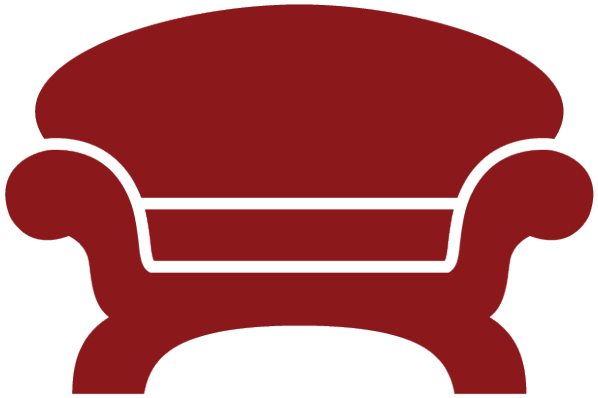
Couchbase Server

Chris Anderson
Chief Architect @jchris

Couchbase

Couchbase Server



Couchbase

=

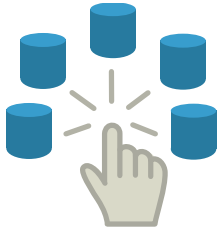
- Simple
- Fast
- Elastic
- NoSQL Database

Formerly known as Membase Server

Couchbase Server Features

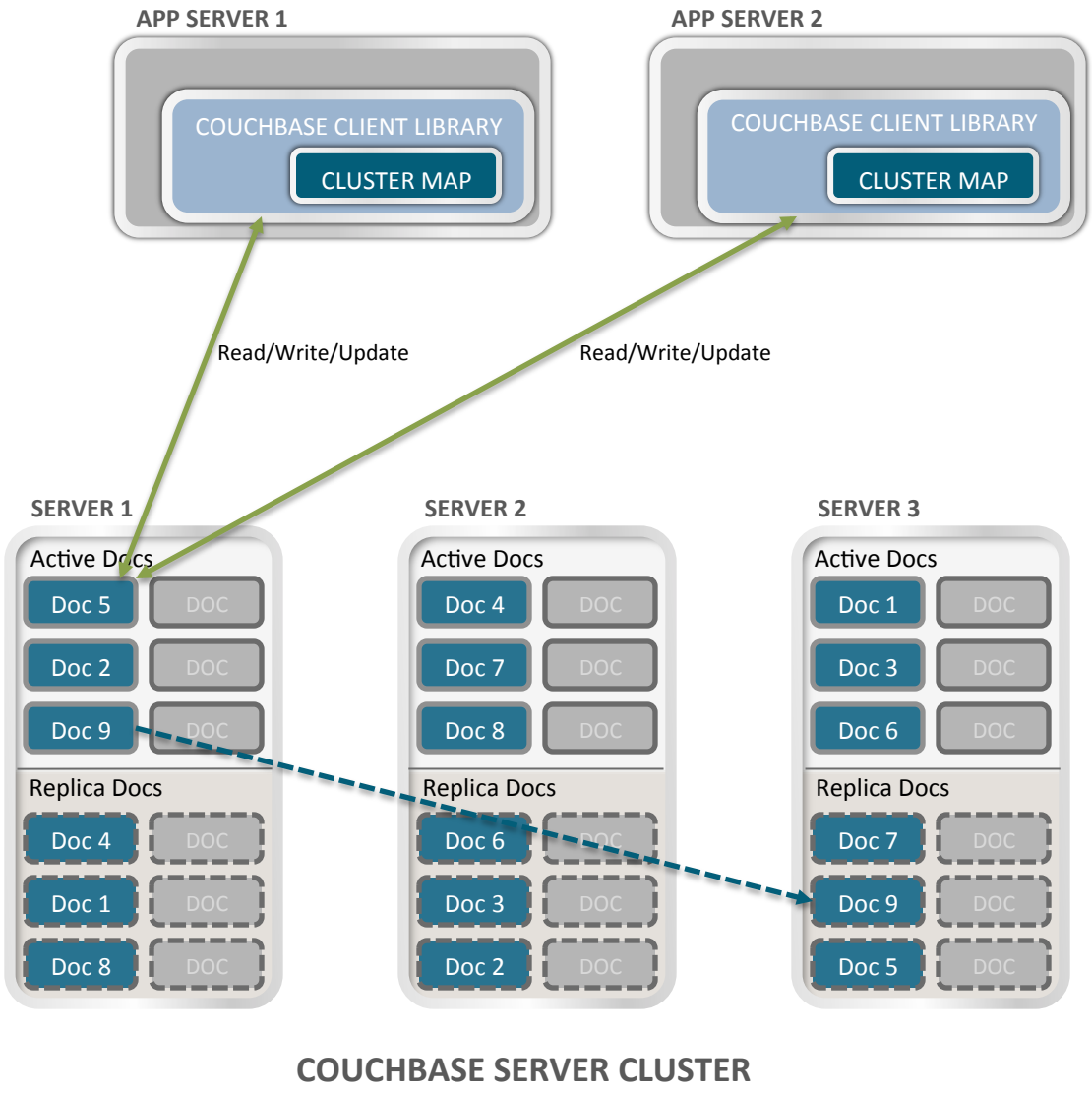
- Built-in clustering – All nodes equal
- Data replication with auto-failover

- Zero--downtime maintenance



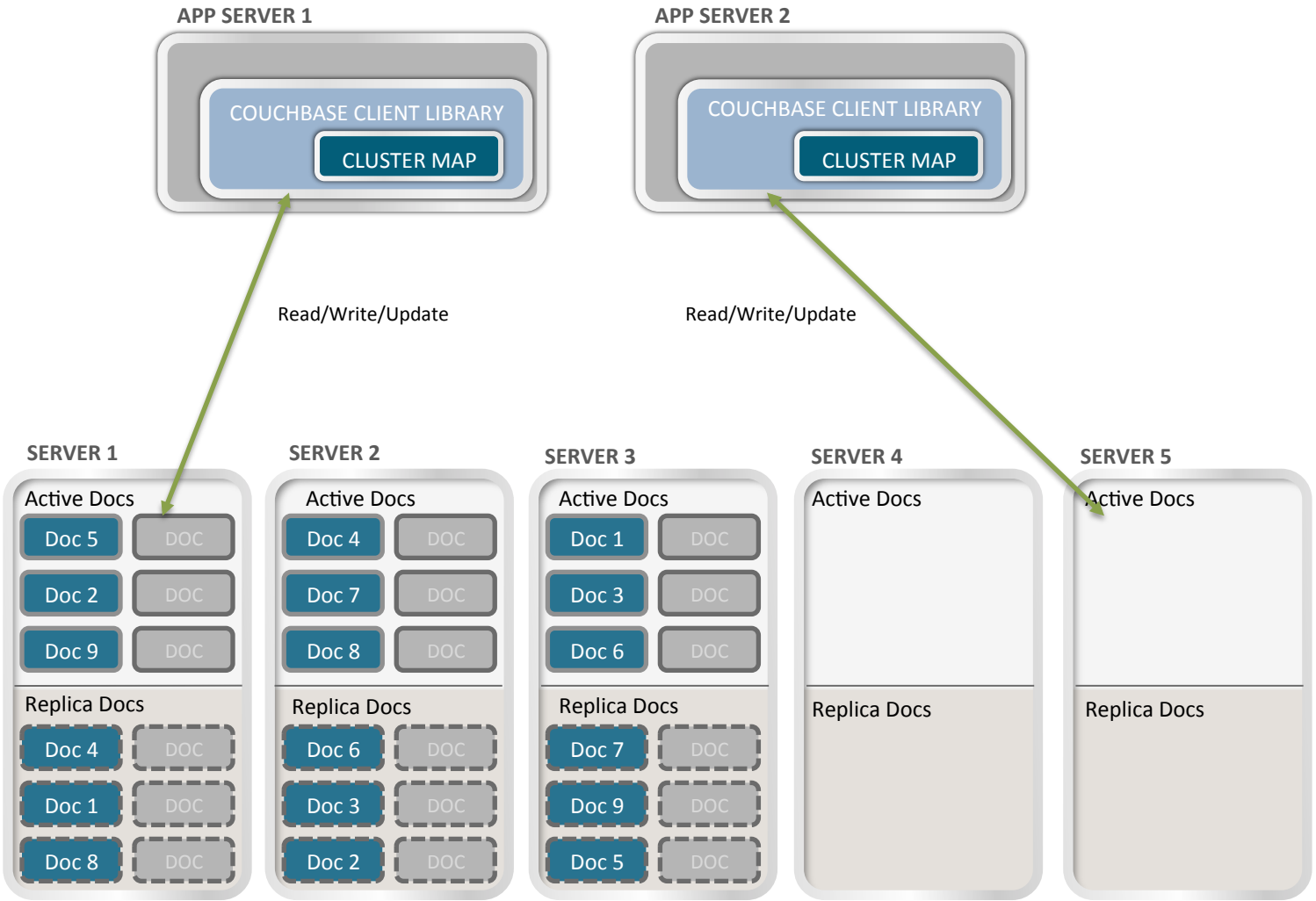
- Clone to grow and scale horizontally

Couchbase Server Basic Operation



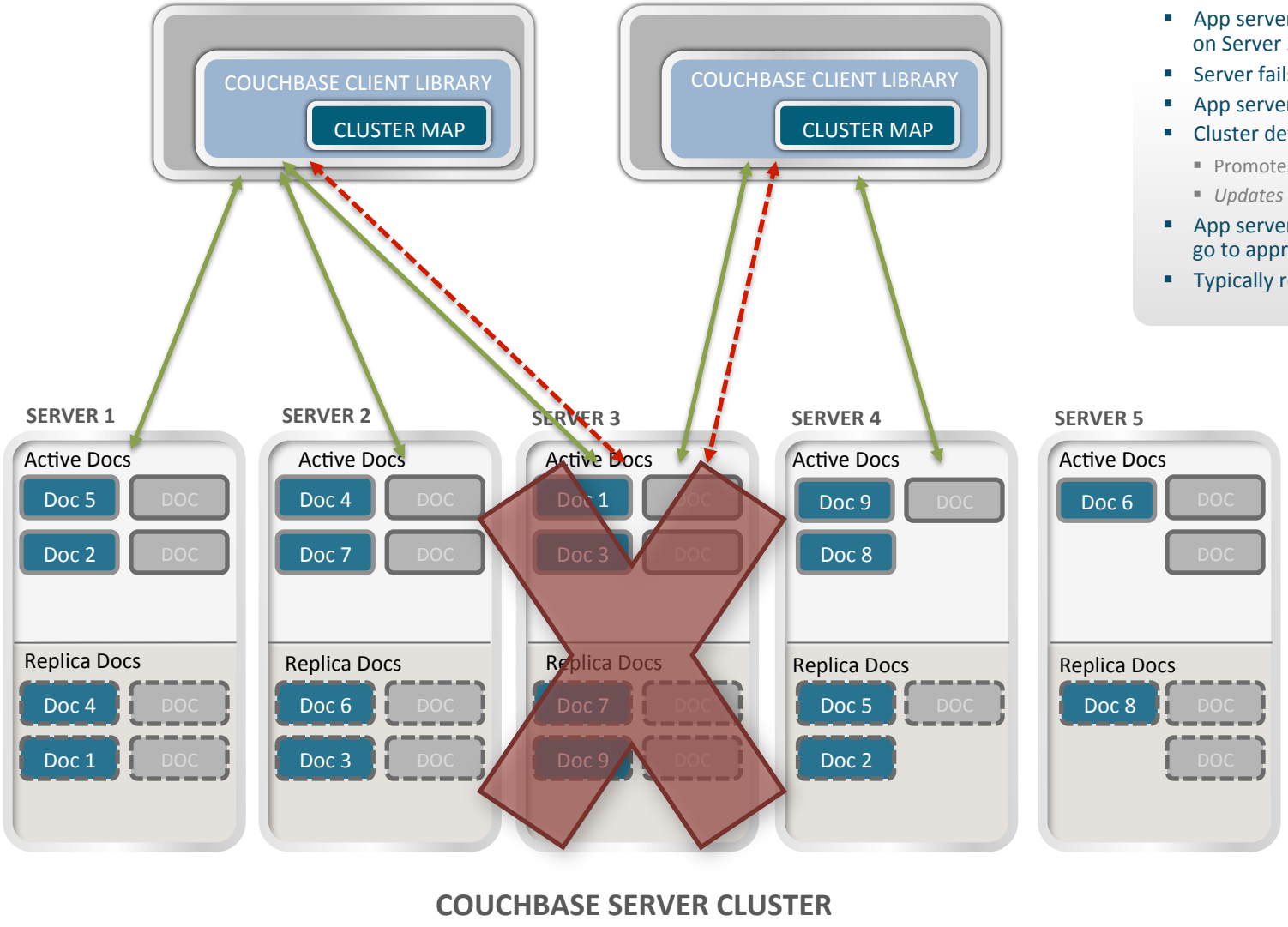
- Docs distributed evenly across servers in the cluster
- Each server stores both *active* & *replica* docs
  - Only one server active at a time
- Client library provides app with simple interface to database
- Cluster map provides map to which server doc is on
  - App never needs to know
- App reads,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

Add Nodes



- 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

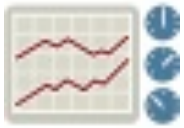
Fail Over Node



- 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 Features

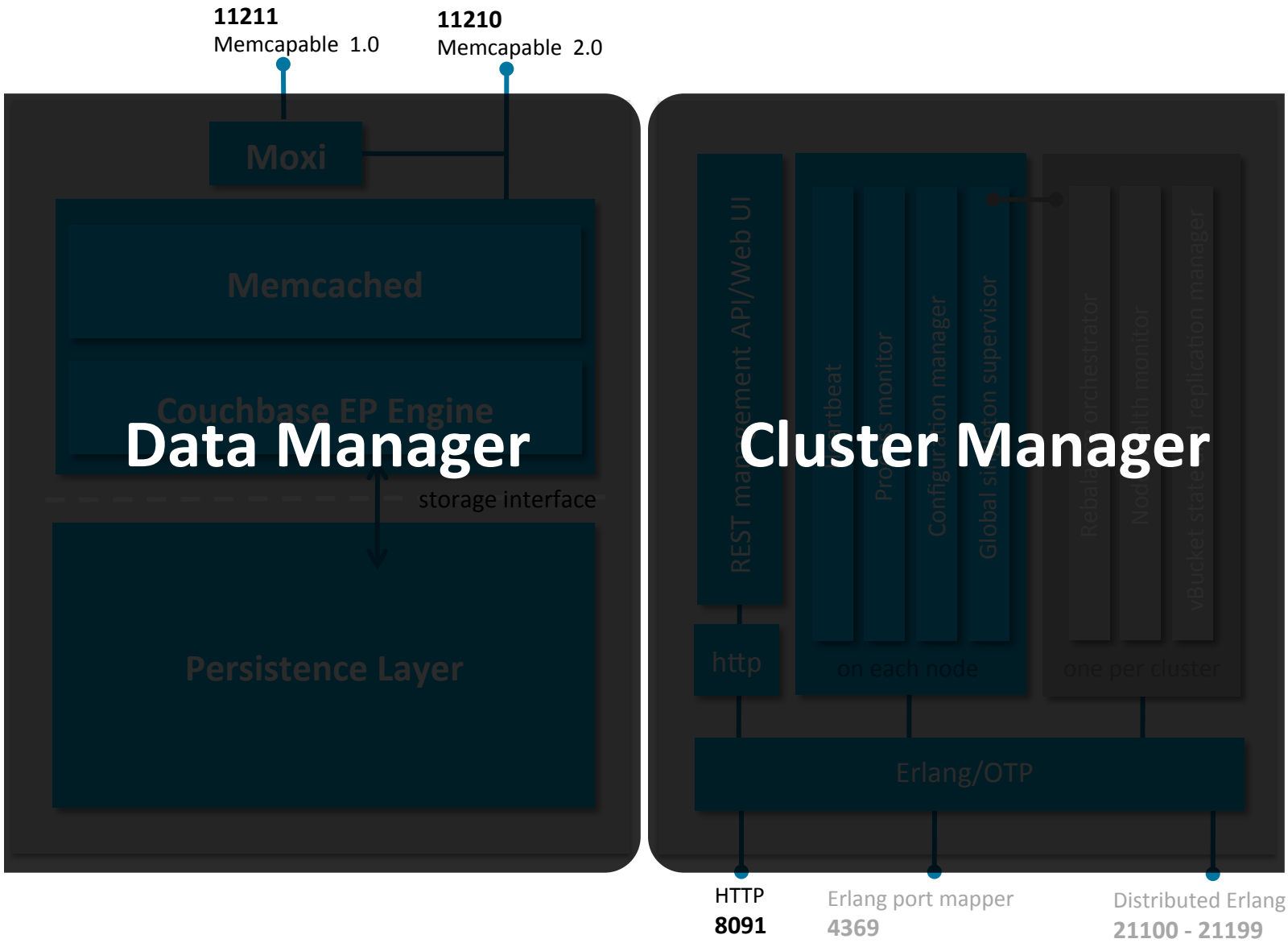
- Memcached compatible (built-in caching)



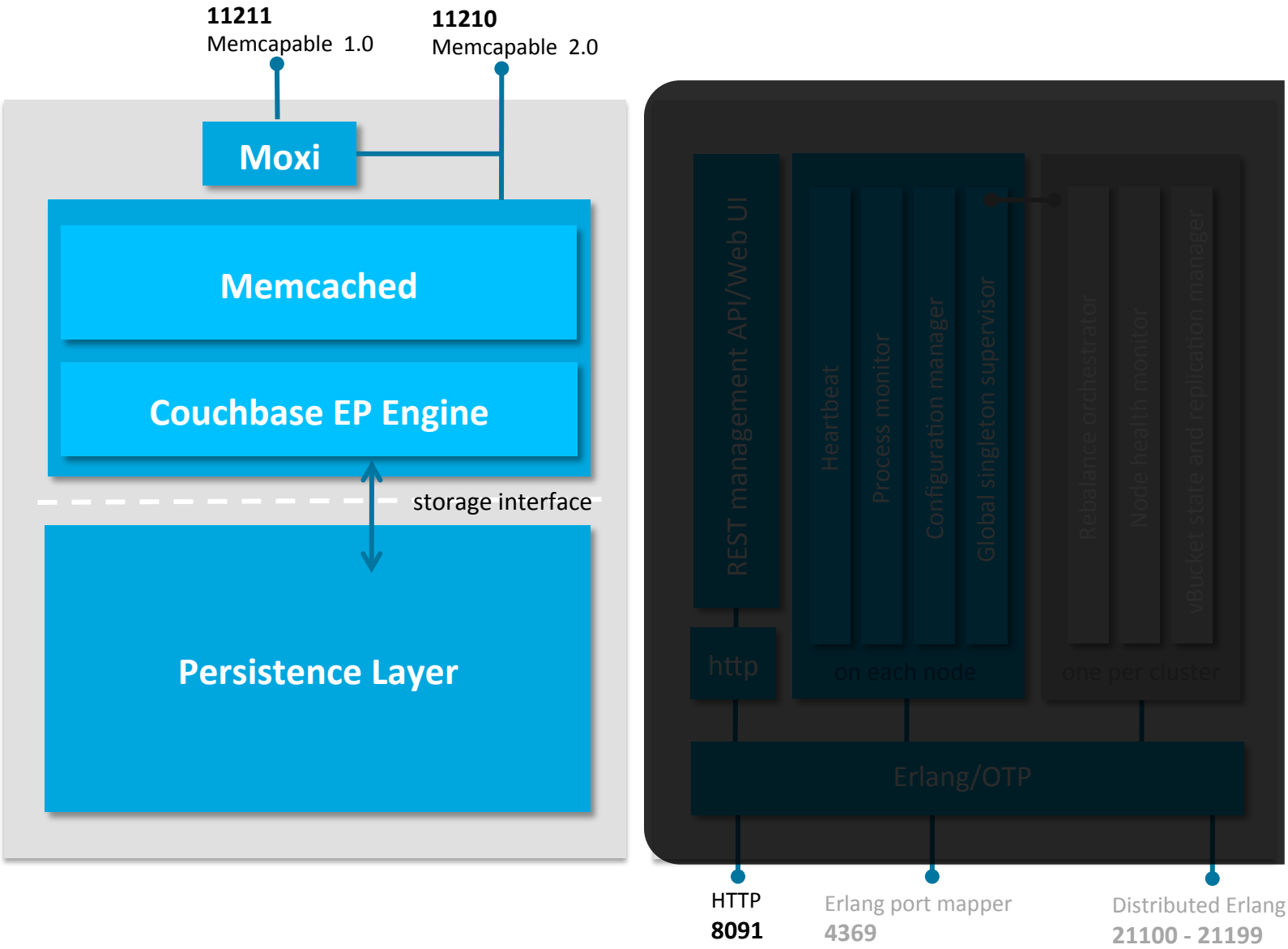
- Monitoring and administration APIs and GUI

- Reliable storage architecture

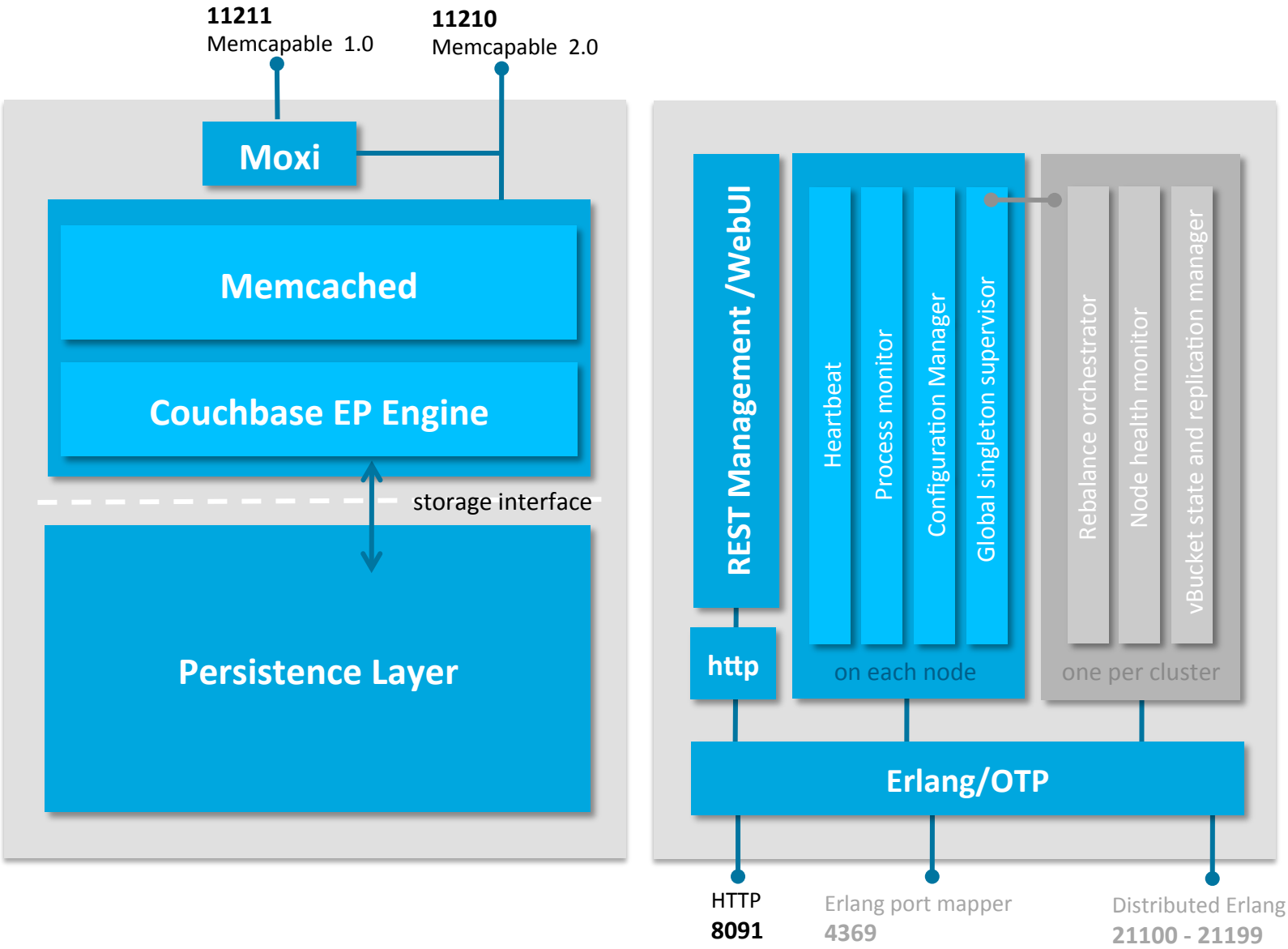
Couchbase Server 1.8 Architecture



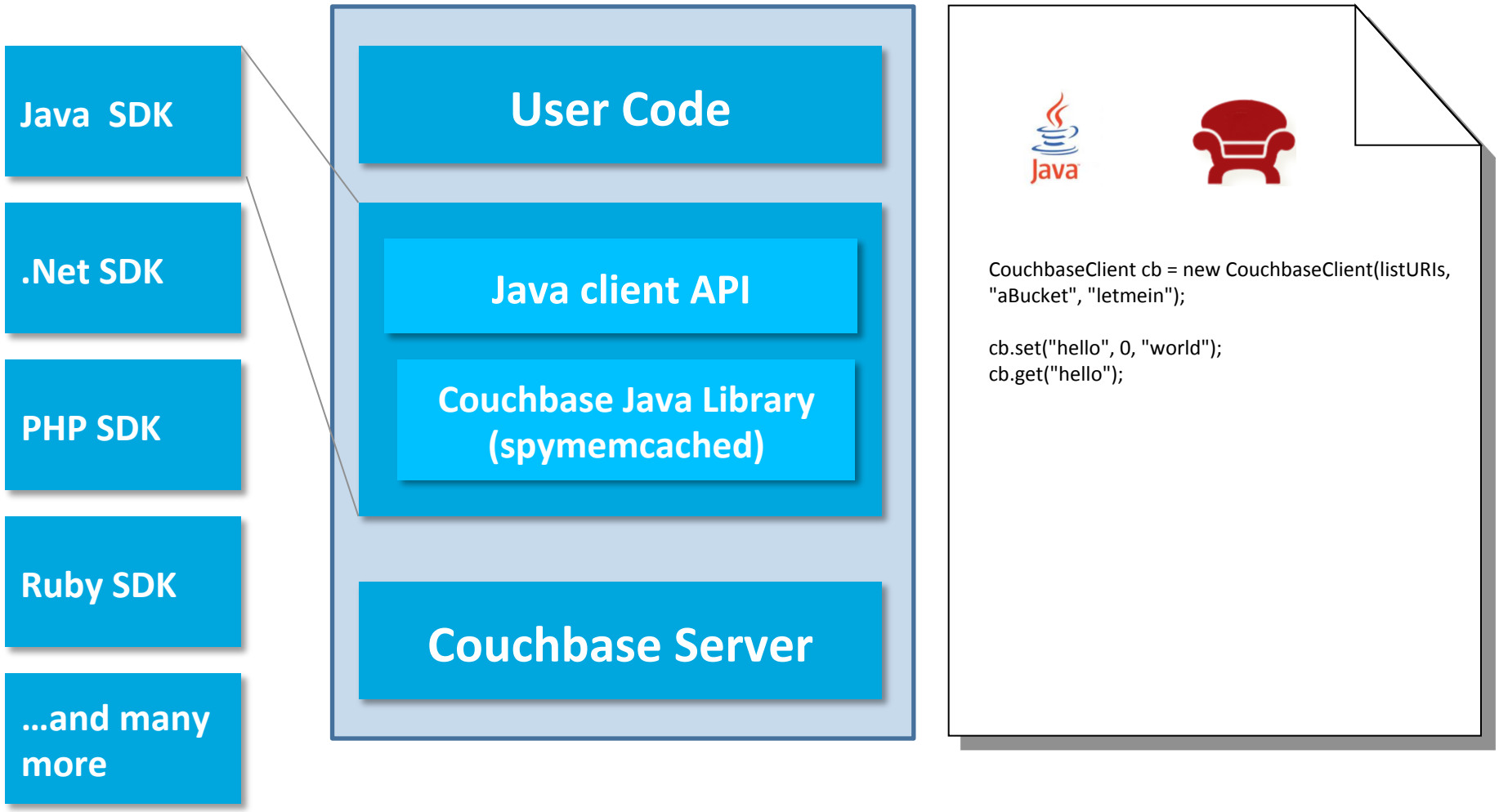
Couchbase Server 1.8 Architecture



Couchbase Server 1.8 Architecture



Couchbase SDKs



http://www.couchbase.com/develop

DEMO TIME

Couchbase Server 2.0

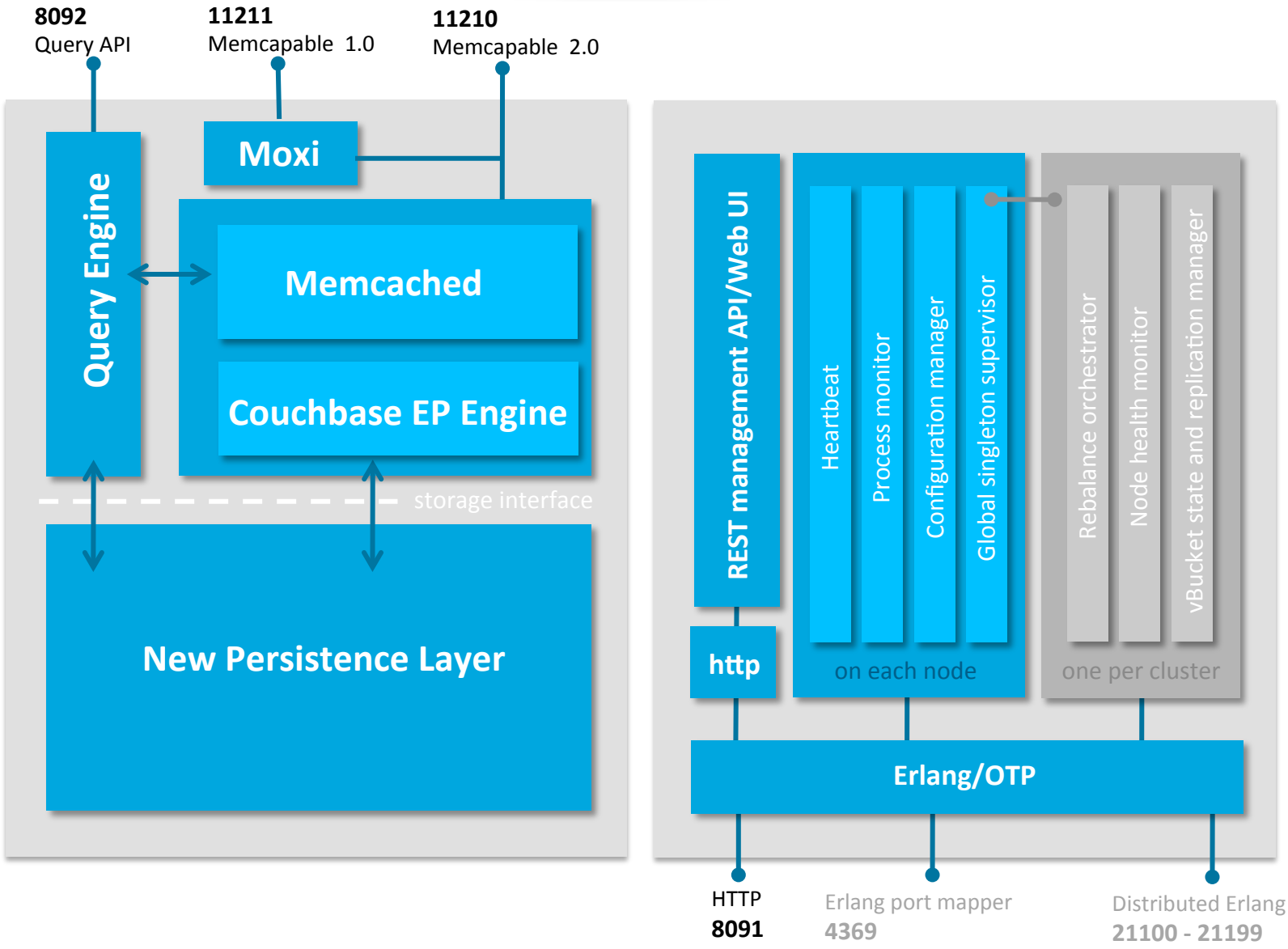
- Next major release of Couchbase Server
- Currently in Developer Preview

What's new:

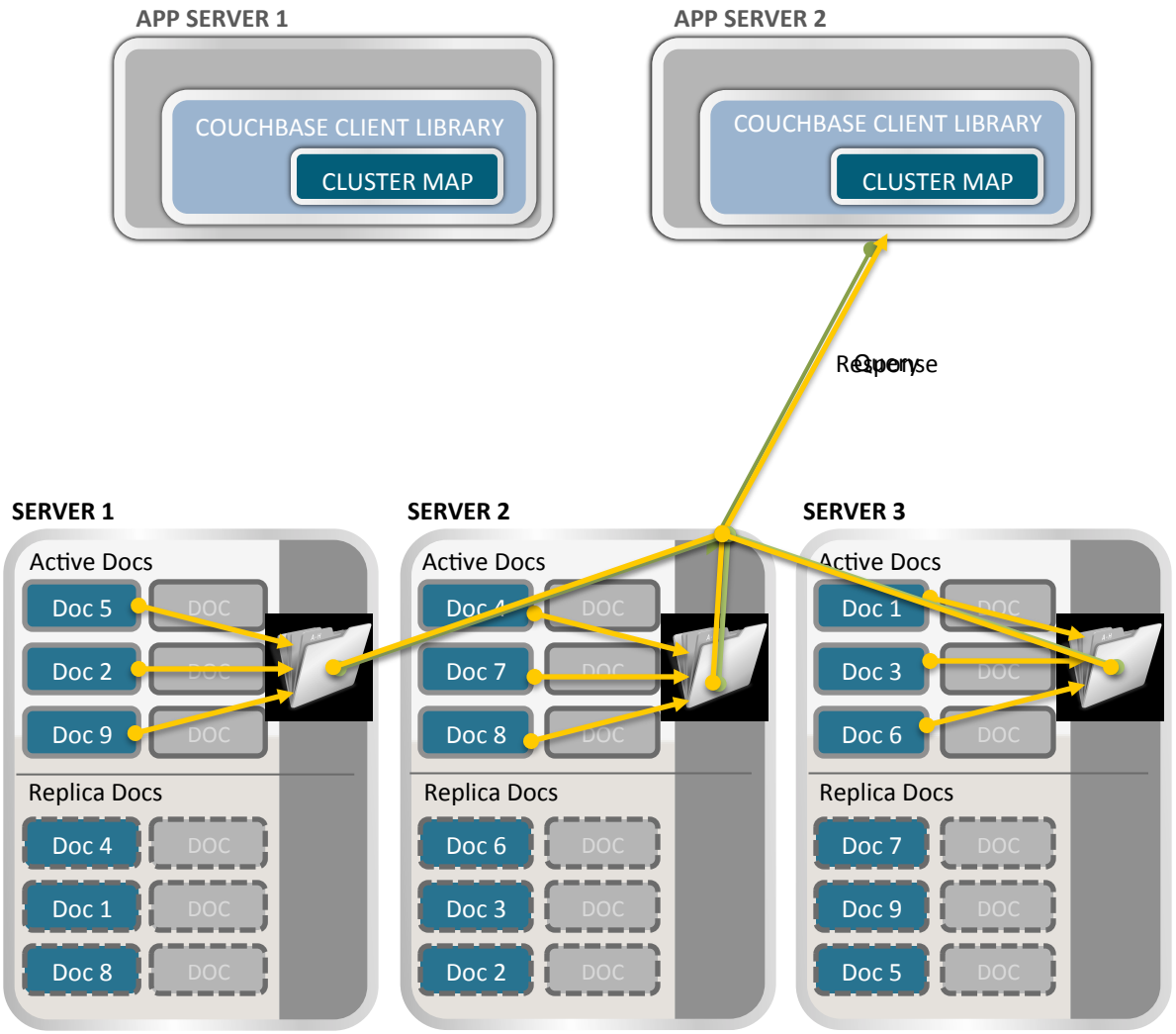
- Indexing and Querying
- Incremental Map Reduce
- Cross Data Center Replication

- Fully backwards compatible with existing Couchbase Server

Couchbase Server 2.0 Architecture

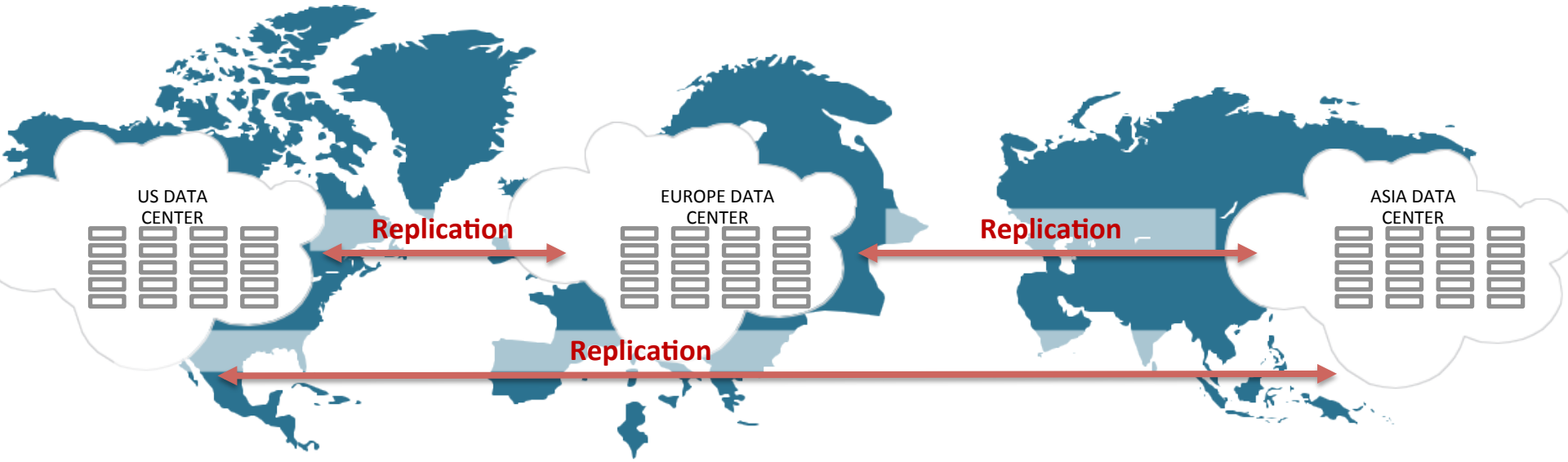


Indexing and Querying



- 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



- Data close to users
- Multiple locations for disaster recovery
- Independently managed clusters serving local data

DEMO TIME

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",
        "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",
        "jsonType": "item",
        "name": "Katana_e5890c94-11c6-65746ce6c560",
        "uuid": "e5890c94-11c6-4856-a7a6-65746ce6c560",
        "ownerId": "Dale9887"
}
```


Player “_id”

Monster Document


```
{
        "_id": "Bauchan9932”,
        "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.com/forums

NOSQL KNOCKOUT
#GOTOCPH