The No-Marketing Bullshit Introduction to Couchbase Server 2.0 and ext/couchbase

Jan Lehnardt
@janl / jan@couchbase.com

CouchBase

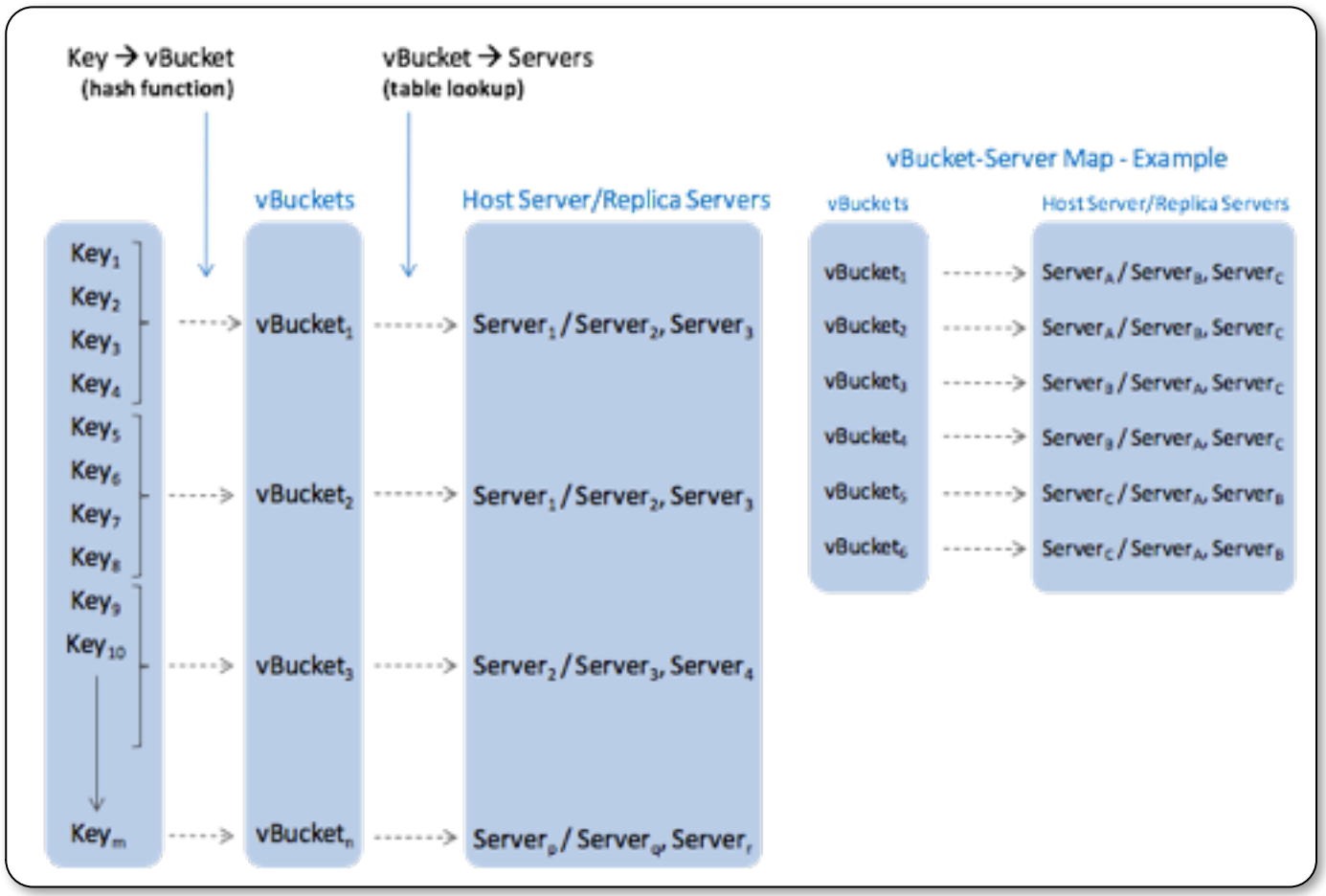
Couchbase Server Features

- Built-in clustering – All nodes equal
- Data replication with auto-failover

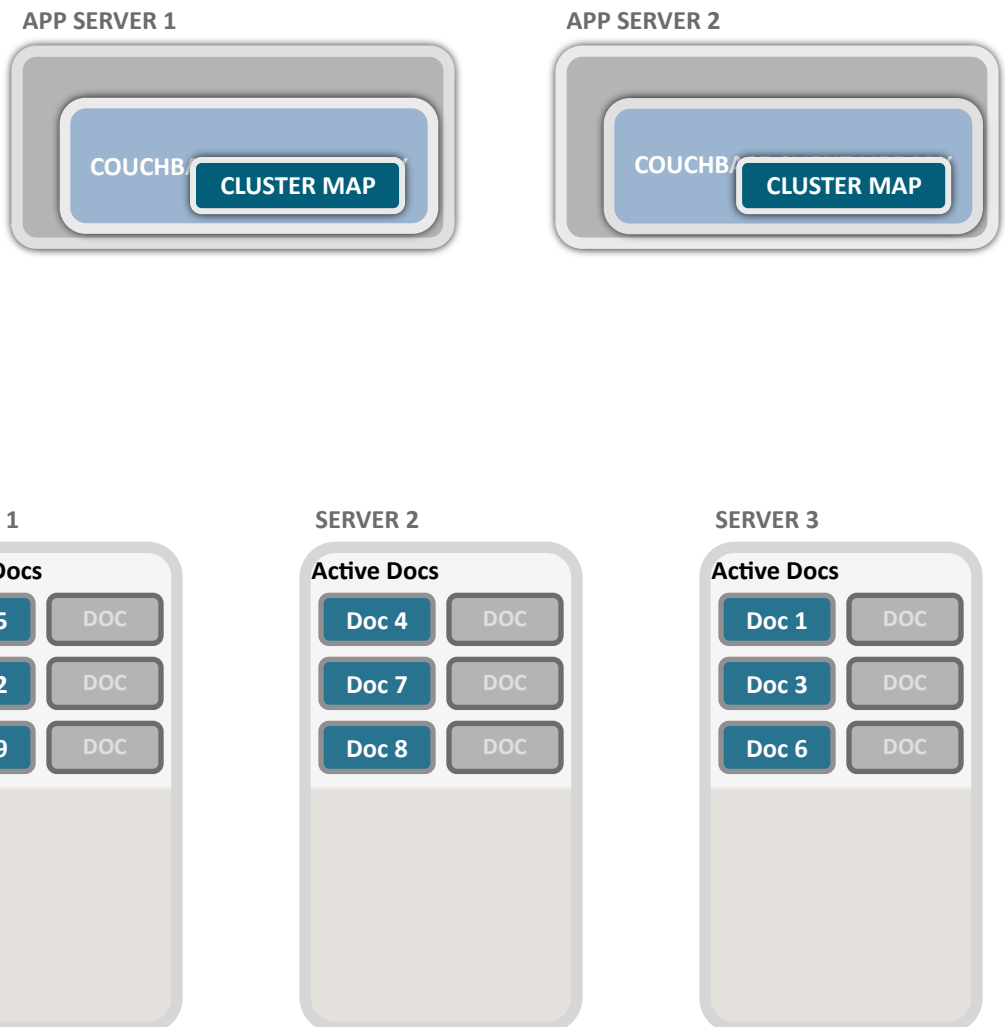
- Zero-downtime maintenance

- Clone to grow and scale horizontally

Auto-sharding: vBuckets

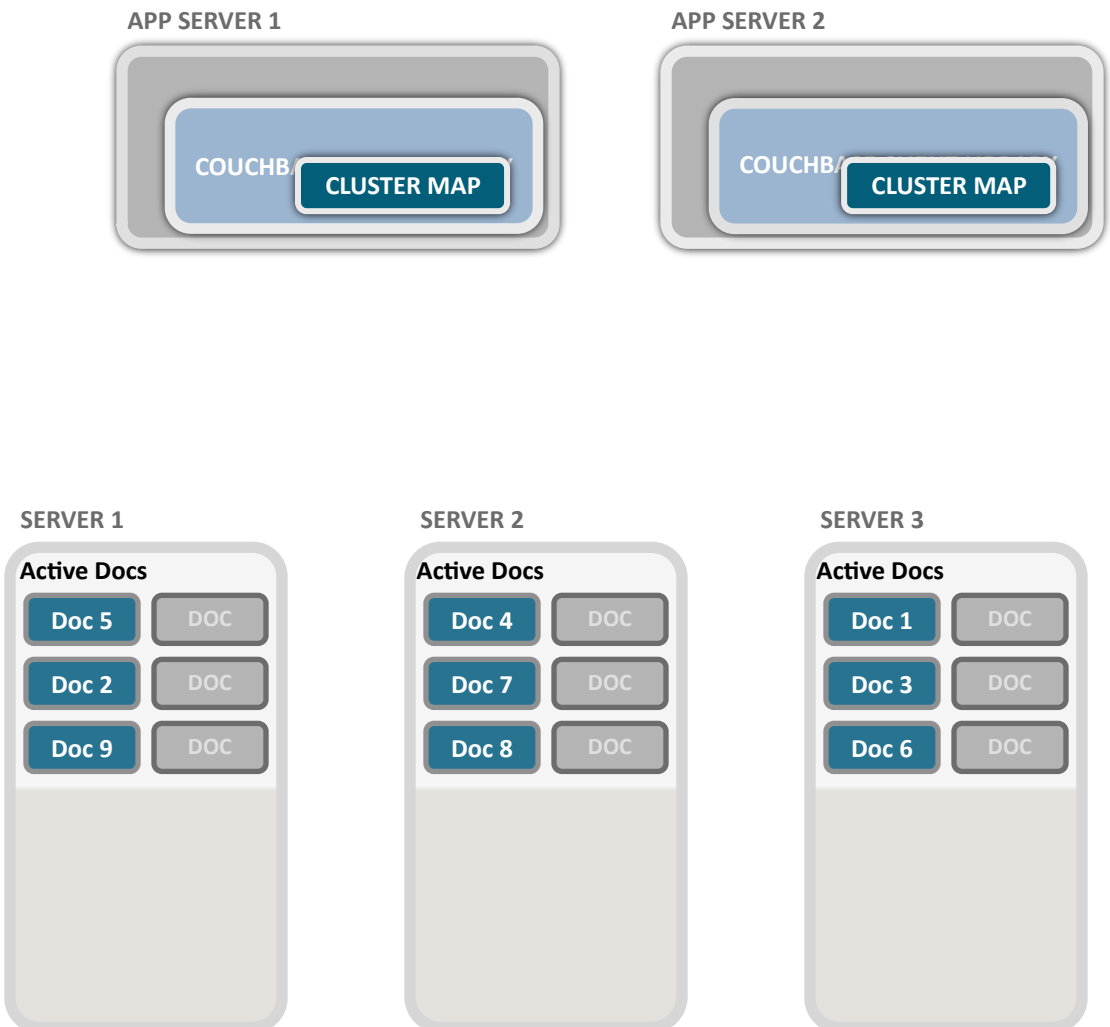


Couchbase Server Basic Operation



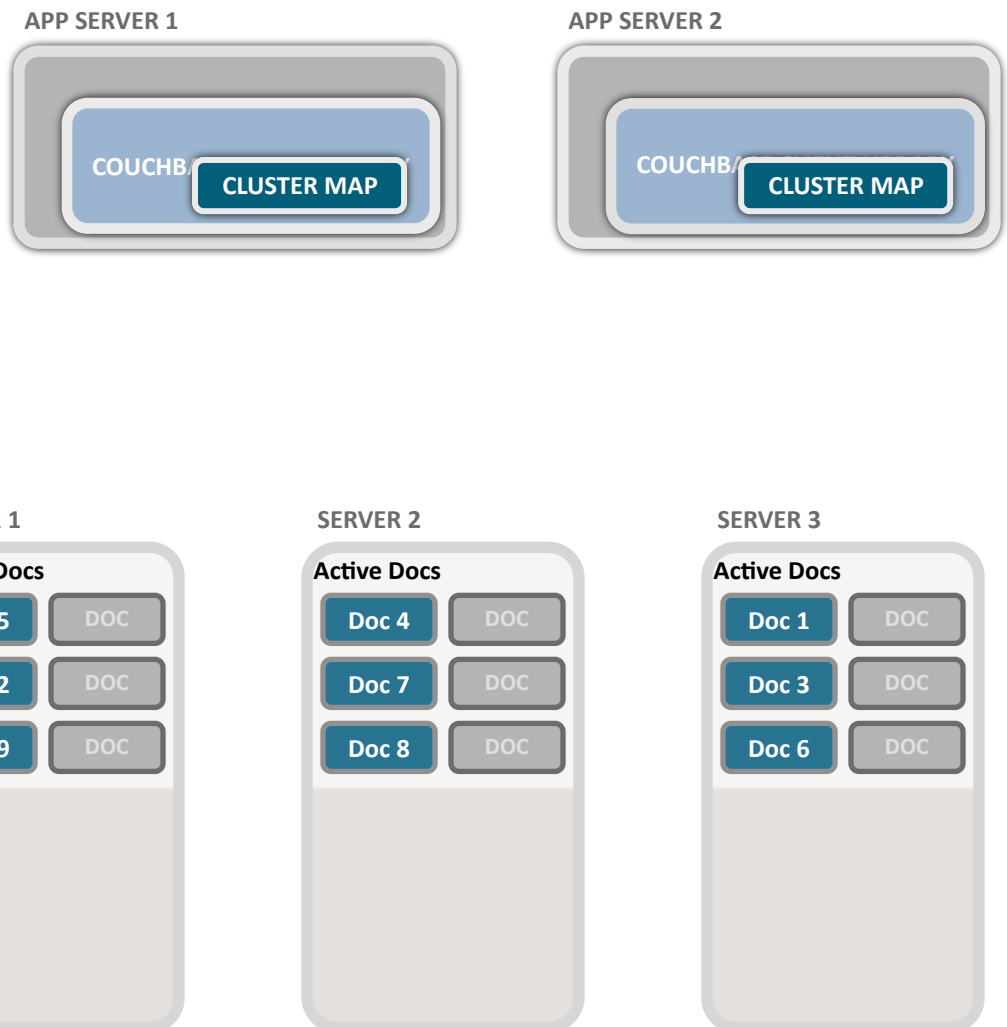
- Docs distributed evenly across servers in the cluster

Couchbase Server Basic Operation



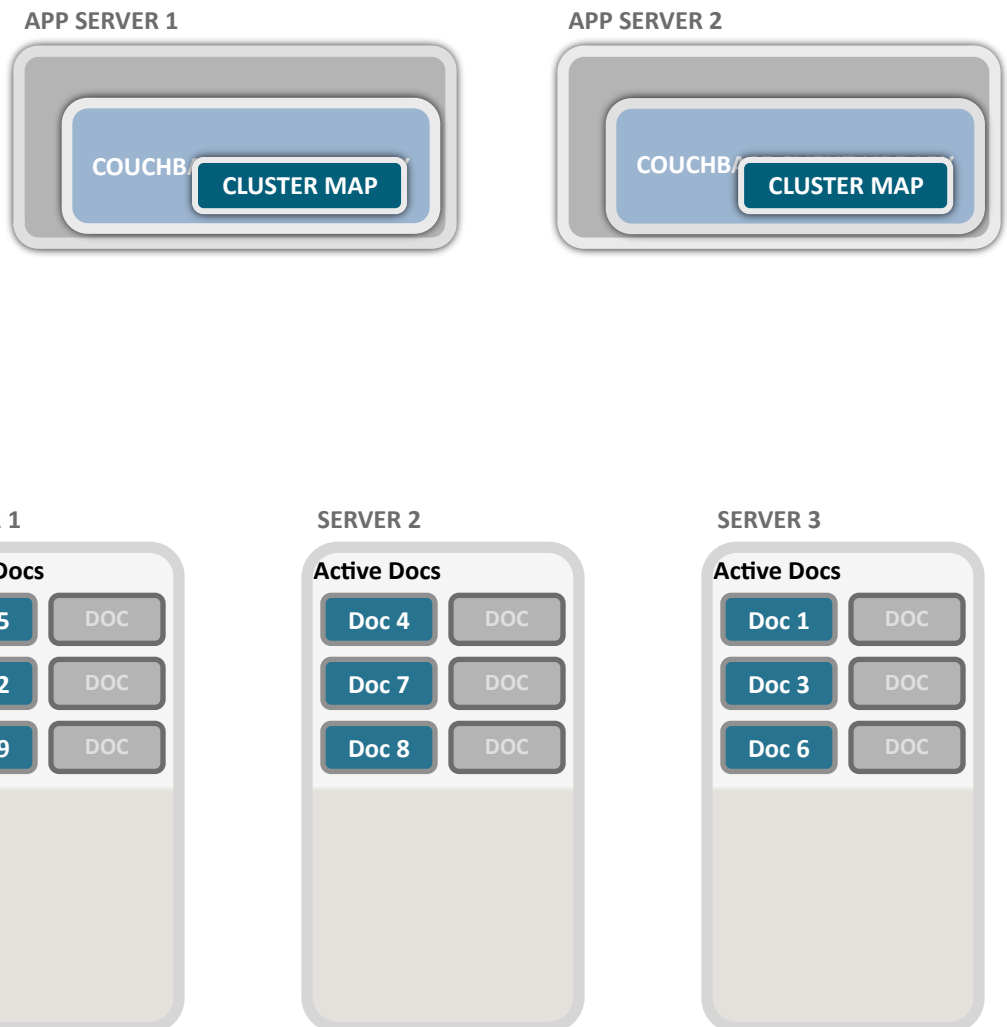
- Docs distributed evenly across servers in the cluster
- Each server stores both *active* & *replica* docs
  - Only one server active at a time

Couchbase Server Basic Operation



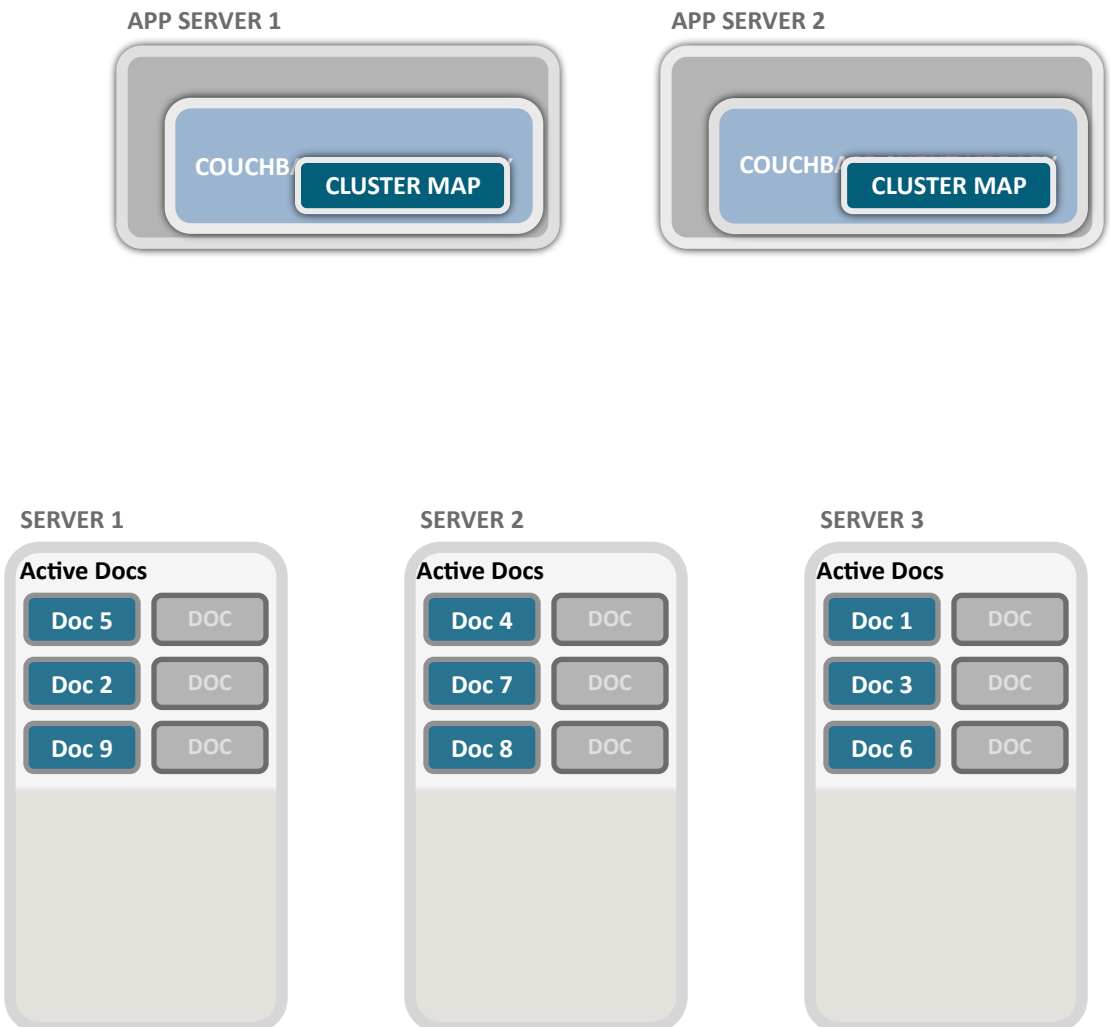
- Docs distributed evenly across servers in the cluster
- Each server stores both *active* & *replica* docs
  - Only one server active at a time
- Client library provides app with simple interface to database

Couchbase Server Basic Operation



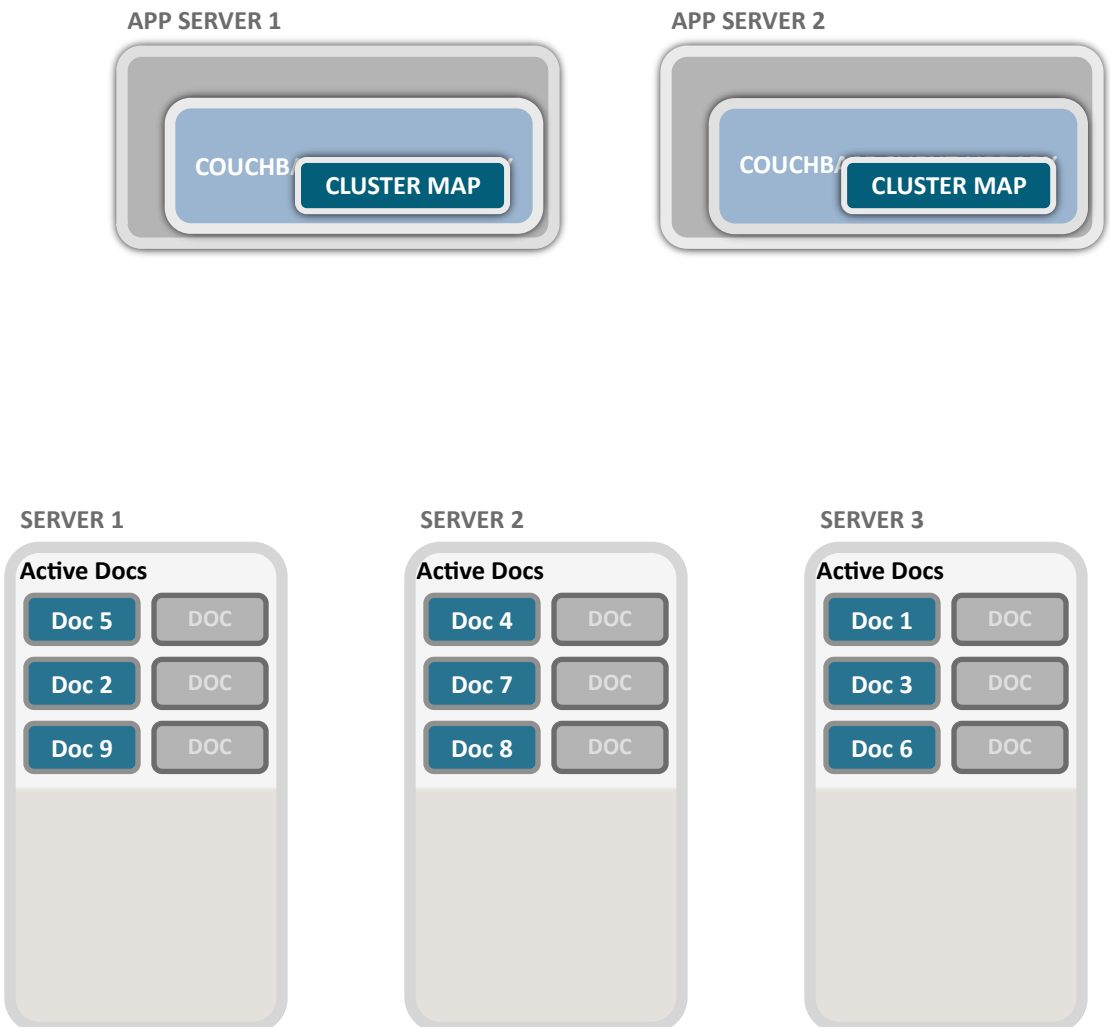
- Docs distributed evenly across servers in the cluster
- Each server stores both *active* & *replica* docs
  - Only one server active at a time
- Client library provides app with simple interface to database
- Cluster map provides map to which server doc is on
  - App never needs to know

Couchbase Server Basic Operation



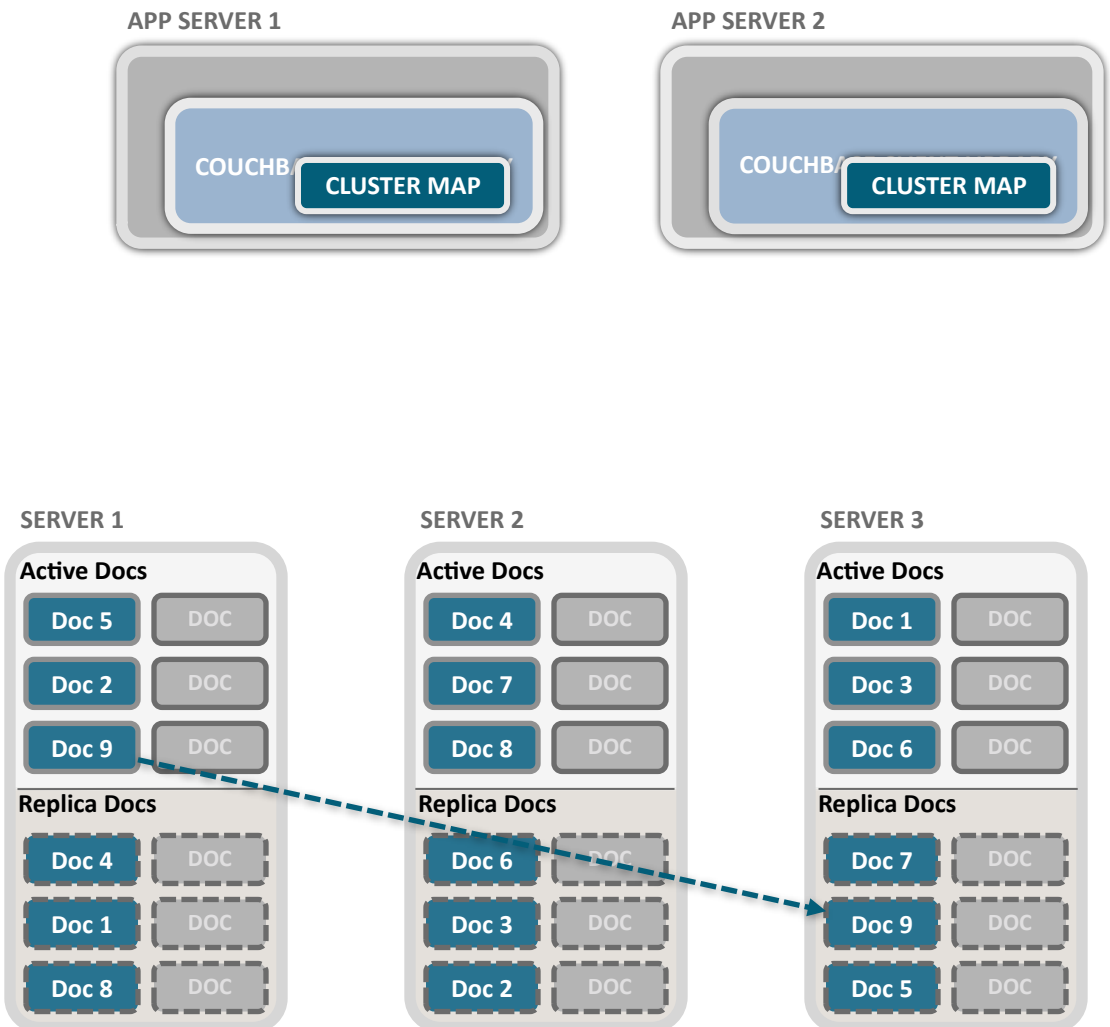
- Docs distributed evenly across servers in the cluster
- Each server stores both *active* & *replica* docs
  - Only one server active at a time
- Client library provides app with simple interface to database
- Cluster map provides map to which server doc is on
  - App never needs to know
- App reads,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

Couchbase Server Basic Operation



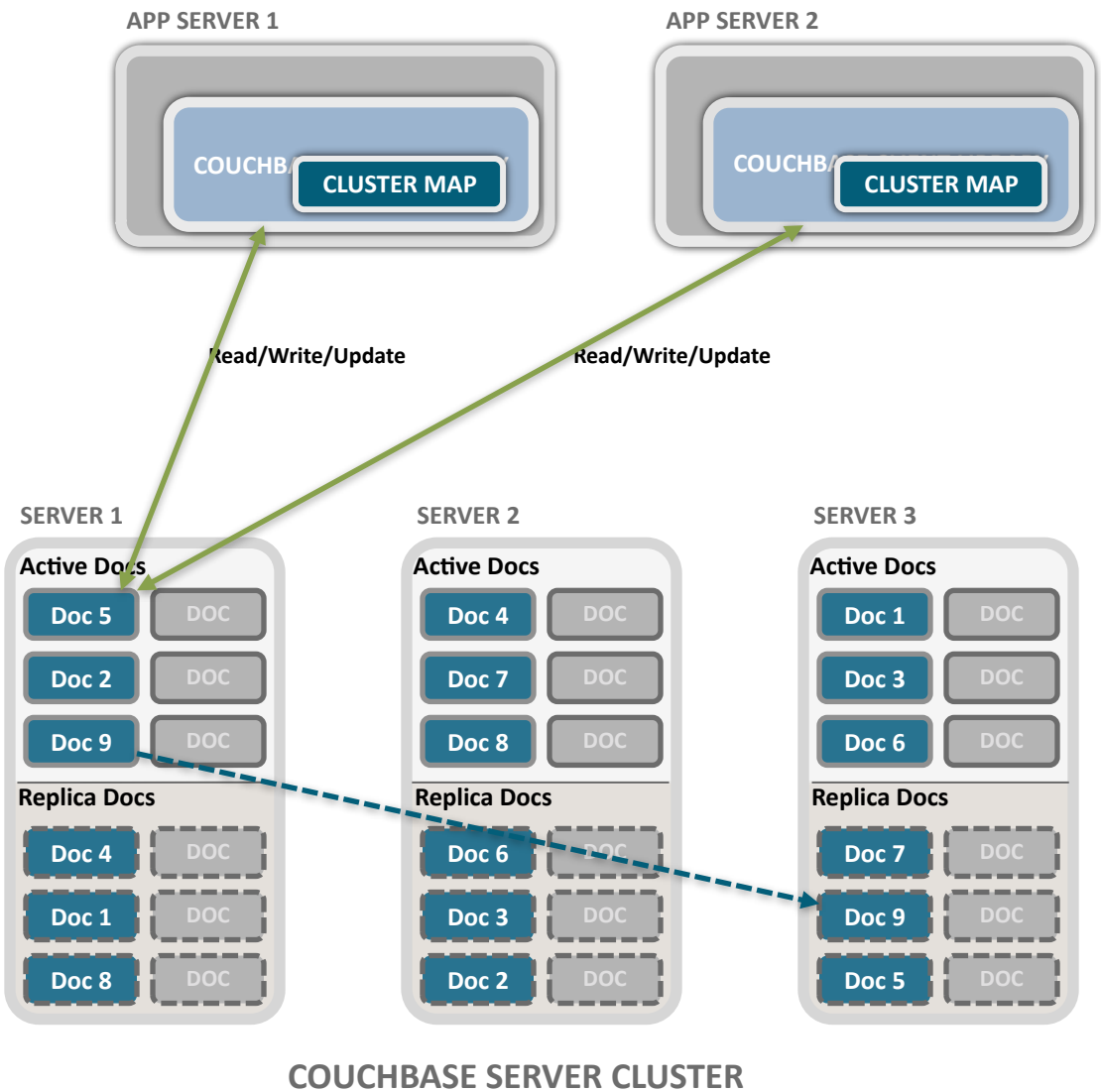
- Docs distributed evenly across servers in the cluster
- Each server stores both *active* & *replica* docs
  - Only one server active at a time
- Client library provides app with simple interface to database
- Cluster map provides map to which server doc is on
  - App never needs to know
- App reads,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

Couchbase Server Basic Operation



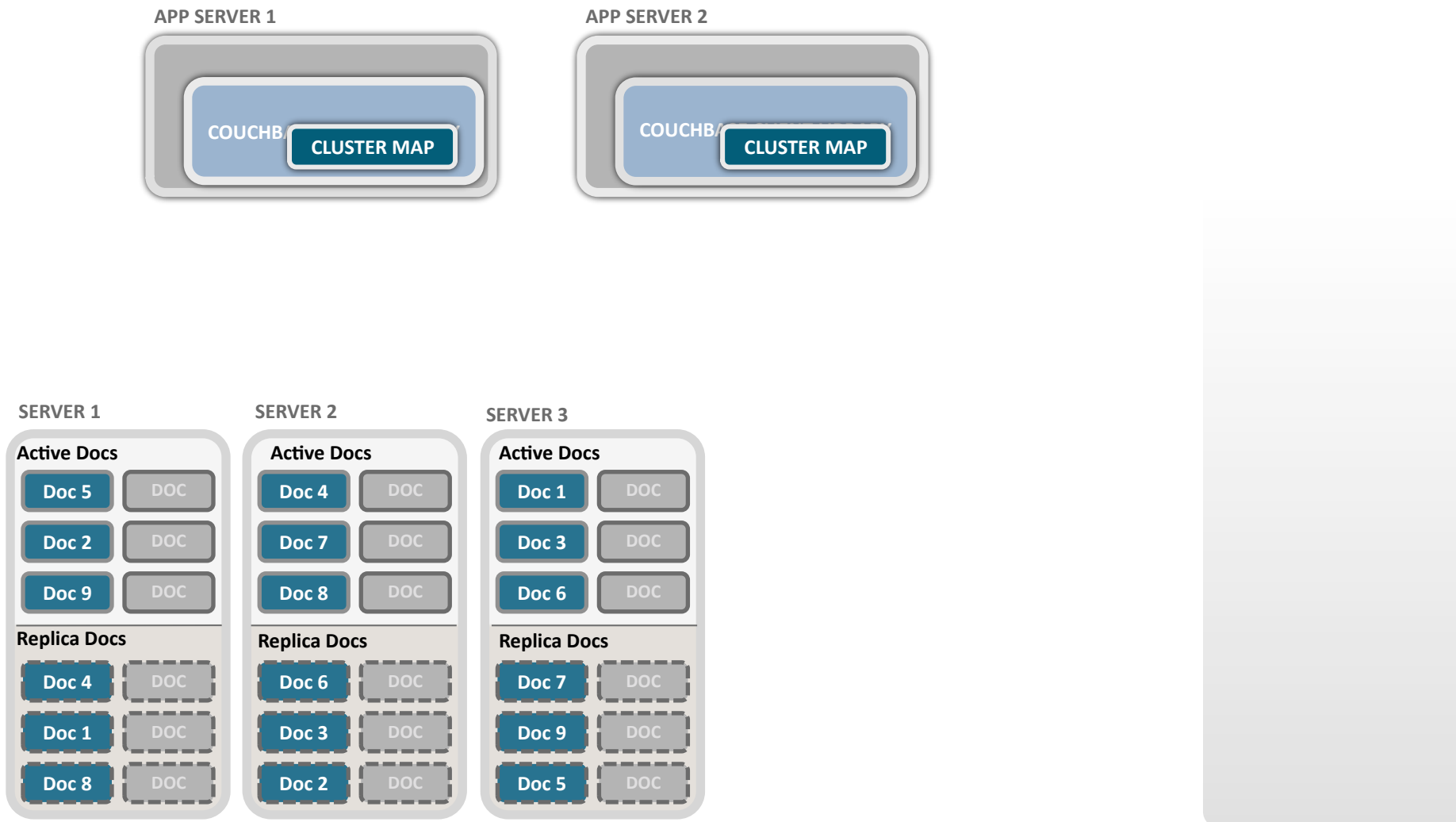
- Docs distributed evenly across servers in the cluster
- Each server stores both *active* & *replica* docs
  - Only one server active at a time
- Client library provides app with simple interface to database
- Cluster map provides map to which server doc is on
  - App never needs to know
- App reads,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

Couchbase Server Basic Operation

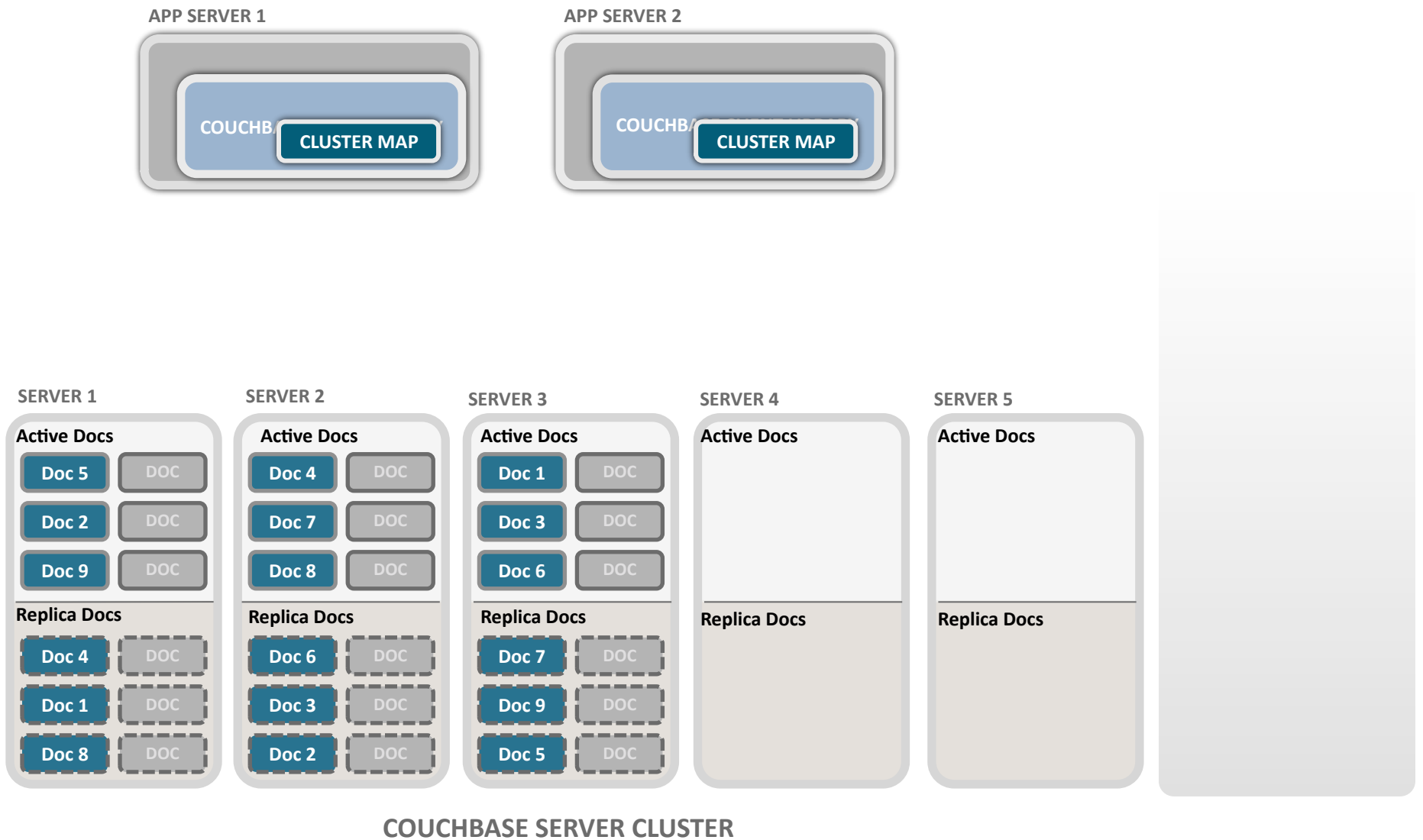


- Docs distributed evenly across servers in the cluster
- Each server stores both *active* & *replica* docs
  - Only one server active at a time
- Client library provides app with simple interface to database
- Cluster map provides map to which server doc is on
  - App never needs to know
- App reads,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

Add Nodes

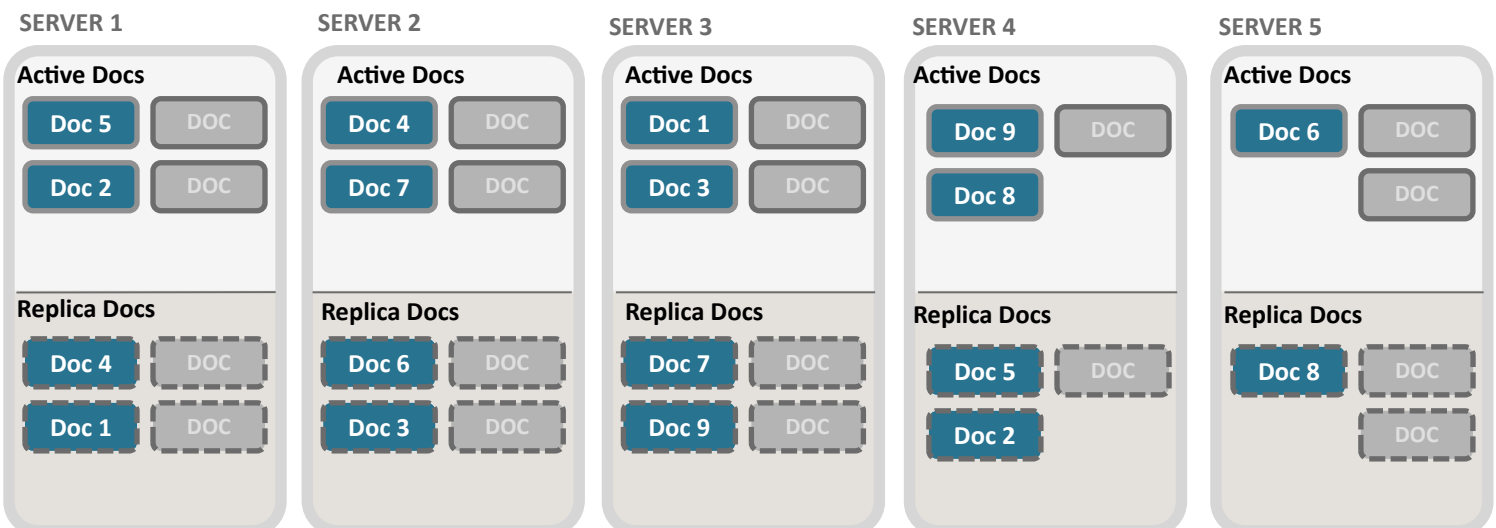

APP SERVER 1
COUCHB
CLUSTER MAP
APP SERVER 2
COUCHB
CLUSTER MAP
SERVER 1
Active Docs
Doc 5
DOC
Doc 2
DOC
Doc 9
DOC
Replica Docs
Doc 4
DOC
Doc 1
DOC
Doc 8
DOC
SERVER 2
Active Docs
Doc 4
DOC
Doc 7
DOC
Doc 8
DOC
Replica Docs
Doc 6
DOC
Doc 3
DOC
Doc 2
DOC
SERVER 3
Active Docs
Doc 1
DOC
Doc 3
DOC
Doc 6
DOC
Replica Docs
Doc 7
DOC
Doc 9
DOC
Doc 5
DOC
COUCHBASE SERVER CLUSTER

Add Nodes


APP SERVER 1
COUCHB
CLUSTER MAP
APP SERVER 2
COUCHB
CLUSTER MAP
SERVER 1
Active Docs
Doc 5
DOC
Doc 2
DOC
Doc 9
DOC
Replica Docs
Doc 4
DOC
Doc 1
DOC
Doc 8
DOC
SERVER 2
Active Docs
Doc 4
DOC
Doc 7
DOC
Doc 8
DOC
Replica Docs
Doc 6
DOC
Doc 3
DOC
Doc 2
DOC
SERVER 3
Active Docs
Doc 1
DOC
Doc 3
DOC
Doc 6
DOC
Replica Docs
Doc 7
DOC
Doc 9
DOC
Doc 5
DOC
SERVER 4
Active Docs
Replica Docs
SERVER 5
Active Docs
Replica Docs
COUCHBASE SERVER CLUSTER

Add Nodes

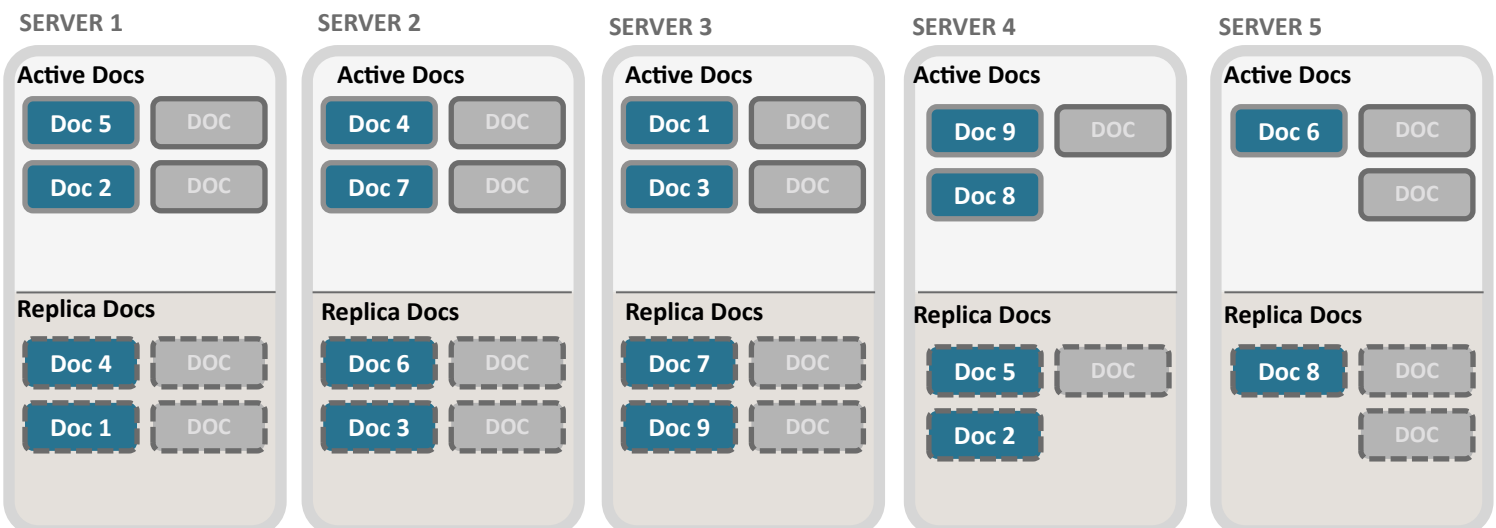
- Two servers added to cluster
  - One-click operation



COUCHBASE SERVER CLUSTER

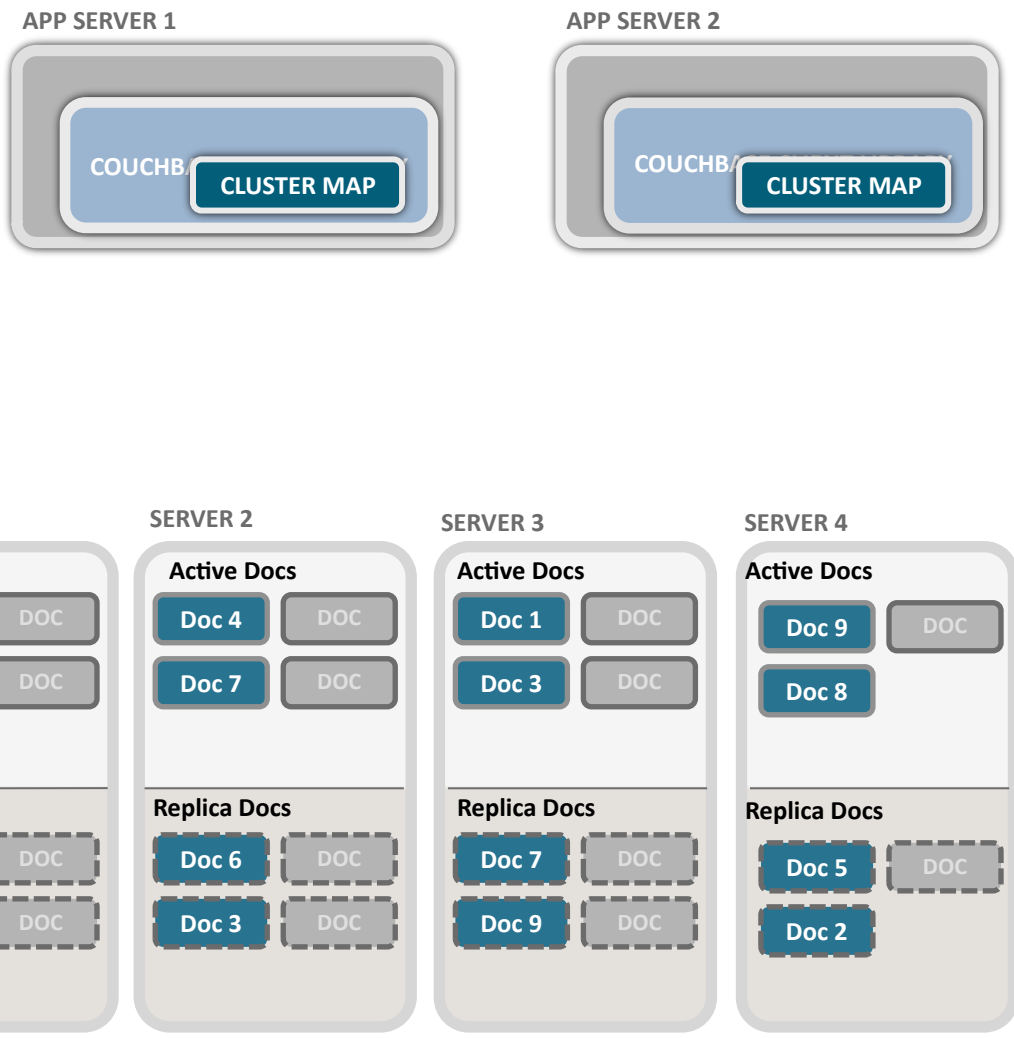
Add Nodes

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



COUCHBASE SERVER CLUSTER

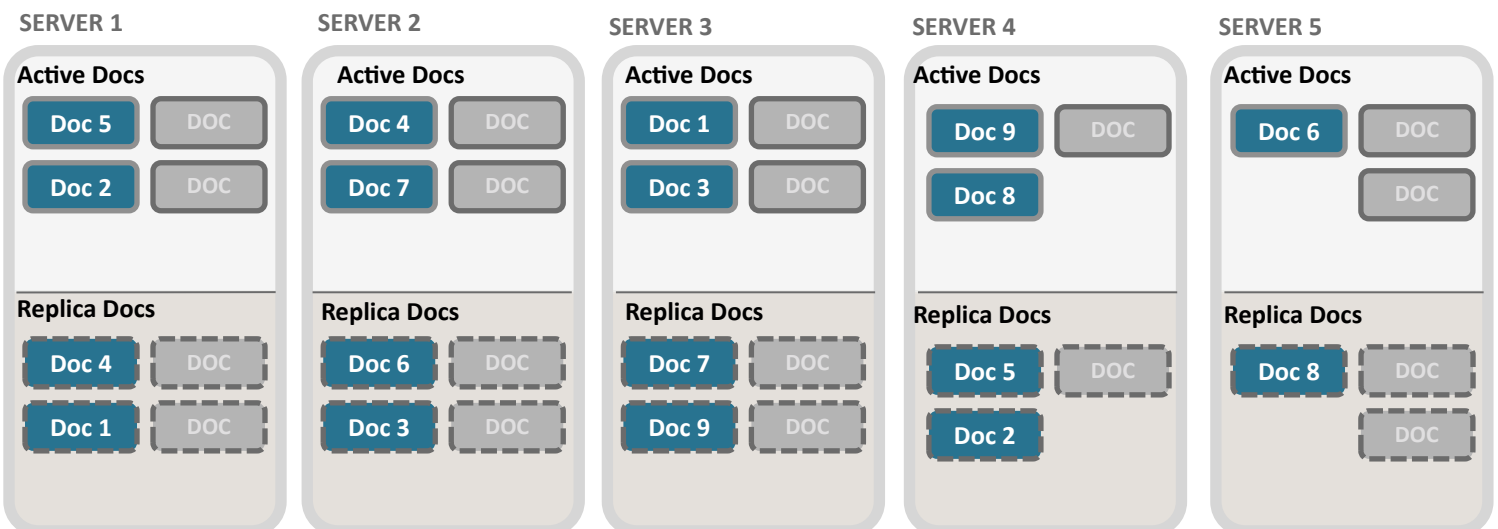
Add Nodes



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

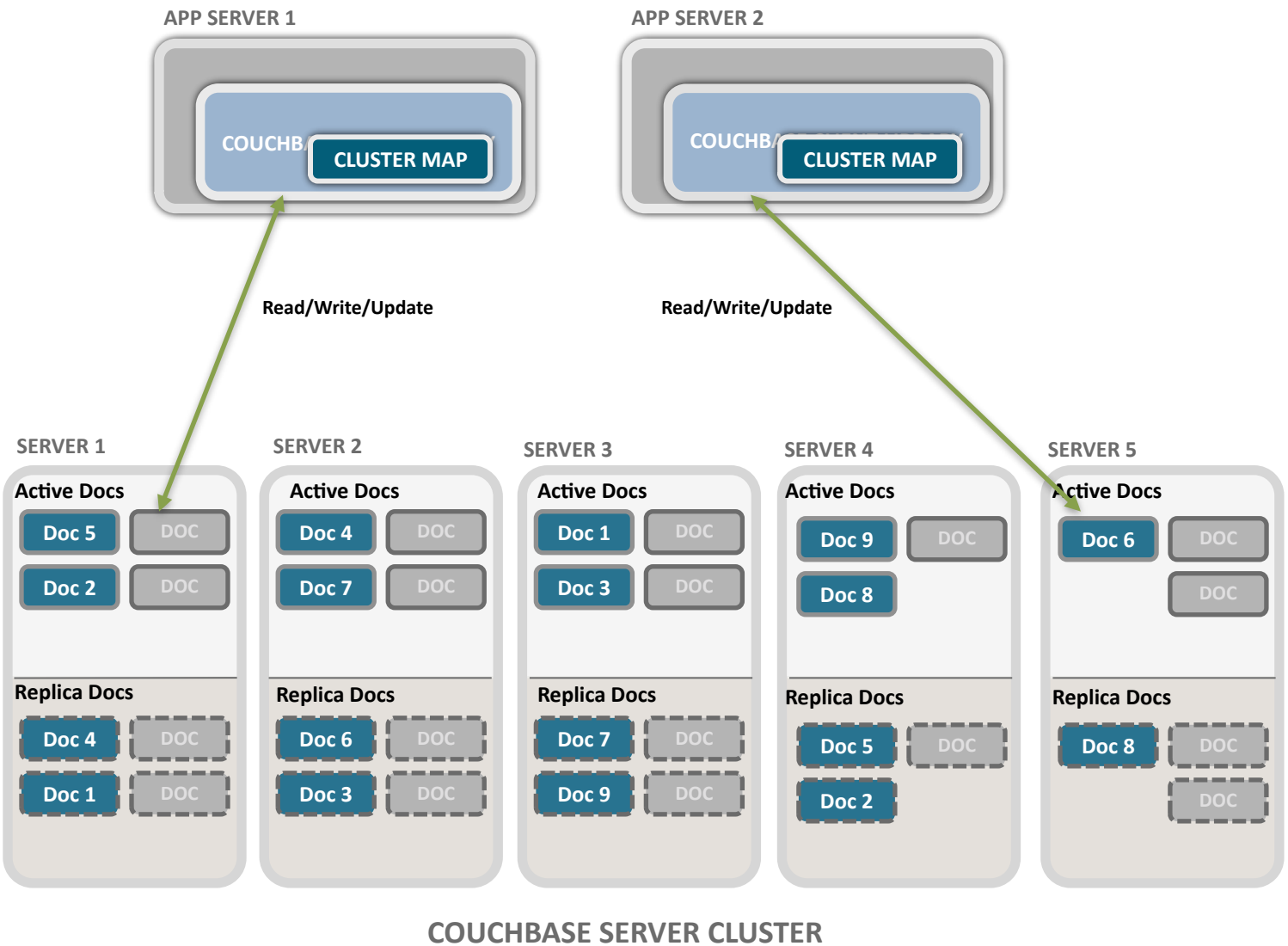
Add Nodes

- 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



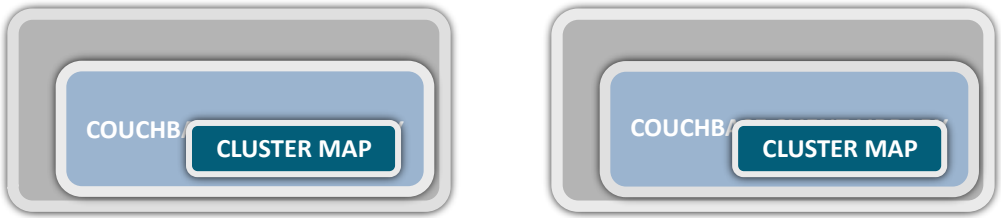
COUCHBASE SERVER CLUSTER

Add Nodes

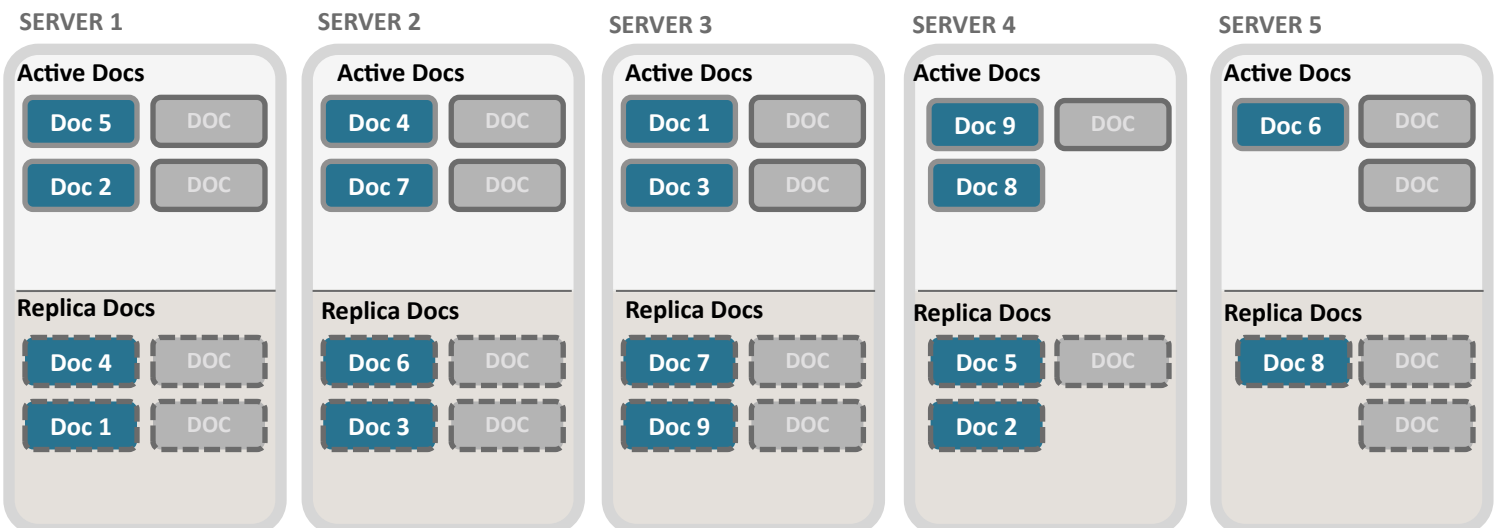


- Two servers added to cluster
  - One-click operation
- Docs automatically rebalanced across cluster
  - Even distribution of docs
  - Minimum doc movement
- Cluster map updated
- App database calls now distributed over larger # of servers

Fail Over Node

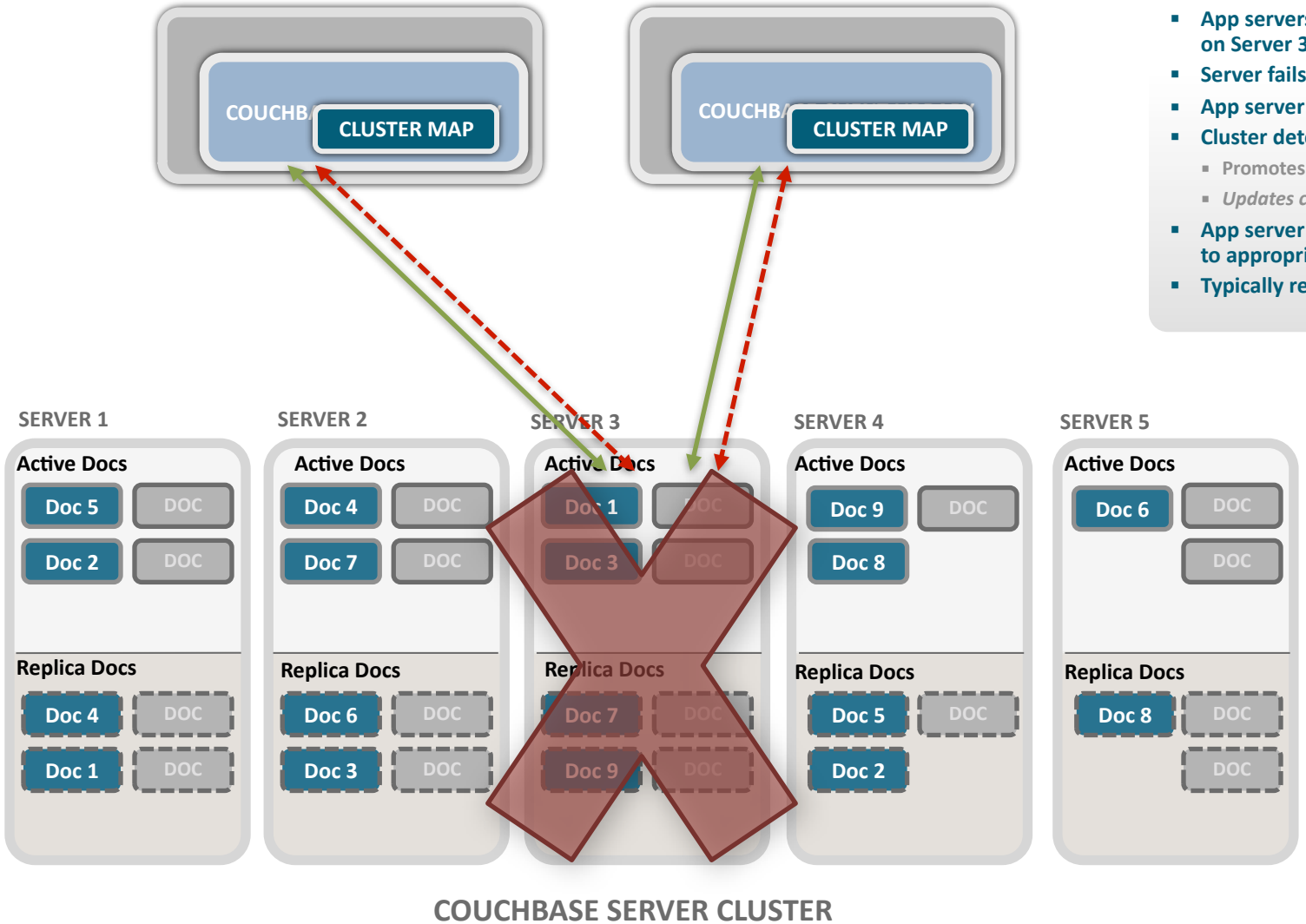


- App servers happily accessing docs on Server 3
- Server fails
- App server requests to server 3 fail
- Cluster detects server has failed
  - Promotes *replicas* of docs to *active*
  - *Updates cluster map*
- App server requests for docs now go to appropriate server
- Typically rebalance would follow



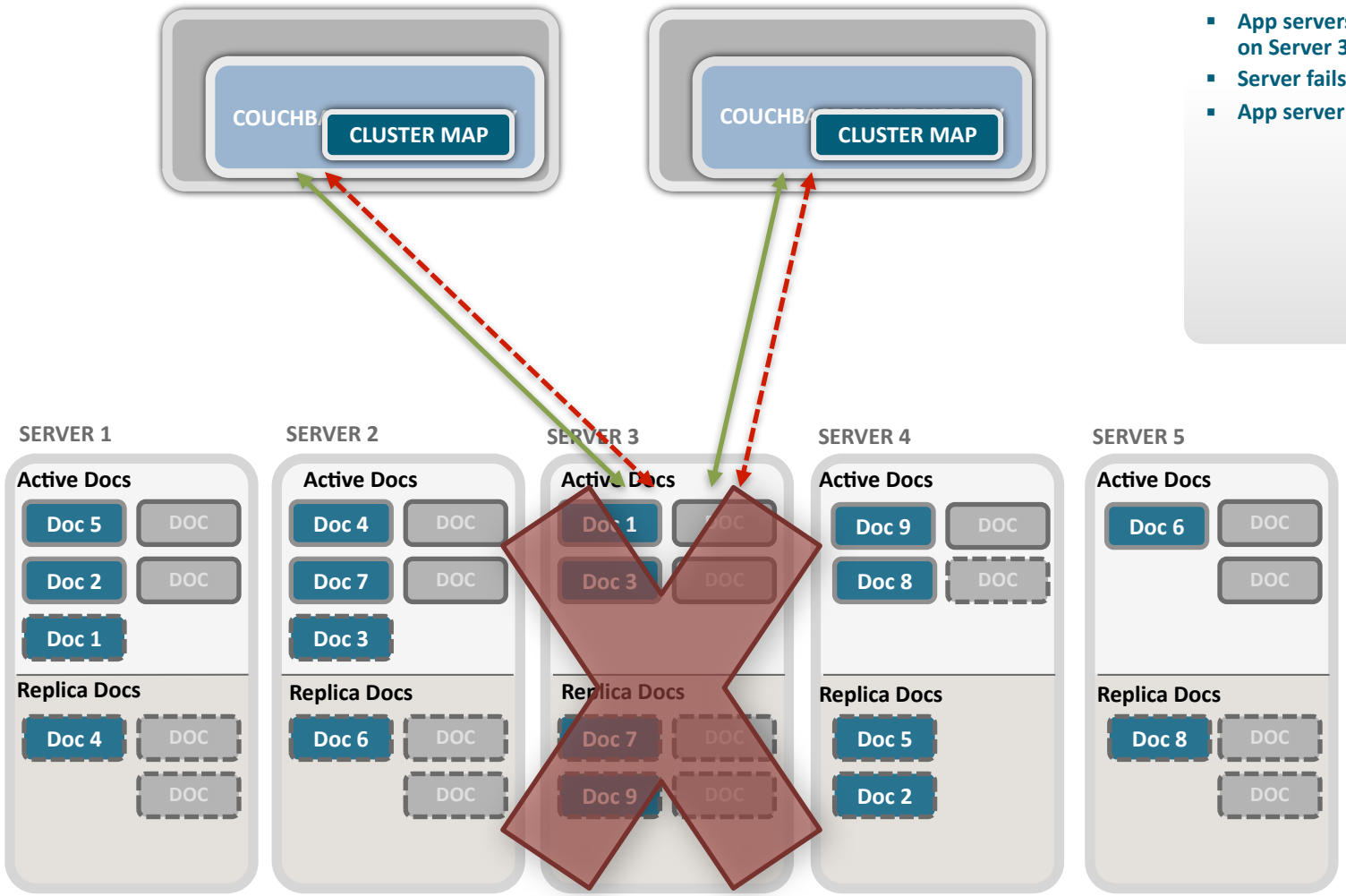
COUCHBASE SERVER CLUSTER

Fail Over Node



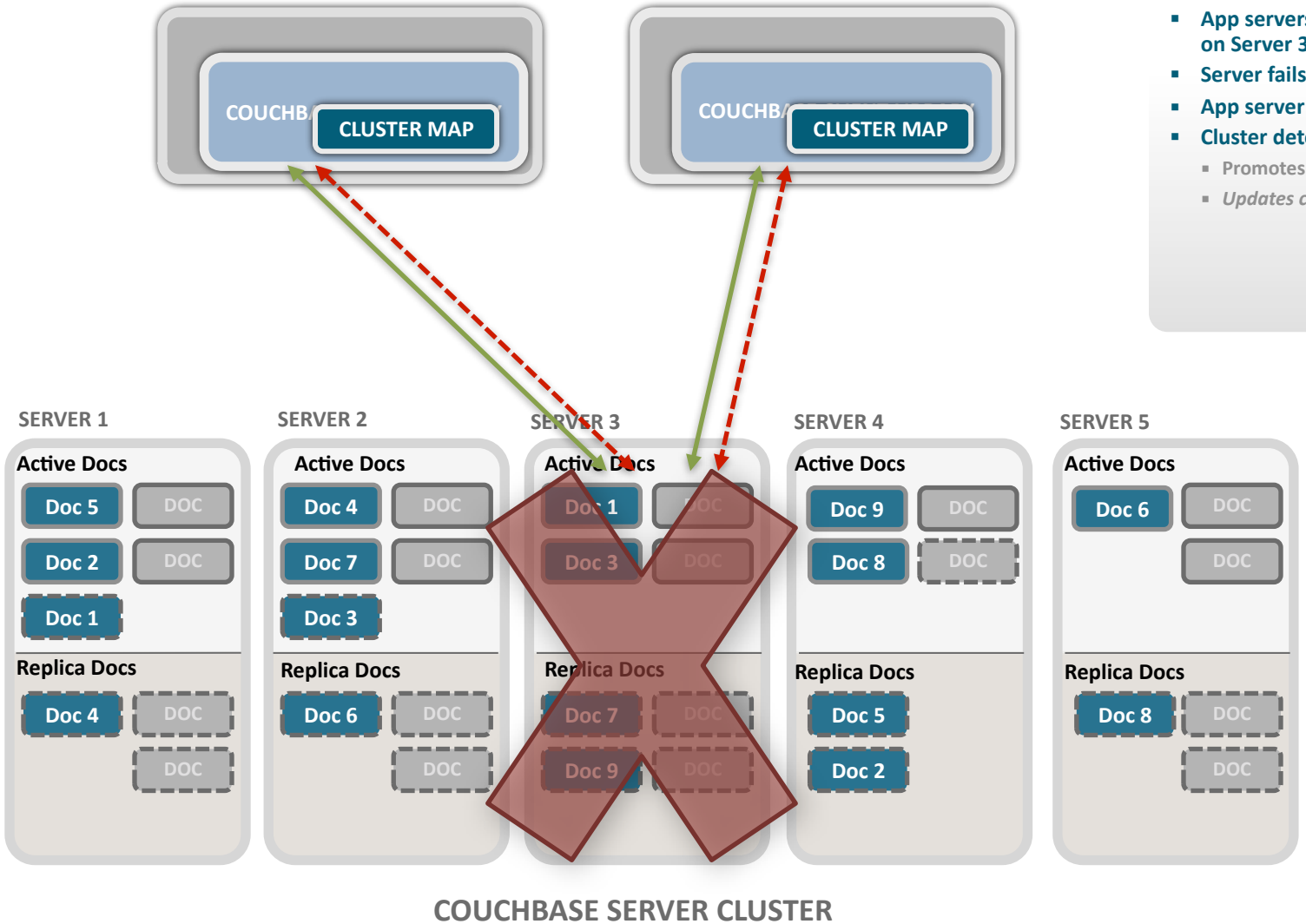
- App servers happily accessing docs on Server 3
- Server fails
- App server requests to server 3 fail
- Cluster detects server has failed
  - Promotes *replicas* of docs to *active*
  - *Updates cluster map*
- App server requests for docs now go to appropriate server
- Typically rebalance would follow

Fail Over Node



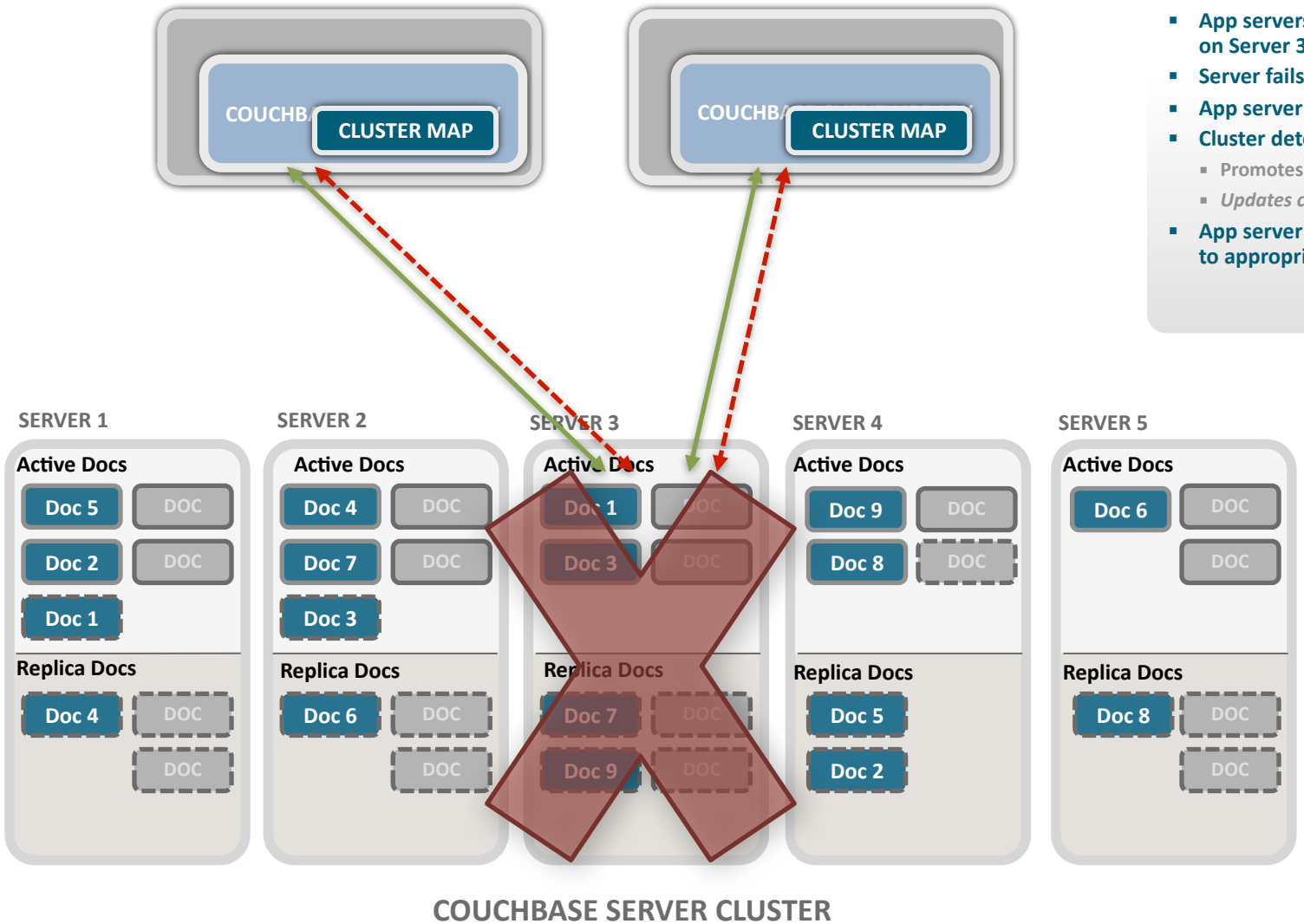
- App servers happily accessing docs on Server 3
- Server fails
- App server requests to server 3 fail

Fail Over Node



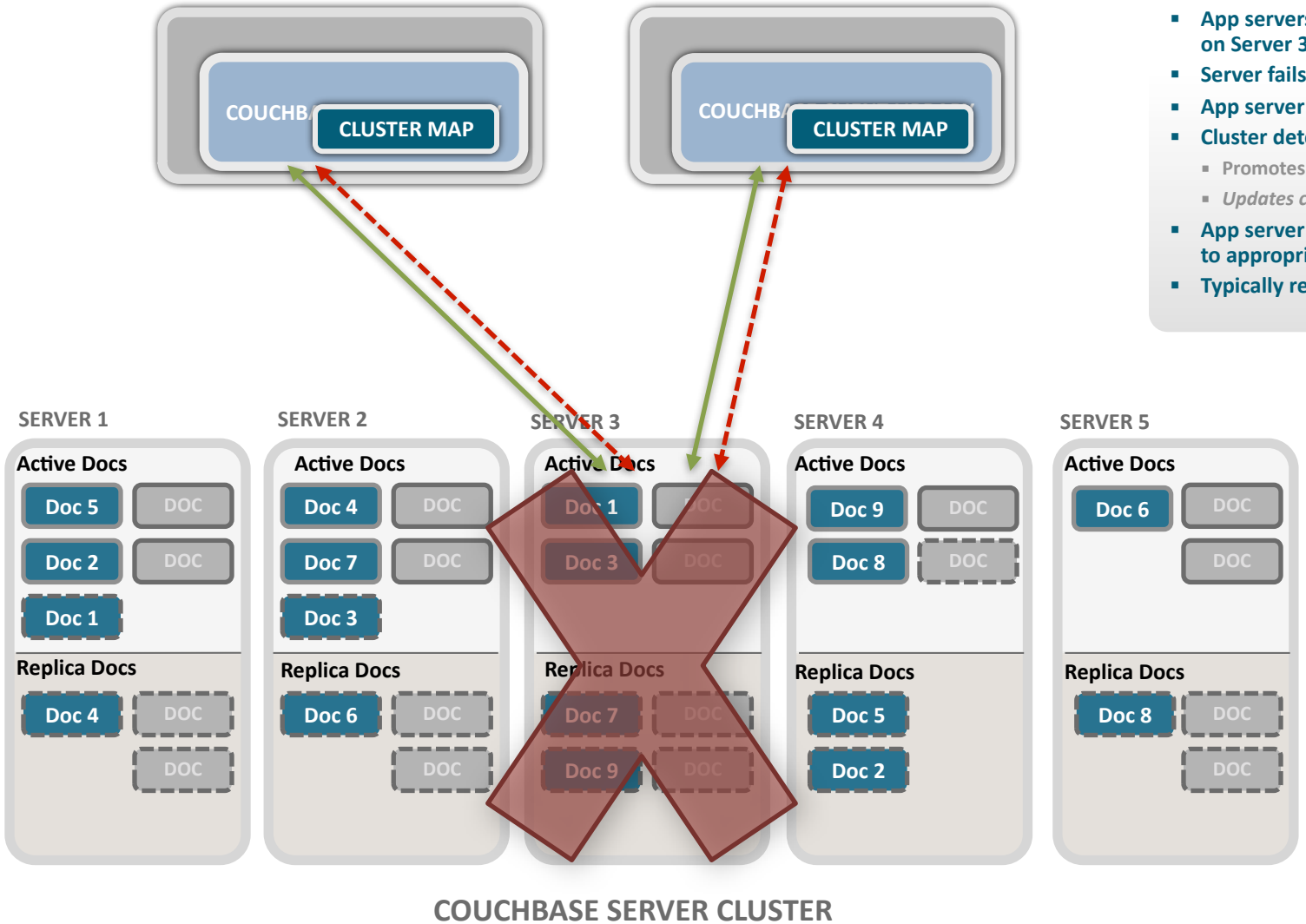
- App servers happily accessing docs on Server 3
- Server fails
- App server requests to server 3 fail
- Cluster detects server has failed
  - Promotes *replicas* of docs to *active*
  - *Updates cluster map*

Fail Over Node



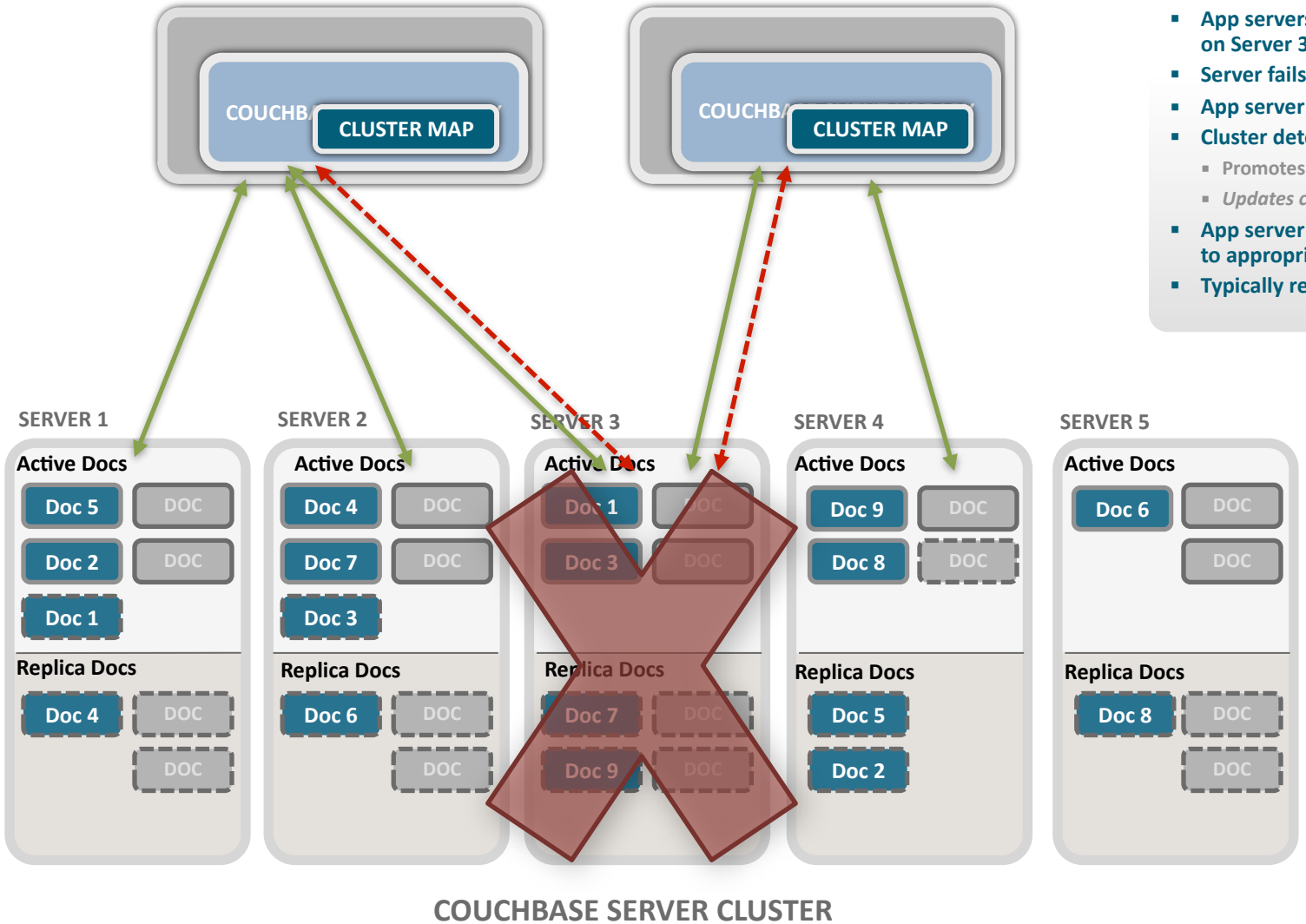
- App servers happily accessing docs on Server 3
- Server fails
- App server requests to server 3 fail
- Cluster detects server has failed
  - Promotes *replicas* of docs to *active*
  - *Updates cluster map*
- App server requests for docs now go to appropriate server

Fail Over Node



- 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

Fail Over Node



- 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

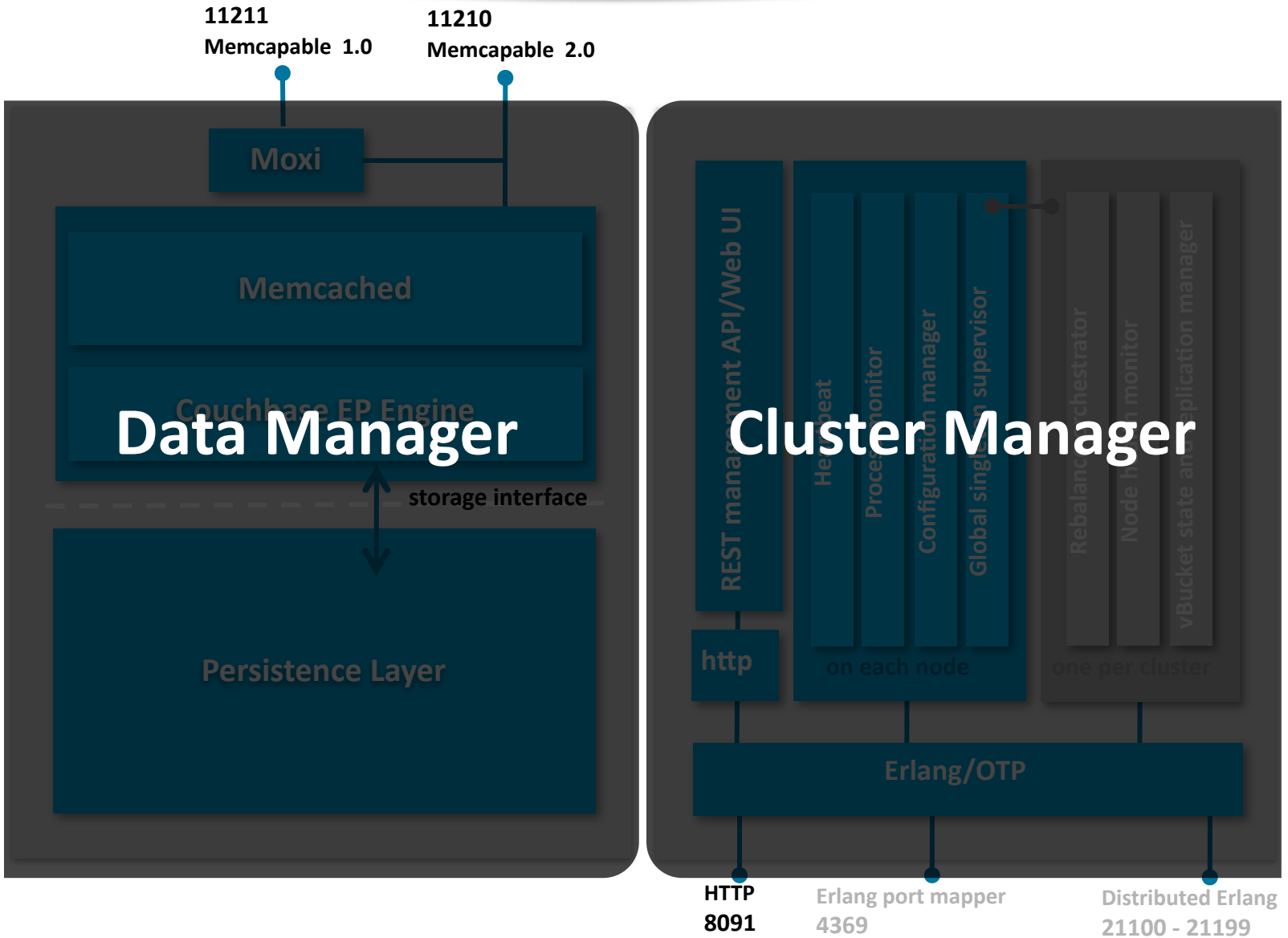
Couchbase Server Features

- Memcached compatible (built-in caching)

- Monitoring and administration APIs and GUI

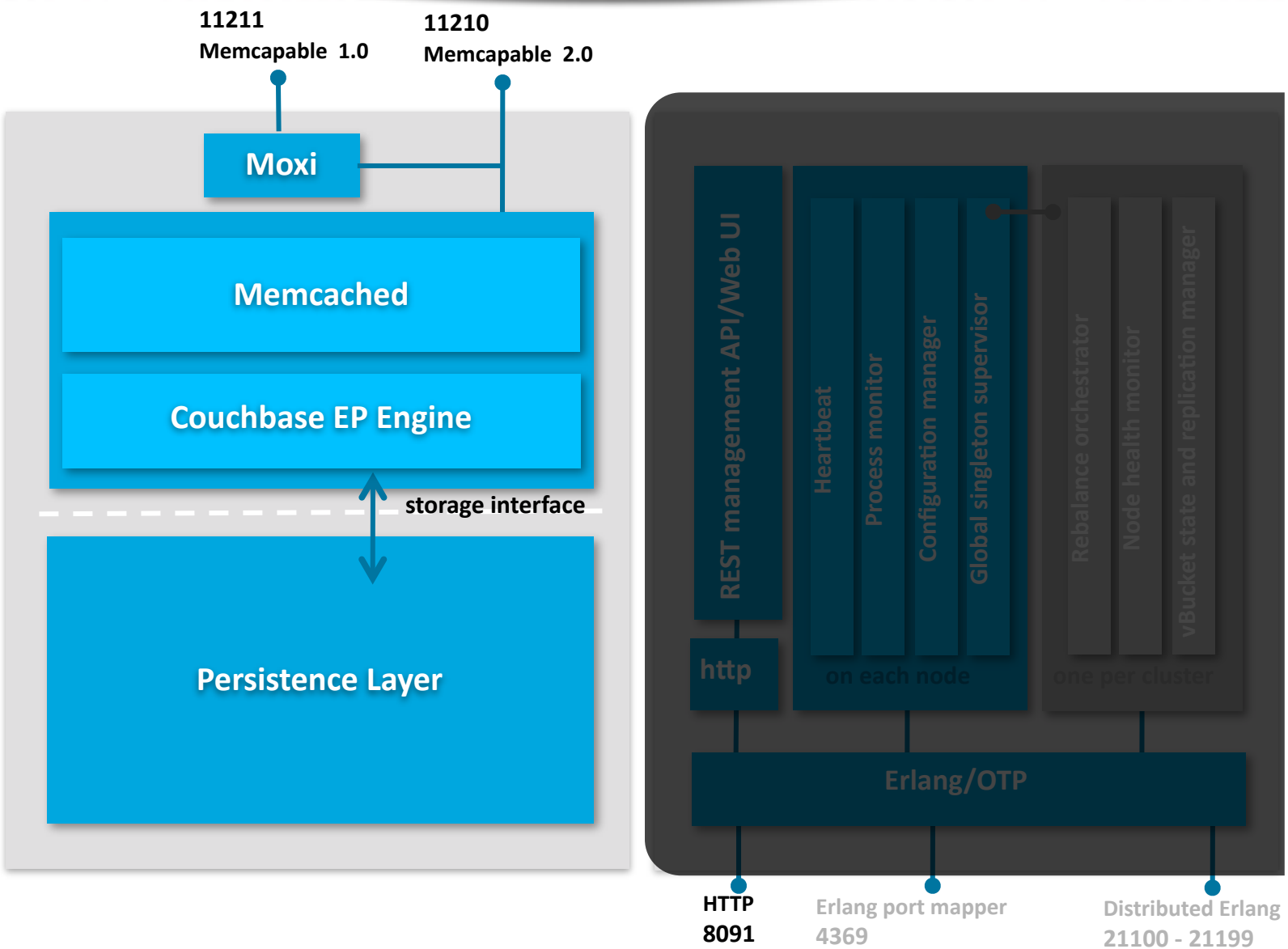
- Reliable storage architecture

Couchbase Server 1.8 Architecture



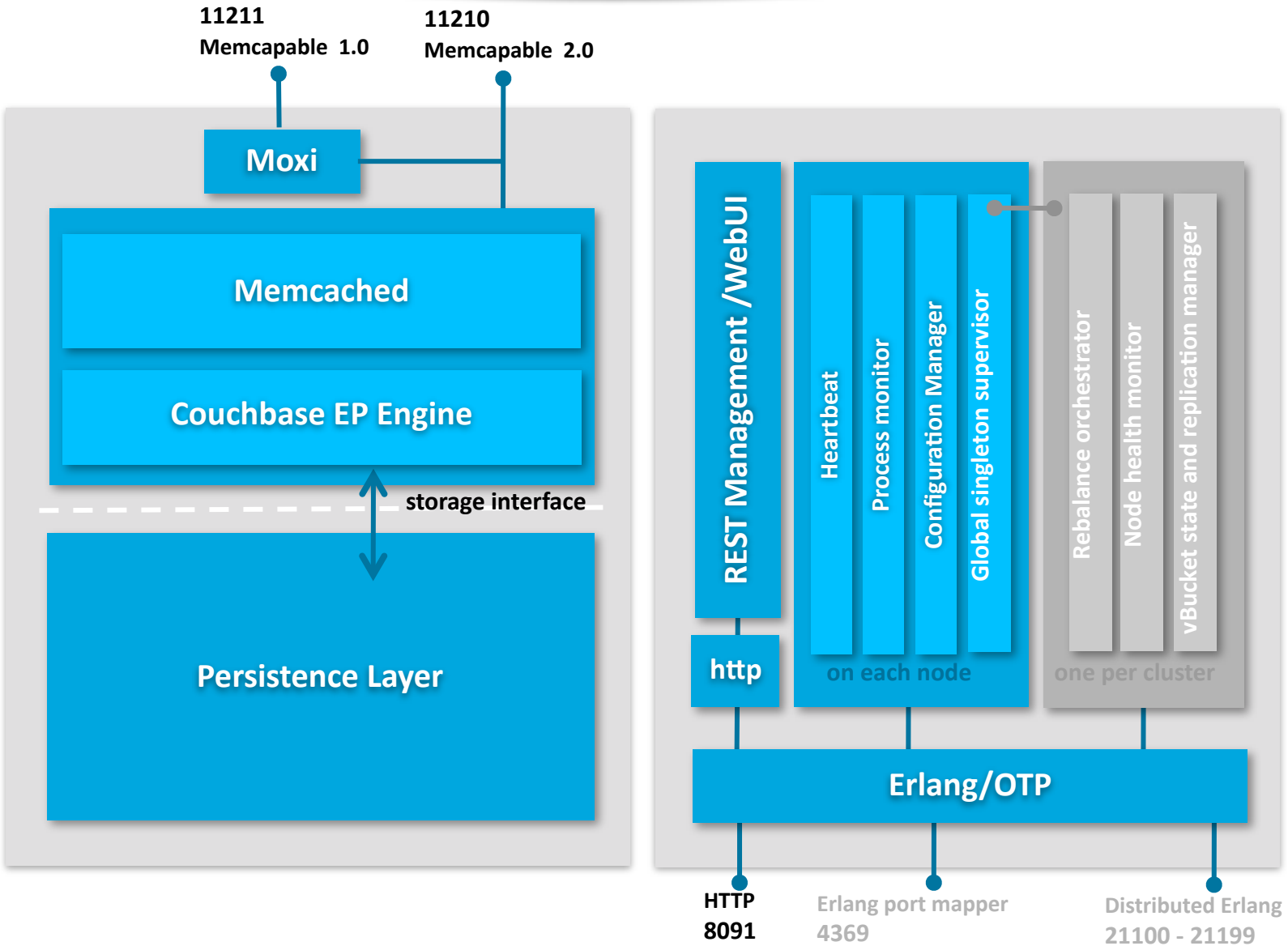
CAPI interface – basic Couch API of which some goes through the caching layer (CRUD), some goes directly to Couch (Views)

Couchbase Server 1.8 Architecture

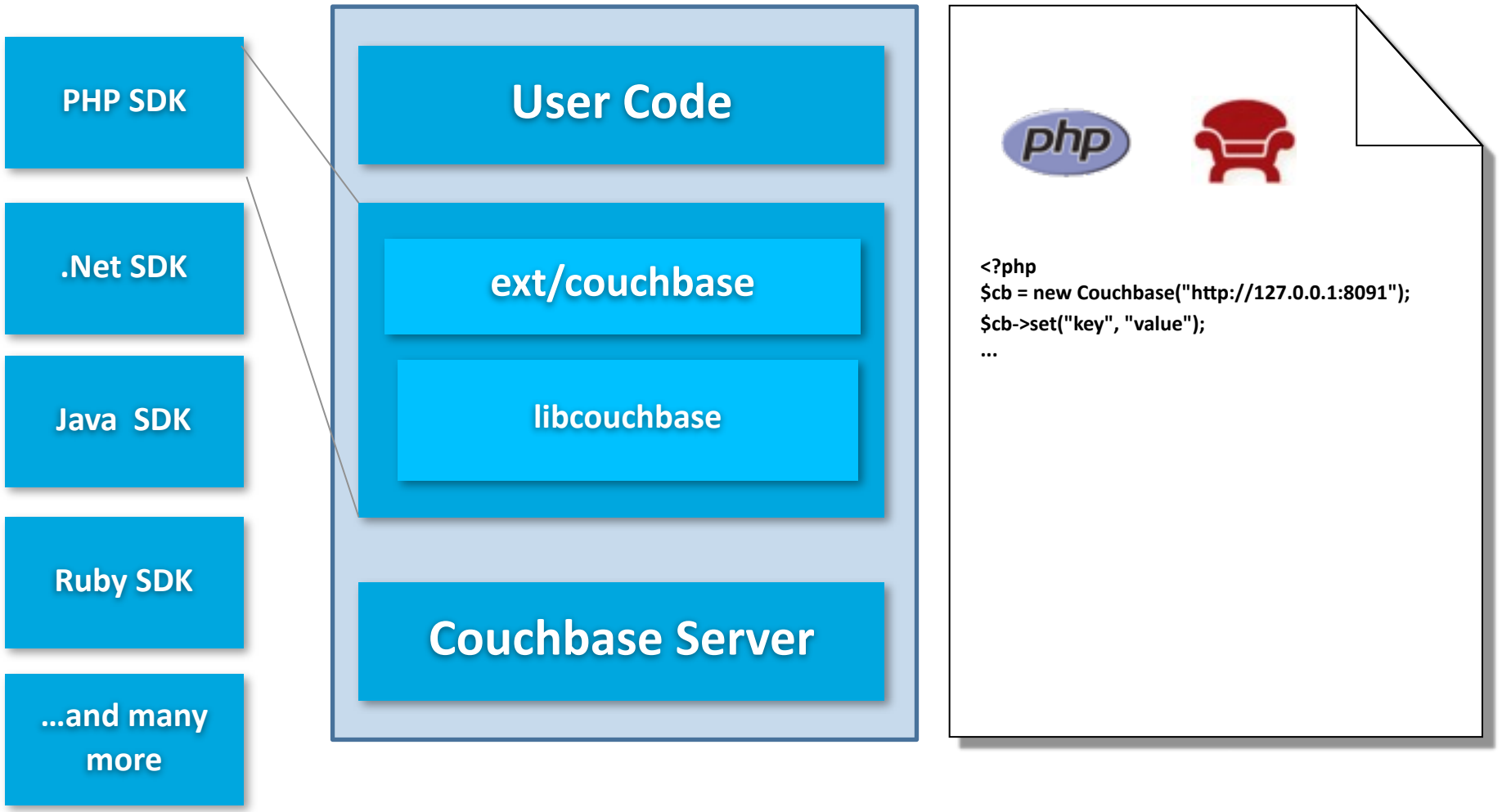


CAPI interface – basic Couch API of which some goes through the caching layer (CRUD), some goes directly to Couch (Views)

Couchbase Server 1.8 Architecture



Couchbase SDKs



http://www.couchbase.com/develop

Couchbase Server 2.0

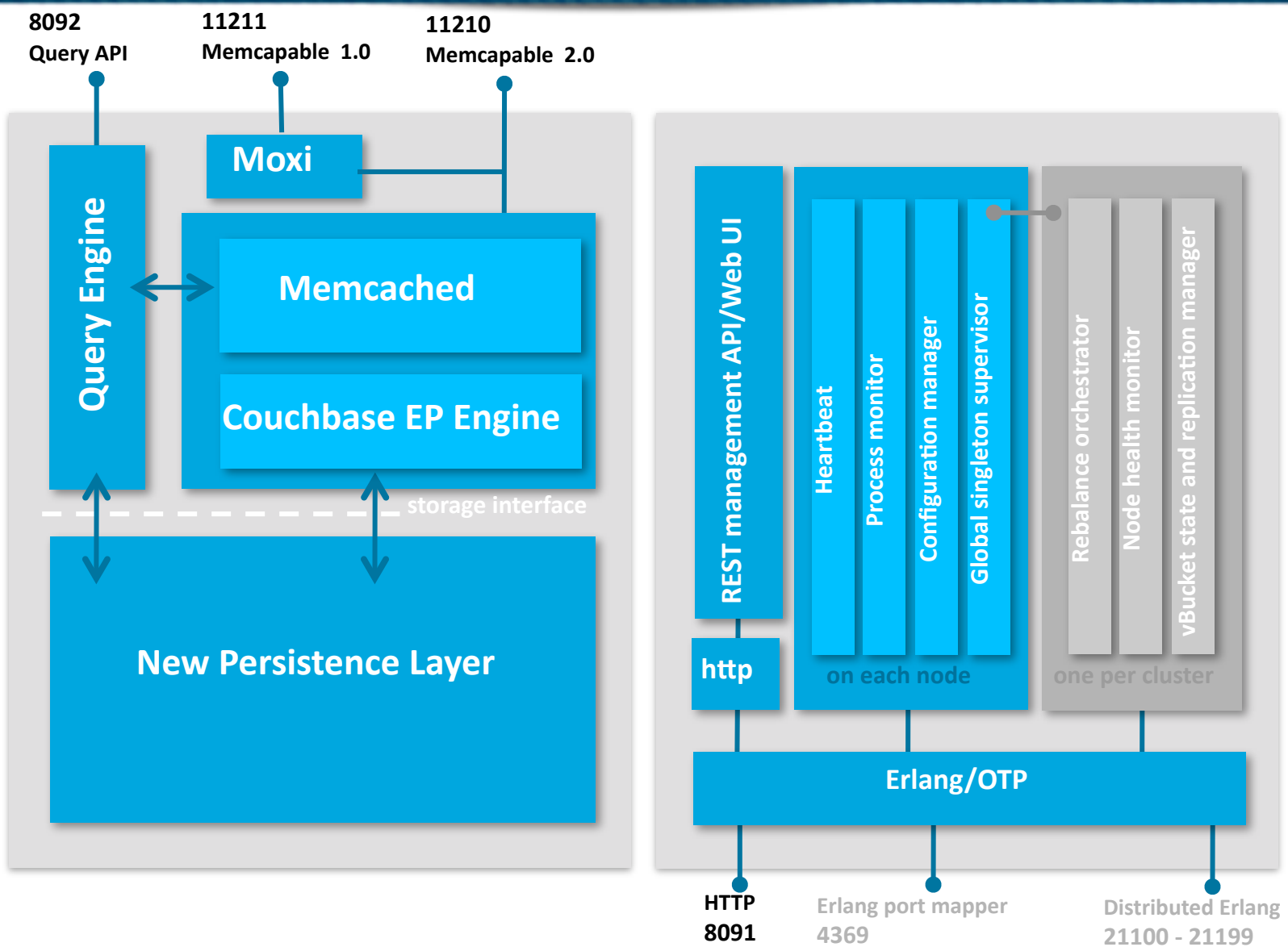
- Next major release of Couchbase Server
- Currently in Developer Preview

What's new:

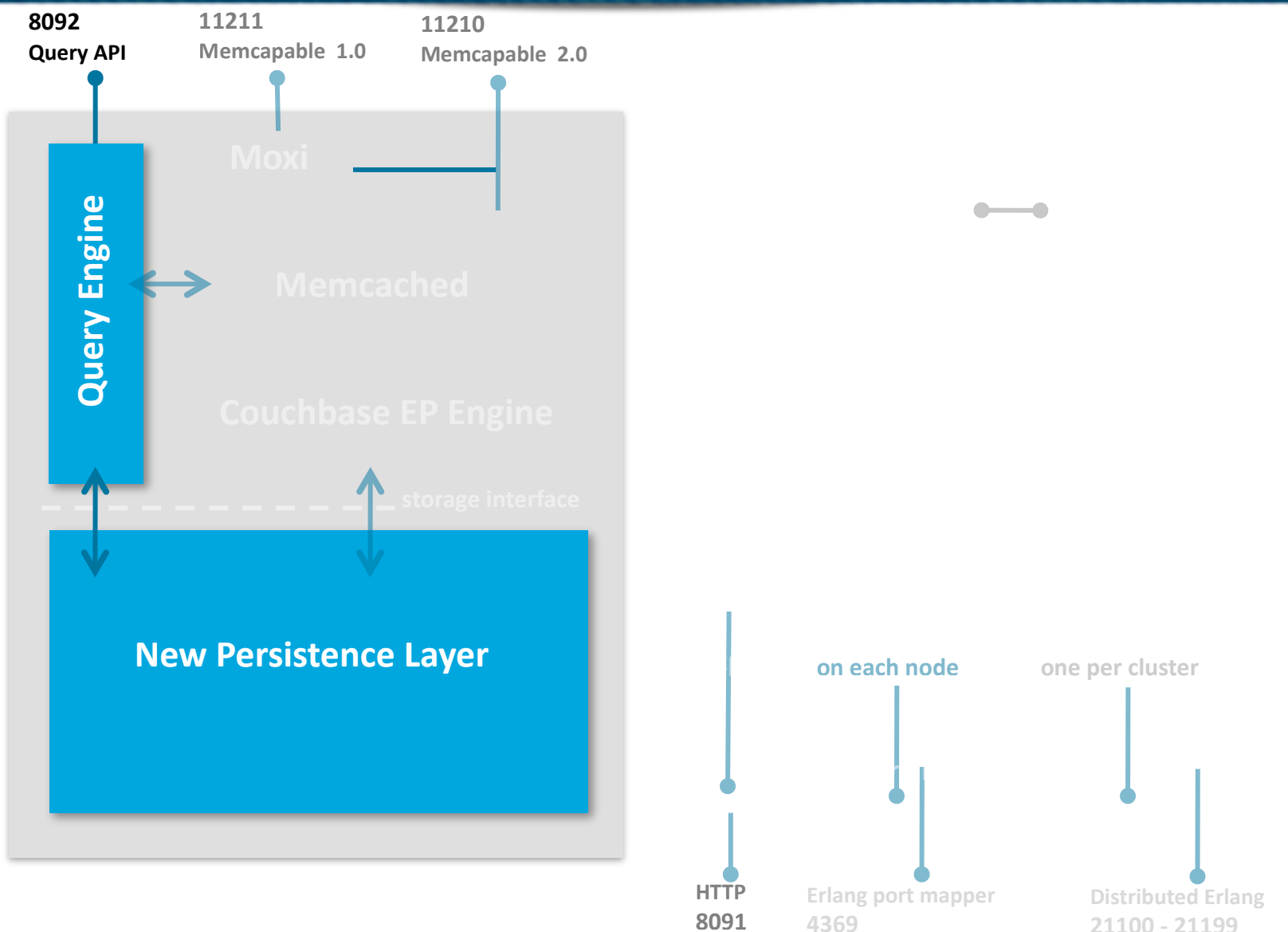
- Indexing and Querying
- Incremental Map Reduce
- Cross Data Center Replication

- Fully backwards compatible with existing Couchbase Server

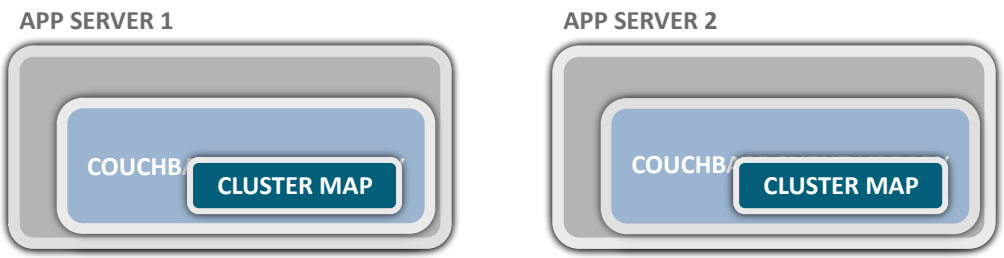
Couchbase Server 2.0 Architecture



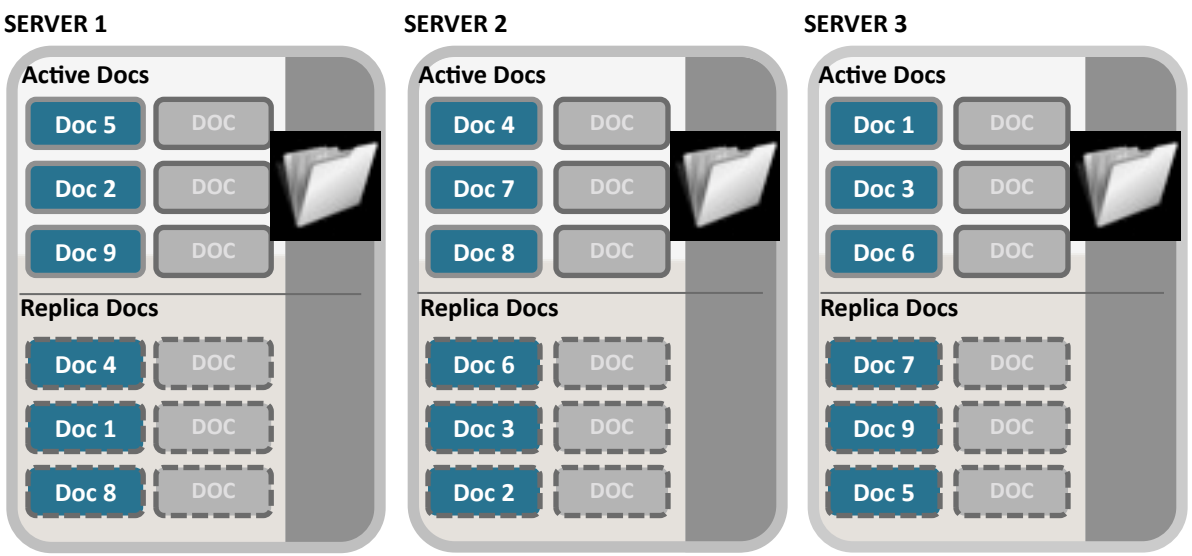
Couchbase Server 2.0 Architecture



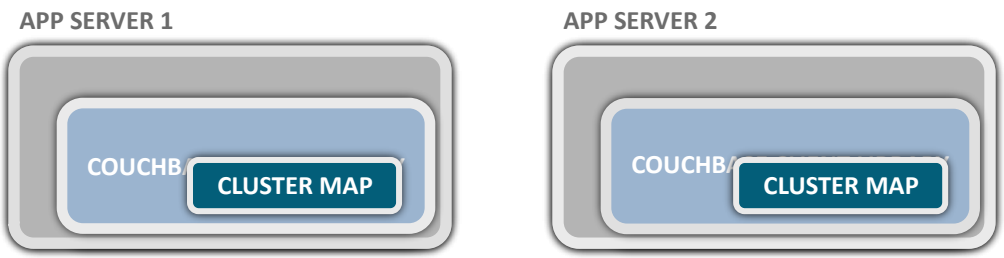
Indexing and Querying



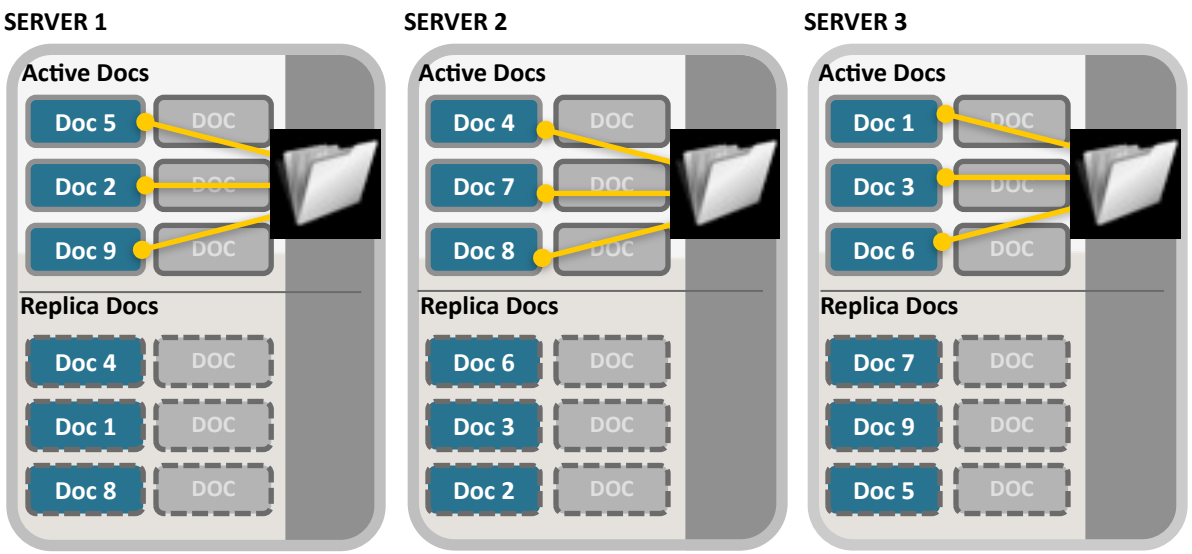
- **Indexing work is distributed amongst nodes**
  - Large data set possible
  - Parallelize the effort



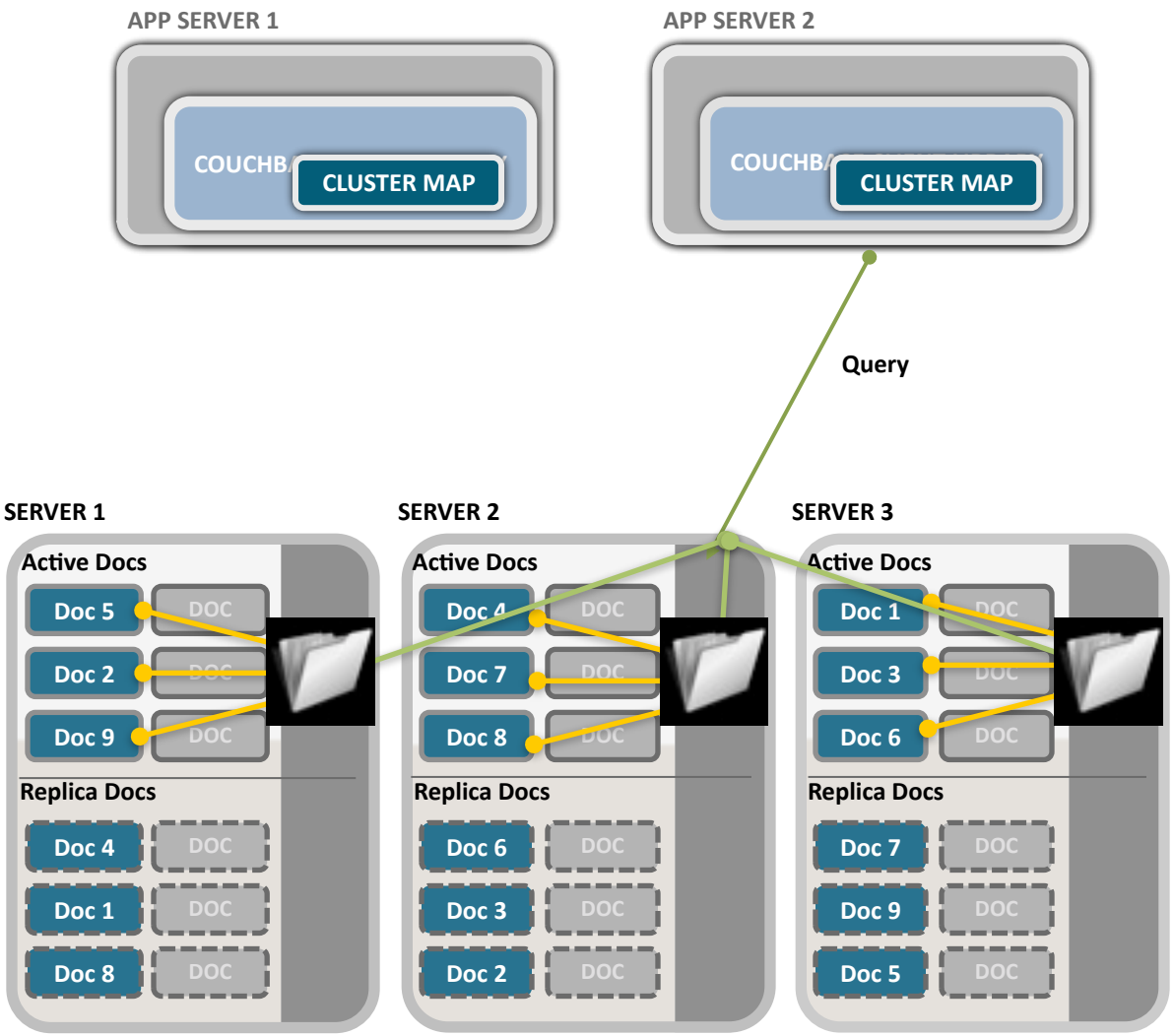
Indexing and Querying



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

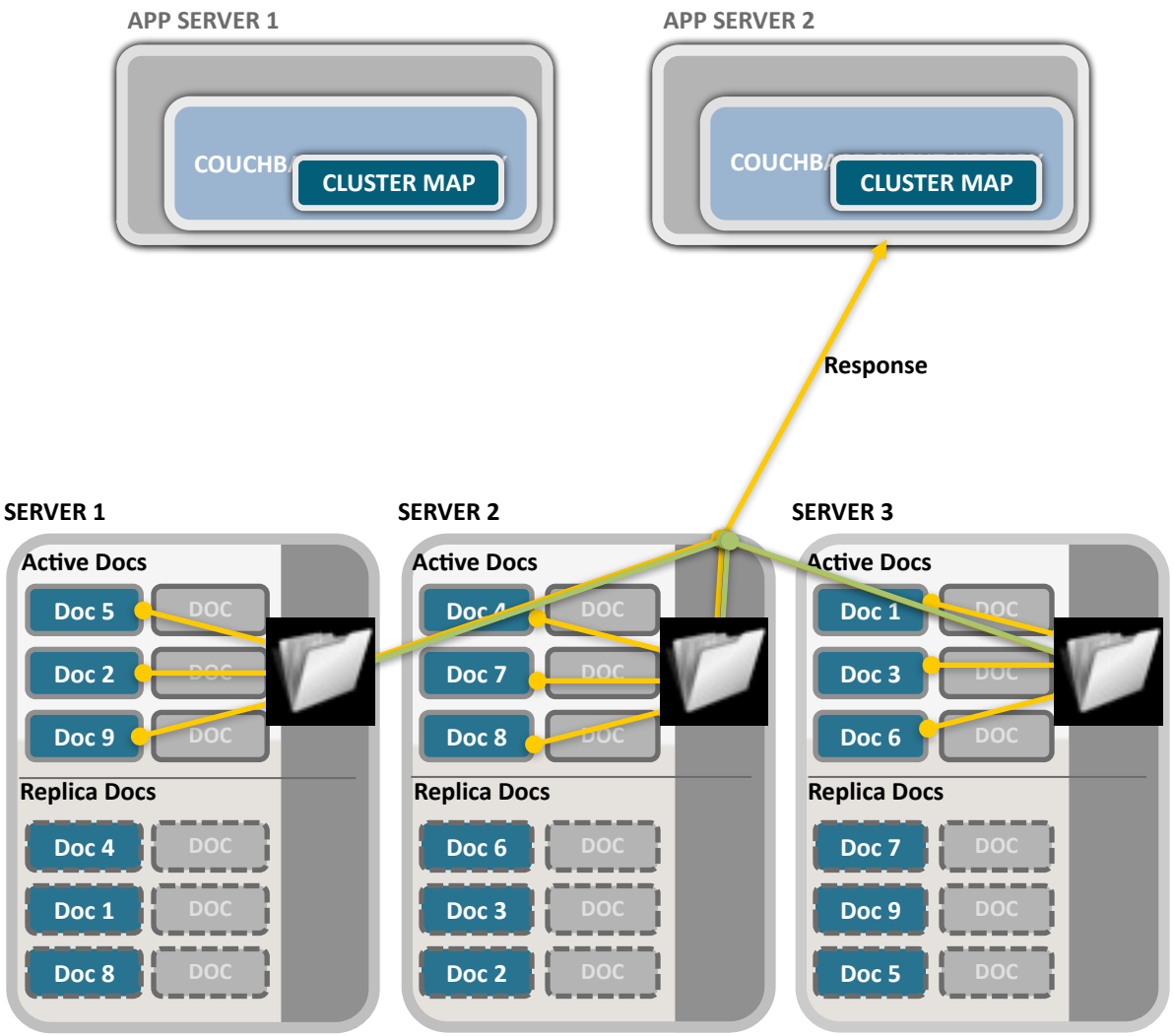


Indexing and Querying



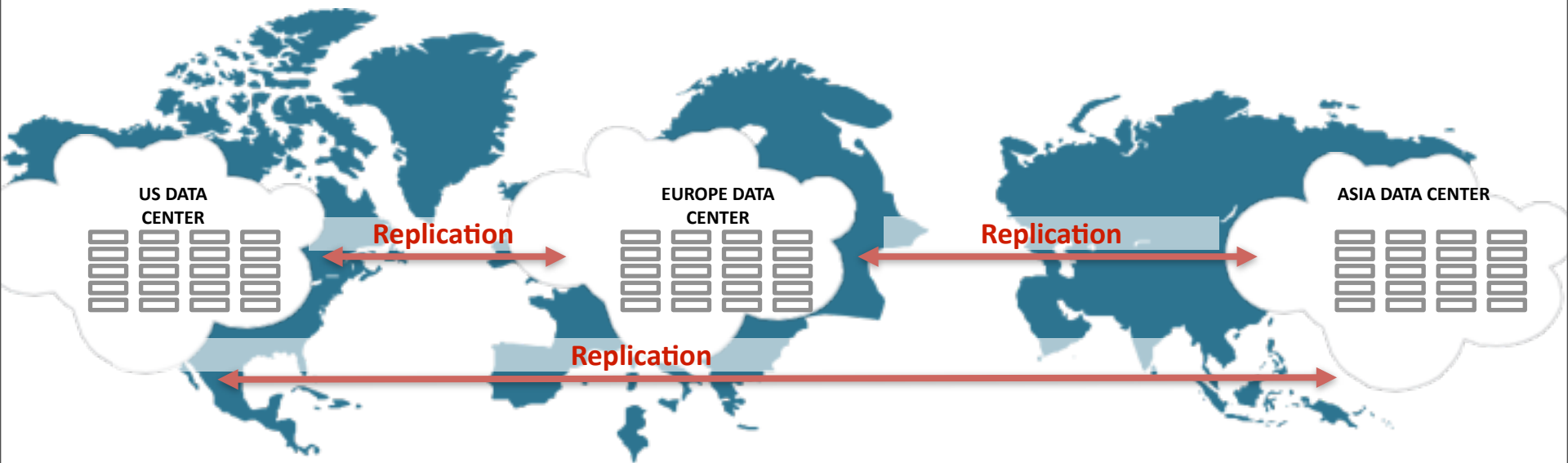
- Indexing work is distributed amongst nodes
  - Large data set possible
  - Parallelize the effort
- Each node has index for data stored on it

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 combine the results from required nodes

Cross Data Center Replication



- Want data close to user
- Want multiple locations for disaster recovery

- Multi-Master: Can write to same document in all different regions & it will sync (eventually consistent, always available)

Not yet enabled in current DP, will be available for Beta

EXT/COUCHBASE

Setup

```
<?php
$url = "http://localhost:8091/";
$cb = new Couchbase($url);
```

Basics

```
<?php
// setup
$cb->set("a", 1);

$a = $cb->get("a");
echo $a; // prints 1

$cb->increment("a");
echo $cb->get("a"); // prints 2
$cb->delete("a"); // booya
```

Storage Operations

```
<?php
// set "a" to 1
$cb->set("a", 1);

// fails if "b" exists
$cb->add("b", 1);

// fails if "c" doesn't exit
$cb->replace("c", 1);
```

Expiration

```
<?php
// set "a" to 1 for 10 seconds
$cb->set("a", 1, 10);

// most other ops can set expire


// just update the expiry
$cb->touch("a", 10);
```

Compare And Swap

```
<?php
$cas = -1;
$cb->get("a", null, &$cas);

echo $cas; // prints 76324827359

// fails if cas doesn't match
$cb->cas($cas, "a", 2);
```

Arithmetic

```
<?php
$cb->set("a", 1); // "a" is 1
$cb->increment("a"); // "a" is 2
$cb->increment("a", 2); // "a" is 4
$cb->decrement("a", 4); // "a" is 0
```

Arithmetic 2

```
<?php
// increment "b" by 3, create "b"
// if it doesn't exist in which
// case, initialise it with 2, set
// no expiry

// increment($key, $offset,
//     $create, $expiry, $initial)

$cb->increment("b", 3, true, 0, 2);
```

QUESTIONS?

THANK YOU!

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

Give us feedback at:
http://www.couchbase.com/forums