Introduction to Couchbase Server 3.0

Ramzi Alqrainy

Search / Data Engineer

Couchbase

Couchbase Server

NoSQL Document Database

Couchbase

Couchbase Server

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

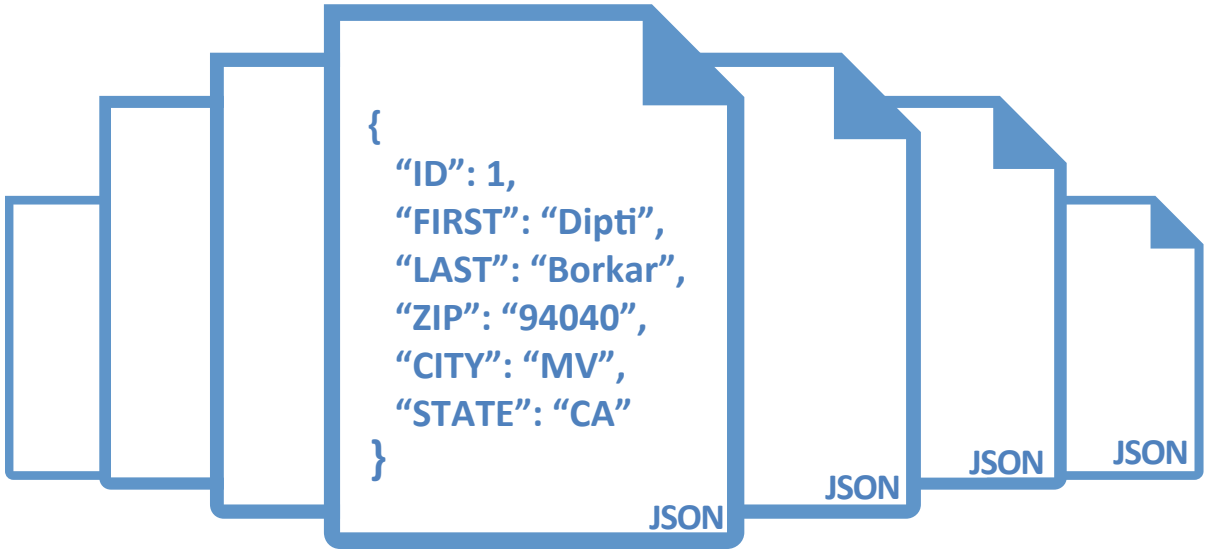
Always On 24x365

No downtime for software upgrades, hardware maintenance, etc.

Flexible Data Model

JSON document model with no fixed schema.

Flexible Data Model



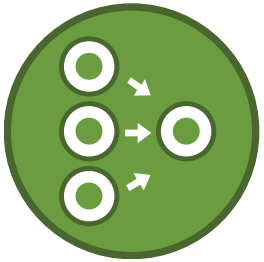
- No need to worry about the database when changing your application
- Records can have different structures, there is no fixed schema
- Allows painless data model changes for rapid application development

New in 3.0

JSON support

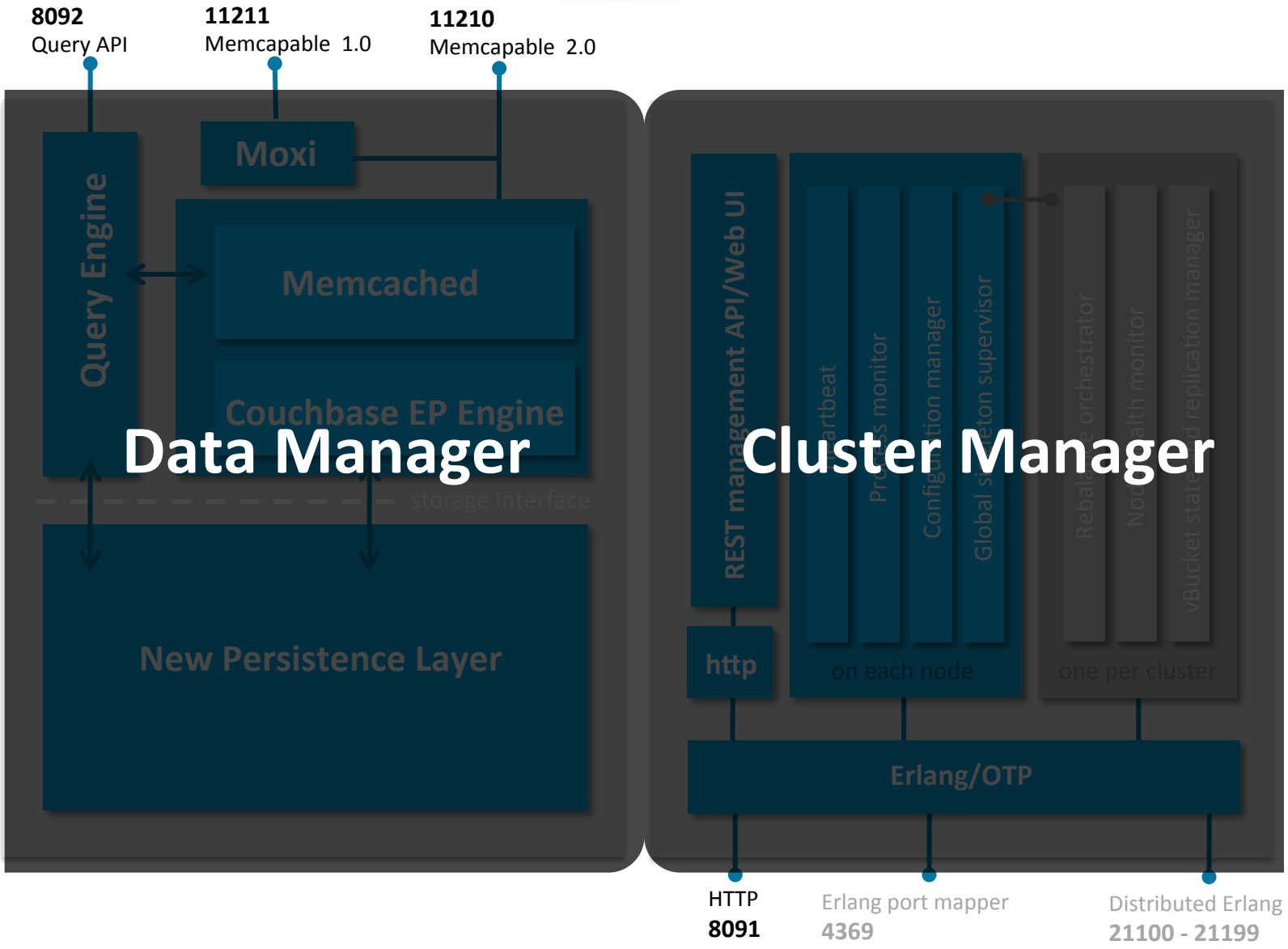
Indexing and Querying

Incremental Map Reduce

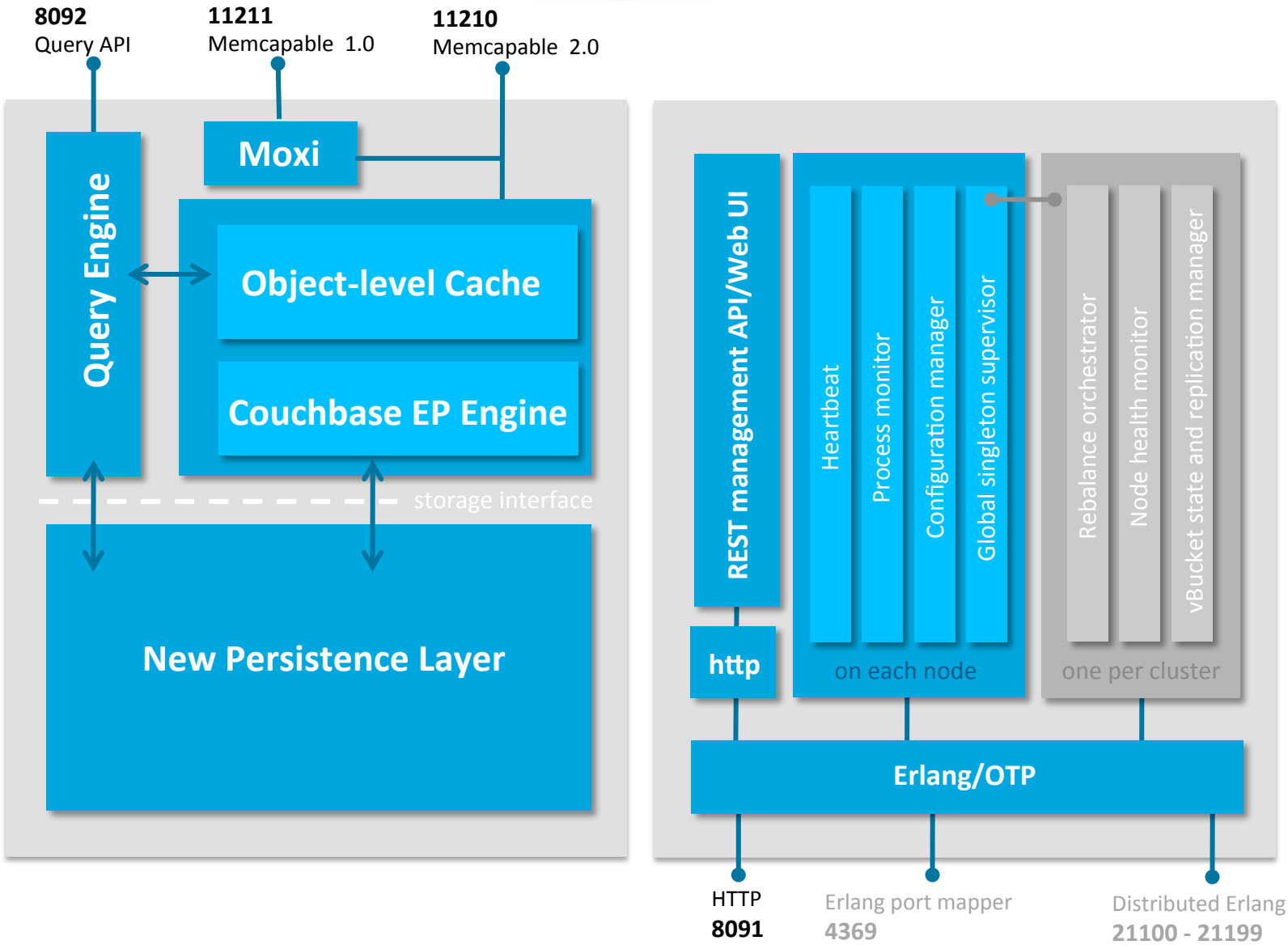


Cross data center replication

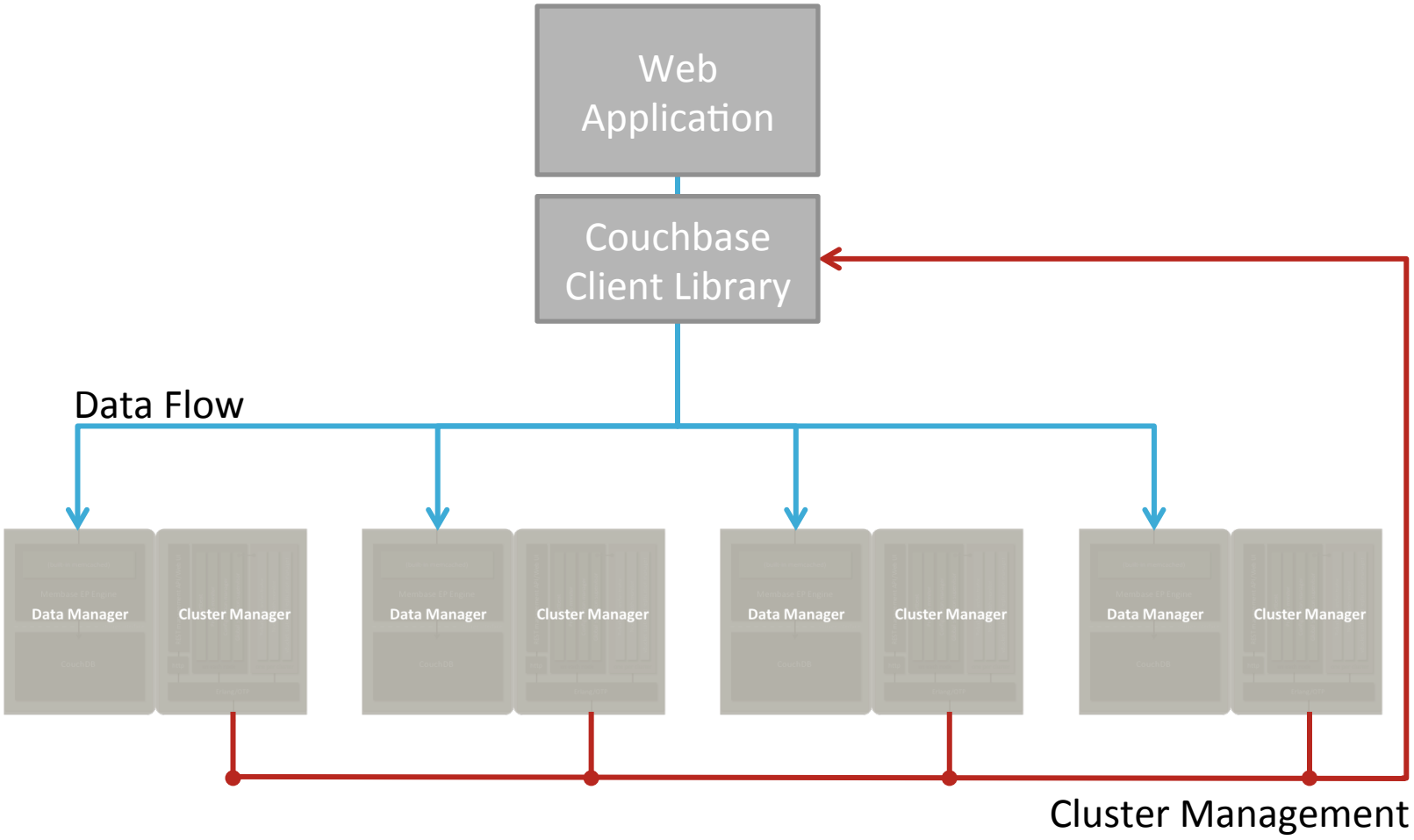
Couchbase Server 3.0 Architecture



Couchbase Server 3.0 Architecture

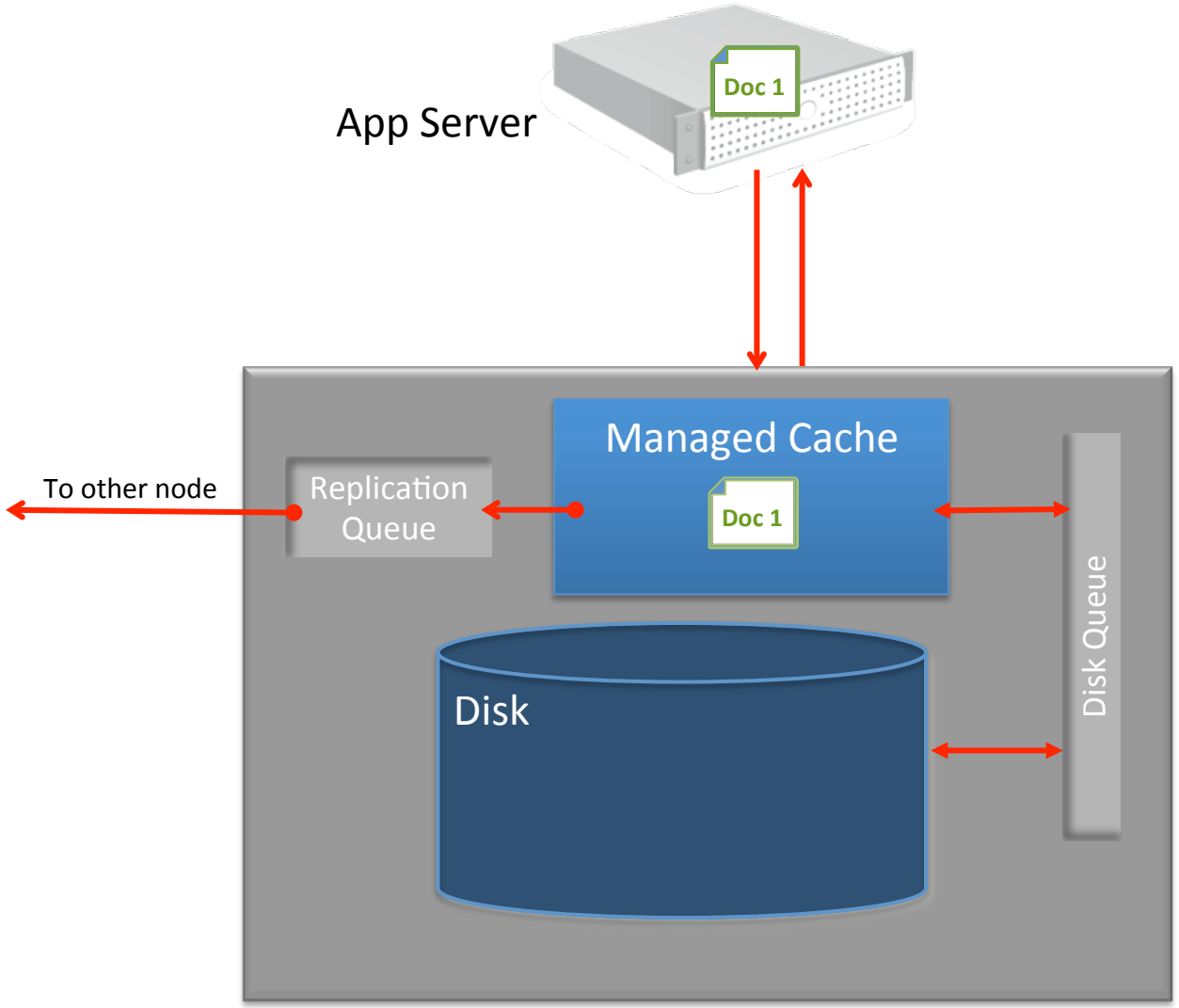


Couchbase deployment

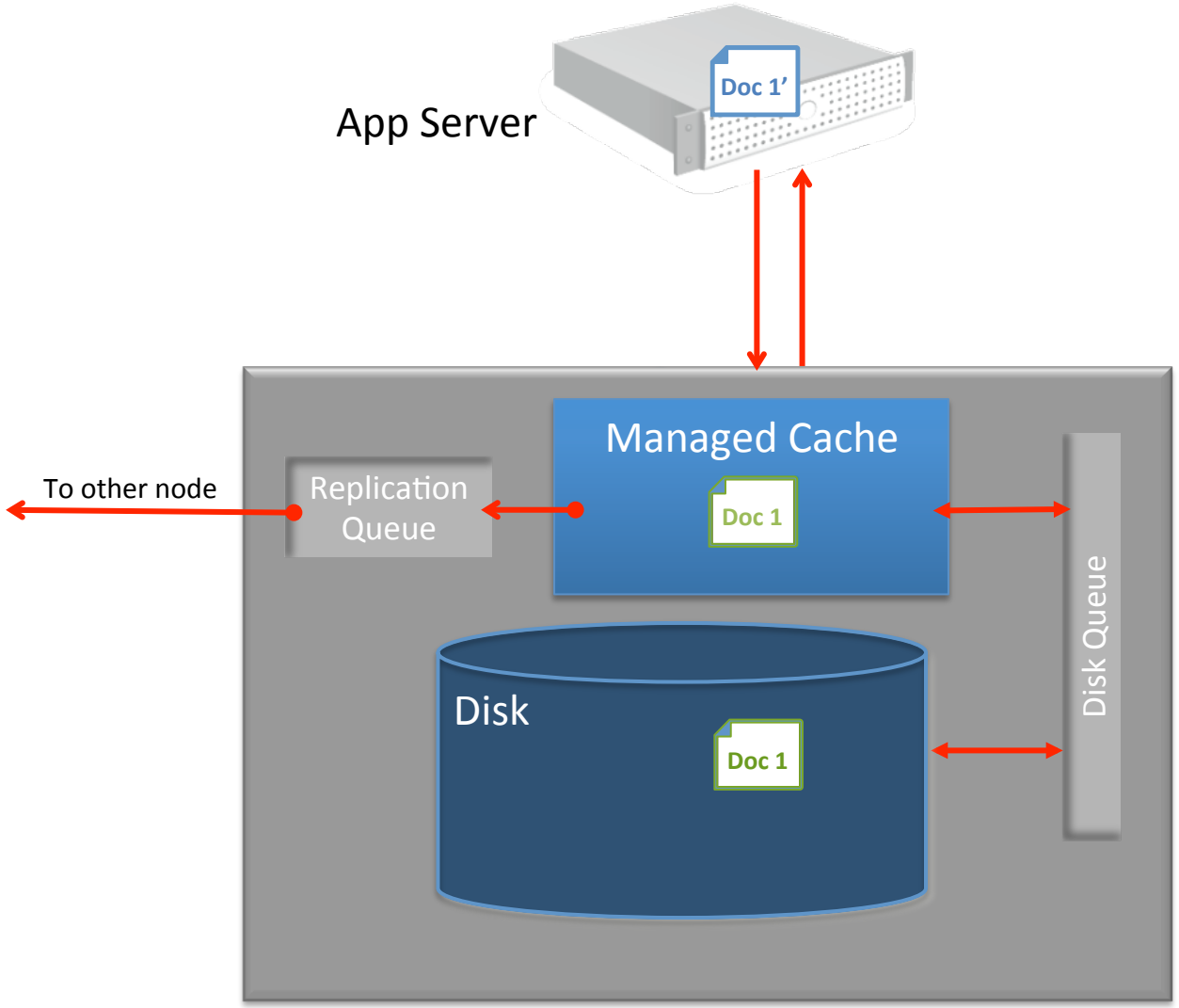


COUCHBASE OPERATIONS

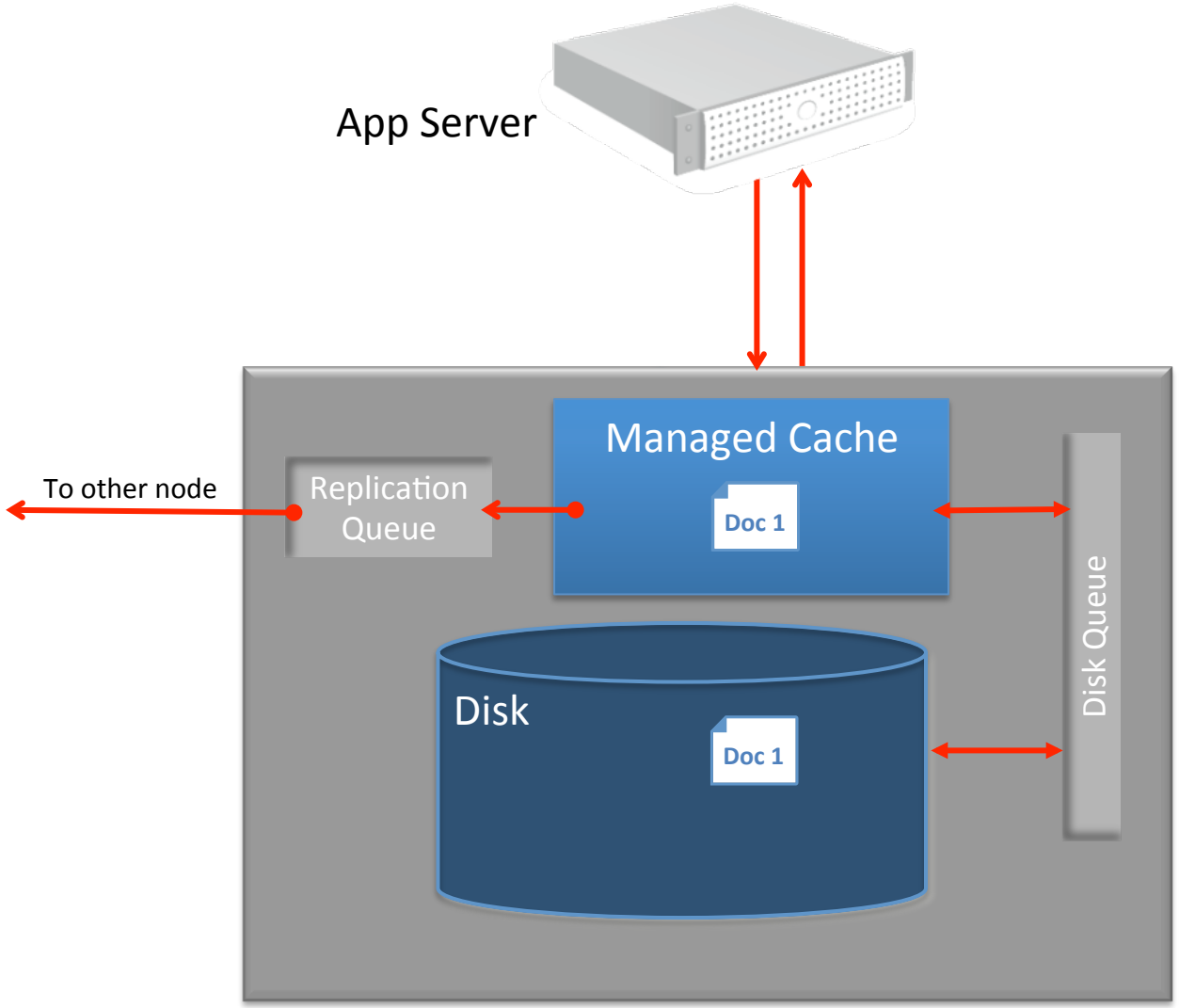
Single node - Couchbase Write Operation



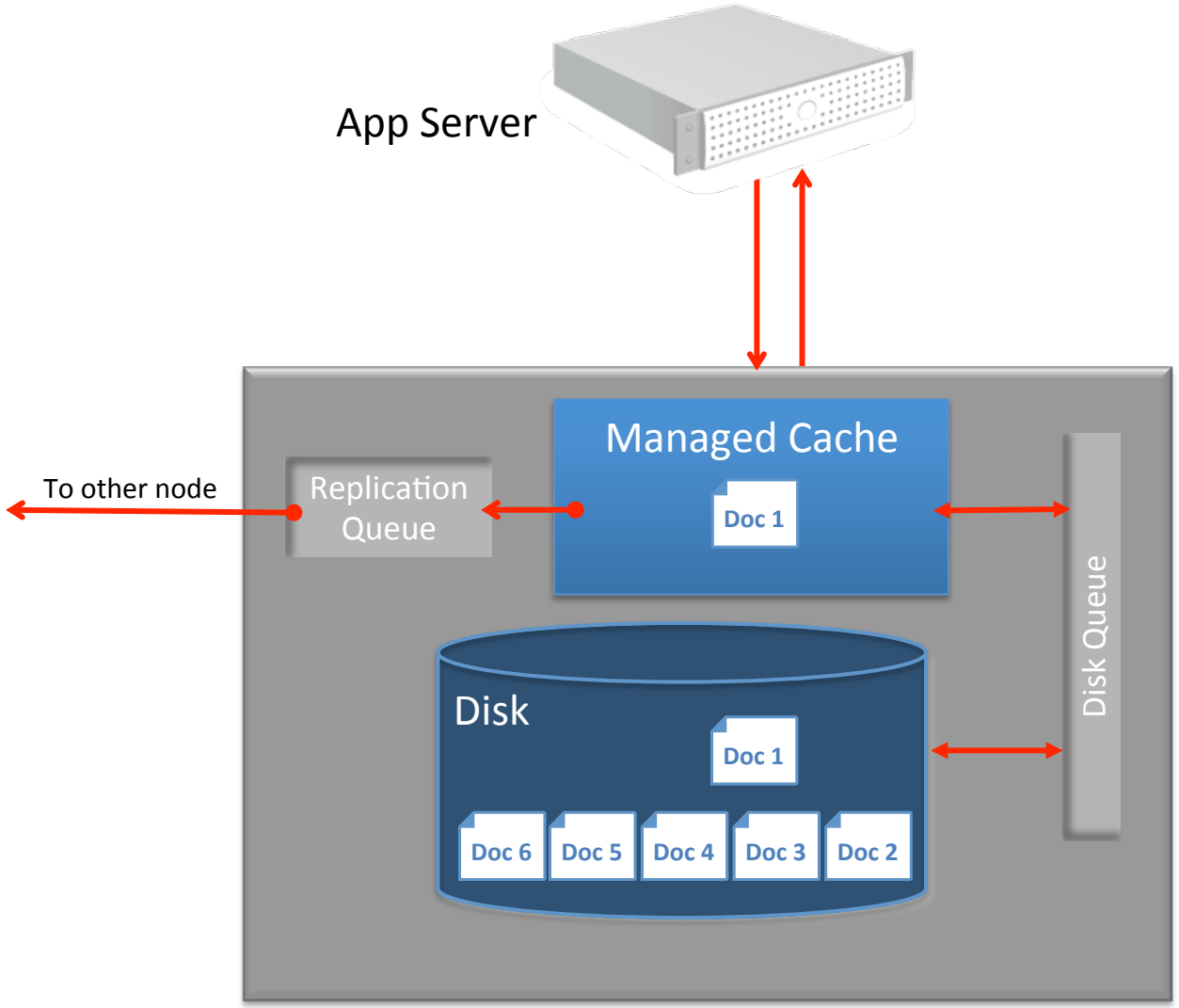
Single node - Couchbase Update Operation



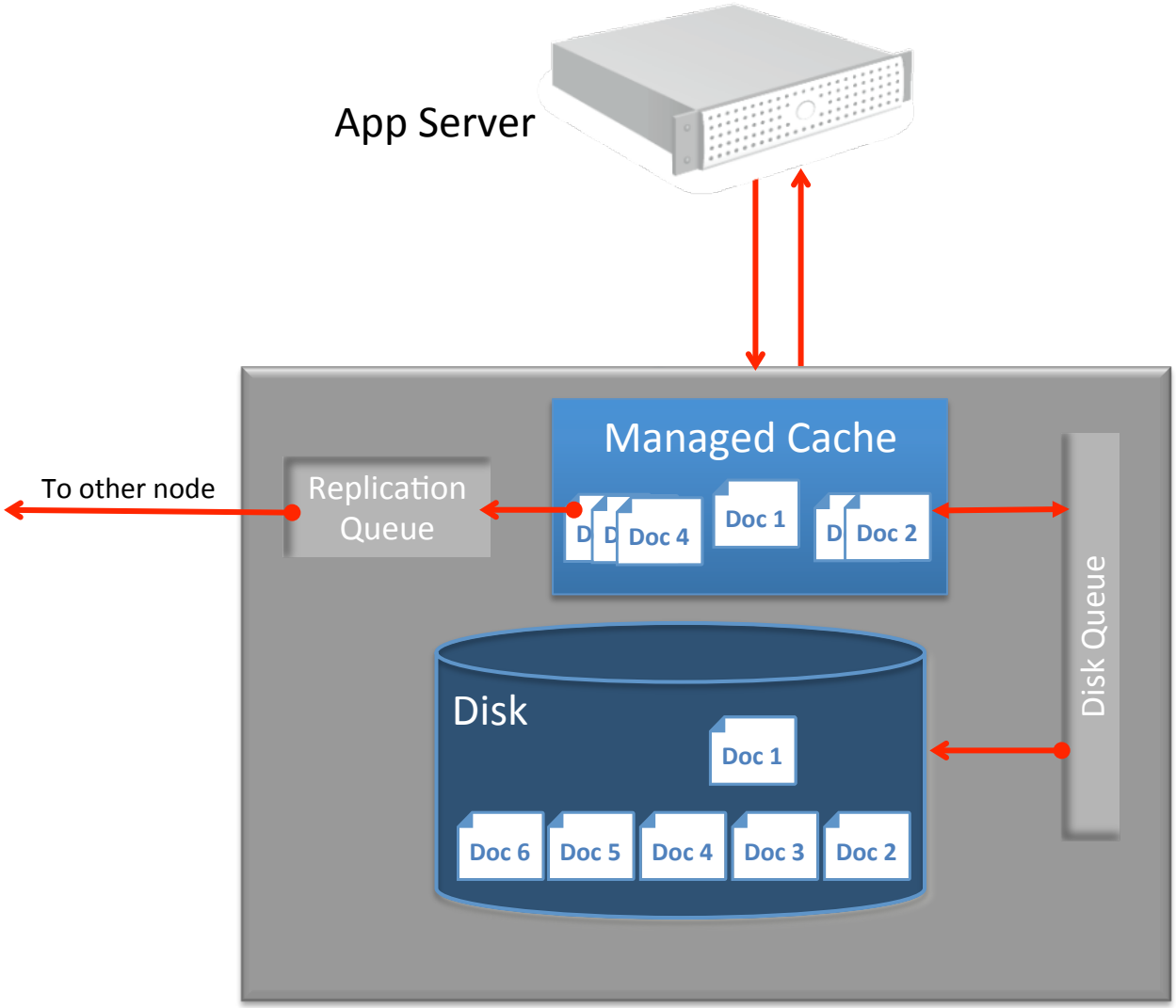
Single node - Couchbase Read Operation



Single node - Couchbase Cache Eviction

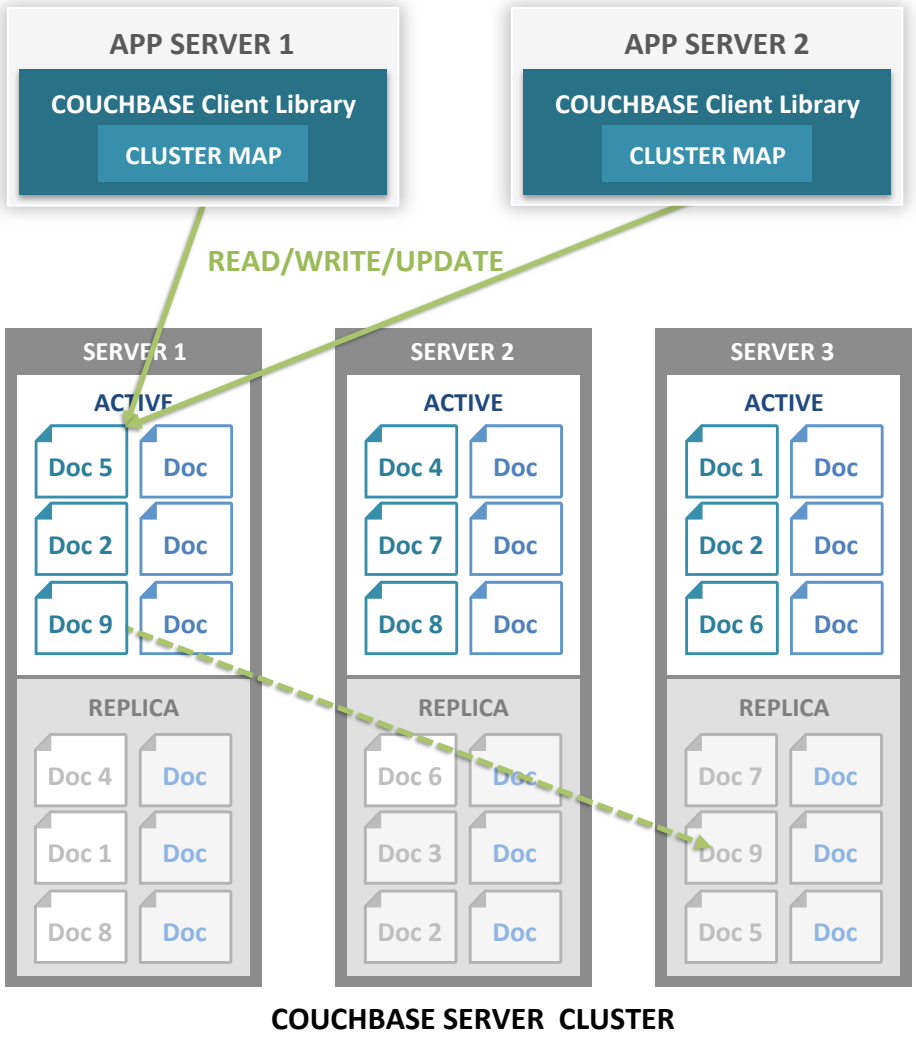


Single node – Couchbase Cache Miss



Couchbase Server Node

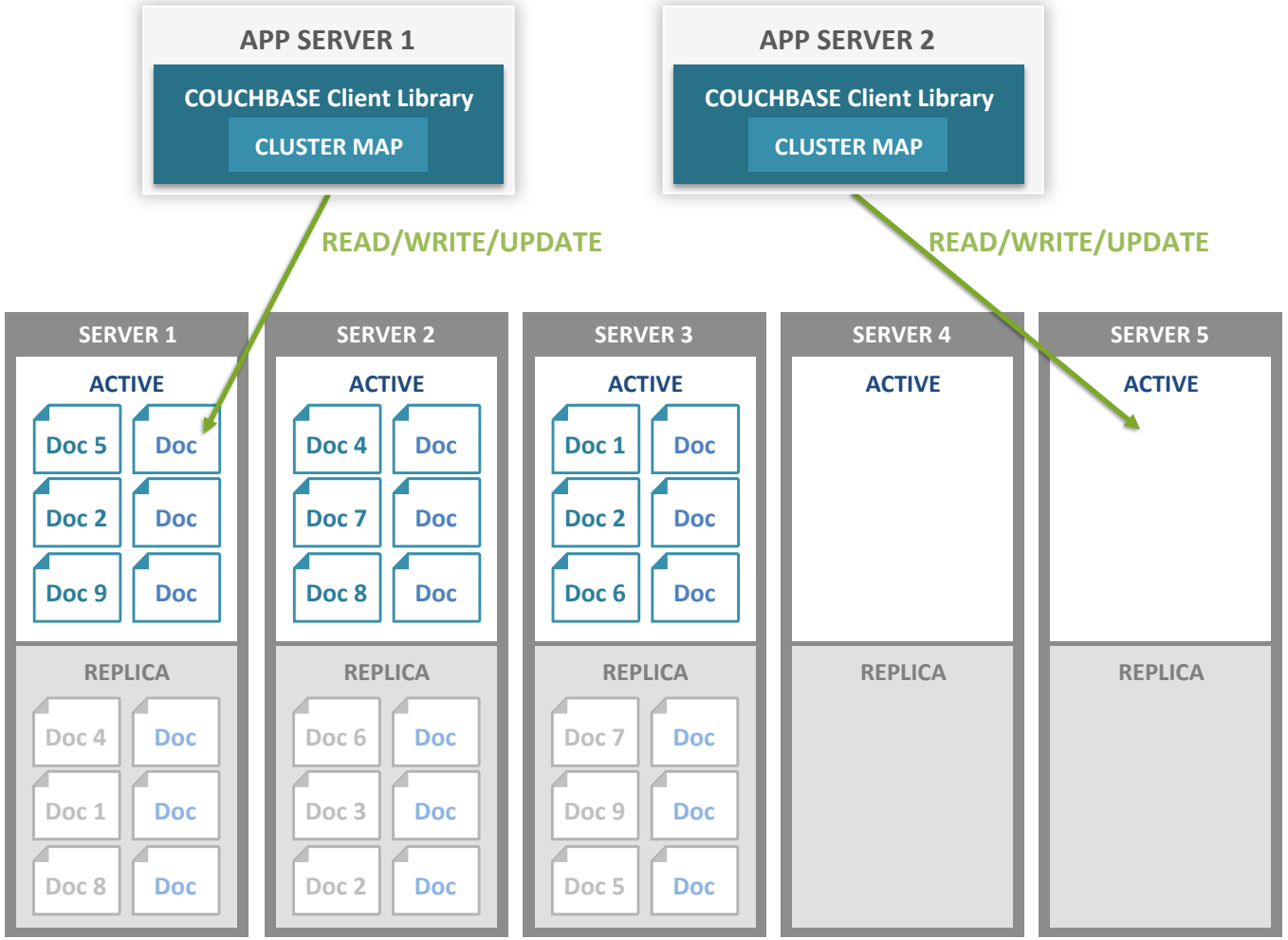
Cluster wide - Basic Operation



- **Docs distributed evenly across servers**
- **Each server stores both active and 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**

User Configured Replica Count = 1

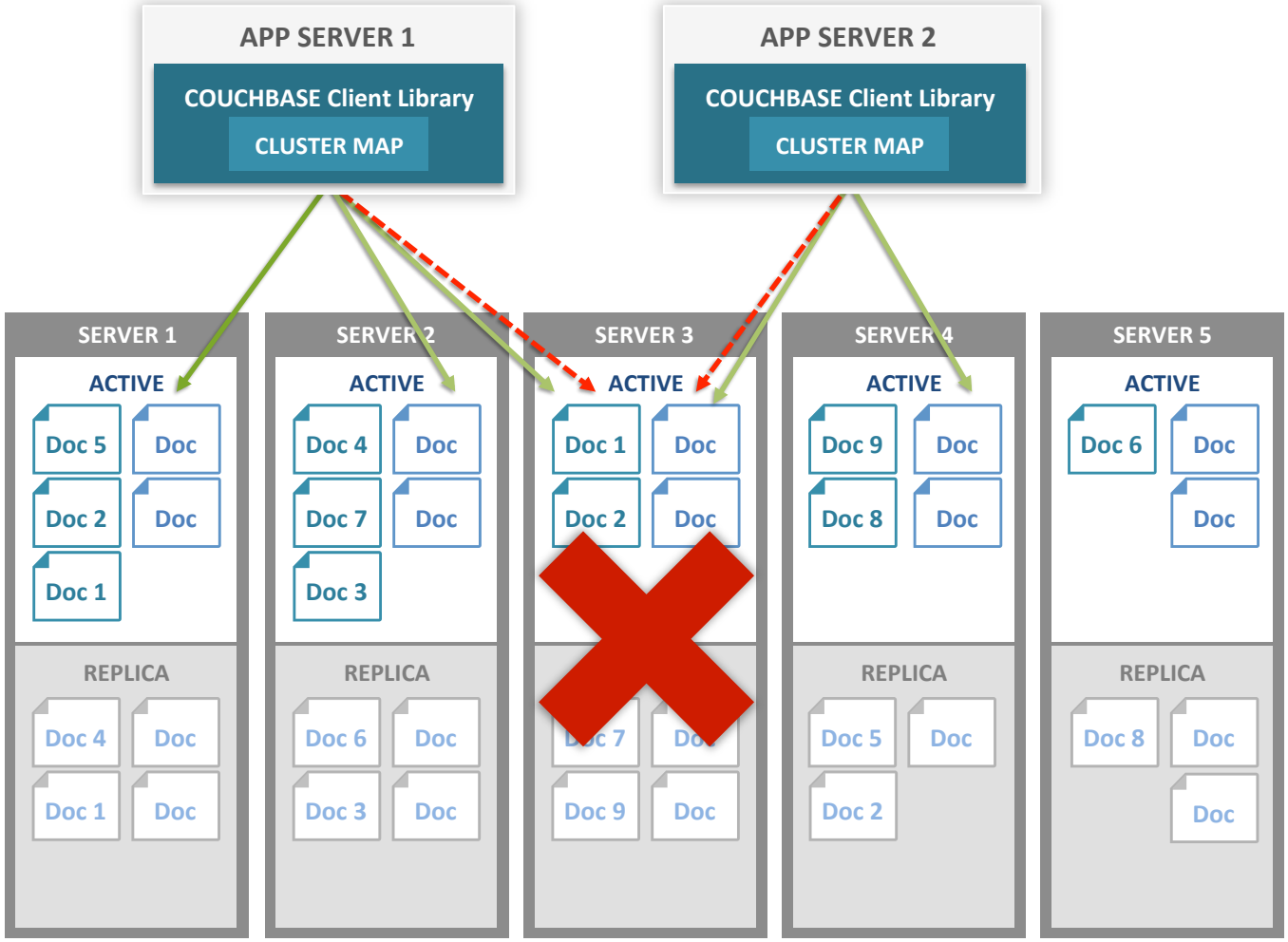
Cluster wide - Add Nodes to Cluster



- **Two servers added 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 number of servers**

User Configured Replica Count = 1

Cluster wide - Fail Over Node



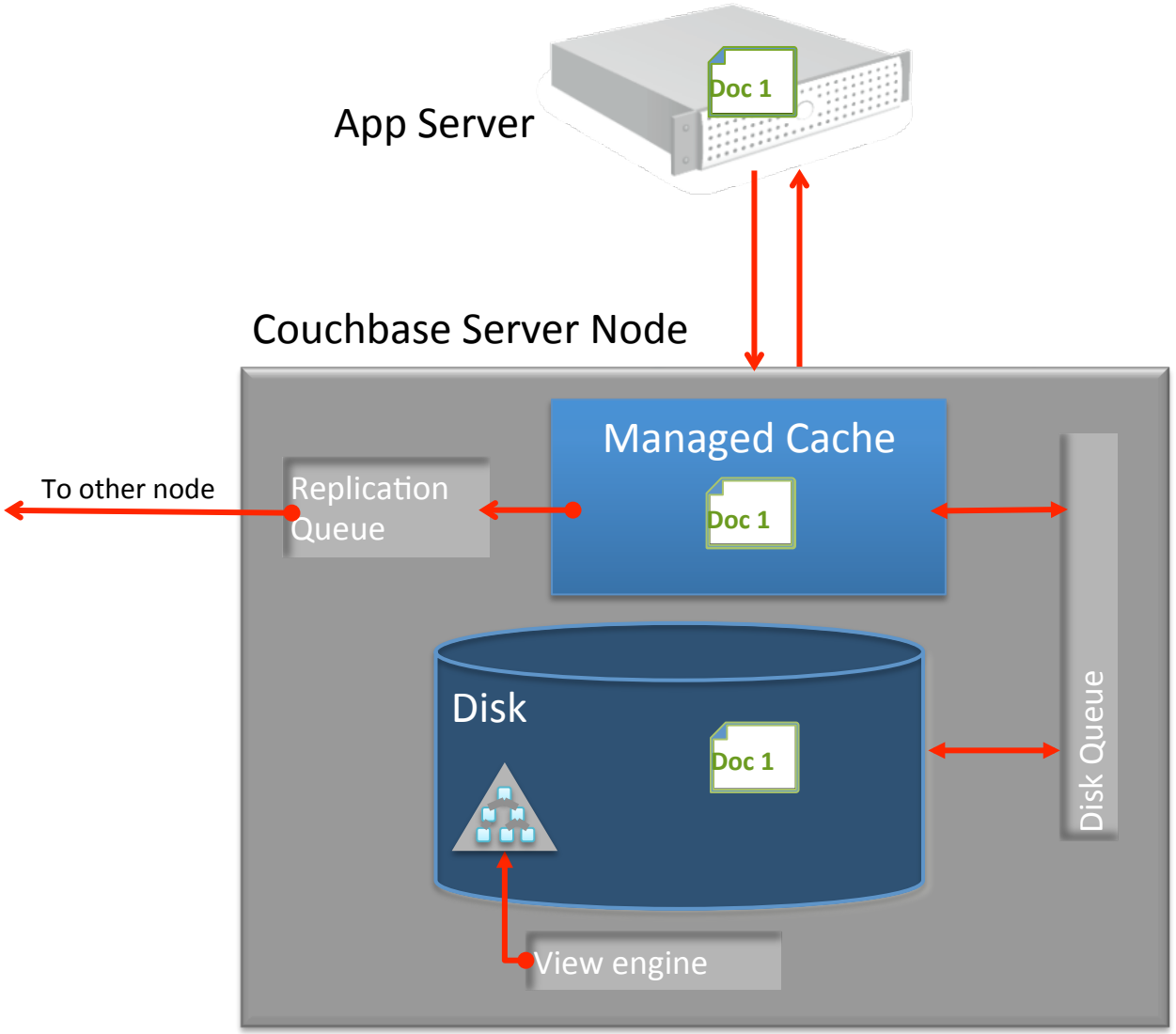
- **App servers accessing docs**
- **Requests to Server 3 fail**
- **Cluster detects server failed**
  Promotes replicas of docs to active
  Updates cluster map
- **Requests for docs now go to appropriate server**
- **Typically rebalance would follow**

User Configured Replica Count = 1

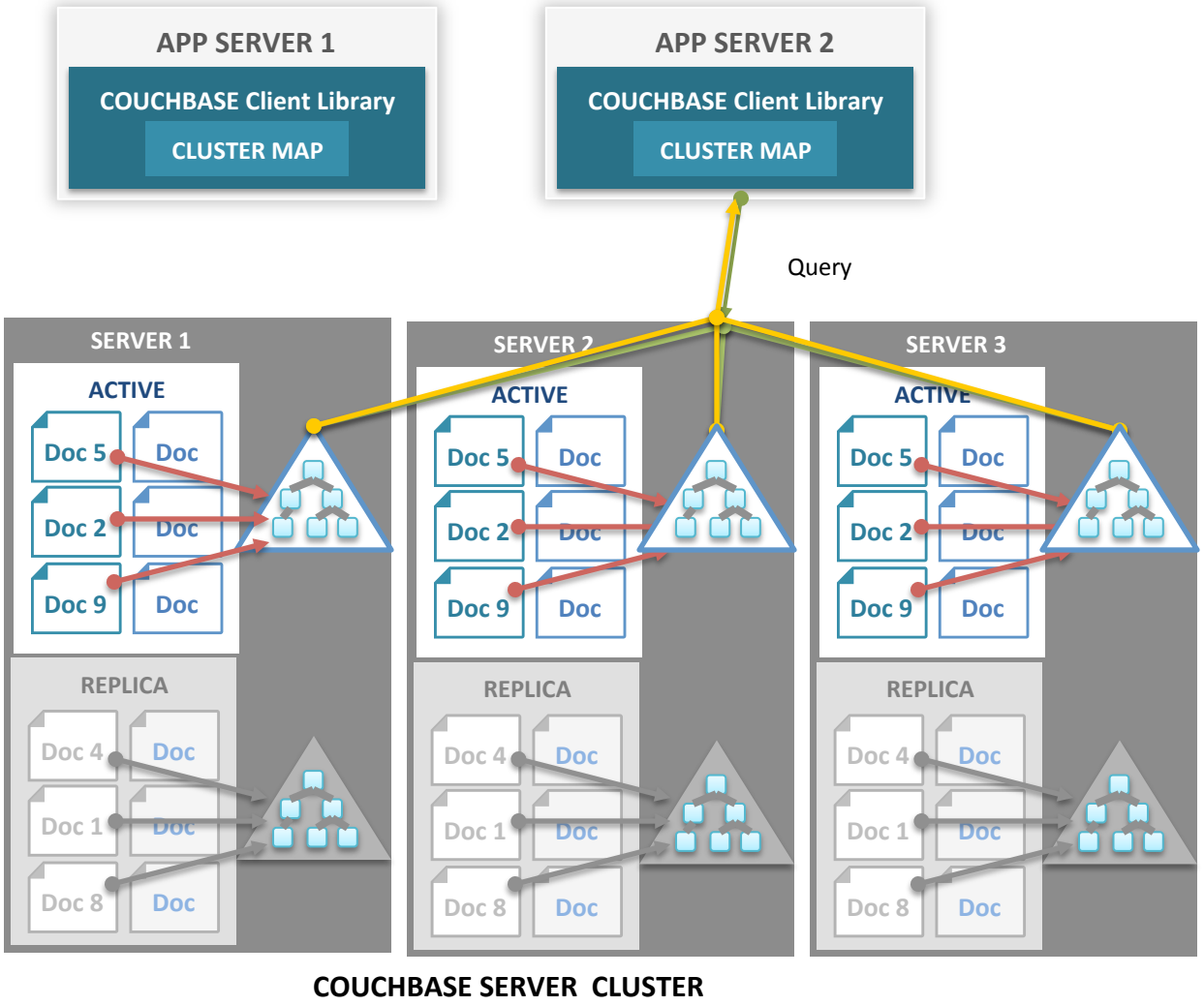
Indexing and Querying – The basics

- 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

Eventually indexed Views – Data flow



Cluster wide - Indexing and Querying



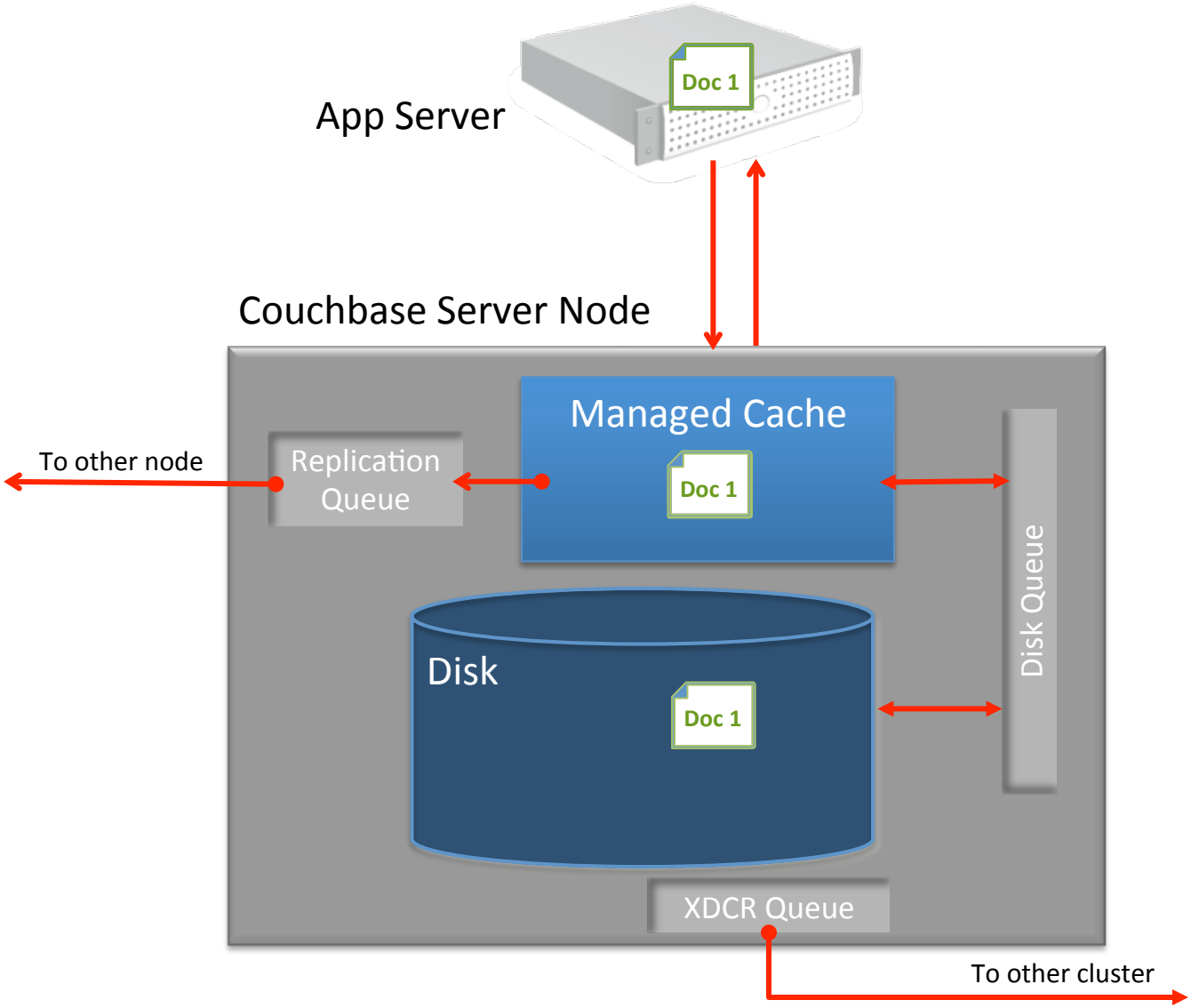
- **Indexing work is distributed amongst nodes**
- **Large data set possible**
- **Parallelize the effort**
- **Each node has index for data stored on it**
- **Queries combine the results from required nodes**

User Configured Replica Count = 1

Cross Data Center Replication – The basics

- Replicate your Couchbase data **across clusters**
- Clusters may be spread across geos
- Configured on a per-bucket basis
- Supports unidirectional and bidirectional operation
- Application can read and write from both clusters (active – active replication)
- Replication throughput scales out linearly
- Different from intra-cluster replication

Cross data center replication – Data flow



DEMO TIME