



Couchbase



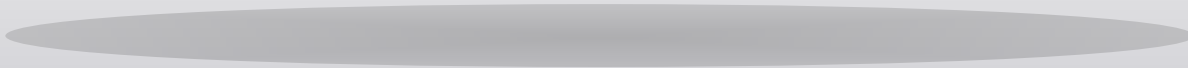
Couchbase & Javascript MapReduce, Node.js, Angular MapReduce, Node.js, Angular

Tugdual “Tug” Grall

Technical Evangelist



Couchbase



- Tugdual “Tug” Grall
 - Couchbase
 - Technical Evangelist
 - eXo
 - CTO
 - Oracle
 - Developer/Product Manager
 - Mainly Java/SOA
 - Developer in consulting firms

- Web



grall



<http://blog.grallandco.com>



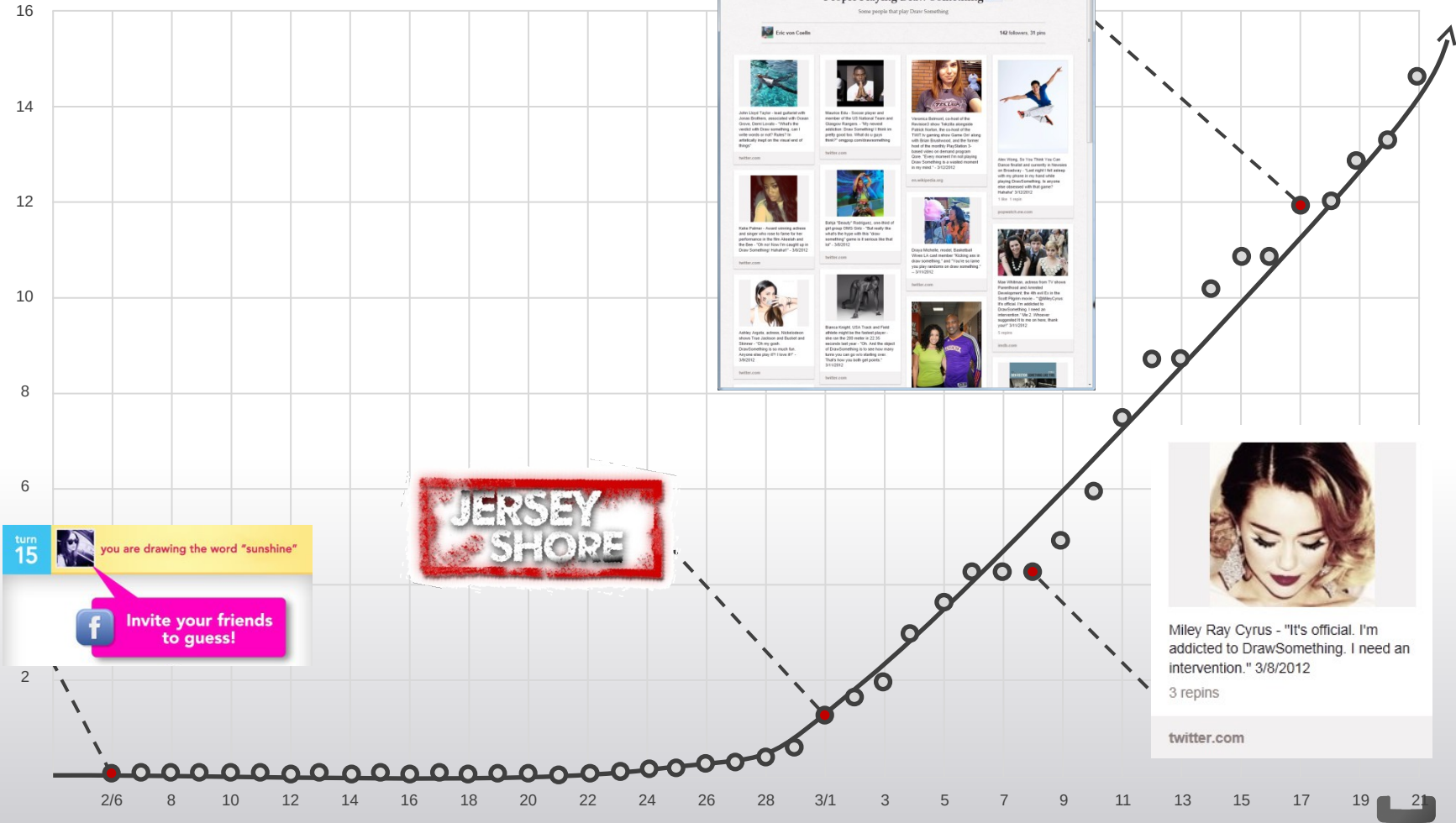
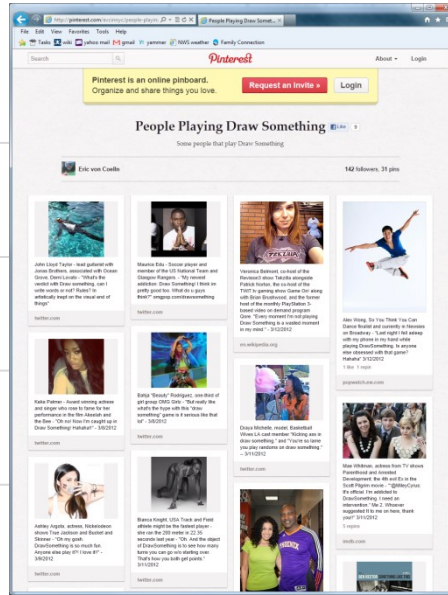
itesJUG co-founder

- Pet Project :
- <http://www.resultri.com>



Couchbase

Draw Something by OMGPOP Daily Active Users (millions)



turn 15

you are drawing the word "sunshine"

Invite your friends to guess!

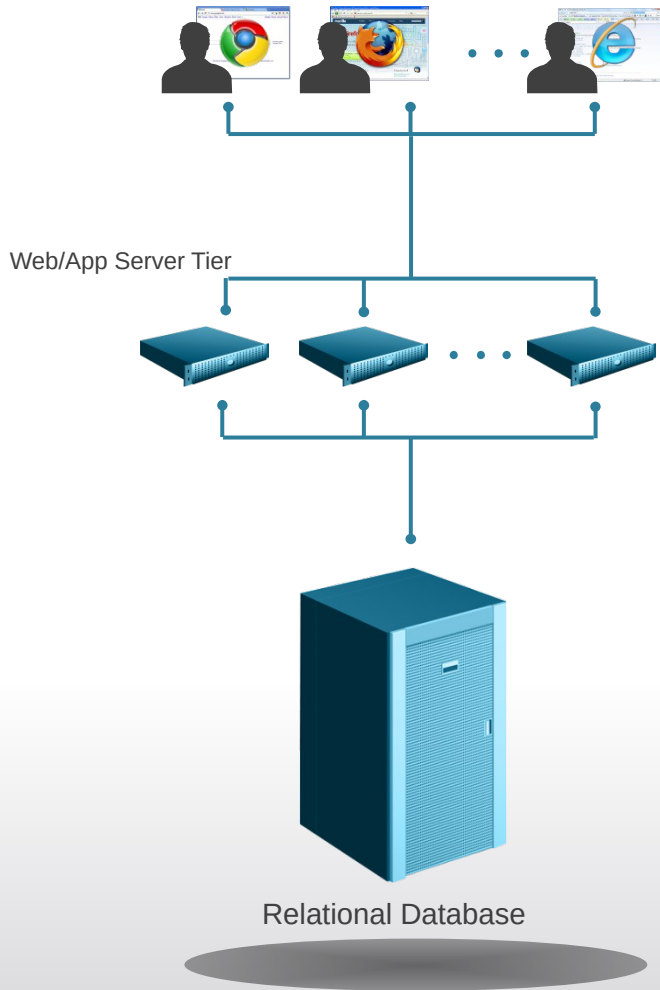


Miley Ray Cyrus - "It's official. I'm addicted to DrawSomething. I need an intervention." 3/8/2012

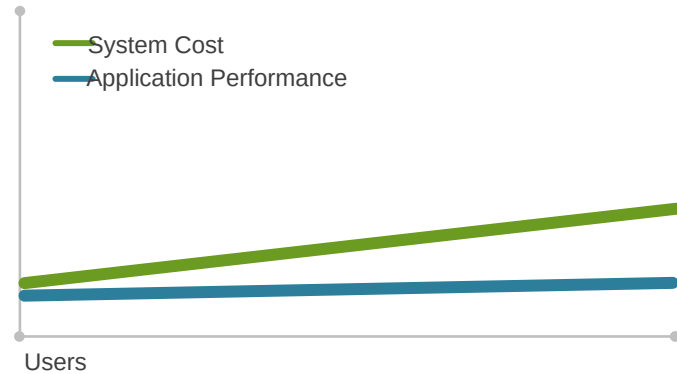
3 repins

twitter.com

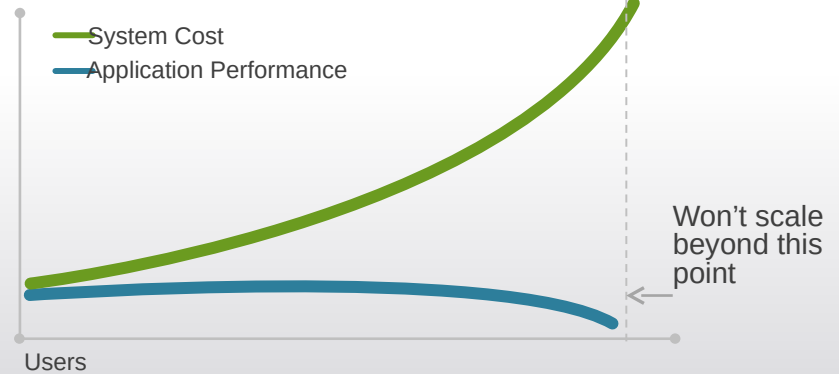
How do you take the growth?



Application Scales Out
Just add more commodity web servers

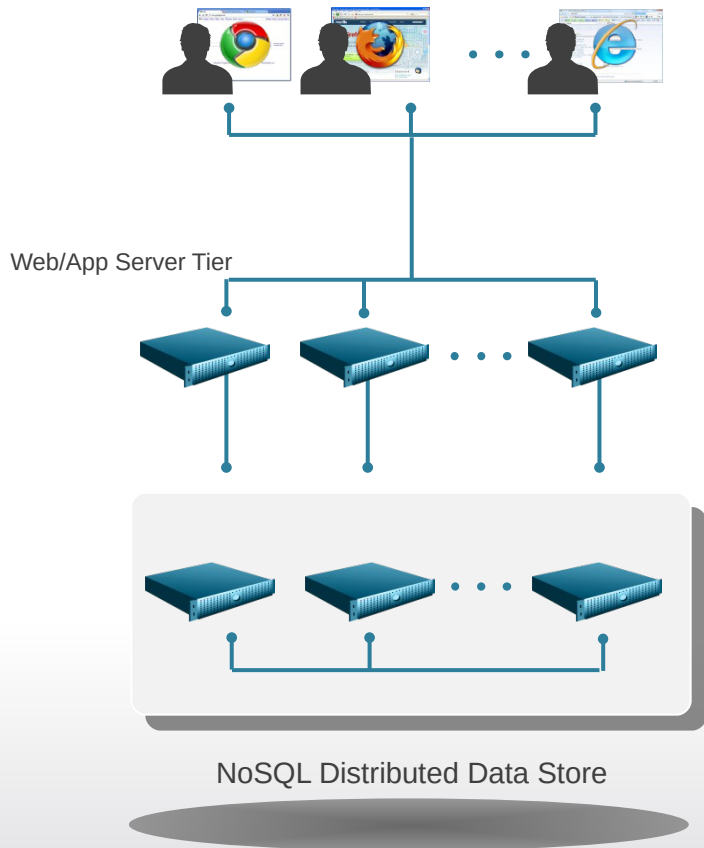


RDBMS Scales Up
Get a bigger, more complex server

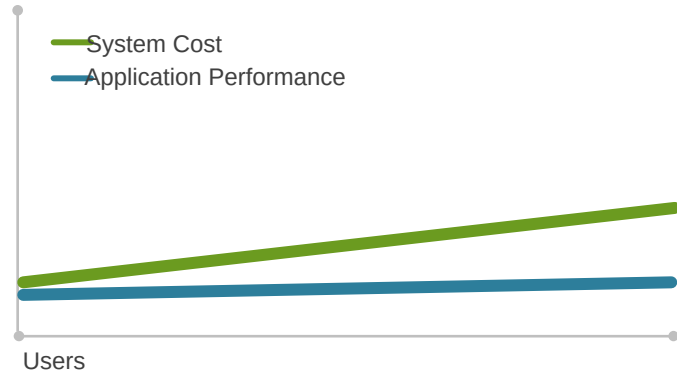


RDBMS is good for many thing, but hard to scale

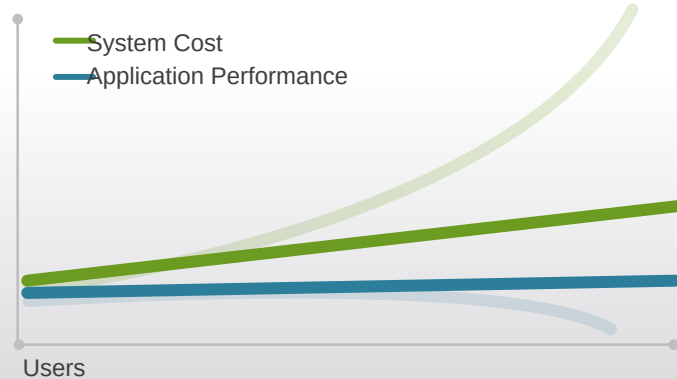
NoSQL Technology Scales Out



Application Scales Out
Just add more commodity web servers



NoSQL Database Scales Out
Cost and performance mirrors app tier



Scaling out flattens the cost and performance curves

A new technology?



Bigtable
November 2006



Dynamo
October 2007



Cassandra
August 2008

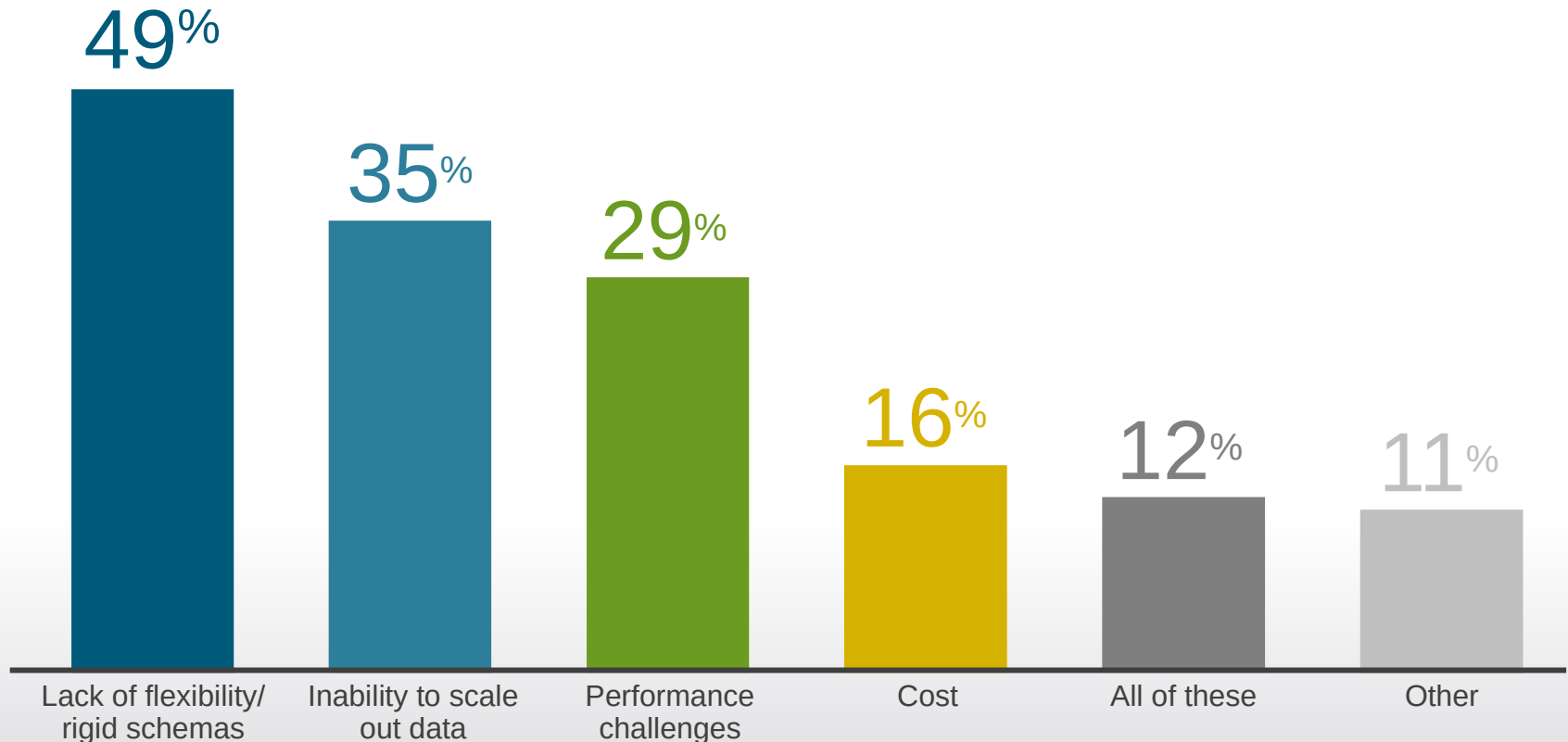


Voldemort
February 2009

- Building new database to answer the following requirements
 - No schema required before inserting data
 - No schema change required to change data format
 - Auto-sharding without application participation
 - Distributed queries
 - Integrated main memory caching
 - Data synchronization (multi-datacenter)

Very few organizations want to (fewer can) build and maintain database software technology.
But every organization building interactive web applications needs this technology.

What is driving NoSQL adoption?



Source: Couchbase Survey, December 2011, n = 1351.

Couchbase Open Source Project

- Leading NoSQL database project focused on distributed database technology and surrounding ecosystem
- Supports both key-value and document-oriented use cases
- All components are available under the Apache 2.0 Public License
- Obtained as packaged software in both enterprise and community editions.



COUCHBASE
OPEN SOURCE PROJECT

Couchbase Server Core Principles



Easy Scalability

Grow cluster without application changes, without downtime with a single click



Consistent High Performance

Consistent sub-millisecond read and write response times with consistent high throughput



Always On 24x365

No downtime for software upgrades, hardware maintenance, etc.

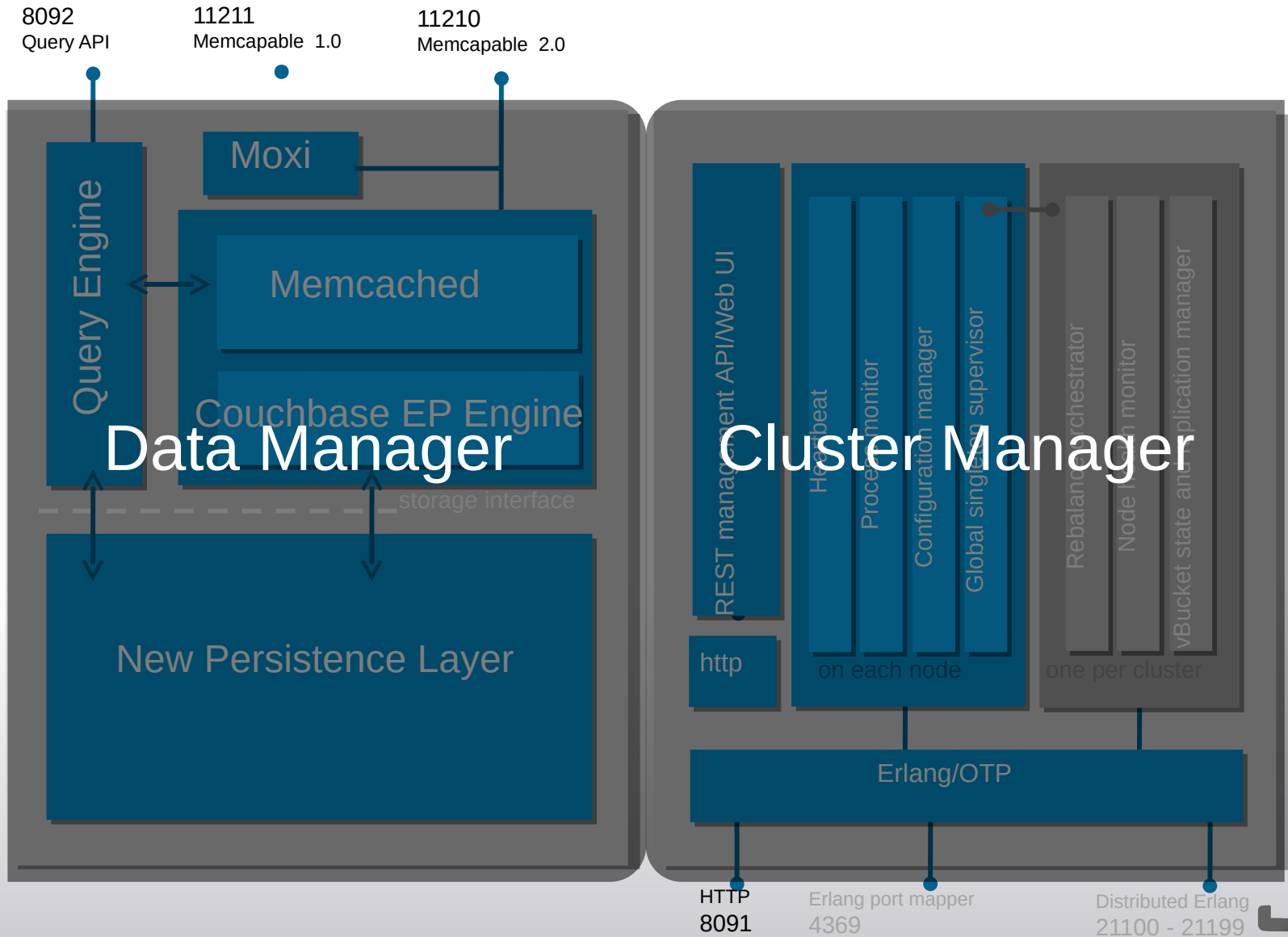


Flexible Data Model

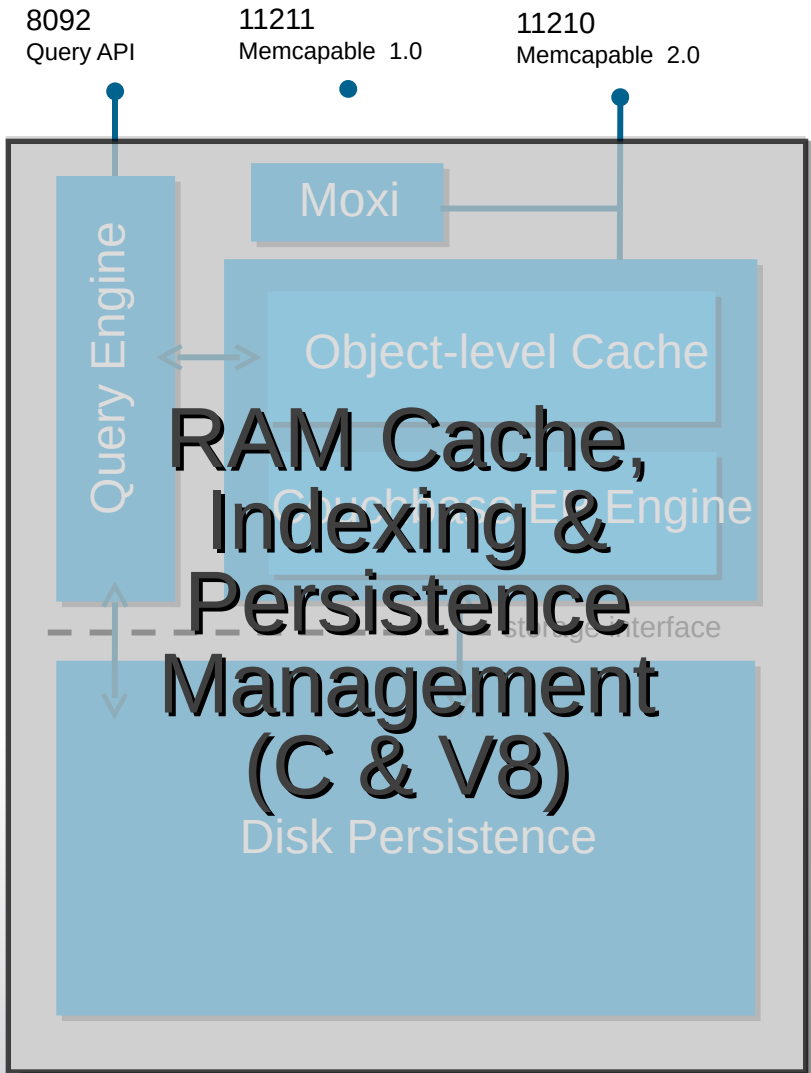
JSON document model with no fixed schema.



Couchbase Server 2.0 Architecture



Couchbase Server 2.0 Architecture



[The Unreasonable Effectiveness of C by Damien Katz](#)

Open Source Project

Apache 2.0

The screenshot shows the GitHub repository page for Couchbase. The Couchbase logo is prominently displayed. Below it, the repository 'testrunner' is highlighted, showing its description and commit history. Other repositories listed include 'couchbase', 'couchstore', 'php-ext-couchbase', 'manifest', and 'couchbase-ruby-client'. The page includes navigation links for Repositories and Members, and a search bar for finding repositories.

The screenshot shows the Gerrit code review interface for Couchbase. It displays a list of recent code reviews with columns for ID, Subject, Owner, Project, Branch, Updated, and V. The reviews are sorted by updated time, with the most recent at the top. The interface includes a search bar and navigation links for Documentation, Register, and Sign In.

ID	Subject	Owner	Project	Branch	Updated	V	R
1dcb31502	PR#83: Add persist to and replicate to	Trond Norbye	php-ext-couchbase	master	12:17 PM	1	1
1e8ec8684	Beef up options code	Trond Norbye	php-ext-couchbase	master	8:29 AM		✓
1e3b15638	Remove debugger development dependency	Sergey Avseyev	couchbase-ruby-client	master	12:05 AM		
1481769a7	CR#6-848 Print out server response after do_maint	Chang Hong	testrunner	master	Dec 1		✓
10817211e	Add manpages for all of cbc-* commands	Trond Norbye	libcouchbase	master	Dec 1		
1e3b7786e	Added support for libevent in win32	Brett Harrison	libcouchbase	master	Dec 1		
1c453a709	MB-7283 - docs for upgrade when hostnames configured	Steve Yen	voltron	couchbase	Dec 1		
16857c88a	MB-7271 - docs on restoring via mxi	Steve Yen	membase-cli	master	Dec 1		
117bd8c00	Sort file listing	Trond Norbye	php-ext-couchbase	master	Dec 1		✓
1677e0f6b	refactor: move version to separate file	Trond Norbye	php-ext-couchbase	master	Dec 1		✓
10e09f6ea	move stats to a separate file	Trond Norbye	php-ext-couchbase	master	Dec 1		✓
12c550897	Simplify flush logic	Trond Norbye	php-ext-couchbase	master	Dec 1		✓
1d663eb36	wp: using wait_backfill_complete	Aleksey Kandratsenka	ns_server	master	Nov 30		
1810b948c	[bucketmigrator] added ability to wait for end of backfill	Aleksey Kandratsenka	ns_server	master	Nov 30		
13182b67d	MB-6726: added internal settings for vbucket scheduling variables	Aleksey Kandratsenka	ns_server	master	Nov 30		
152b5c055	MB-100: store just binaries in master.activity.events.keeper	Aleksey Kandratsenka	ns_server	master	Nov 30		✓
14c768ea3b	MB-100: added scripts/reformat-master-events.sh	Aleksey Kandratsenka	ns_server	master	Nov 30		
15410d11d	MB-6726: added script to draw timeline of new-style rebalance	Aleksey Kandratsenka	ns_server	master	Nov 30		
1d033a864	MB-6726: added vbucket scheduling simulation/visualization tool	Aleksey Kandratsenka	ns_server	master	Nov 30		
1ac069b3b	MB-6726: added master activity events for in-rebalance compaction	Aleksey Kandratsenka	ns_server	master	Nov 30		
1b4433876	MB-6726: added rebalance ignore view compactions setting	Aleksey Kandratsenka	ns_server	master	Nov 30		
1d46aa7d9	MB-6726: avoid doing obviously duplicate set view config changes	Aleksey Kandratsenka	ns_server	master	Nov 30		
1a46da1be	MB-6726: 99% implemented index aware vbucket move scheduling	Aleksey Kandratsenka	ns_server	master	Nov 30		
1a5134529	MB-100: added master events support for new style rebalance	Aleksey Kandratsenka	ns_server	master	Nov 30		
19cc836fa	MB-7285 Fix badmatch in index updater on cleanup	Elipe David Borba Marano	couchdb	2.0.0	Nov 30		✓

<https://github.com/couchbase/>
<https://github.com/couchbaselabs/>

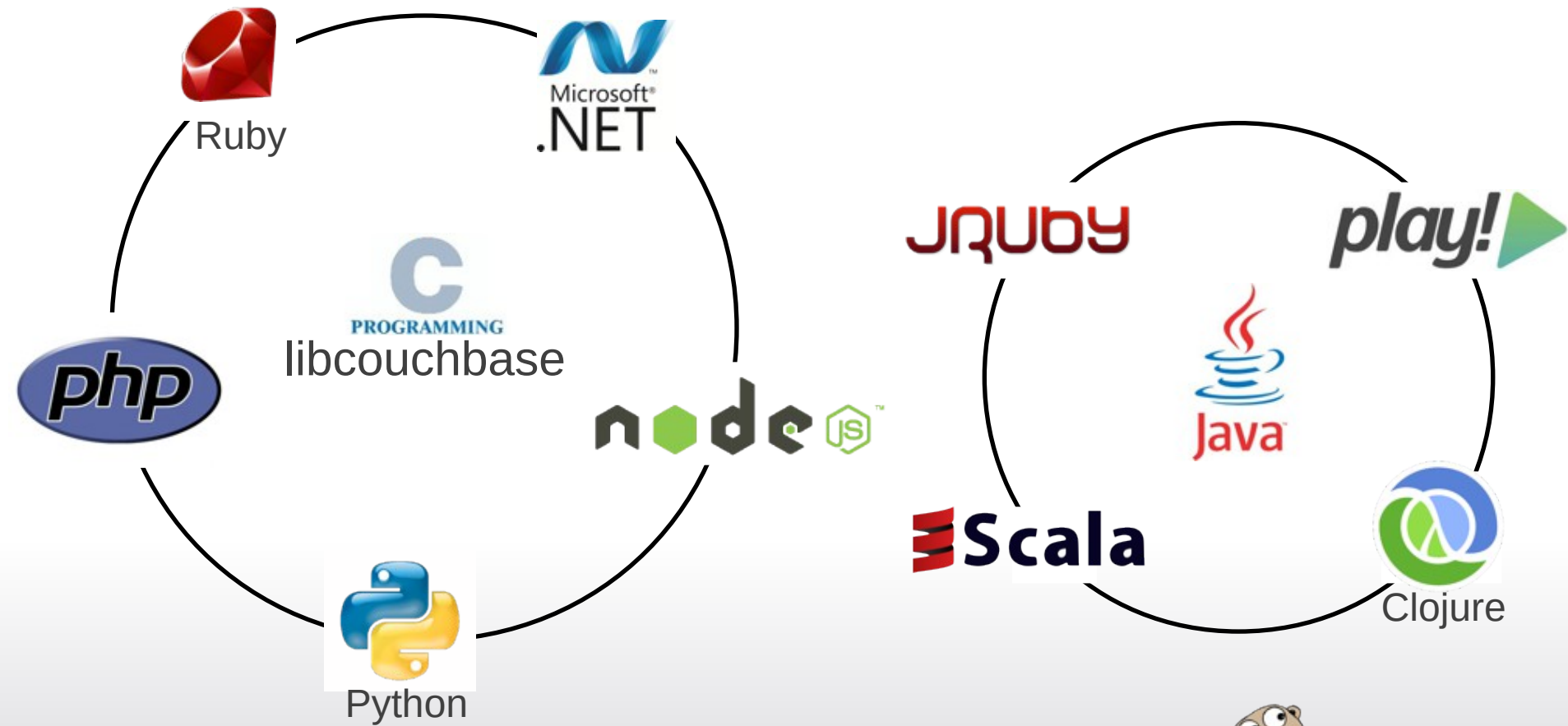
Gerrit: <http://review.couchbase.org/>



WHAT ABOUT MY APP?



Couchbase SDK



www.couchbase.com/develop

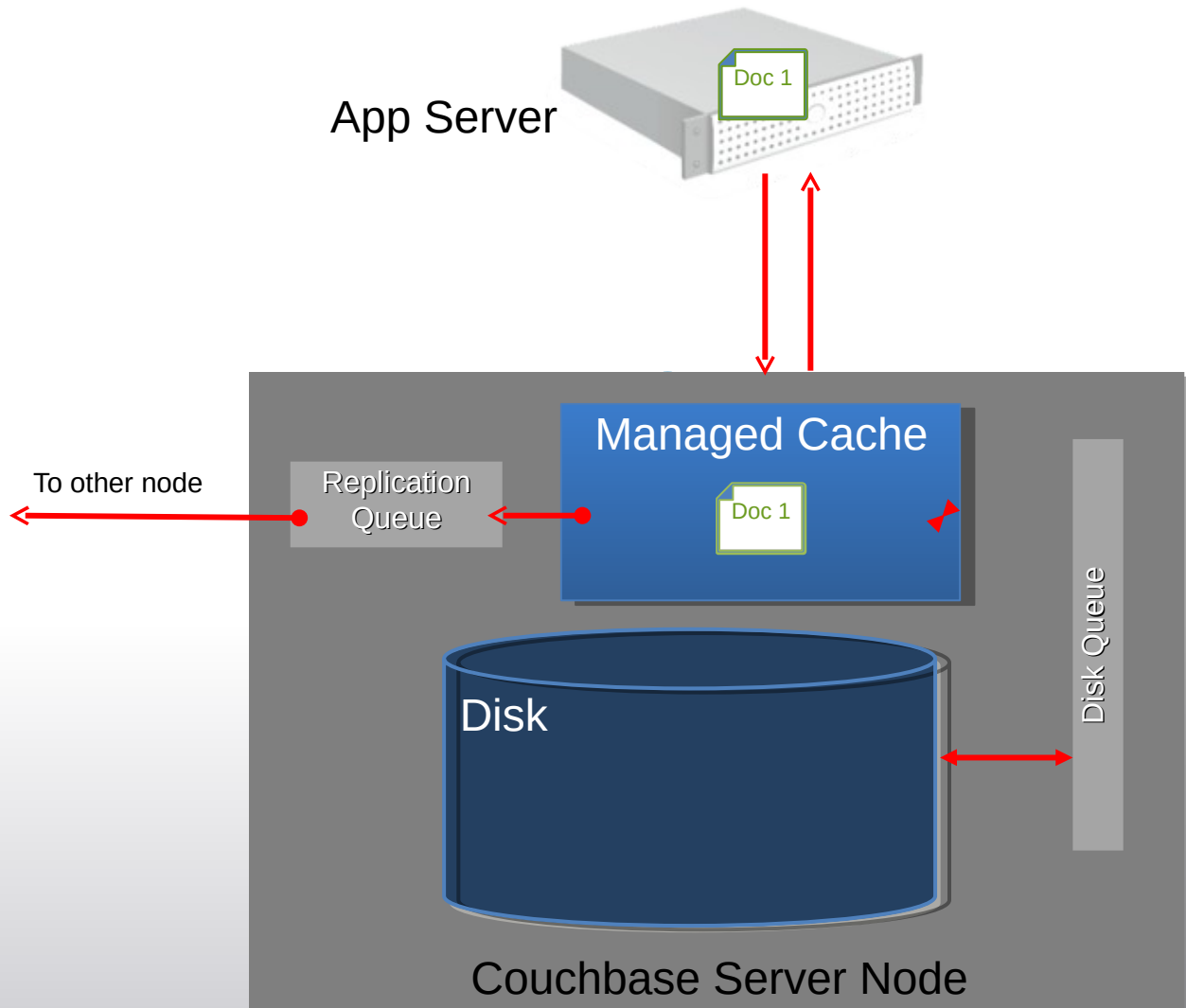


Go



Couchbase

Write Operation



Basic Operations



- Docs distributed evenly across servers
- Each server stores both active and replica docs
Only one doc active at a time
- Client library provides app with simple interface to database
- Cluster map provides map to which server doc is on
App never needs to know
- App reads, writes, updates docs
- Multiple app servers can access same document at same time

Store & Retrieve Operations

- **get (key)**
 - Retrieve a document
- **set (key, value)**
 - Store a document, overwrites if exists
- **add (key, value)**
 - Store a document, error/exception if exists
- **replace (key, value)**
 - Store a document, error/exception if doesn't exist
- **cas (key, value, cas)**
 - Compare and swap, mutate document only if it hasn't changed while executing this operation

Atomic Counter Operations

These operations are always executed in order atomically.

- **set (key, value)**

- Use set to initialize the counter
 - `cb.set("my_counter", 1)`

- **incr (key)**

- Increase an atomic counter value, default by 1
 - `cb.incr("my_counter")` # now it's 2

- **decr (key)**

- Decrease an atomic counter value, default by 1
 - `cb.decr("my_counter")` # now it's 1



Mental Adjustments

- In SQL we tend to want to avoid hitting the database as much as possible
 - Even with caching and indexing tricks, and massive improvements over the years, SQL still gets bogged down by complex joins and huge indexes, so we avoid making database calls
- In Couchbase, get's and set's are so fast they are trivial, not bottlenecks, this is hard for many people to accept at first; Multiple get statements are commonplace, don't avoid it!



Operations with Node



NPM

```
{  
  "name": "my-node-application",  
  "version": "1.0.0",  
  "private": true,  
  "dependencies":  
  {  
    "express": "3.x",  
    "couchbase": "0.0.11",  
    "ejs": ">= 0.0.1"  
  }  
}
```

Connect to the cluster

```
var driver = require('couchbase');

dbConfiguration = {
  "hosts": ["localhost:8091"],
  "bucket": "ideas"
};

driver.connect(dbConfiguration, function(err, cb) {
  if (err) {
    throw (err)
  }

  // your application code here
})
```



Insert Data

```
var meetup = {"type" : "meetup", "language" : "javascript"};
cb.set("barcelonajs", meetup, function(err, meta) {});

var tmp = {"message" : "hello world!"};
cb.set("tmp", tmp, {"expiry" : 5}, function(err, meta) {});
```

Insert / Delete Data

```
var meetup = {"type" : "meetup", "language" : "javascript"};  
cb.set("barcelonajs", meetup, function(err, meta) {});
```

```
var tmp = {"message" : "hello world!"};  
cb.set("tmp", tmp, {"expiry" : 5}, function(err, meta) {});
```

```
cb.set("todelete", tmp, function(err, meta) {});  
cb.remove("todelete", function(err, meta) {});
```

Retrieve the Data

```
cb.get("product:45", function(errs, doc, metas) {
  console.log("=== get the document ===");
  console.log( doc );  });      var keys = new Array();
keys.push("product:1");      keys.push("product:45");
keys.push("product:65");      keys.push("product:80");
cb.get(keys, null, function(errs, docs, metas) {
  console.log("\n=== get List of documents ===");
  console.log( docs );  });
```

Retrieve the Data

What if I want all products or meetups?

```
key : barcelonajs  
{ "type": "meetup",  
  "language": "javascript"}
```

```
key : product:10  
{ "type": "product", "name":  
  "Product with id 10"}
```

Calling a view from your app

```
var queryParams = {
  stale: false,
  key : "meetup"
};

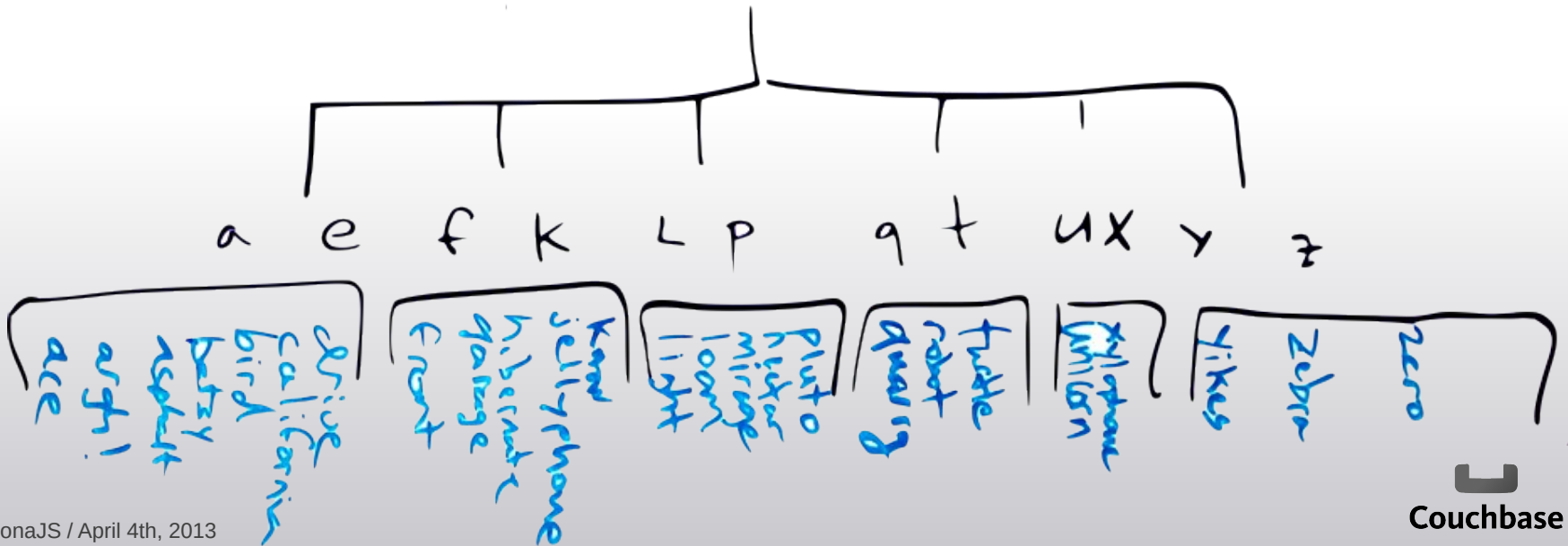
cb.view("my_views", "by_type", queryParams, function(err, view) {
  var keys = new Array();
  for (var i = 0; i < view.length; i++) {
    keys.push(view[i].id);
  }
  cb.get(keys, null, function(errs, docs, metas) {
    console.log(docs);
  });
});
```



Indexing and Querying

- Define materialized views on JSON documents and then query across the data set
- Using views you can define
 - Primary indexes
 - Simple secondary indexes (most common use case)
 - Complex secondary, tertiary and composite indexes
 - Aggregations (reduction)
- Indexes are eventually indexed
- Queries are eventually consistent with respect to documents
- Built using Map/Reduce technology
 - Map and Reduce functions are written in Javascript

Map Function





Q&A

