

CouchBase

Perry Krug, Sr. Solutions Architect, Couchbase Inc.



Couchbase – Company overview

- Elastic Data Management Software
 - For web applications and cloud computing environments
 - Most mature, reliable and widely deployed NoSQL database solution
 - Fully featured, open source document datastore
 - Headquartered in Silicon Valley
 - Couchbase team has authored majority of memcached and CouchDB codebase



Product family spans device to data center

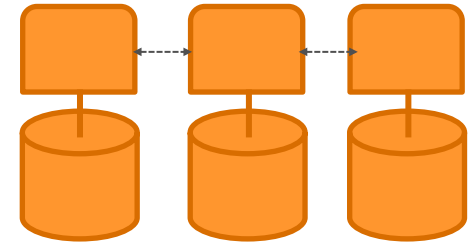


Couchbase Mobile
(iPhone and Android)



Couchbase Single Server
(Single Node)

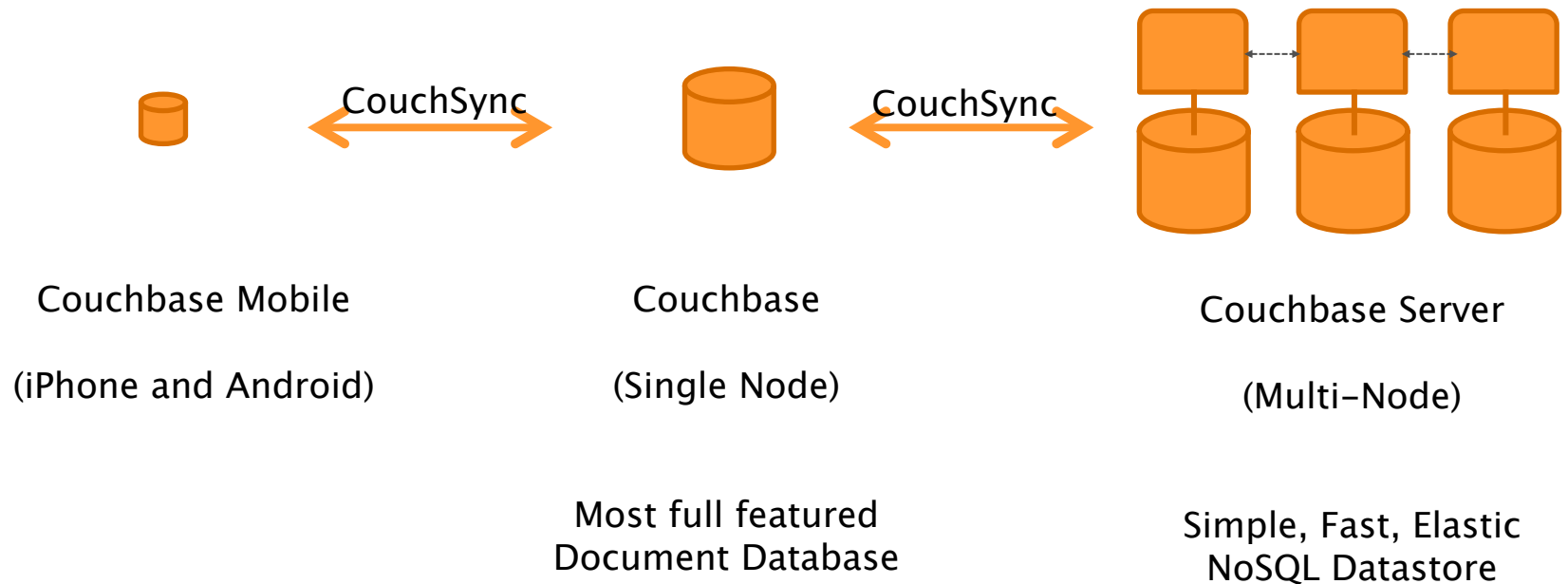
Most full featured
Document Database



Couchbase Server
(Multi-Node)

Simple, Fast, Elastic
NoSQL Datastore

Automated synchronization via CouchSync



Paid Production Deployment Examples



WHY NOSQL?

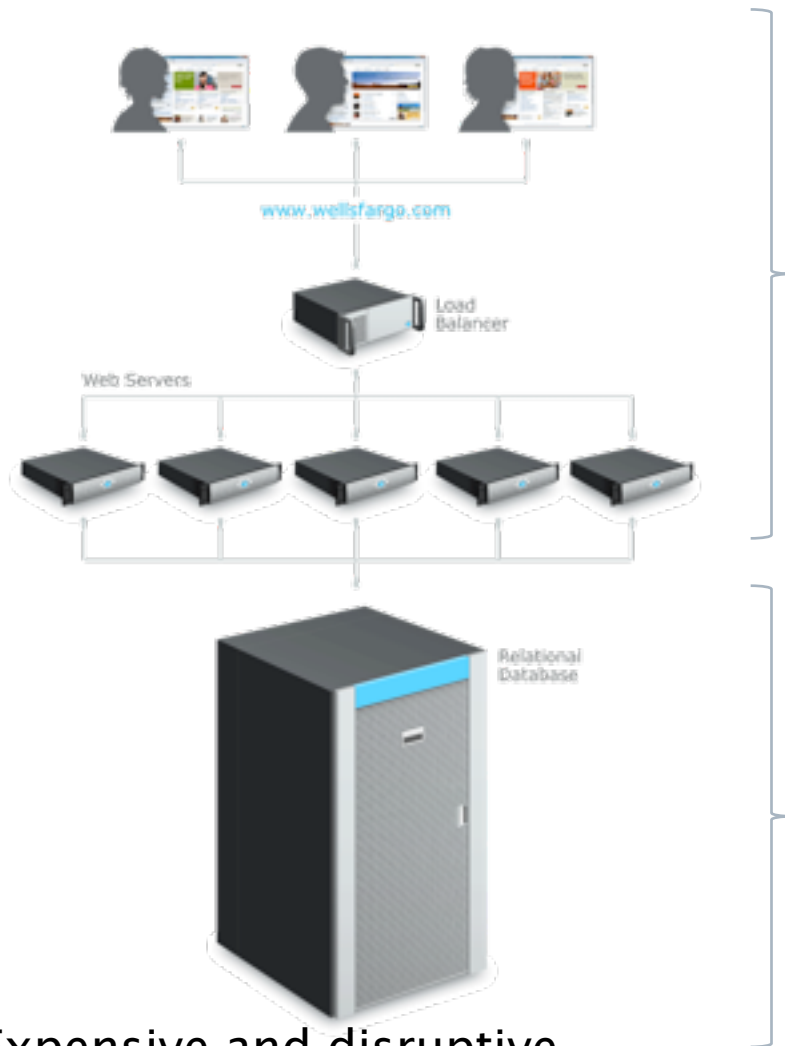
“ Relational database technology has served us well for 40 years, and will likely continue to do so for the foreseeable future to support transactions requiring ACID guarantees. But a large, and increasingly dominant, class of software systems and data do not need those guarantees. Much of the data manipulated by Web applications have less strict transactional requirements but, for lack of a practical alternative, many IT teams continue to use relational technology, needlessly tolerating its cost and scalability limitations. For these applications and data, distributed document cache and database technologies such as Couchbase’s provide a promising alternative.

Carl Olofson

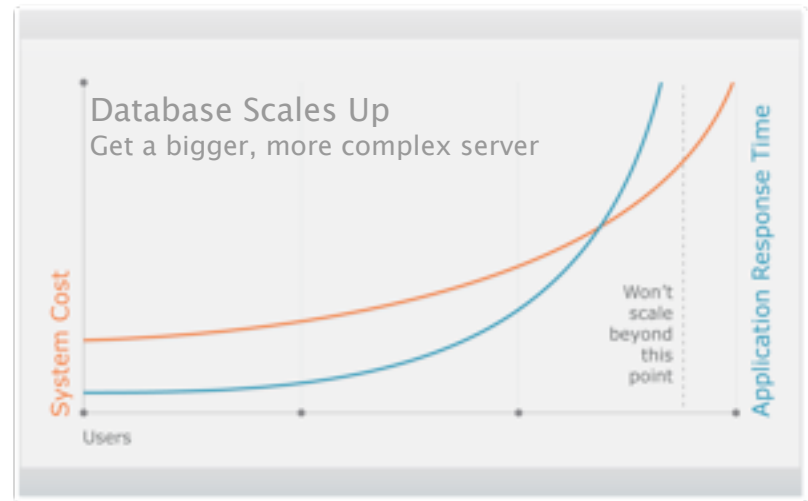
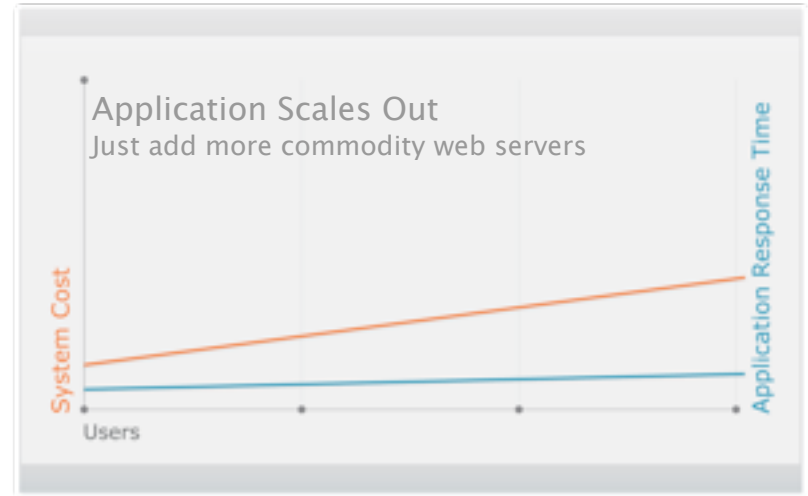
IDC Research Vice President, Information and Data Management



Modern interactive software architecture



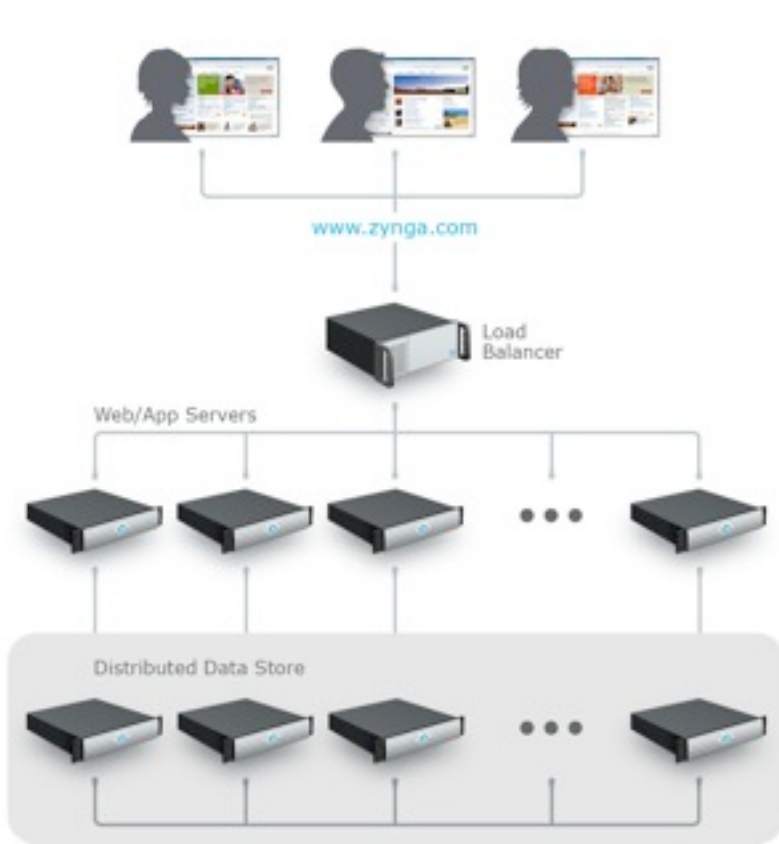
- Expensive and disruptive sharding
- Doesn't perform at Web Scale



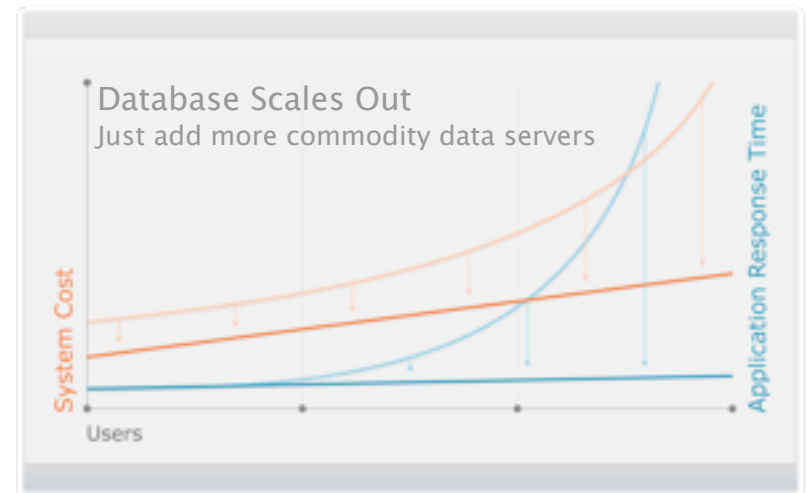
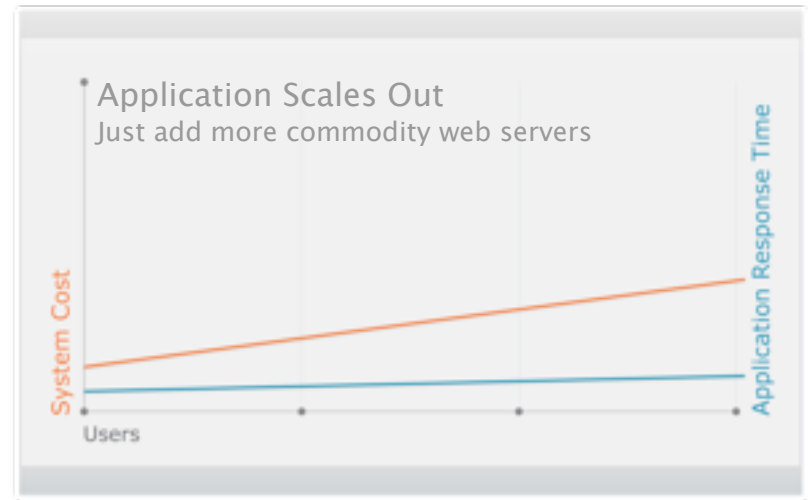
Things Have Changed

- Number of users of apps growing rapidly
 - Was thousands, now often millions or more
- Amount of data stored in apps growing rapidly
 - Was GBs, now often 1000s of GBs or more
- Types of data stored in apps is different
 - Was structured, now often unstructured & user-generated
 - Many apps including social features to increase engagement
- Data you want to store is changing rapidly
 - Fixed data model was okay, now its not flexible enough
- High-speed networking is inexpensive

Data Layer Matches Application Logic Tier Architecture

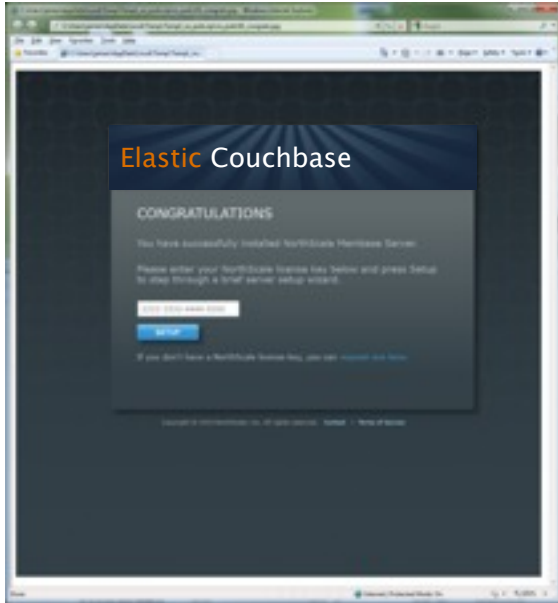


- Horizontally scalable with auto-sharding
- High performance at web scale
- Schema-less for flexibility



► Scaling out flattens the cost and performance curves.

Couchbase is Simple, Fast, Elastic NoSQL



- Simple to:
 - Deploy (Membase ServerTemplate)
 - Develop (memcached)
 - Manage (UI and RESTful API)
- Fast:
 - Predictable low latency
 - Sub-ms response times
 - Built-in memcached technology
- Zero-downtime Elasticity:
 - Spread I/O and data across instances
 - Consistent performance with linear cost

Couchbase Community Worldwide

- Vibrant developer community
- Full documentation and resources available
- Worldwide training events and meet up groups
- CouchConf – User Conference in San Francisco and other major cities



CouchConf World Tour



<http://www.couchbase.com/couchconf-world-tour>

Upcoming CouchConf Events



CouchConf New York City

October 24, 2011

CouchConf is coming to the Big Apple! Join us for a great day of learning and interactive fun. Optional post-conference two-day training classes are also being offered - check it out!

[Learn more](#)



CouchConf Berlin

November 7, 2011

It's going to be a great week in Berlin: first, CouchBlack on November 5-6, then CouchConf on November 7, then Velocity EU on November 8-9. So don't miss this fabulous geek week, and be sure to reserve your seat at CouchConf Berlin now.

[Learn more](#)



CouchConf Chicago

November 16, 2011

CouchConf is on its way to the Windy City! Join us at beautiful Gleacher Center in downtown Chicago for a full day of learning and networking. Pre-conference CouchDB developer training is also available, so be sure to check it out.

[Learn more](#)



CouchConf Bangalore

December 7, 2011

Take a seat at CouchConf Bangalore! We are excited to be working with Zynga to host India's first-ever CouchConf. Join us at Zynga's new Bangalore offices for a day of learning and community fun. You'll hear from Couchbase experts, including Zynga developers who are some of the heaviest users of Couchbase technology.

[Learn more](#)



CouchConf Israel

December 19, 2011

We are looking forward to gathering with the technical community in Tel Aviv to talk about distributed database technology, and its advantages for web applications, mobile, and the cloud. Register now to reserve your place at CouchConf Israel!

[Learn more](#)



CouchConf Tokyo

January 27, 2012

We are excited to come to Tokyo to spread the word about Couchbase and get to know the CouchDB JP community. Early bird registration for this event is now open! Please email couchconf@tokyo@couchbase.com if you are interested in sponsoring this conference.

[Learn more](#)

Couchbase: Leading Open Source NoSQL

- Community Edition
 - Open source build
 - Free forum support
- Enterprise Edition
 - Free for non-production use
 - Certified, QA tested version of open source
 - Case tracking and guaranteed SLA for production environments
 - Annual subscriptions starting at \$999/node
 - Hourly cloud pricing via RightScale

Couchbase Server 2.0



COUCHBASE

=



Memcached



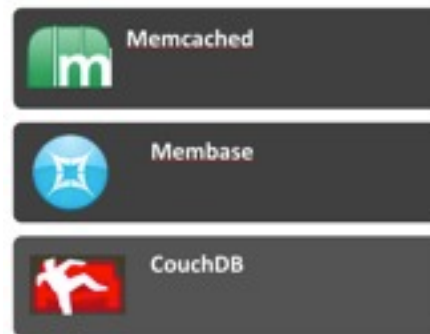
Membase



CouchDB

Simple. Fast. Elastic.

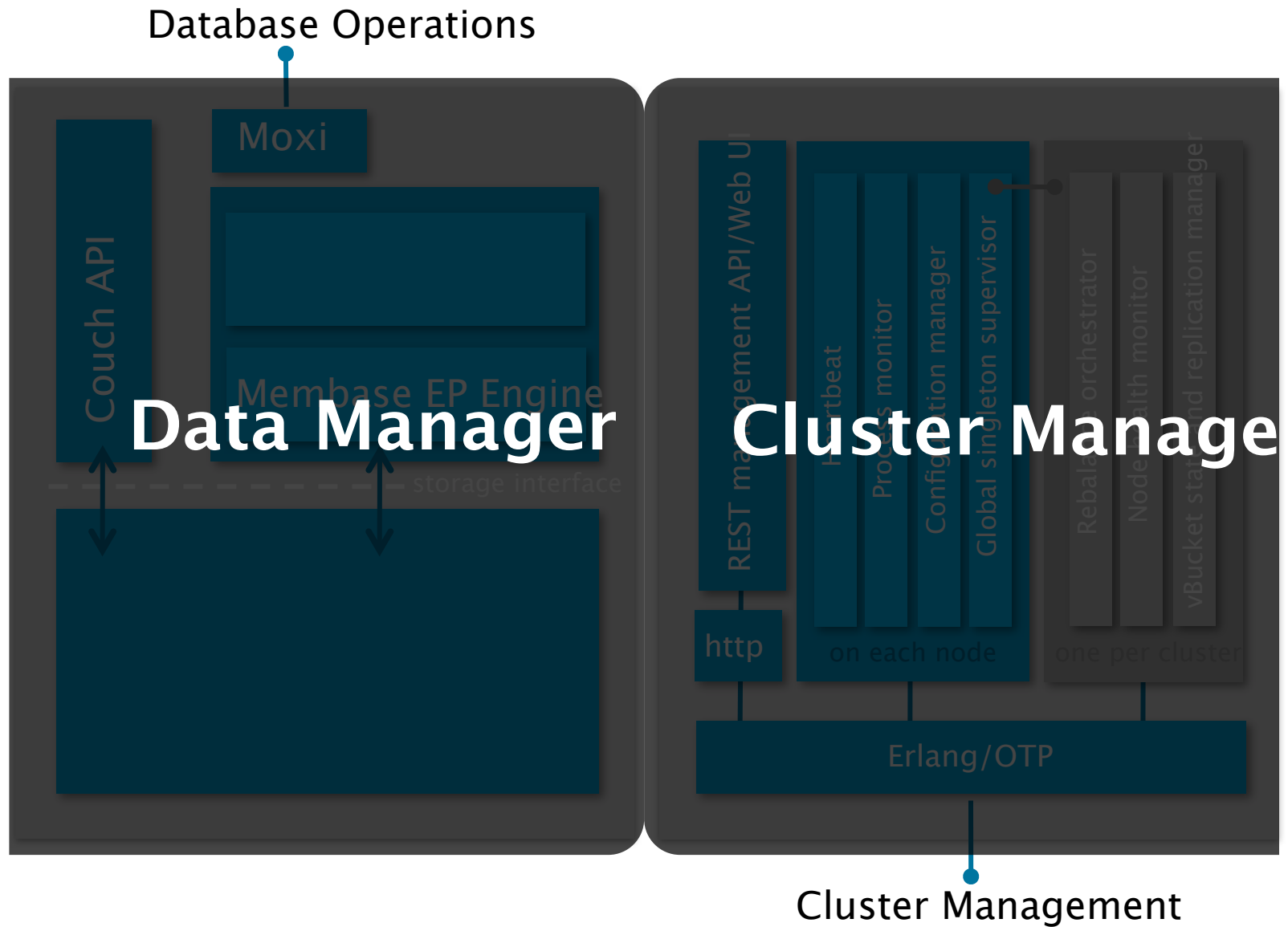
Couchbase Server 2.0: Overview



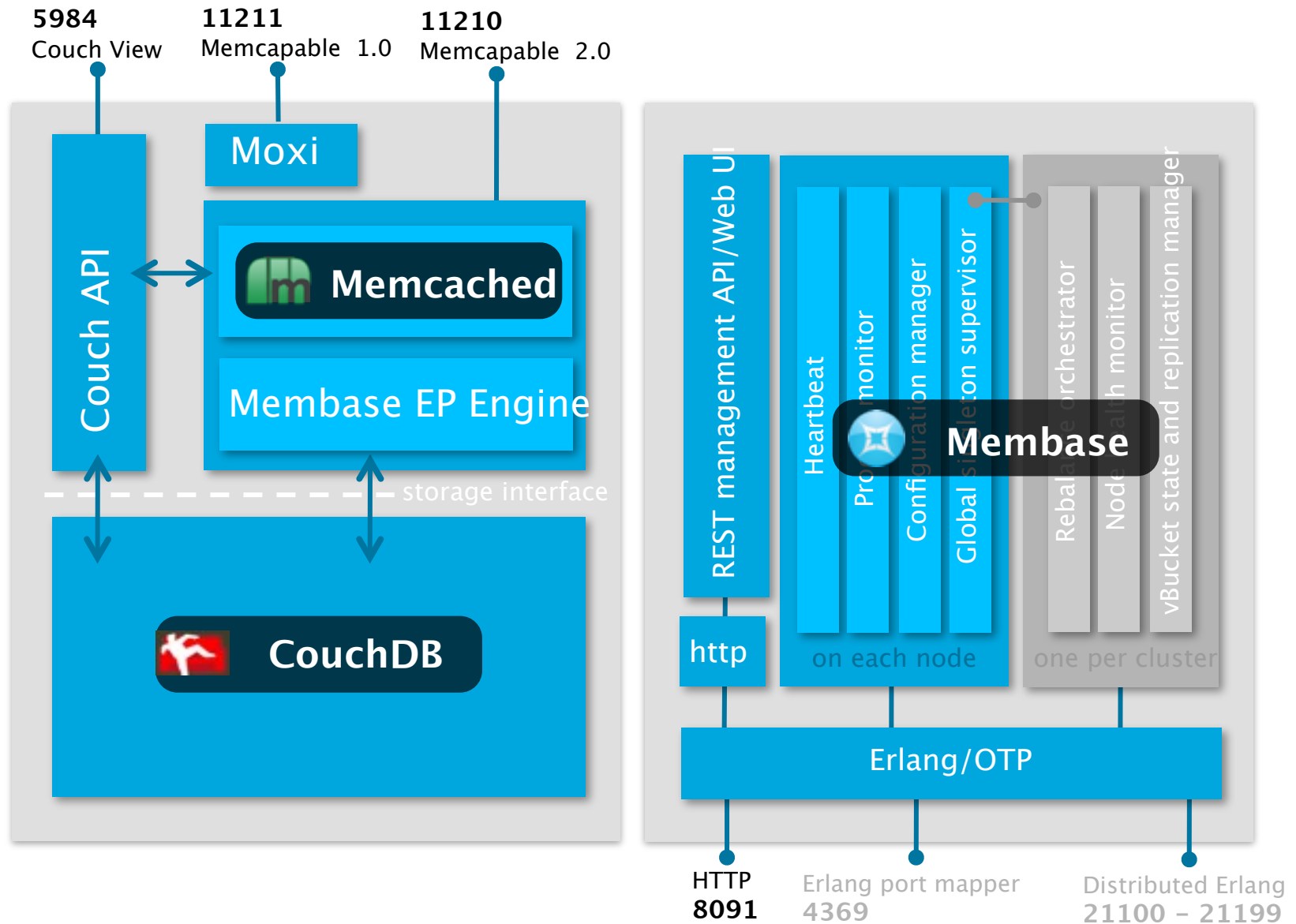
Simple. Fast. Elastic.

- Membase + CouchDB
- Managed memory caching layer = super high performance
- Clustering and online data redistribution (Rebalancing)
- Indexing and Querying via JSON Map-Reduce
- New SDK's and client libraries
- Developer Preview available today!
 - <http://www.couchbase.com/downloads>

Couchbase Server 2.0 Architecture



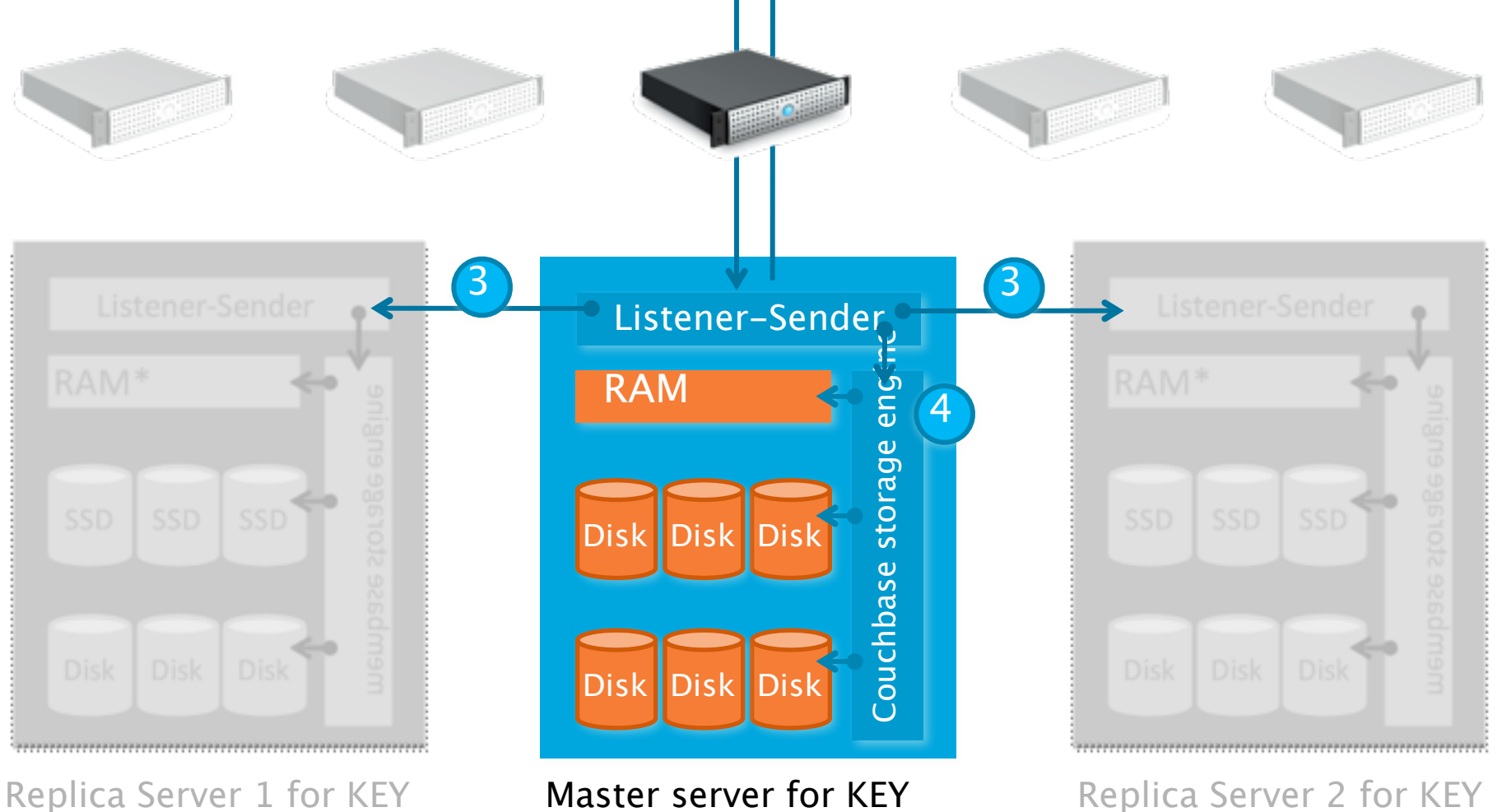
Couchbase Server 2.0 Architecture



Clustering With Couchbase

SET request arrives at KEY's master server 1

2 SET acknowledgement returned to application



Couchbase Client SDKs

Java Client SDK

.Net SDK

PHP SDK

Ruby SDK

Python SDK

User Code

Java client API

spymemcached Connection

HTTP
couchDB
connection

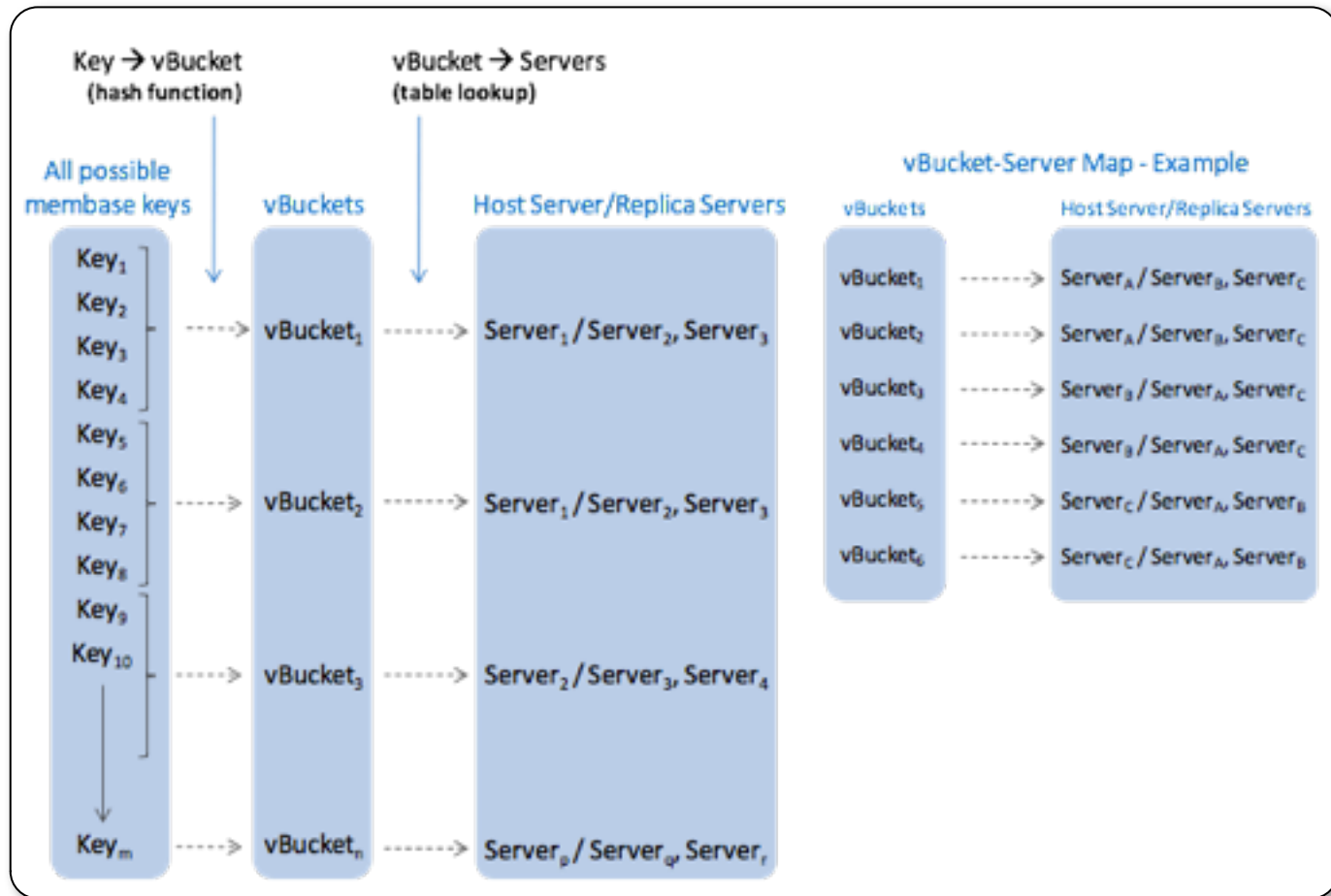
Couchbase Server



```
CouchbaseClient cb =  
new CouchbaseClient(listURIs,  
"aBucket", "letmein");  
// this is all the same as before  
cb.set("hello", 0, "world");  
cb.get("hello");  
Map<String, Object> manyThings =  
cb.getBulk(Collection<String> keys);  
/* accessing a view  
View view = cb.getView("design_document",  
"my_view");  
Query query = new Query();  
query.getRange("abegin", "theend");
```

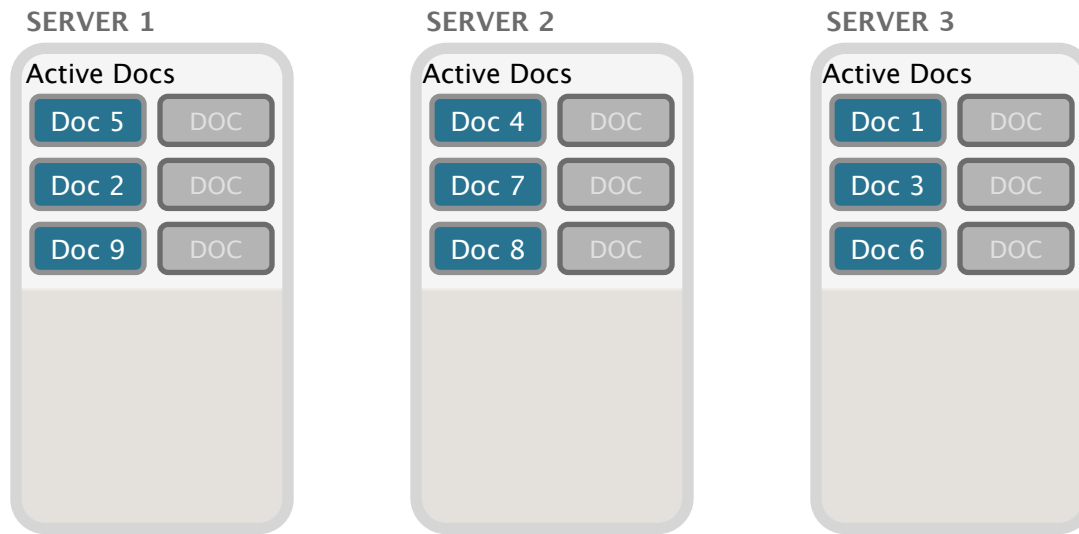
<http://www.couchbase.org/code>

Partitioning The Data – vbucket map



Basic Operation

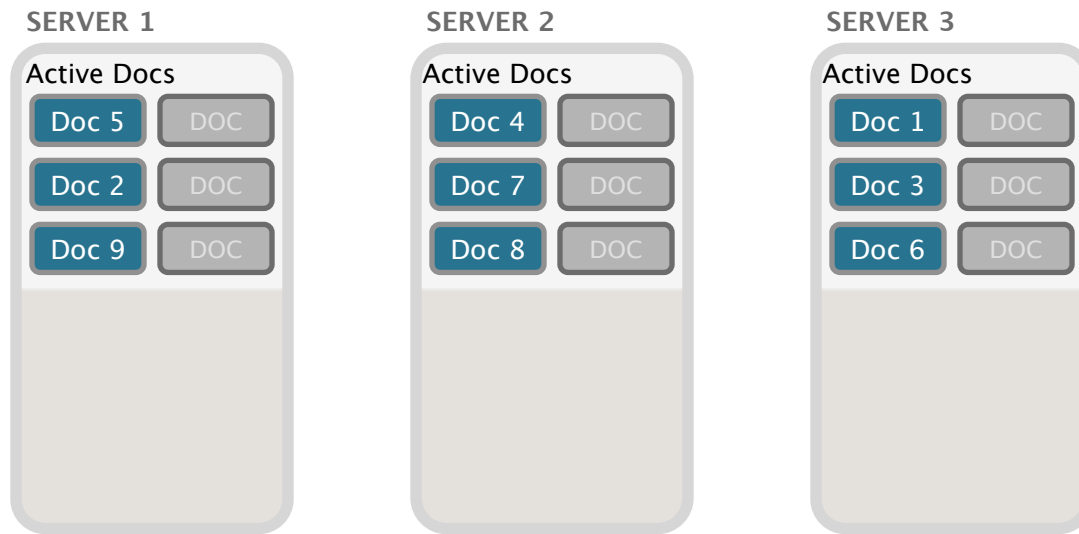
- Docs distributed evenly across servers in the cluster



COUCHBASE SERVER CLUSTER

Basic Operation

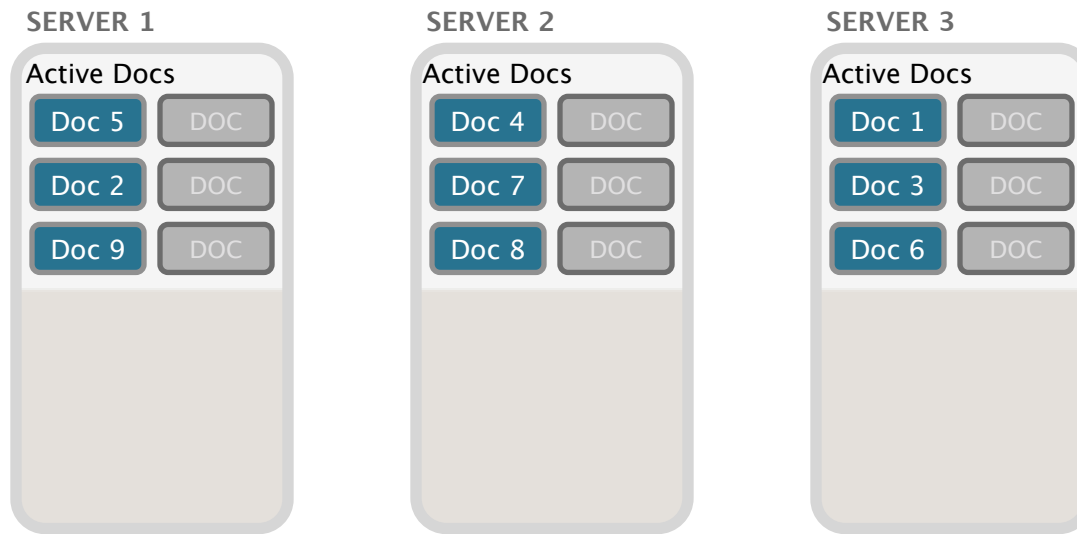
- Docs distributed evenly across servers in the cluster
- Each server stores both active & replica docs
 - Only one server active at a time



COUCHBASE SERVER CLUSTER

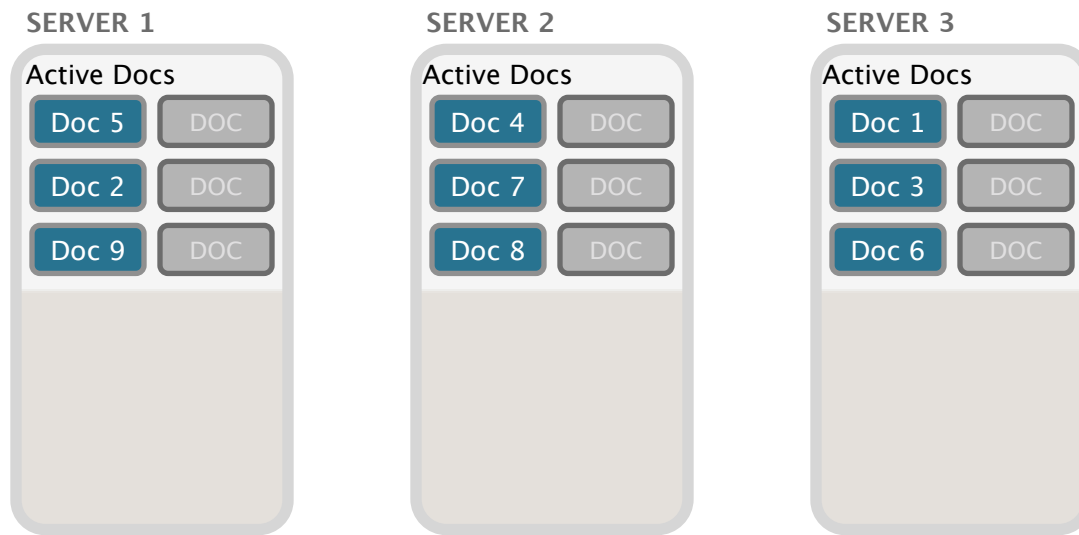
Basic Operation

- Docs distributed evenly across servers in the cluster
- Each server stores both active & replica docs
 - Only one server active at a time
- Client library provides app with simple interface to database



COUCHBASE SERVER CLUSTER

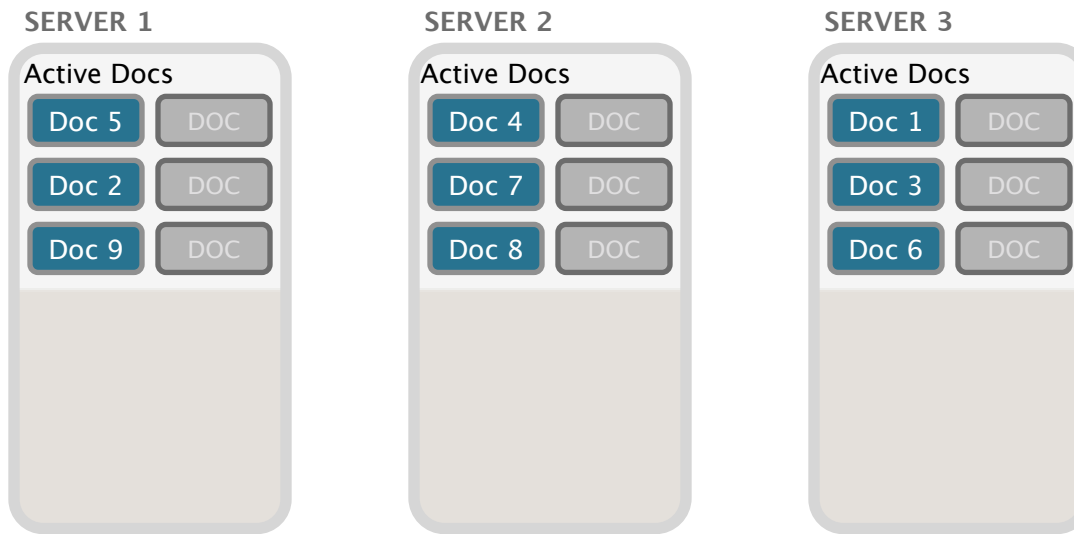
Basic Operation



COUCHBASE SERVER CLUSTER

- Docs distributed evenly across servers in the cluster
- Each server stores both active & replica docs
 - Only one server 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

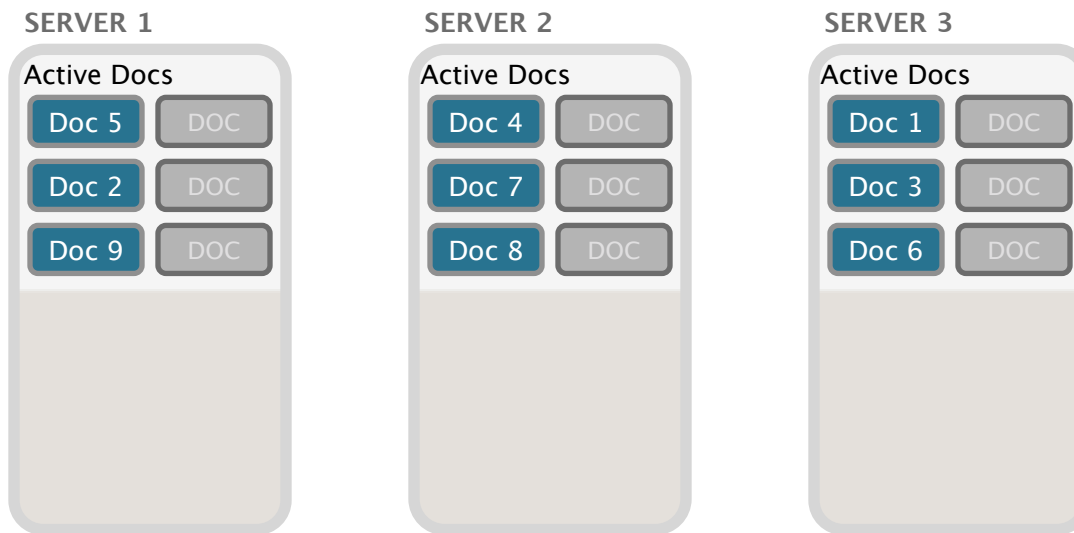
Basic Operation



COUCHBASE SERVER CLUSTER

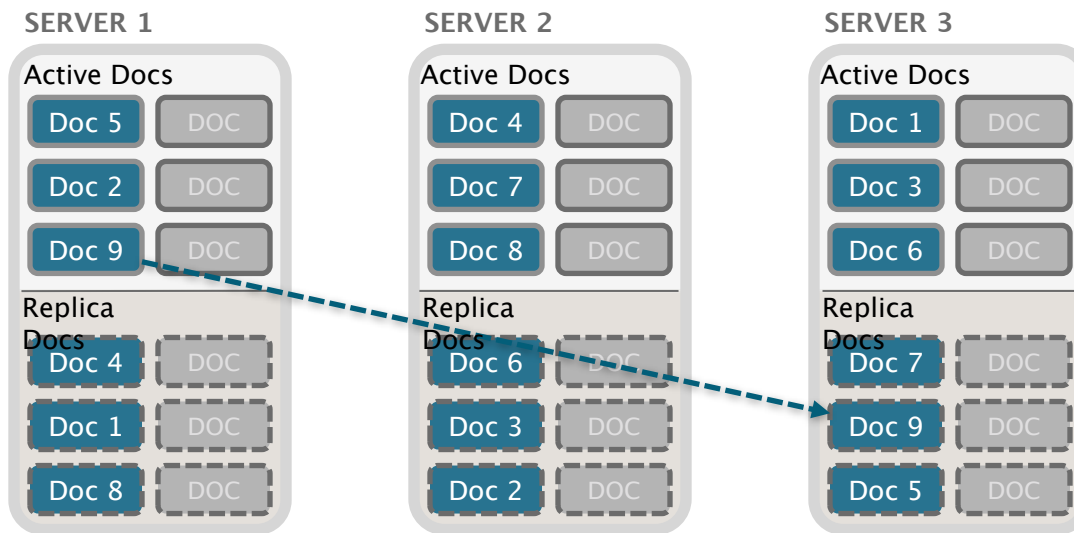
- Docs distributed evenly across servers in the cluster
- Each server stores both active & replica docs
 - Only one server active at a time
- Client library provides app with simple interface to database
- Cluster map provides map to which server doc is on
 - App never needs to know
- App reads, writes,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

Basic Operation



- Docs distributed evenly across servers in the cluster
- Each server stores both active & replica docs
 - Only one server 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

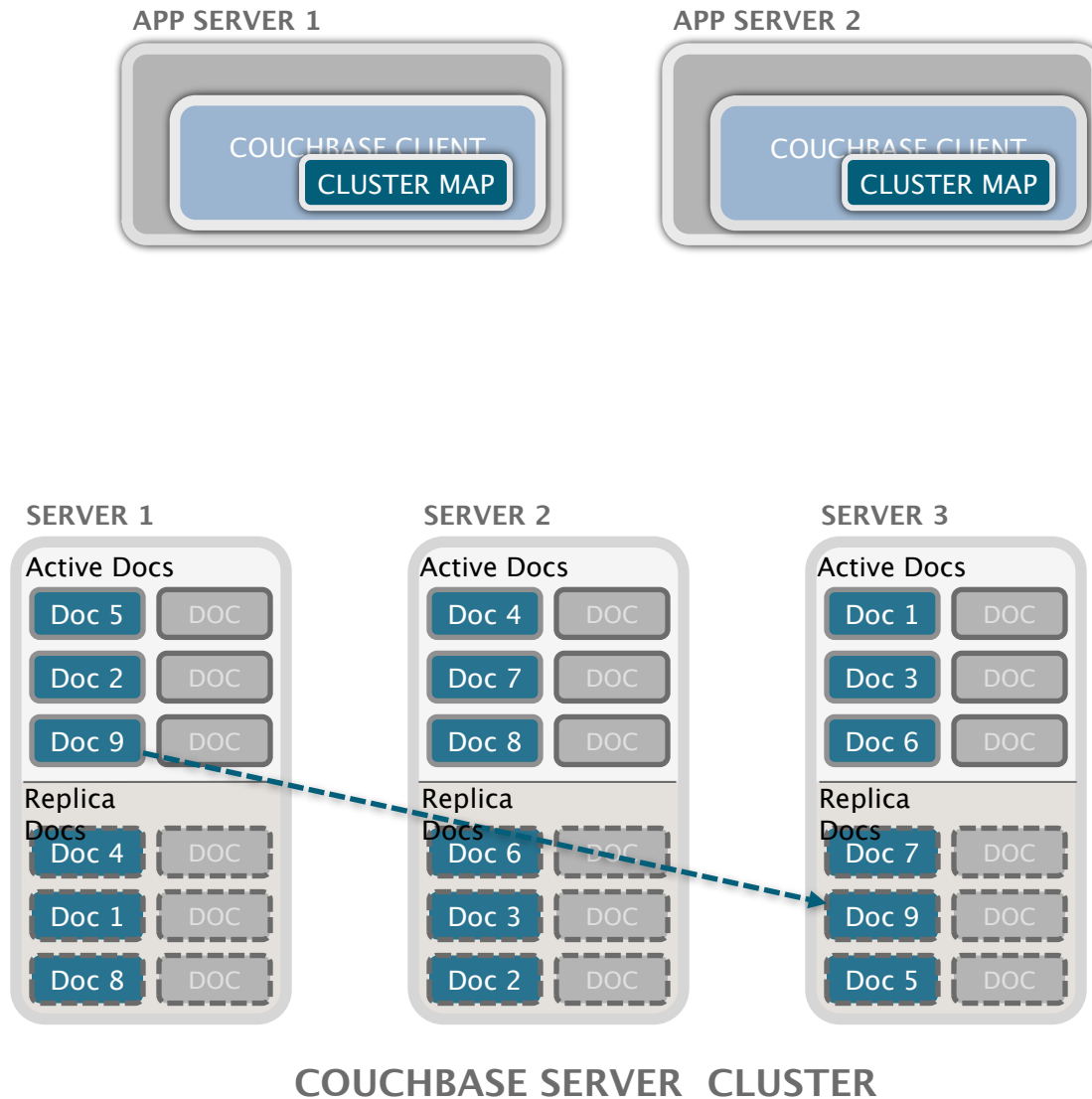
Basic Operation



COUCHBASE SERVER CLUSTER

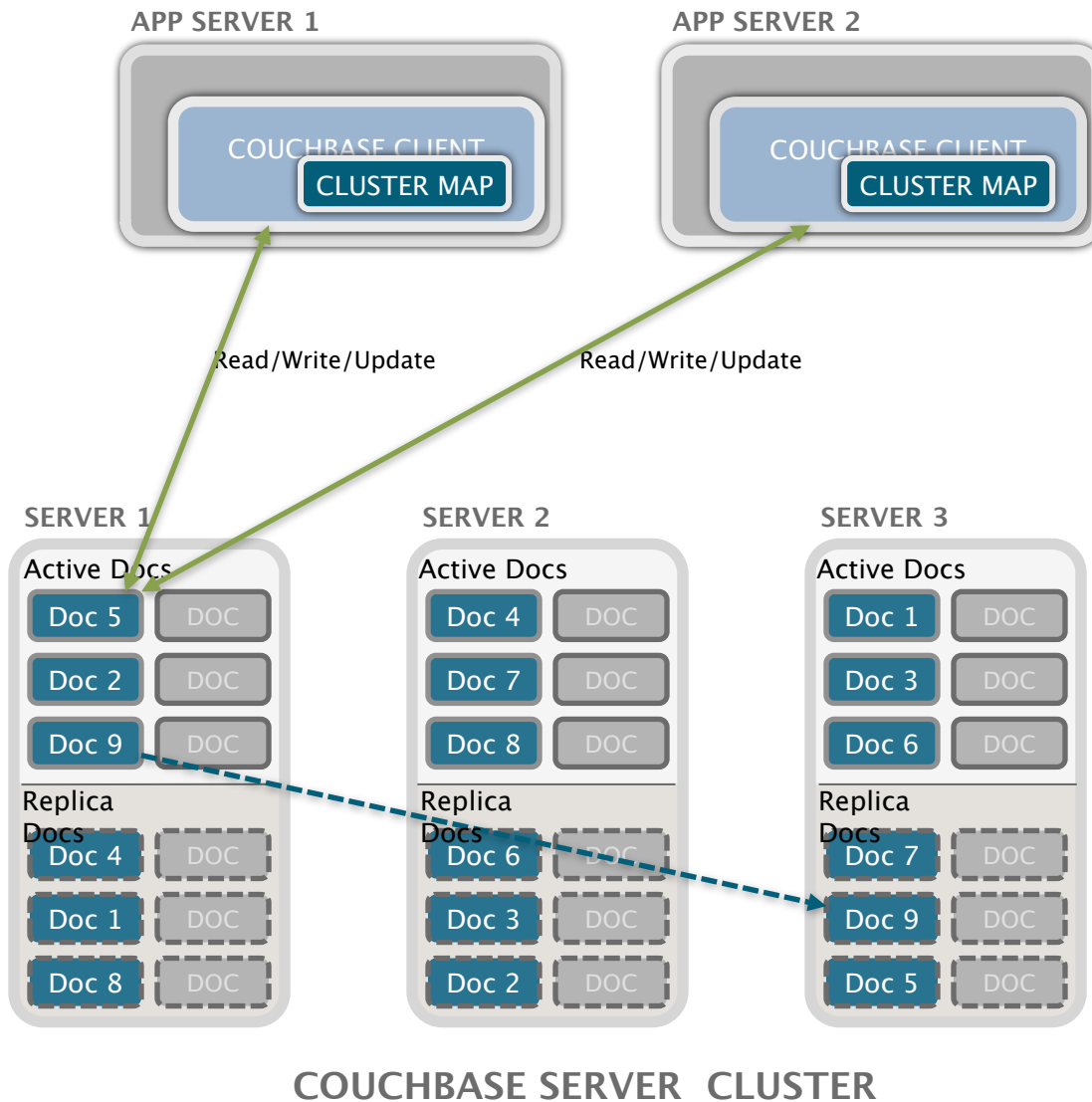
- Docs distributed evenly across servers in the cluster
- Each server stores both active & replica docs
 - Only one server active at a time
- Client library provides app with simple interface to database
- Cluster map provides map to which server doc is on
 - App never needs to know
- App reads, writes,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

Basic Operation



- Docs distributed evenly across servers in the cluster
- Each server stores both active & replica docs
 - Only one server 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

Basic Operation



- Docs distributed evenly across servers in the cluster
- Each server stores both active & replica docs
 - Only one server 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

Indexing and Querying

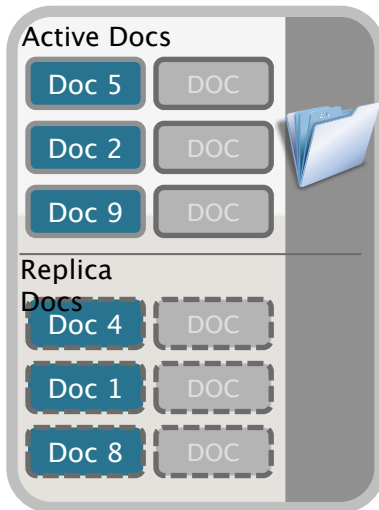
APP SERVER 1



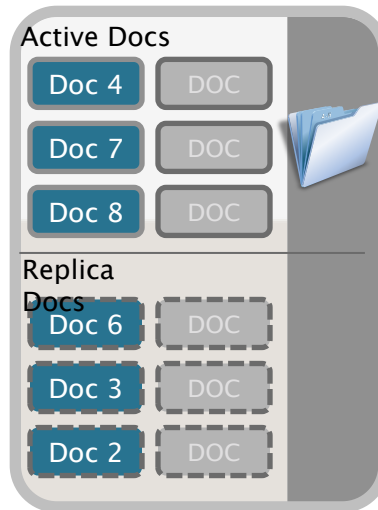
APP SERVER 2



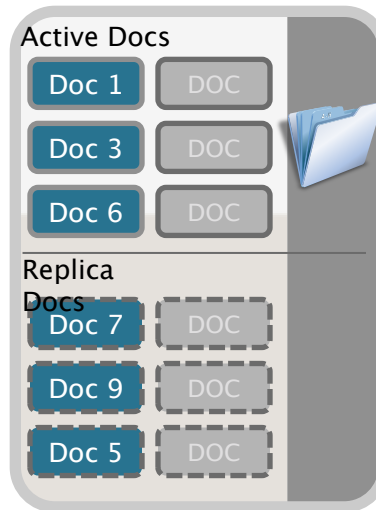
SERVER 1



SERVER 2



SERVER 3



Indexing and Querying

APP SERVER 1



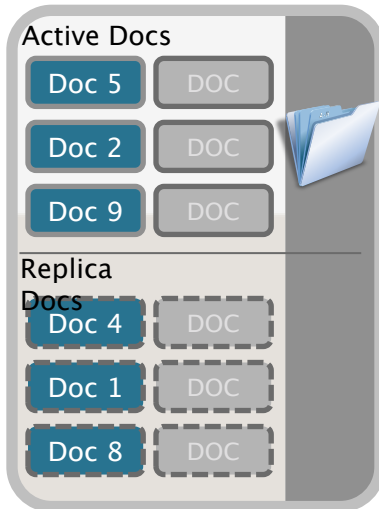
APP SERVER 2



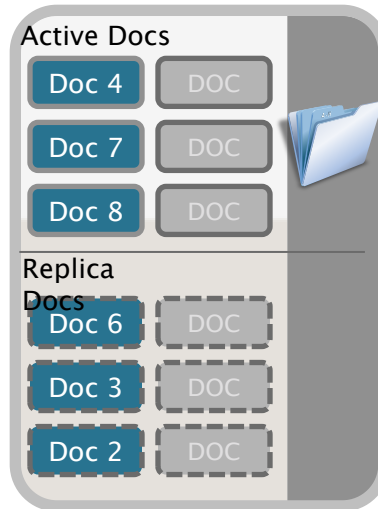
- Indexing work is distributed amongst nodes

- Large data set possible
- Parallelize the effort

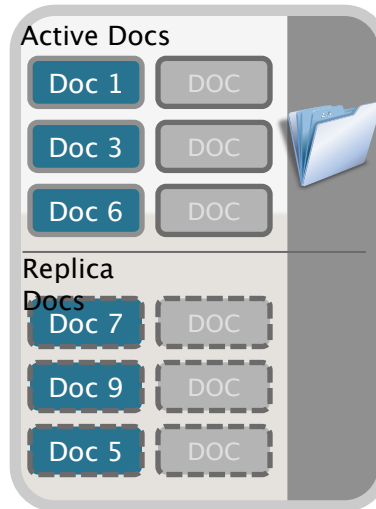
SERVER 1



SERVER 2



SERVER 3



Indexing and Querying

APP SERVER 1

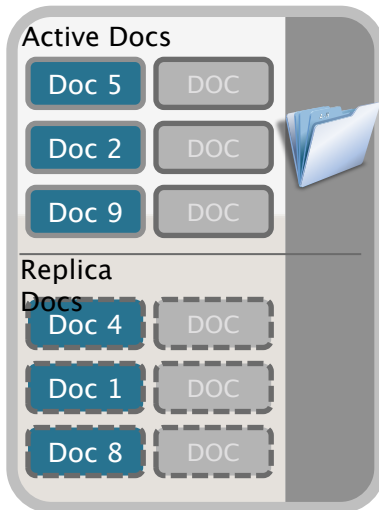


APP SERVER 2

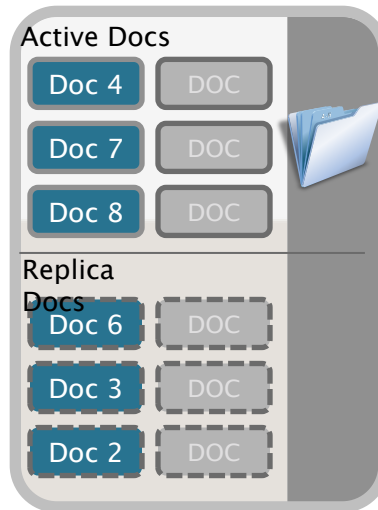


- Indexing work is distributed amongst nodes
 - Large data set possible
 - Parallelize the effort
- Each node has index for data stored on it

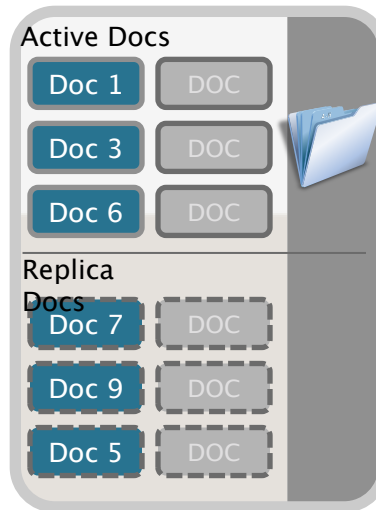
SERVER 1



SERVER 2



SERVER 3



Indexing and Querying

APP SERVER 1

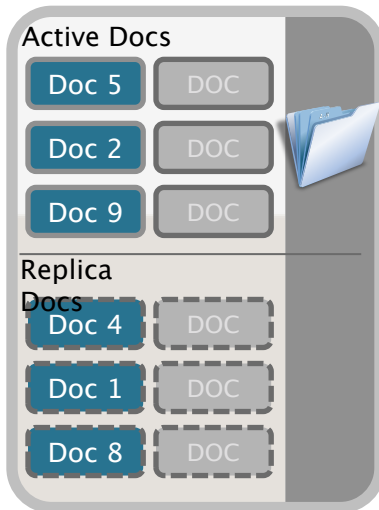


APP SERVER 2

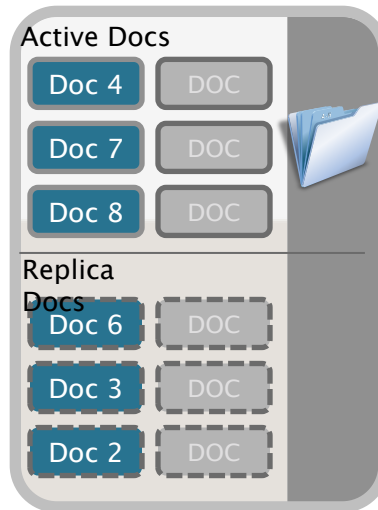


- Indexing work is distributed amongst nodes
 - Large data set possible
 - Parallelize the effort
- Each node has index for data stored on it
- Queries combined the results from required nodes

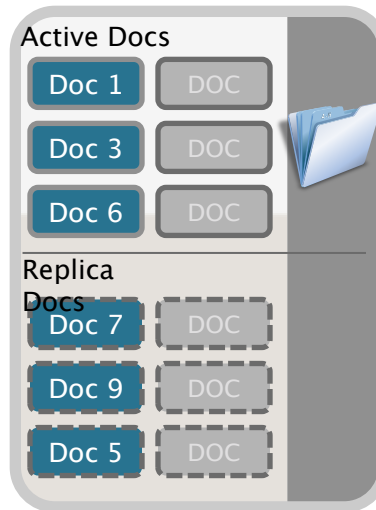
SERVER 1



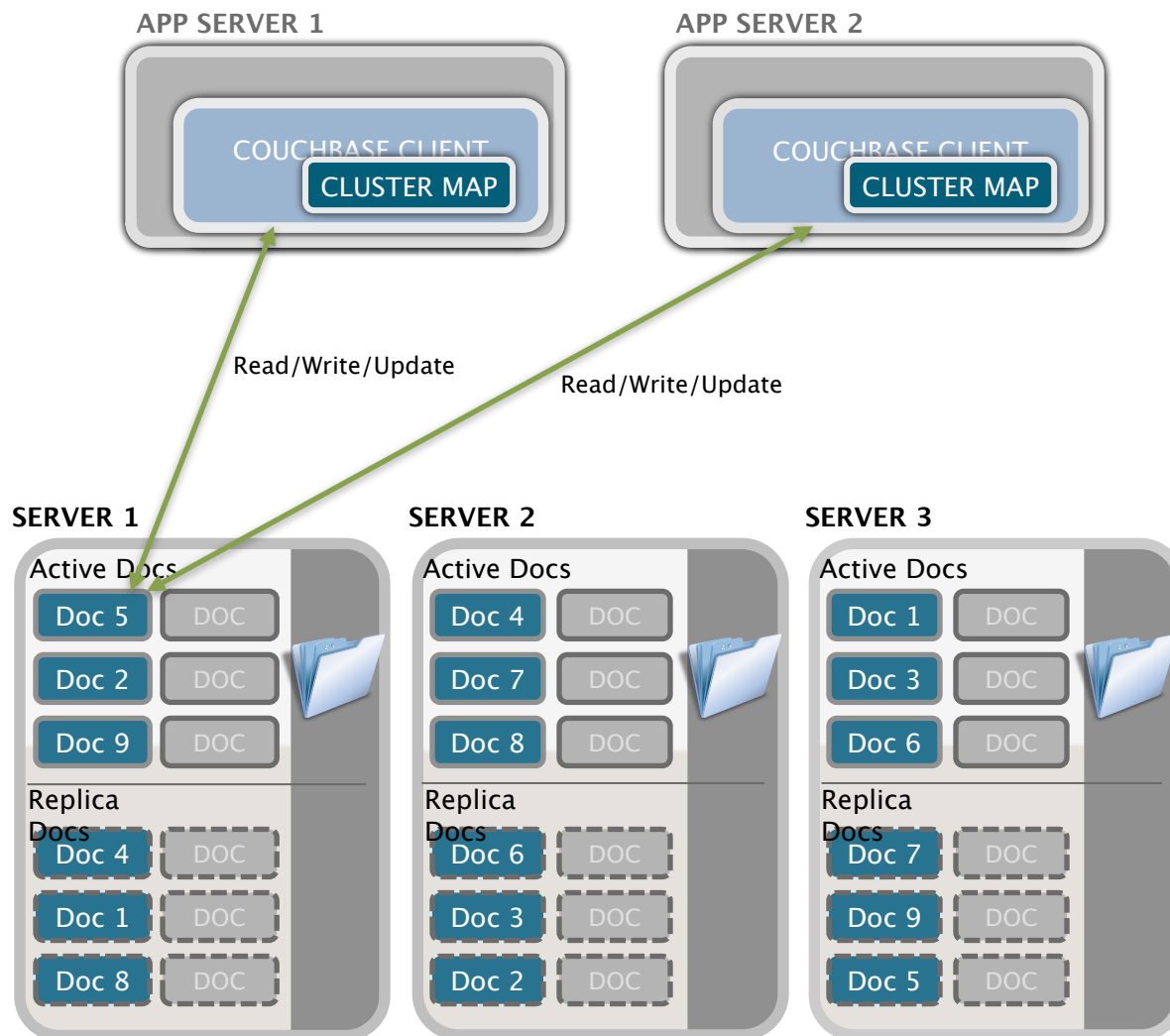
SERVER 2



SERVER 3

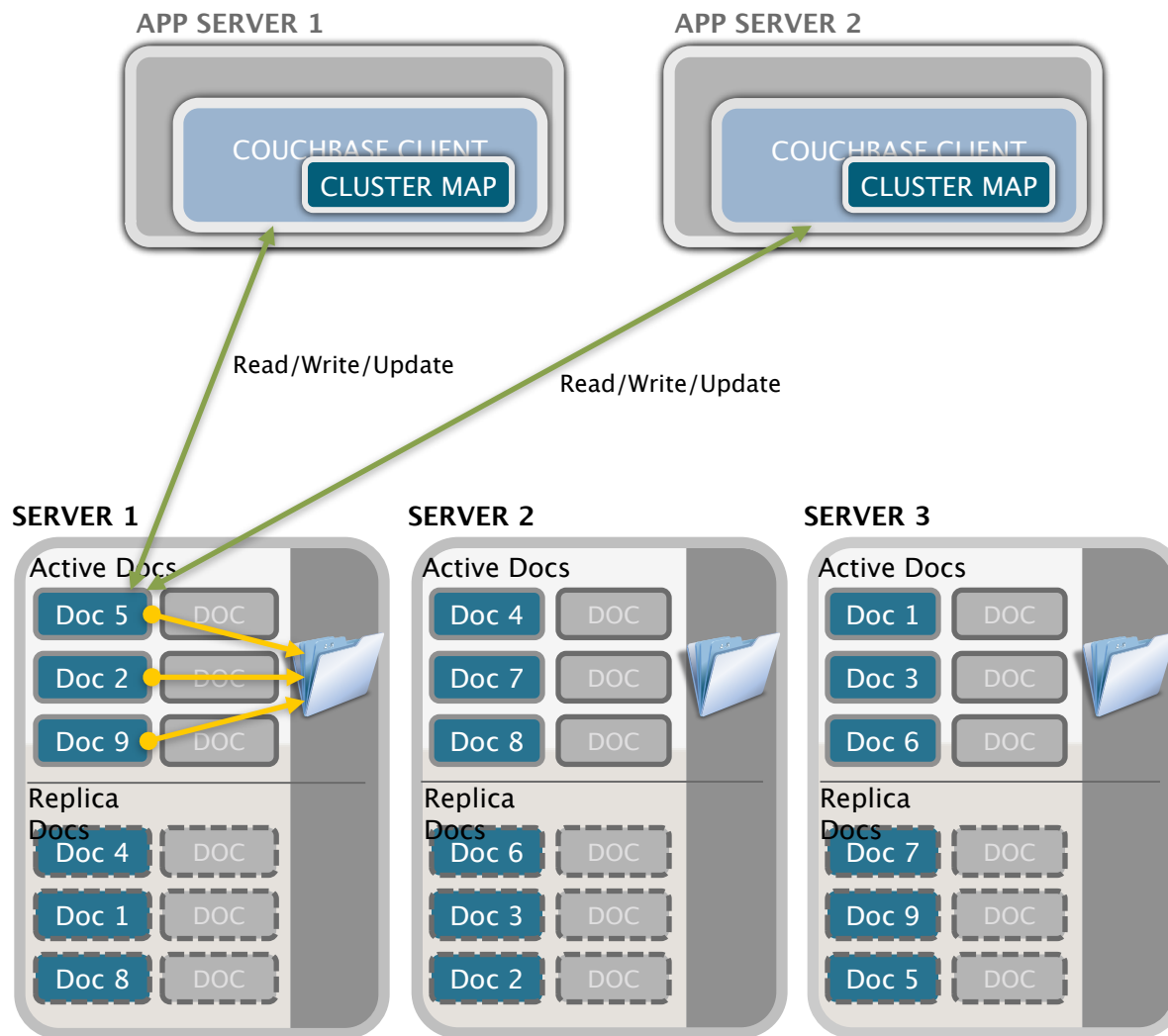


Indexing and Querying



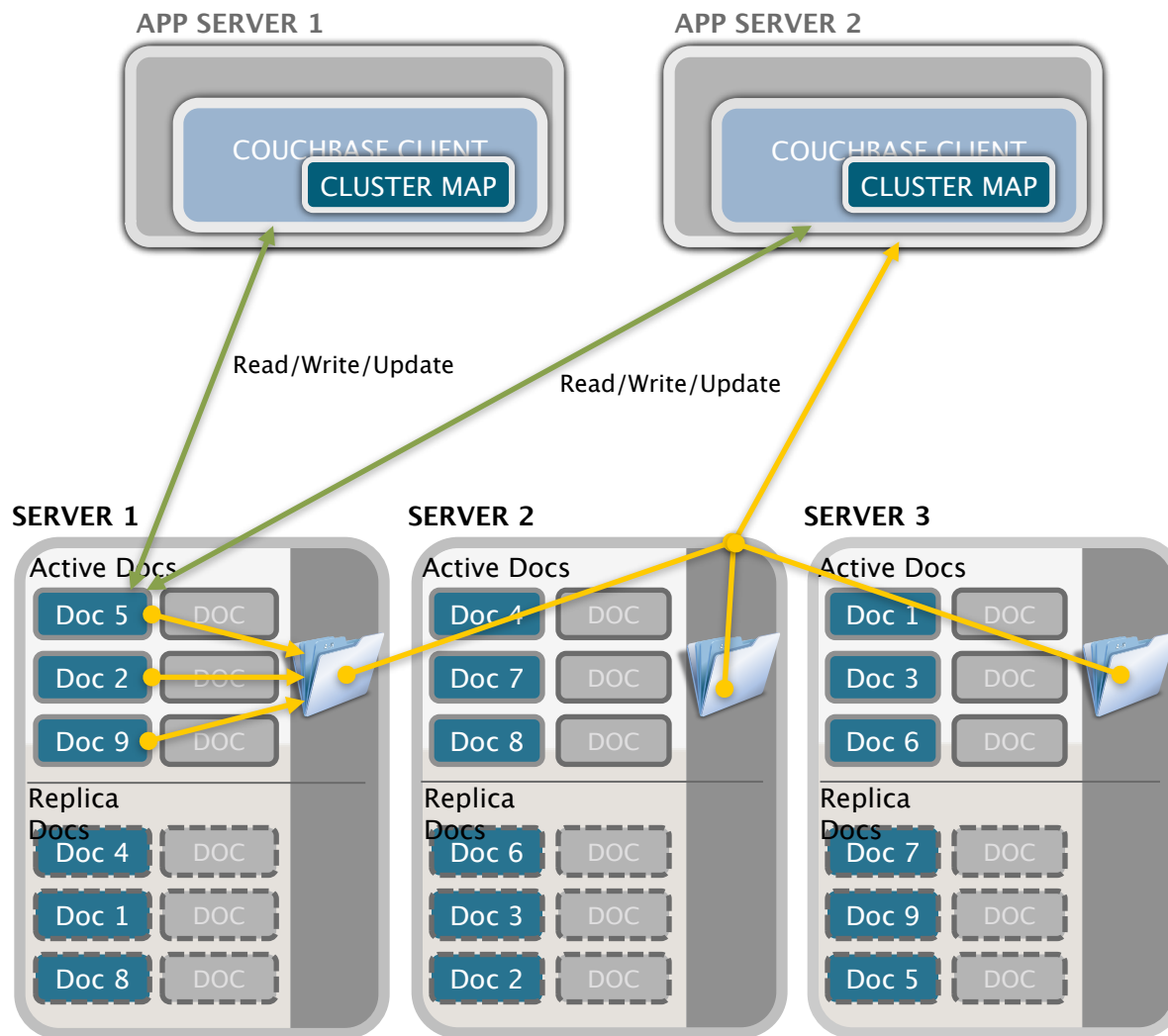
- Indexing work is distributed amongst nodes
 - Large data set possible
 - Parallelize the effort
- Each node has index for data stored on it
- Queries combined the results from required nodes

Indexing and Querying



- Indexing work is distributed amongst nodes
 - Large data set possible
 - Parallelize the effort
- Each node has index for data stored on it
- Queries combined the results from required nodes

Indexing and Querying



- Indexing work is distributed amongst nodes
 - Large data set possible
 - Parallelize the effort
- Each node has index for data stored on it
- Queries combined the results from required nodes

Add Nodes

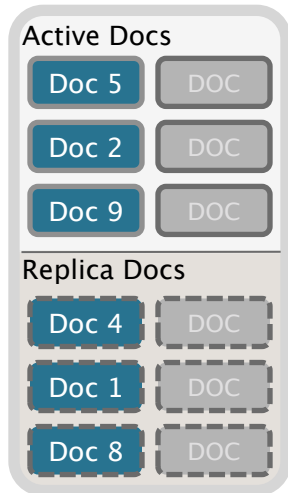
APP SERVER 1



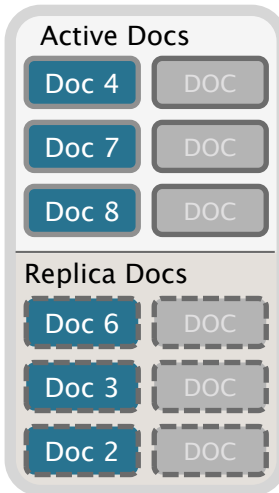
APP SERVER 2



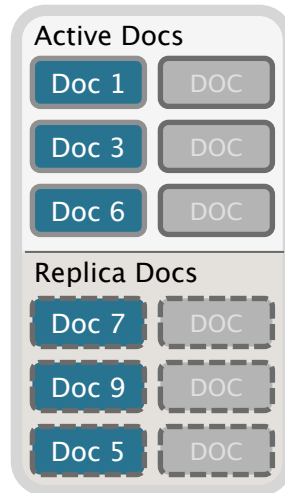
SERVER 1



SERVER 2



SERVER 3



COUCHBASE SERVER CLUSTER

Add Nodes

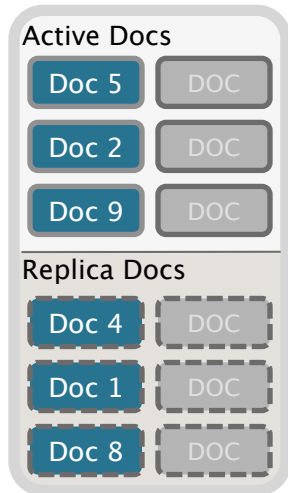
APP SERVER 1



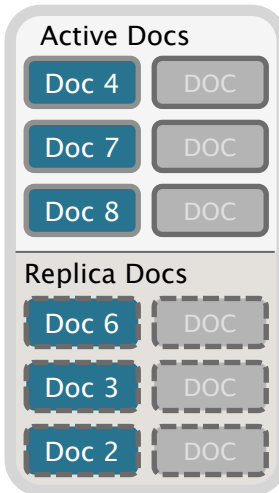
APP SERVER 2



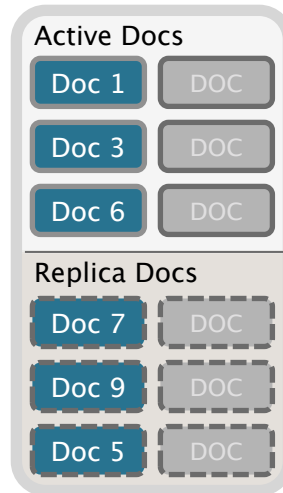
SERVER 1



SERVER 2



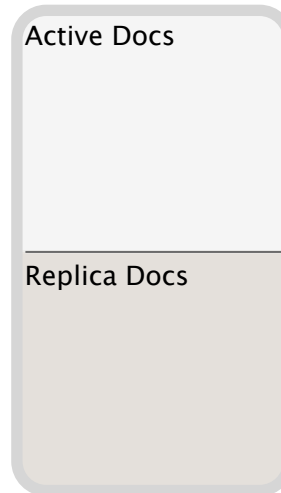
SERVER 3



SERVER 4



SERVER 5



COUCHBASE SERVER CLUSTER

Add Nodes

APP SERVER 1

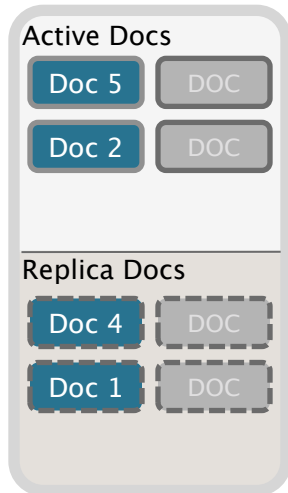


APP SERVER 2

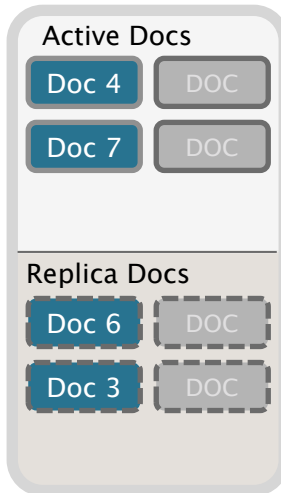


- Two servers added to cluster
 - One-click operation

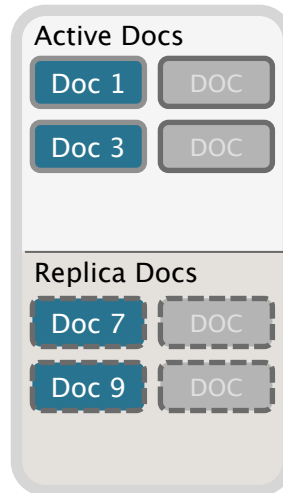
SERVER 1



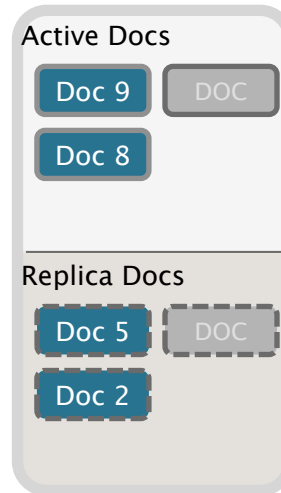
SERVER 2



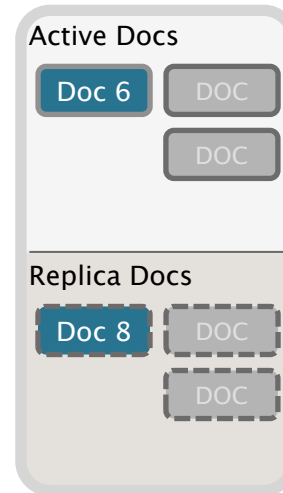
SERVER 3



SERVER 4



SERVER 5



COUCHBASE SERVER CLUSTER

Add Nodes

APP SERVER 1

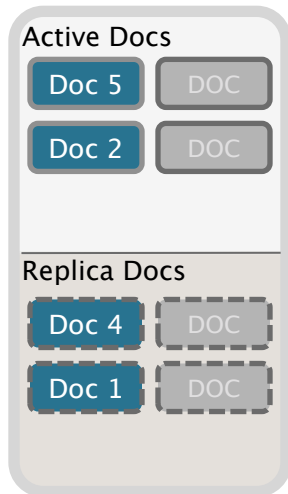


APP SERVER 2

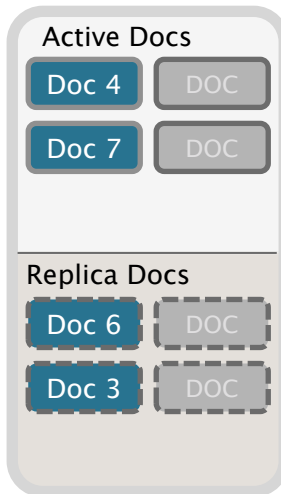


- Two servers added to cluster
 - One-click operation
- Docs automatically rebalanced across cluster
 - Even distribution of docs
 - Minimum doc movement

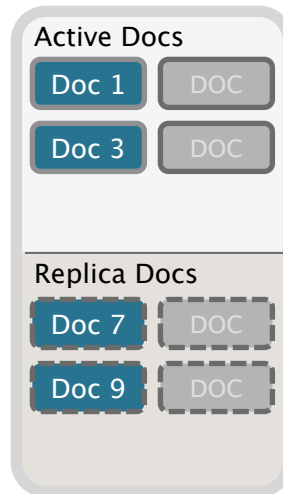
SERVER 1



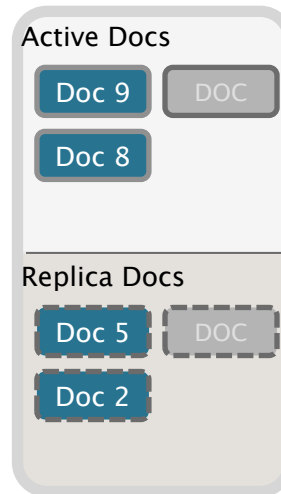
SERVER 2



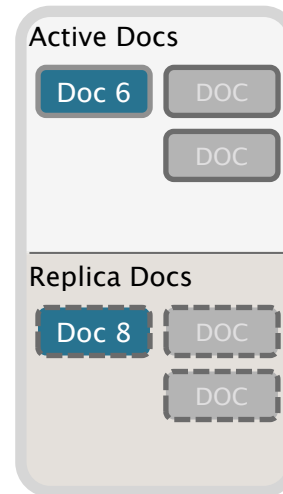
SERVER 3



SERVER 4



SERVER 5



COUCHBASE SERVER CLUSTER

Add Nodes

APP SERVER 1



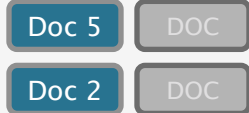
APP SERVER 2



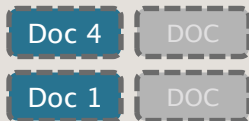
- Two servers added to cluster
 - One-click operation
- Docs automatically rebalanced across cluster
 - Even distribution of docs
 - Minimum doc movement
- Cluster map updated

SERVER 1

Active Docs



Replica Docs



SERVER 2

Active Docs



Replica Docs



SERVER 3

Active Docs



Replica Docs



SERVER 4

Active Docs



Replica Docs



SERVER 5

Active Docs



Replica Docs



COUCHBASE SERVER CLUSTER

Add Nodes

APP SERVER 1



APP SERVER 2

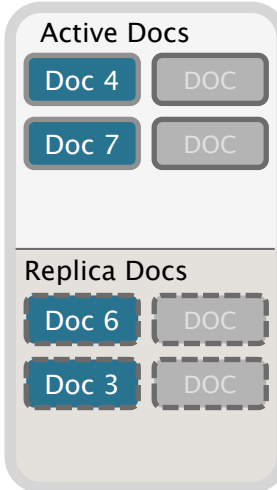


- Two servers added to cluster
 - One-click operation
- Docs automatically rebalanced across cluster
 - Even distribution of docs
 - Minimum doc movement
- Cluster map updated
- App database calls now distributed over larger # of servers

SERVER 1



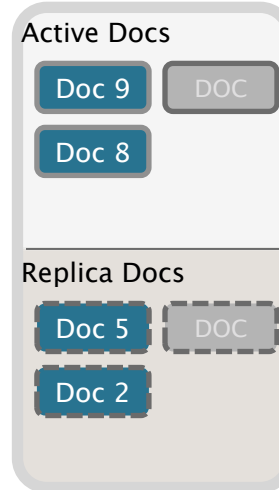
SERVER 2



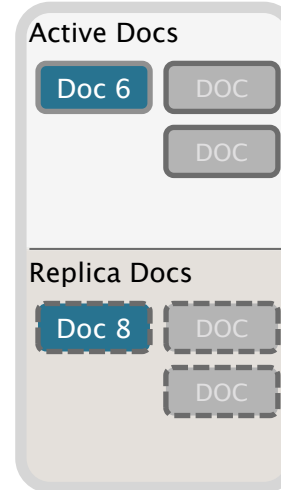
SERVER 3



SERVER 4

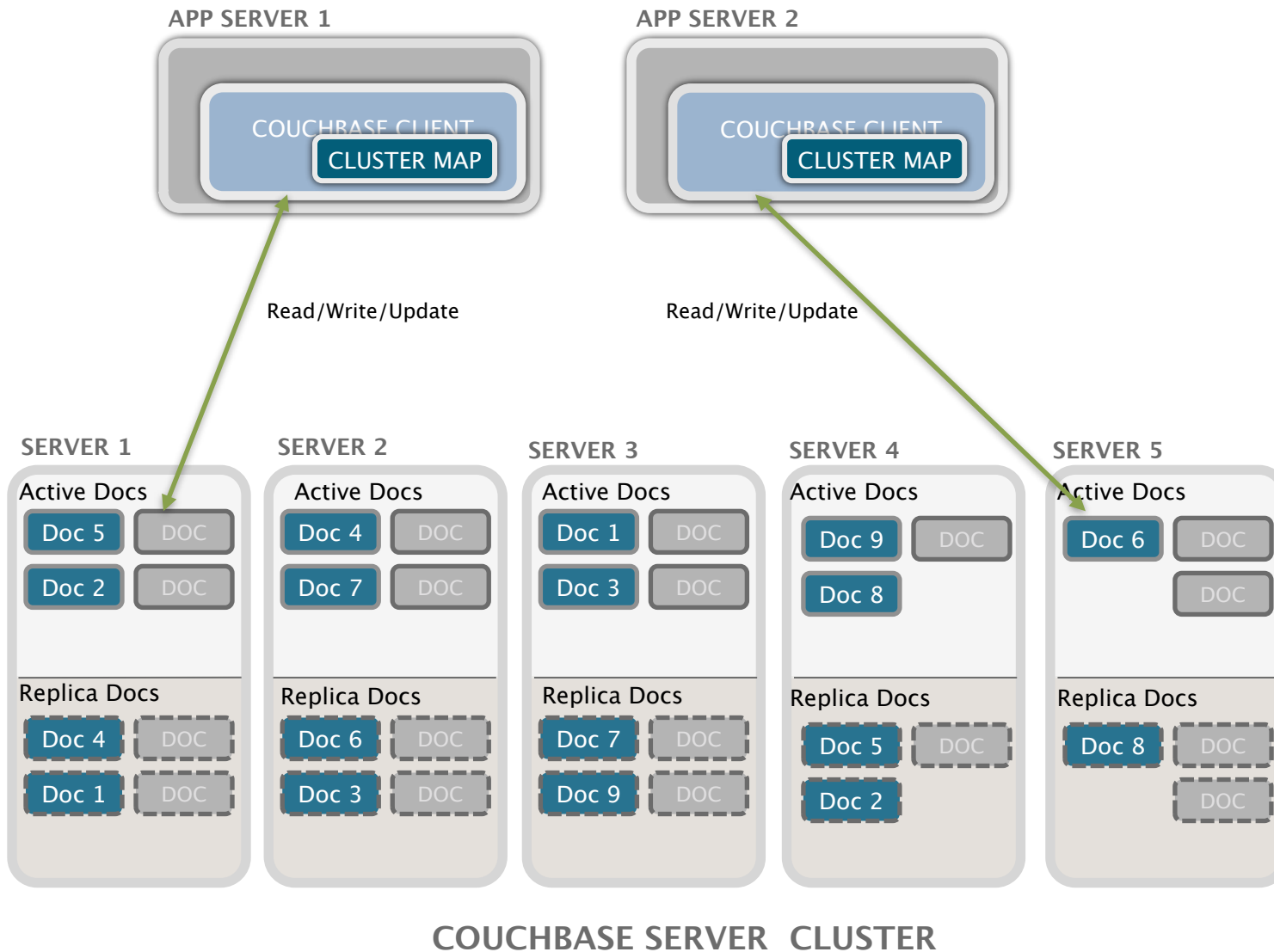


SERVER 5



COUCHBASE SERVER CLUSTER

Add Nodes



- Two servers added to cluster
 - One-click operation
- Docs automatically rebalanced across cluster
 - Even distribution of docs
 - Minimum doc movement
- Cluster map updated
- App database calls now distributed over larger # of servers

COUCHBASE SERVER CLUSTER

Fail Over Node

APP SERVER 1

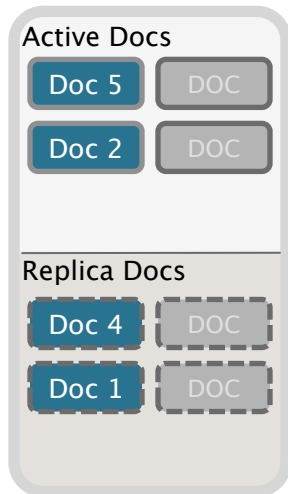


APP SERVER 2

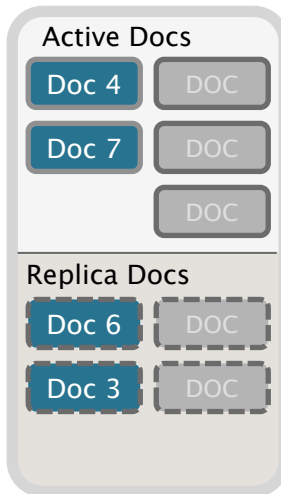


- App servers happily accessing docs on Server 3
- Server fails
- App server requests to server 3 fail
- Cluster detects server has failed
 - Promotes replicas of docs to active
 - Updates cluster map
- App server requests for docs now go to appropriate server
- Typically rebalance would follow

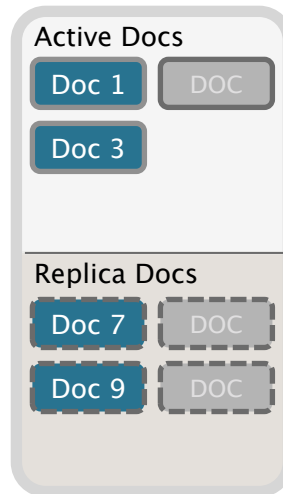
SERVER 1



SERVER 2



SERVER 3



SERVER 4

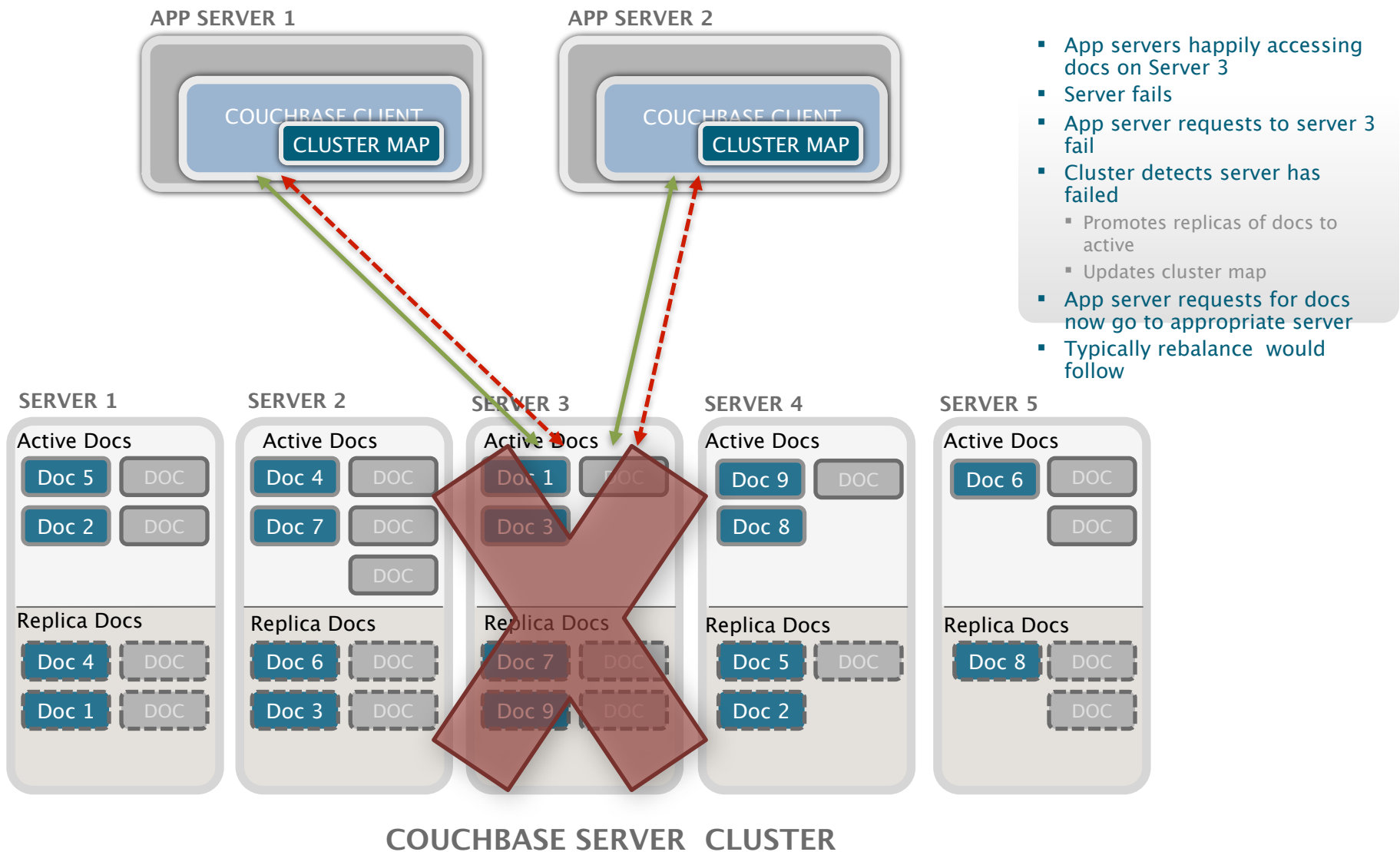


SERVER 5



COUCHBASE SERVER CLUSTER

Fail Over Node



Fail Over Node

APP SERVER 1

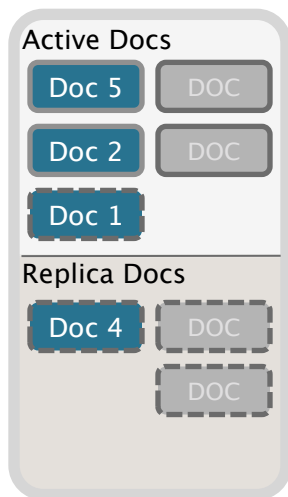


APP SERVER 2

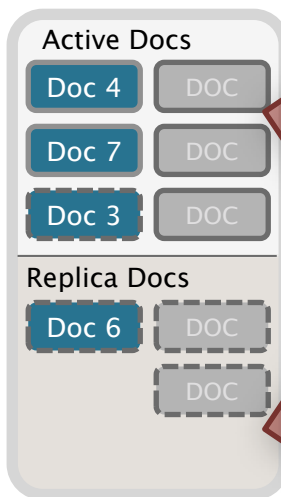


- App servers happily accessing docs on Server 3
- Server fails
- App server requests to server 3 fail

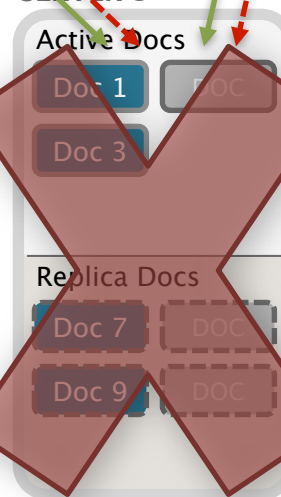
SERVER 1



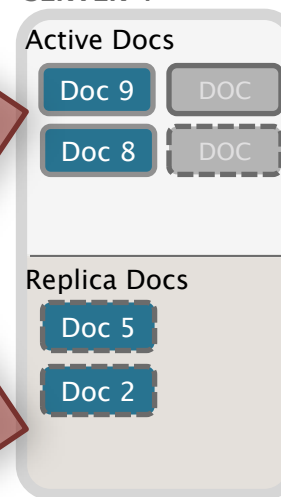
SERVER 2



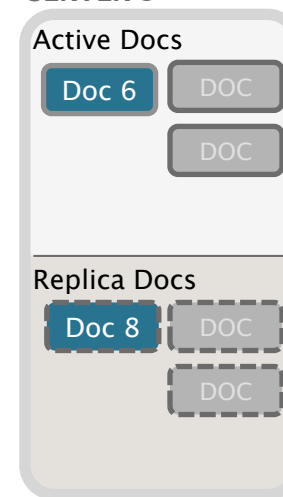
SERVER 3



SERVER 4

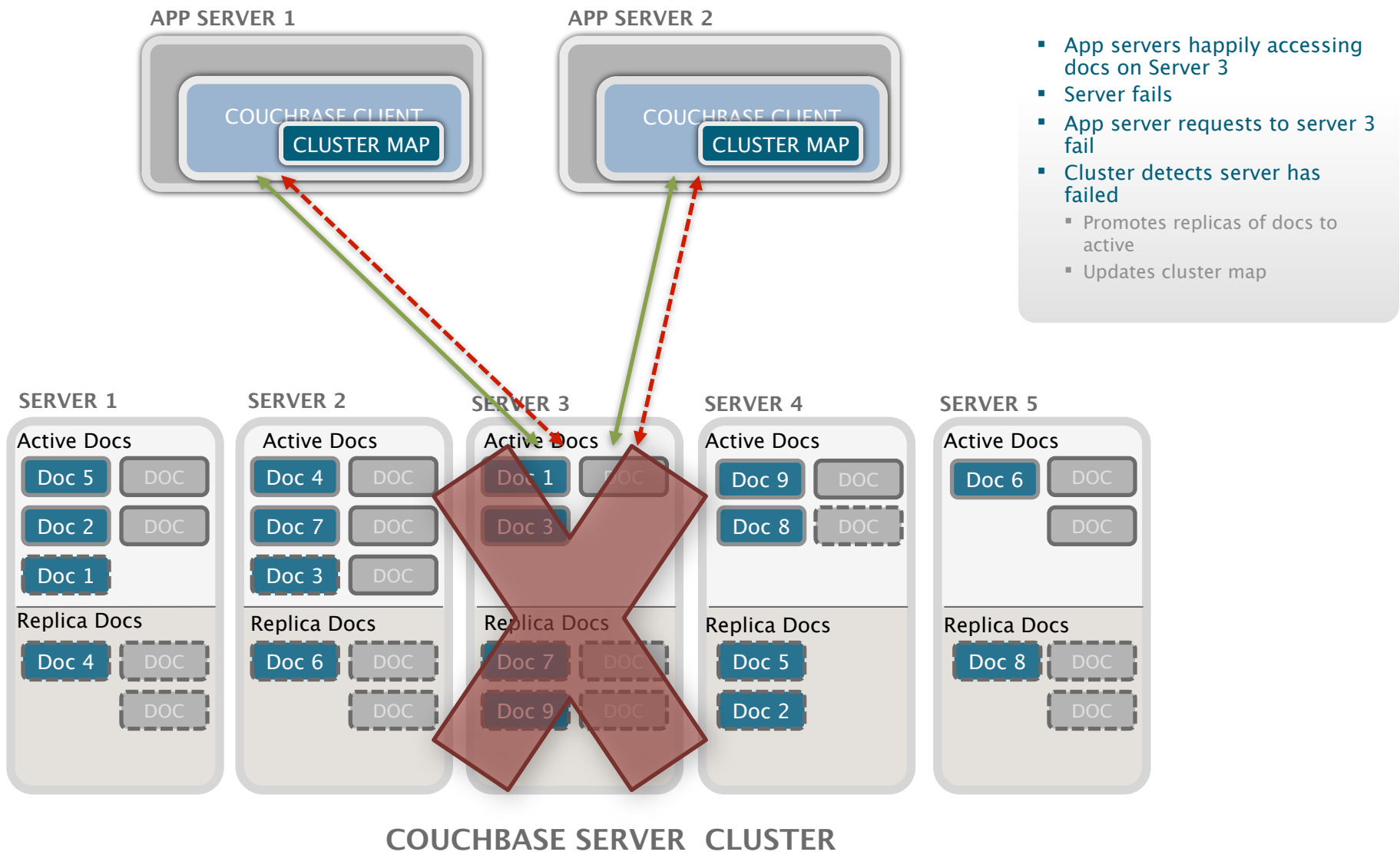


SERVER 5

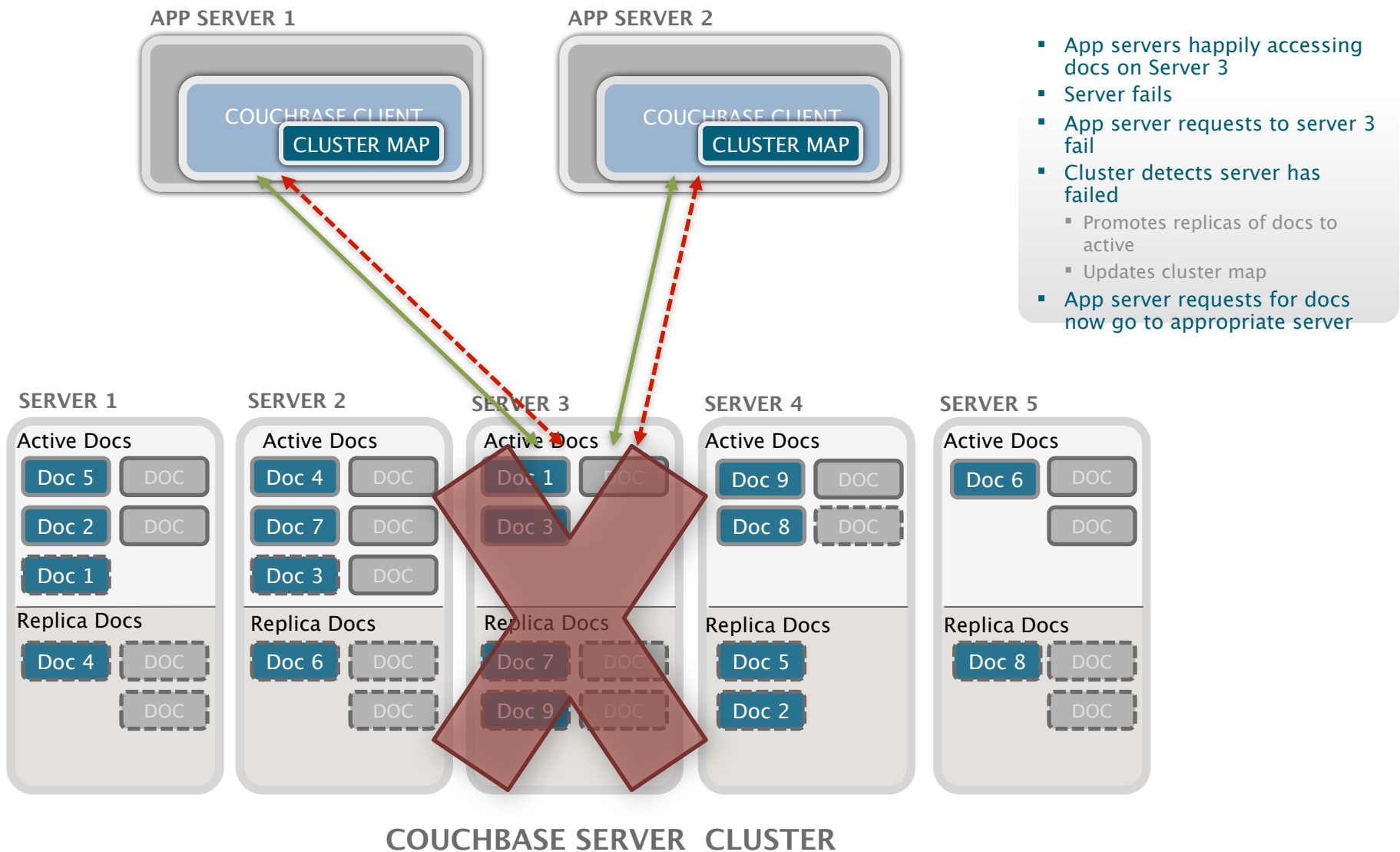


COUCHBASE SERVER CLUSTER

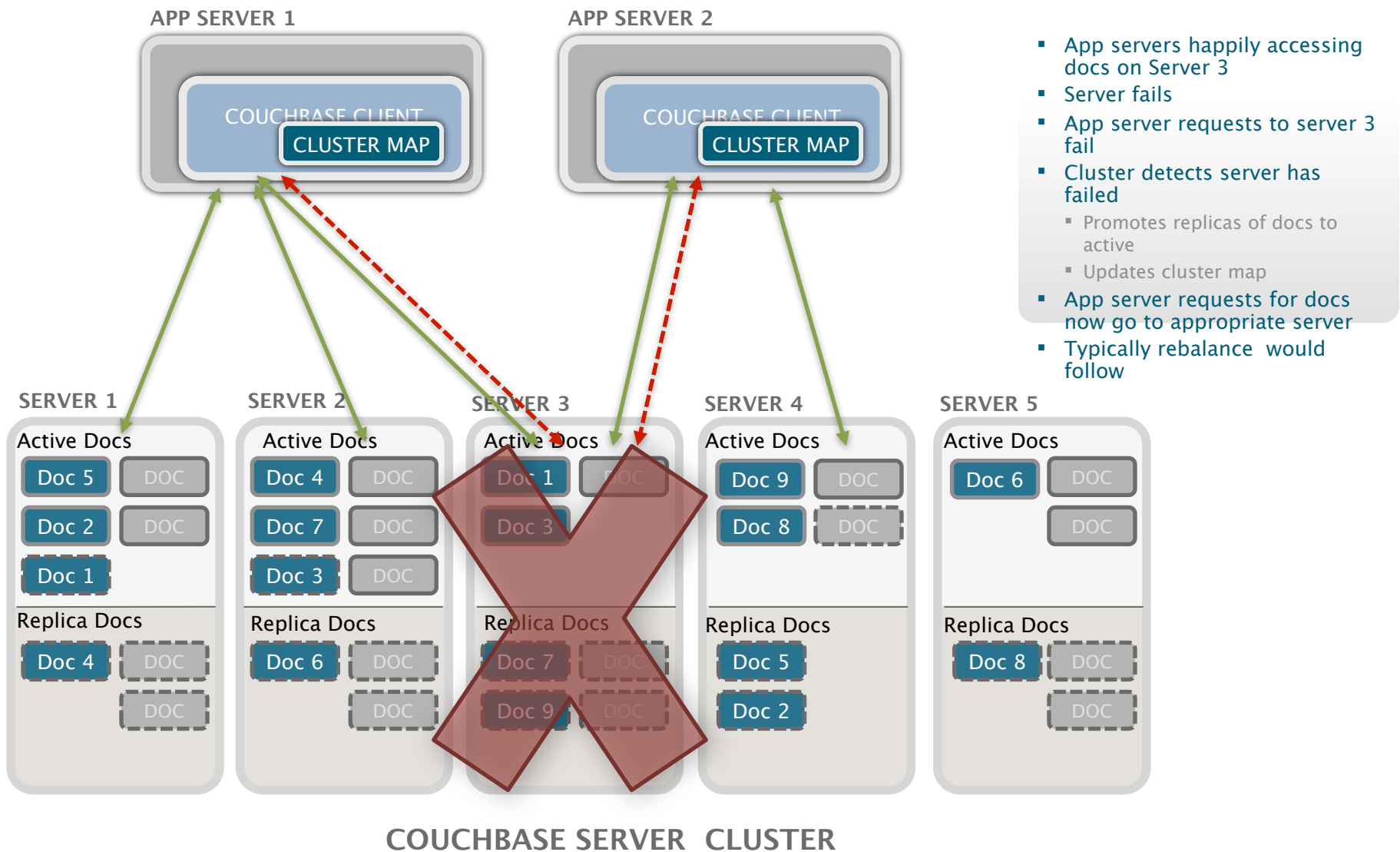
Fail Over Node



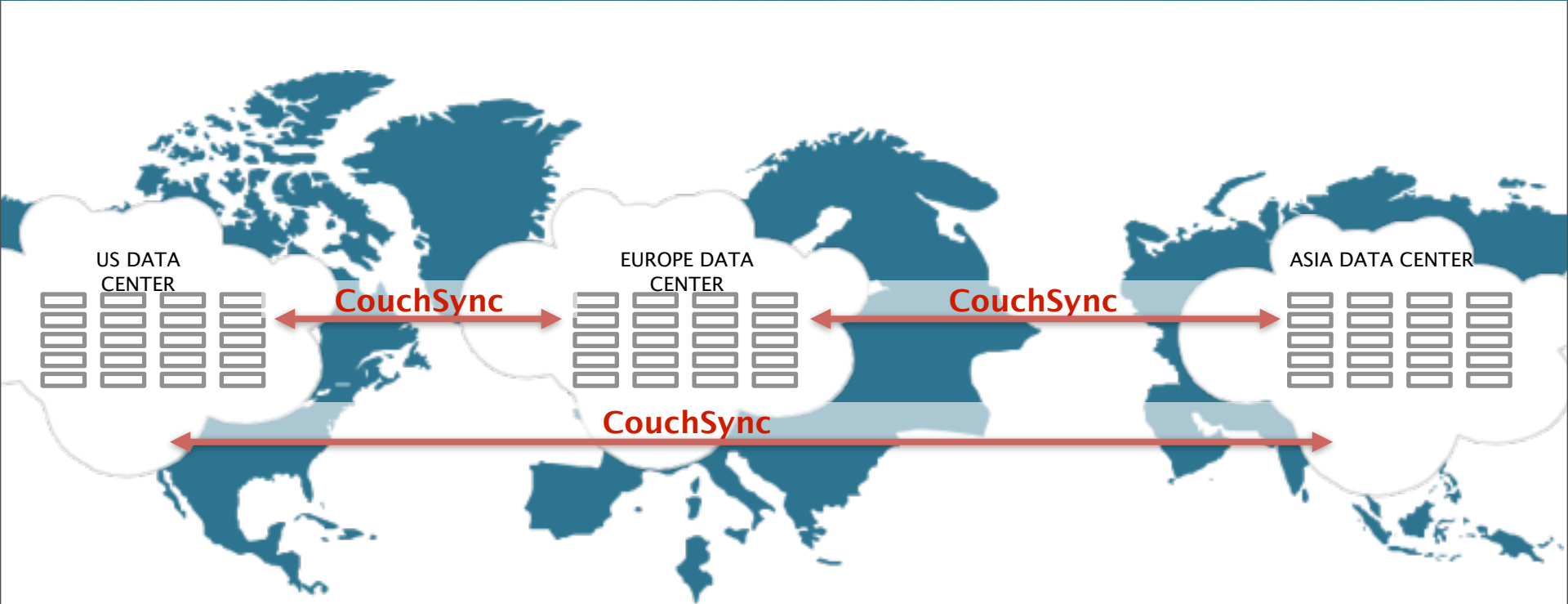
Fail Over Node



Fail Over Node



Cross Data Center Replication



- Want data close to user
- Want multiple locations for disaster recovery
- Can write to same document in all different regions & it will sync

Simple Monitoring and Operations

Couchbase Documentation • Support Forums • About • Sign Out

Cluster Overview Data Buckets Views Server Nodes Log Settings

default > Views > Development > [Level: By level]

SAMPLE DOCUMENT [Key: 14140] [Lookup Id] [Preview a Random Document]

```
{
  "id": "Keith4540",
  "rev": "1-a2354009ce69d198e555b493e05Tadce",
  "flags": 0,
  "expiration": 0,
  "docType": "Player",
  "docLevel": "2576-805-a958-4b83-8179-616816492del",
  "name": "Keith4540",
  "hitpoints": 75,
  "age": 1000,
  "level": 4,
  "loggedIn": false
}
```

VIEW CODE [Save]

```
Map
function (doc) {
  if (doc.docType == "Player") {
    emit([Level], doc.Level);
  }
}
```

Reduce (emit in _sum, _sum)

Filter Results 1descending=true&limit=10&skip=0

Development Time Subset Full Cluster Data Set

Key	Value
[Level, 6] Jens3499	"Jens3499"
[Level, 6] Chyoung5895	"Chyoung5895"
[Level, 5] Vokar5012	"Vokar5012"
[Level, 5] Vokar5063	"Vokar5063"
[Level, 5] Vokar5012	"Vokar5012"

Couchbase Documentation • Support Forums • About • Sign Out

Cluster Overview Data Buckets Views Server Nodes Log Settings

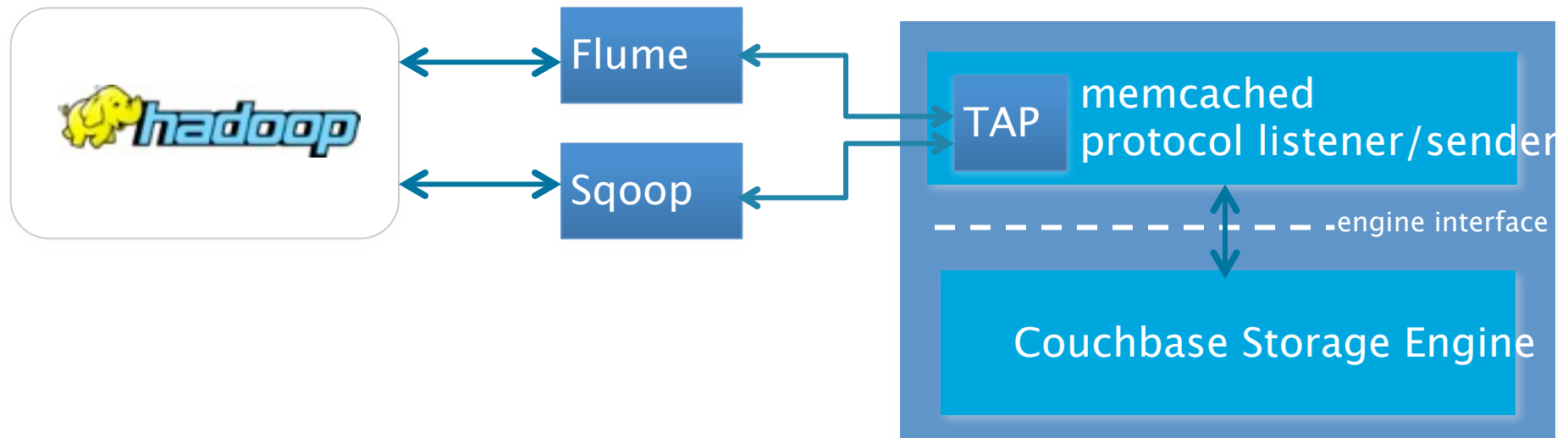
Servers

Active Servers Pending Rebalance [Rebalance] [Add Server]

Server Node Name	RAM Usage	Swap Usage	CPU Usage	Items (Active / Replicat)	Actions
10.32.81.250	43.1%	0%	0.0%	790 / 0	[Fail Over] [Remove]
10.32.136.119	43.3%	0%	1.03%	791 / 0	[Fail Over] [Remove]
10.34.47.57	43.1%	0%	0.98%	791 / 0	[Fail Over] [Remove]
10.38.7.194	43.1%	0%	0.01%	791 / 0	[Fail Over] [Remove]
10.38.11.24	43.8%	0%	0.98%	791 / 0	[Fail Over] [Remove]
10.38.26.191	43.1%	0%	0.02%	791 / 0	[Fail Over] [Remove]
10.38.33.228	43.2%	0%	1.99%	791 / 0	[Fail Over] [Remove]
10.38.34.55	43.3%	0%	0.72%	791 / 0	[Fail Over] [Remove]
10.76.255.249	43.7%	0%	0.29%	791 / 0	[Fail Over] [Remove]
10.83.105.138	43.4%	0%	1.21%	790 / 0	[Fail Over] [Remove]
10.85.73.241	43.3%	0%	0.0%	791 / 0	[Fail Over] [Remove]
10.99.18.238	43.0%	0%	0.0%	790 / 0	[Fail Over] [Remove]
10.112.223.206	46.8%	0%	1%	888 / 0	[Fail Over] [Remove]

Couchbase and Hadoop Integration

- Support large-scale analytics on application data by streaming data from Couchbase to Hadoop
 - Real-time integration using Flume
 - Batch integration using Sqoop
- Examples
 - Various game statistics (e.g., monthly / daily / hourly rankings)
 - Analyze game patterns from users to enhance various game metrics



Couchbase Mobile

- Couchbase on Mobile devices
- Data available offline
- Real-time multi- device synchroni
 - Including to the Cloud
- iOS and Android support
 - SDKs provided (CouchCocoa, Ektorp)



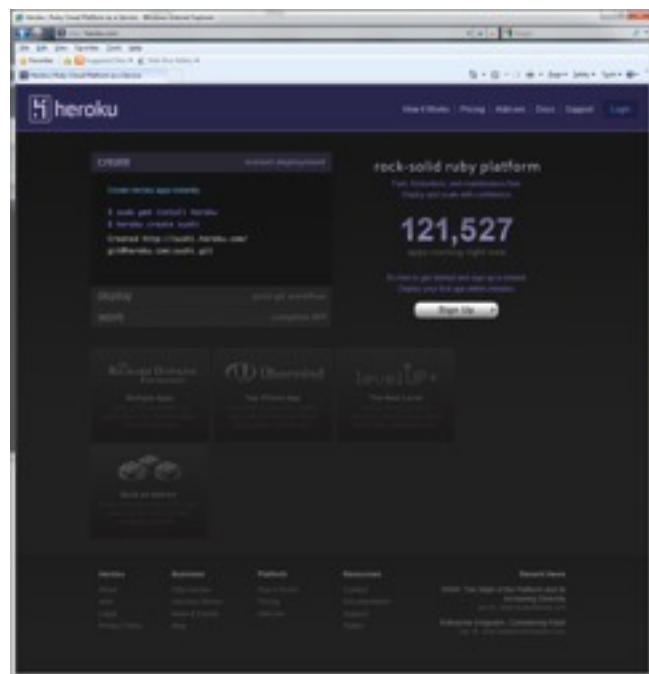
Couchbase
Mobile

Check it out today at: [http://
www.couchbase.org/get/couchbase-mobile-
for-ios/current](http://www.couchbase.org/get/couchbase-mobile-for-ios/current)

COUCHBASE USE CASES



Proven at small, and extra large scale



- Leading cloud service (PAAS) provider
- Over 65,000 hosted applications
- **Membase Server** serving over



- Social game leader – FarmVille, Mafia Wars, Café World
- Over 230 million monthly users
- **Membase Server** is the primary database behind key Zynga properties

Ad and offer targeting



40 milliseconds to respond with the decision.



3 profiles, real time campaign statistics



1
events

2
profiles, campaigns

Social and online gaming



1



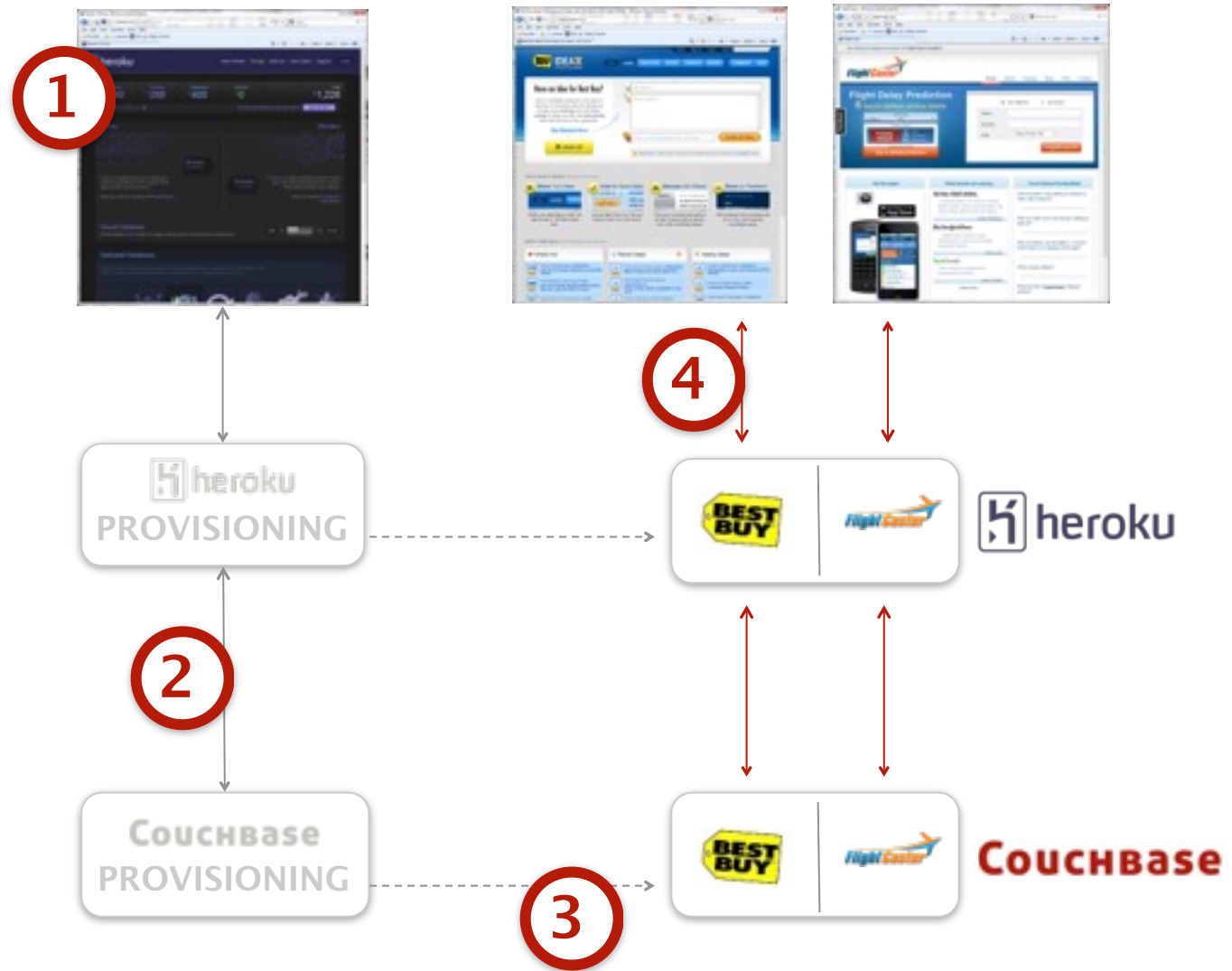
2



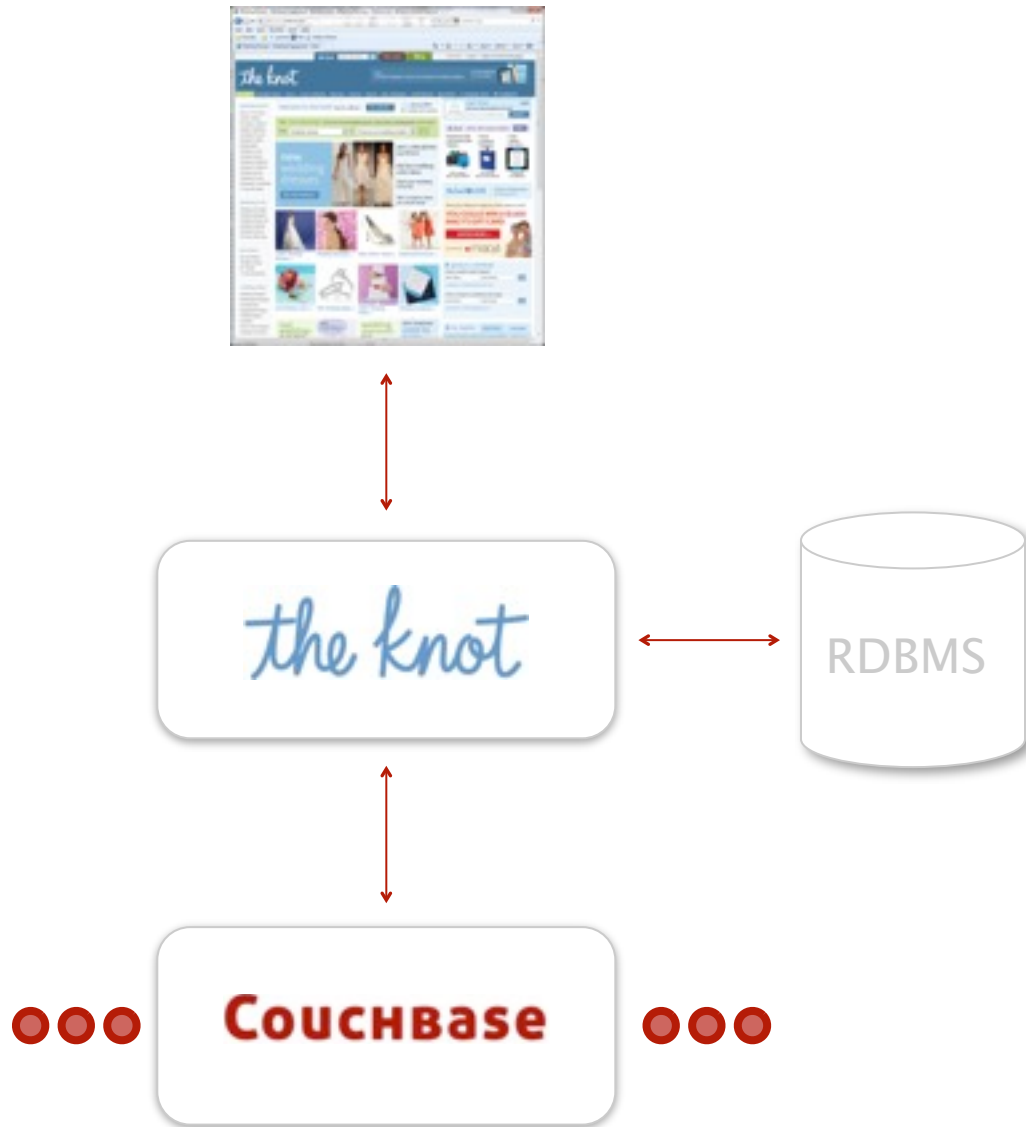
3



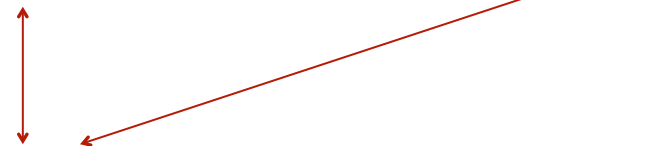
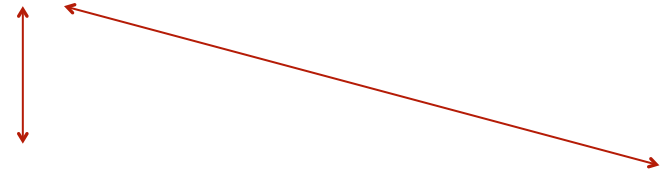
Cloud data services



SQL query caching



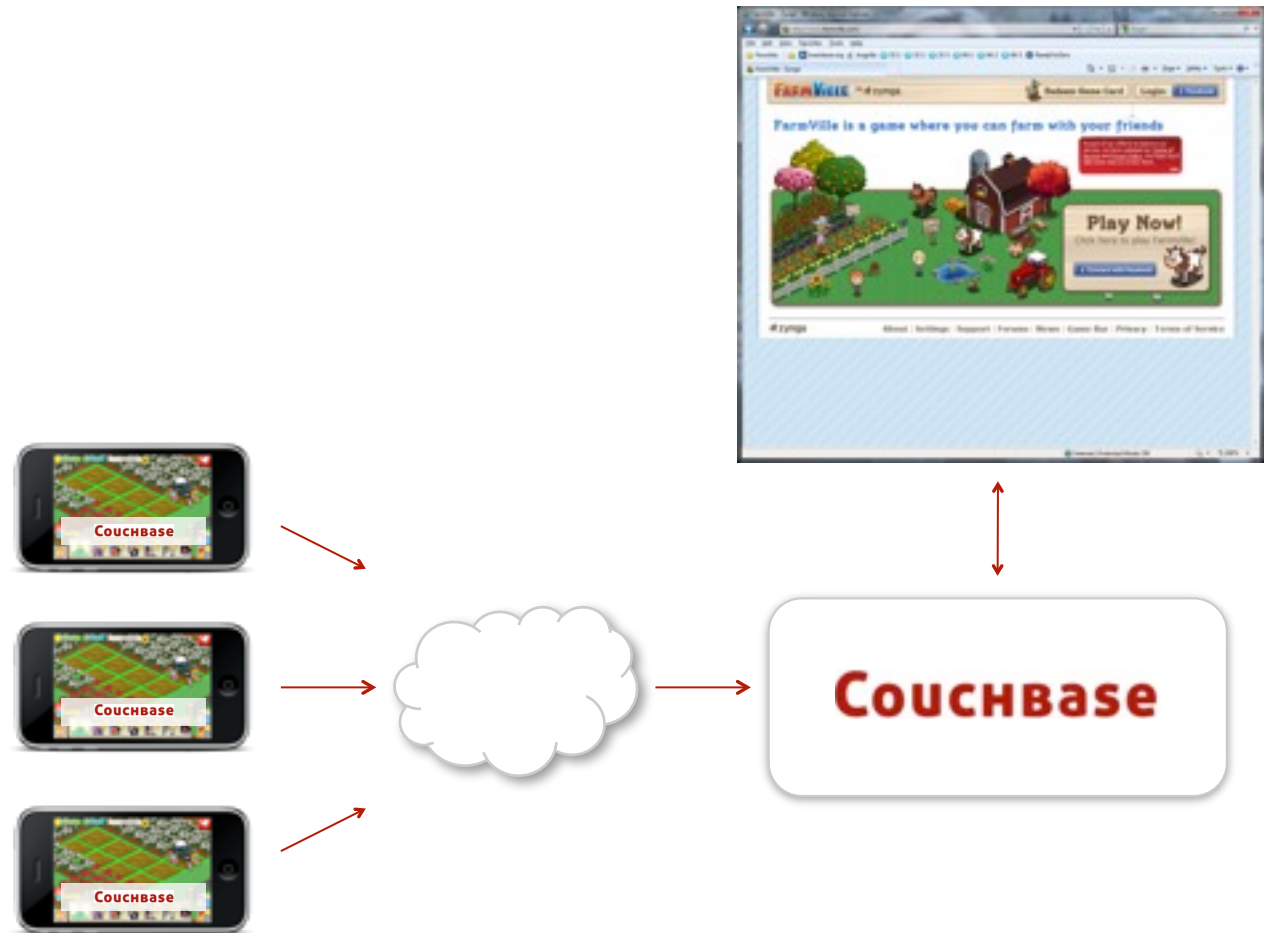
Session state storage



Couchbase



Mobile to cloud data synchronization



group.complete

DEMO TIME



Demo: The next big social game

Demo will show:

- Multiple Couchbase servers under moderate load (10k–15k operations per second)
- Dynamic scalability – adding nodes on the fly
- Secondary indices – Generation, querying, updating

Take-away?

Demo: The next big social game

3 Objects (documents) within game:

- Players
- Monsters
- Items

Gameplay:

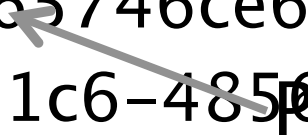
- **Players fight monsters**
- **Monsters drop items**
- **Players own items**

Player Document

```
{
  "_id": "Keith4540",
  "_rev": "1-ab354009ce09f198c555b693e057adce",
  "jsonType": "player",
  "uuid": "35767d02-a958-4b83-8179-616816692de1",
  "name": "Keith4540",
  "hitpoints": 75,
  "experience": 663,
  "level": 4,
  "loggedIn": false
}
```

Item Document

```
{
  "_id":
  "Katana_e5890c94-11c6-48-65746ce6c560",
  "_rev": "1-
d6bbd5e814c32c66e22db2918a2efcd9",
  "jsonType": "item",
  "name":
  "Katana_e5890c94-11c6-65746ce6c560",
  "uuid": "e5890c94-11c6-4856-
a7a6-65746ce6c560",
  "ownerId": "Dale9887"
```



Monster Document

```
{  
  "_id": "Bauchan9932",  
  "_rev":  
"1-5c90be58be58134a0fc5e7db77dab5f2",  
  "jsonType": "monster",  
  "name": "Bauchan9932",  
  "uuid": "d10dfc1b-0412-4140-b4ec-  
affdbf2aa5ec",  
  "hitpoints": 370,  
  "experienceWhenKilled": 52,  
  "itemProbability": 0.5050581341872865
```

GAME ON!



THANK YOU!

Get Couchbase Server 2.0 at
<http://www.couchbase.com/downloads>

Give us feedback at:
<http://www.couchbase.org/forums>



