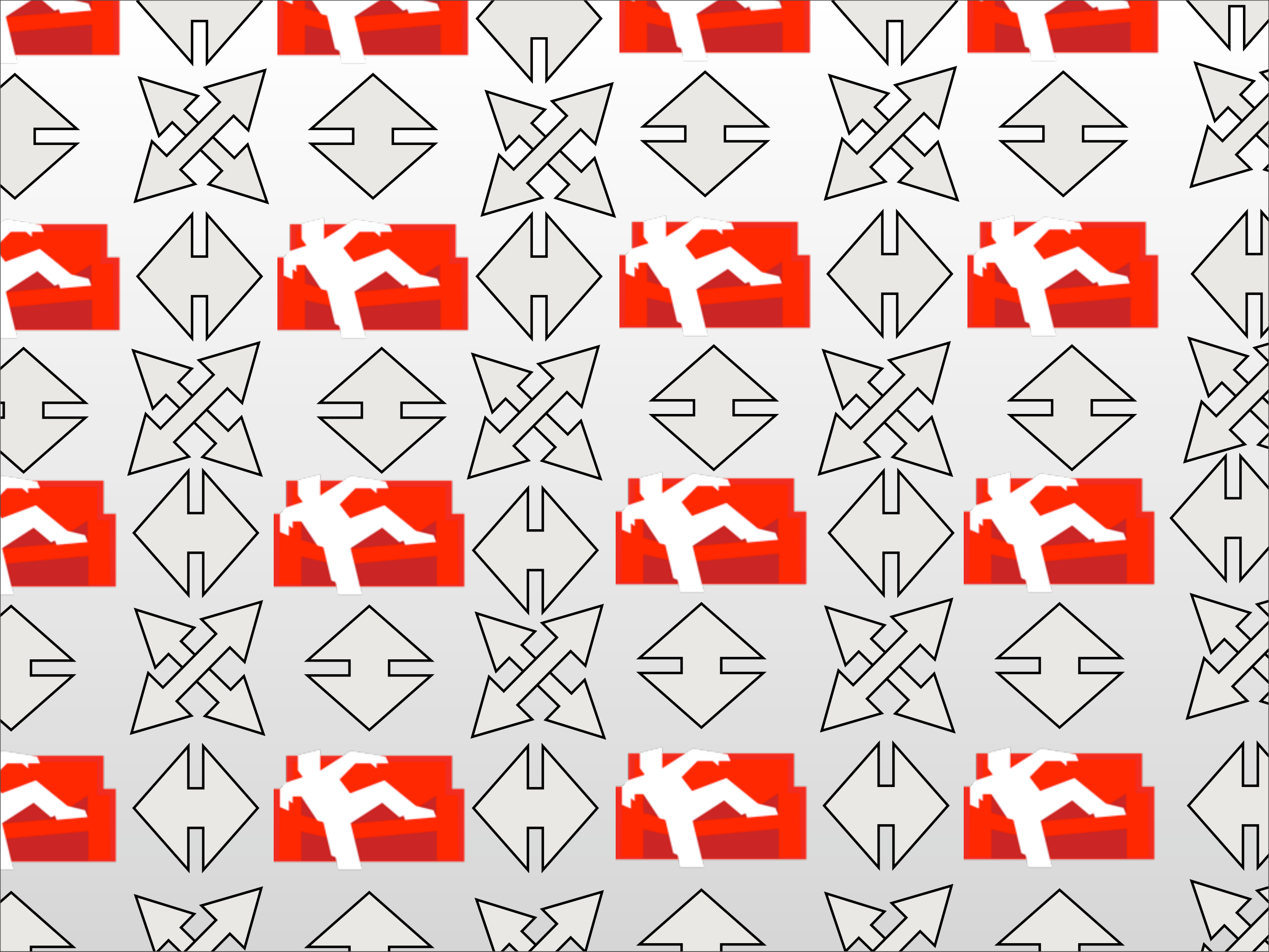










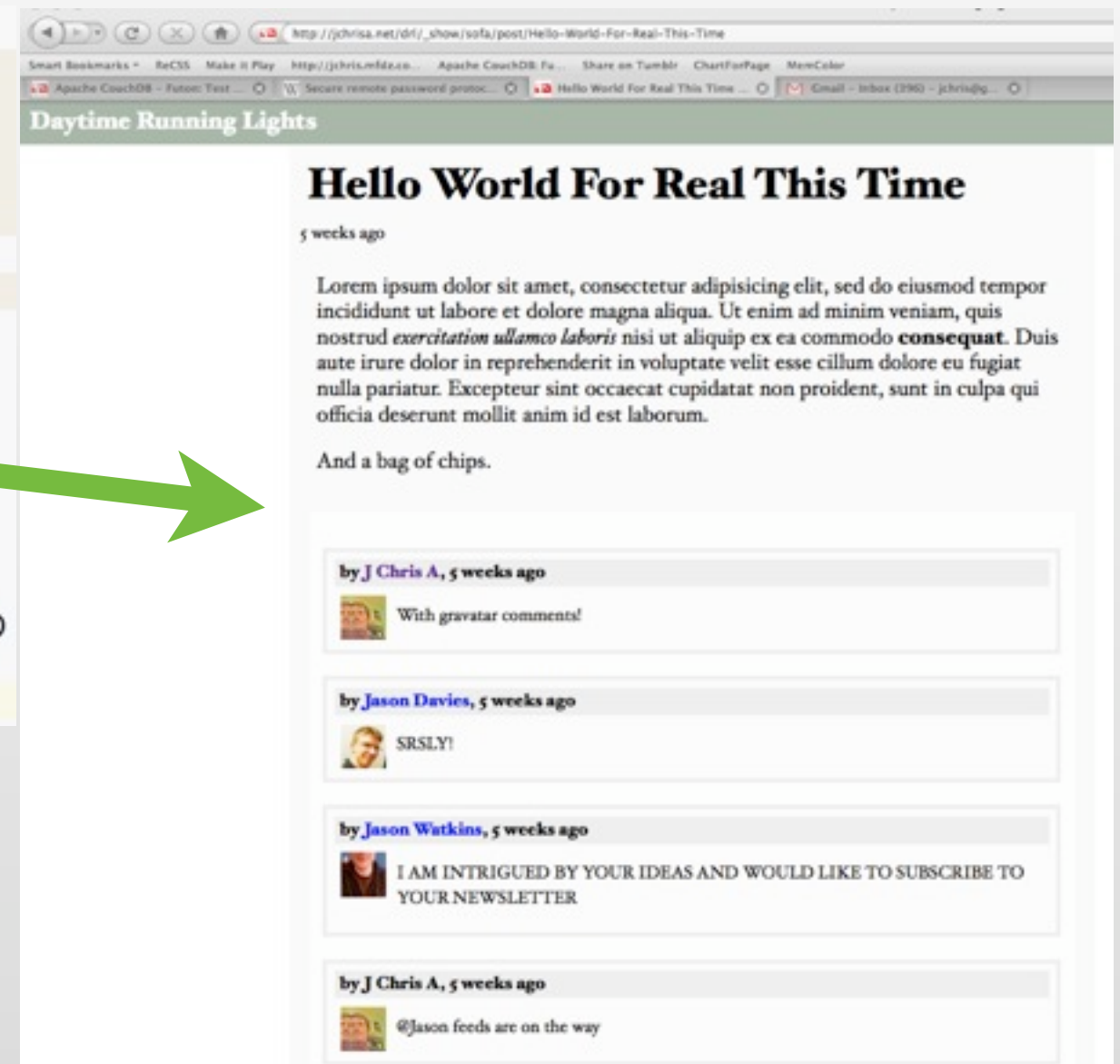


Render JSON Docs as HTML

shows/post.js

```
function(doc, req) {  
  // !json templates.post  
  // !json blog  
  // !code helpers.template  
  // !code helpers.couchapp  
  // log(req.headers.Accept);  
  
  // we only show html  
  return template(templates.post, {  
    title : doc.title,  
    blogName : blog.title,  
    post : doc.html,  
    date : doc.created_at,  
    author : doc.author,  
    assets : assetPath(),  
    editPostPath : showPath('edit', doc._id),  
    index : listPath('index', 'recent-posts', {descending:true, limit:8})  
  });  
}
```

/drl/_show/sofa/post/Hello-World-For-Real-This-Time



Render Views as HTML

lists/index.js

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

```
1 function(head, row, req) {
2   // ljson templates.index
3   // ljson 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.toXMLString().replace(/<\>/, ''));
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]</name></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, beta 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...

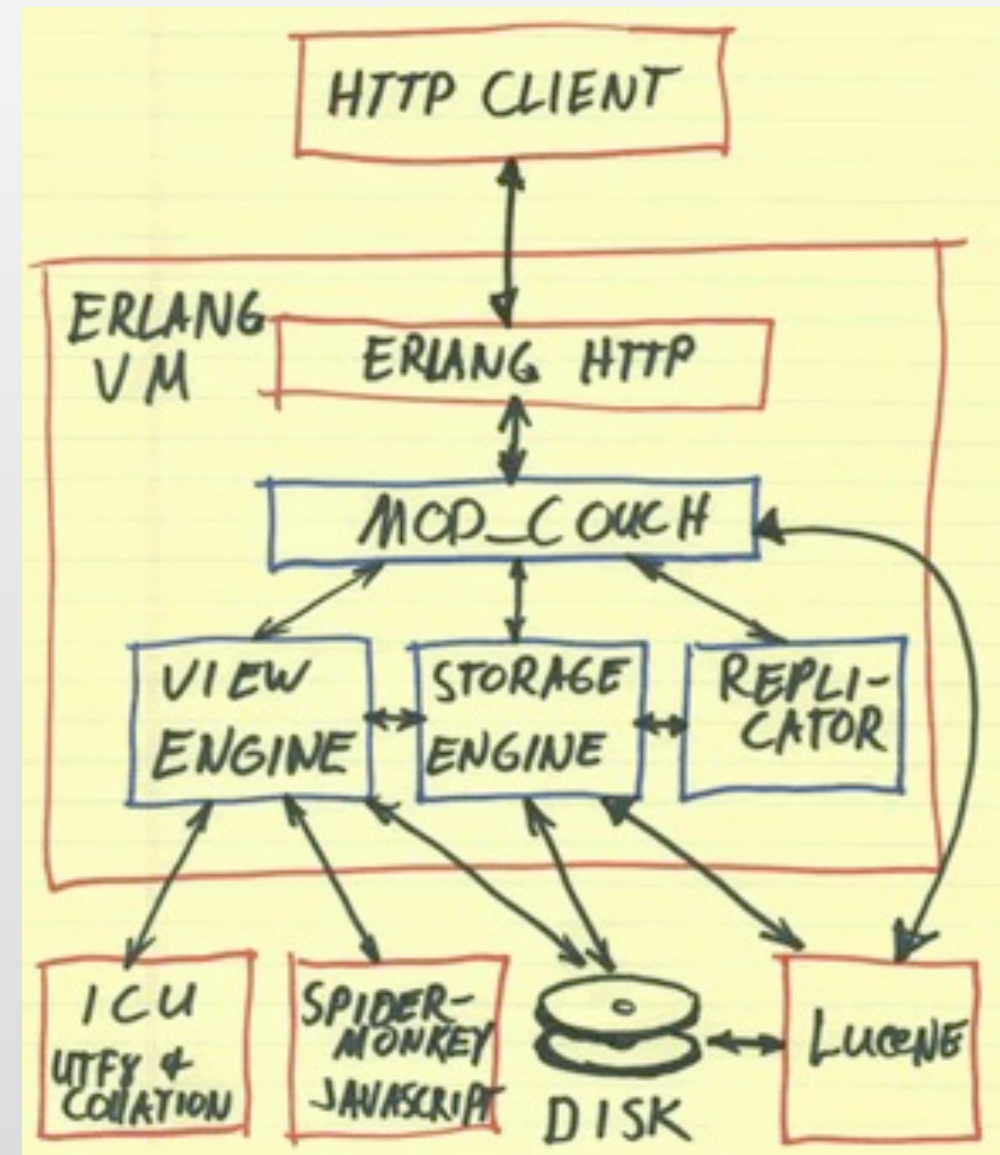
Deployed Sofa
16 days ago
For more information about this blog software, visit the Sofa source code on Github I plan to run my blog from this software, and I've got a few other projects in the pipeline. Also, don't forget about the Twitter client. Same great software, great new address....

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 ...

- 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

Resilient
Replication

Full fsync()



Resources

- Twitter: @CouchDB & <http://couchdb.org/>
- Dress like a Couch:
<http://shop.couchdb.com>
- <http://planet.couchdb.org/>
- <http://books.couchdb.org/relax/>
- <https://peepcode.com/products/couchdb-with-rails>

couch.io

Berlin – London – Portland

Thanks!

