Introduction to Couchbase: A NoSQL Document Database

Daniel Owen

Software Engineer

Couchbase

Couchbase NoSQL Leadership

- **Leading NoSQL database company**
  Open Source development & business model

- **Document-oriented NoSQL database**
  Focused on interactive internet and mobile applications

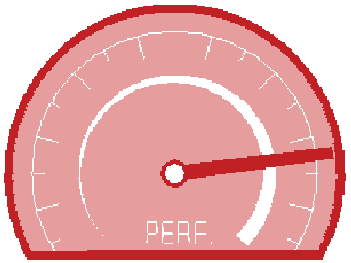
- **Provide more flexible, higher performance, more scalable database than relational alternative**

- **Most mature, reliable and widely deployed solution**
  >7,500 paid production deployments worldwide

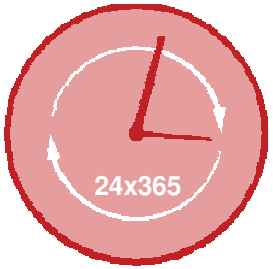
- **Headquarters in Silicon Valley (Mountain View, CA)**
  ~100 employees including >50 in engineering/product
  >80% of commits to Couchbase, memcached, Apache CouchDB

Couchbase

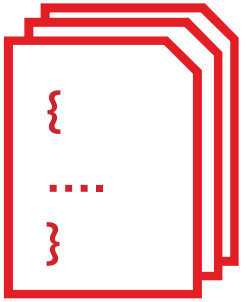
Easy Scalability



Consistent High Performance

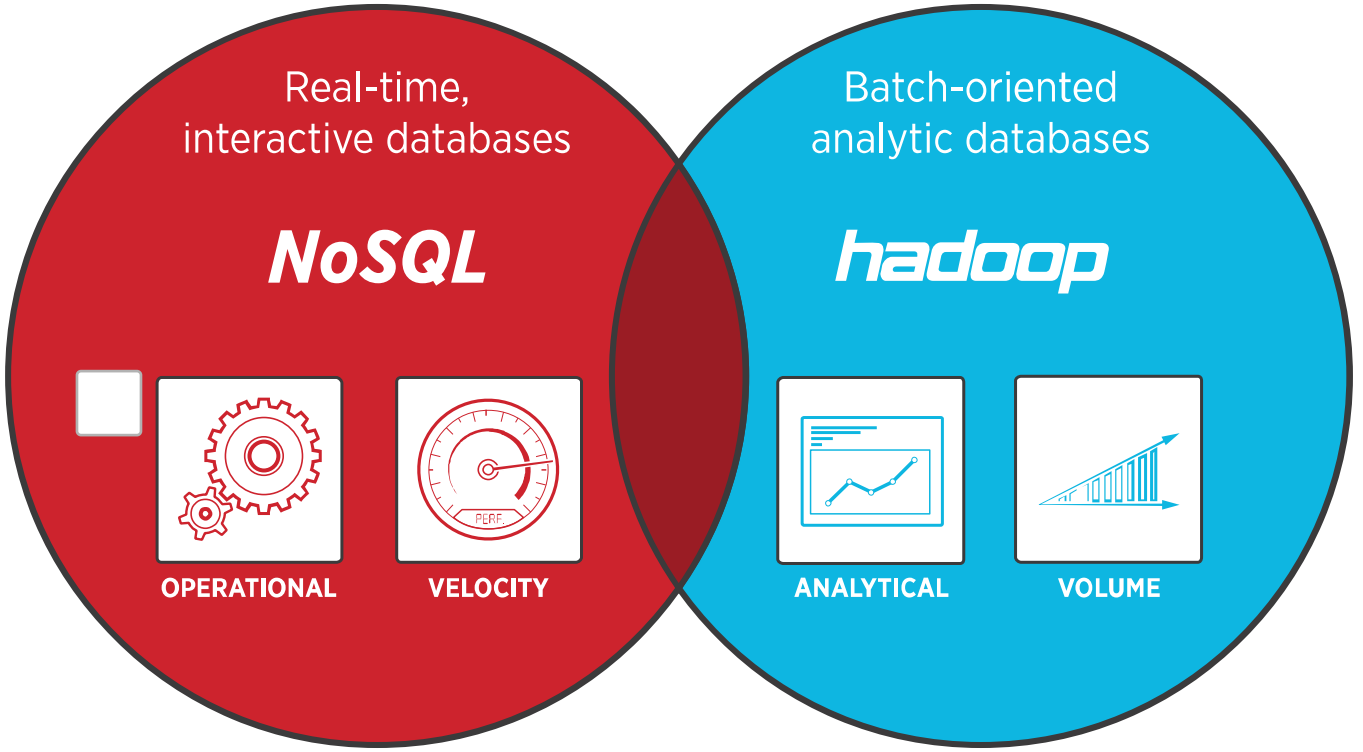


Always On 24x7x365



Flexible Data Model

Big Data = Operational + Analytic (NoSQL + Hadoop)



- Online
- Web/Mobile/IoT apps
- Millions of customers/consumers

- Offline
- Analytics apps
- Hundreds of business analysts

Couchbase Server is Open Source

- All code available on github
- Community Edition for free
- Dedicated to developers
- Try building it yourself!

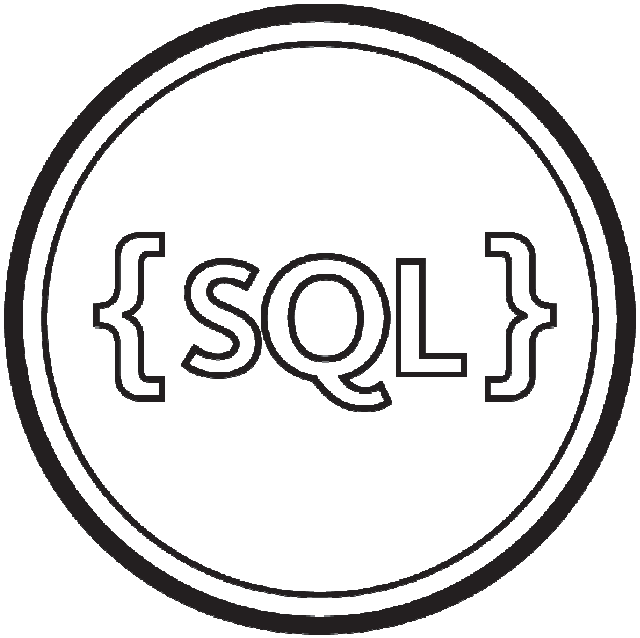
github/couchbase/tlm

NoSQL == “Not Only SQL”

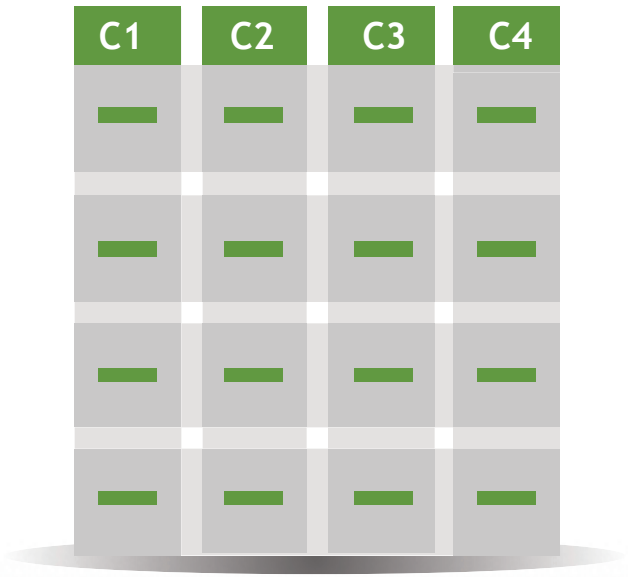
- Users **want** a familiar query language
- SQL has worldwide popularity and recognition
- N1QL allows SQL queries to be run at JSON documentse

Couchbase

N1QL = JSON + SQL



Relational vs Document Data Model



Relational data model

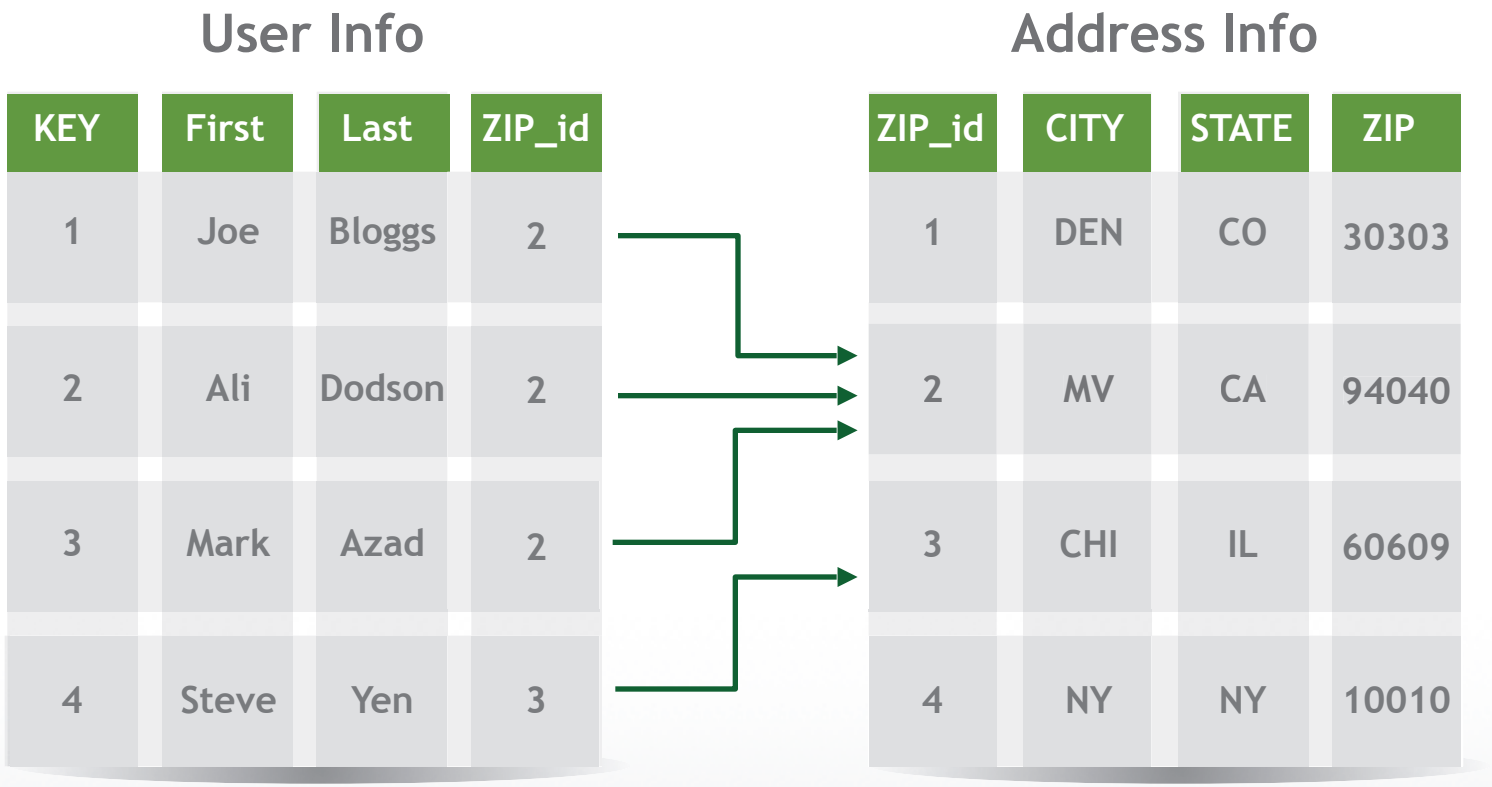
Highly-structured table organization with rigidly-defined data formats and record structure.

Document data model

Collection of complex documents with arbitrary, nested data formats and varying “record” format.

Couchbase

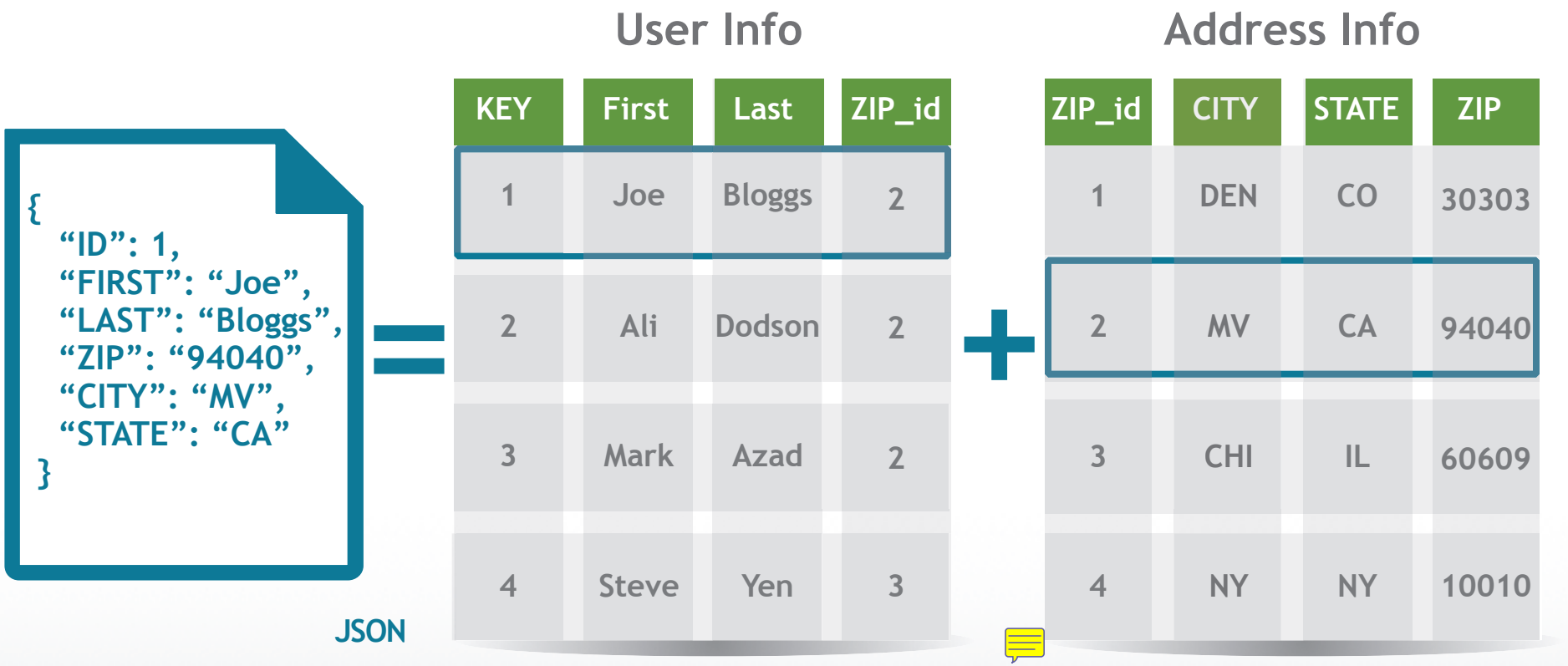
RDBMS Example: User Profile



To get info about specific user, you perform a join across two tables

Couchbase

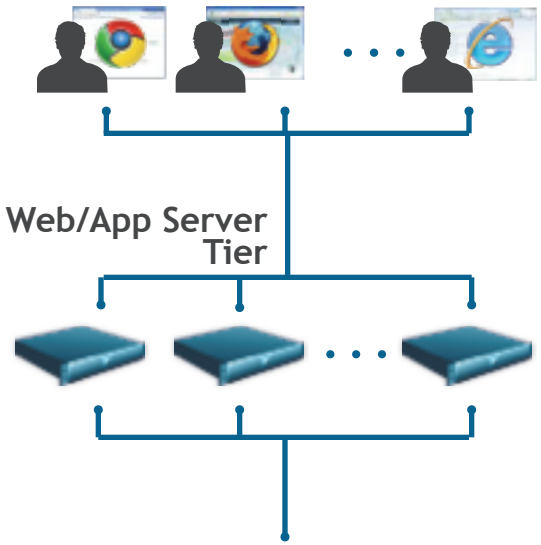
Document Example: User Profile



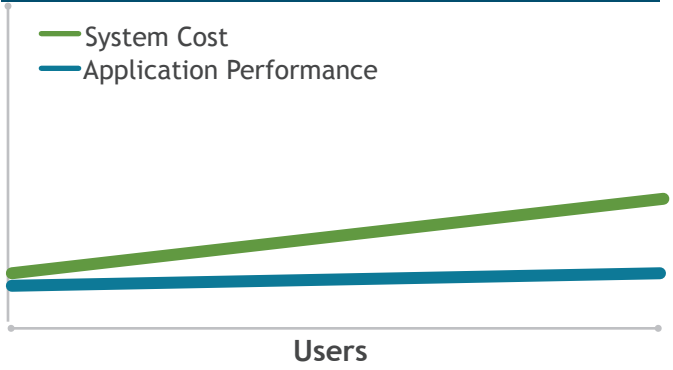
All data in a single document

Couchbase

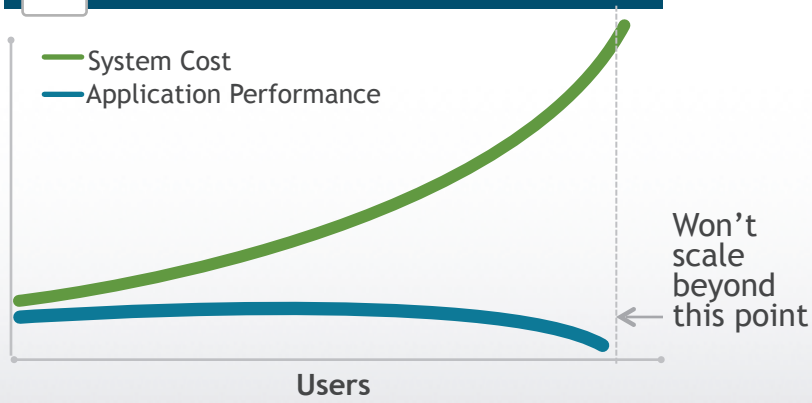
Relational Technology Scales Up



Application Scales Out
Just add more commodity servers

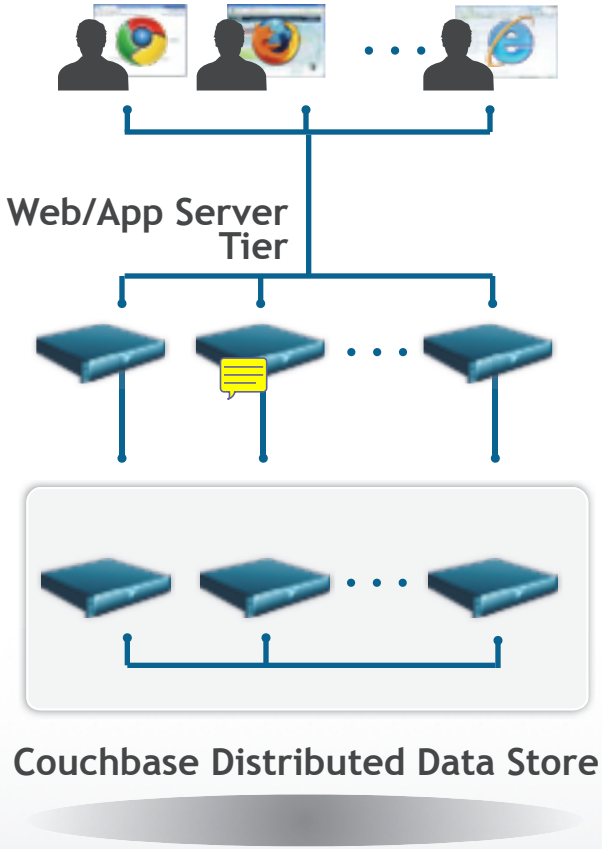


RDBMS Scales Up
a bigger, more complex server

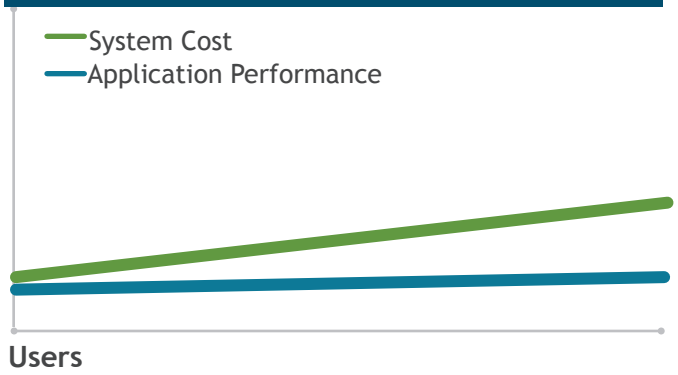


Expensive and disruptive sharding, doesn't perform at web scale

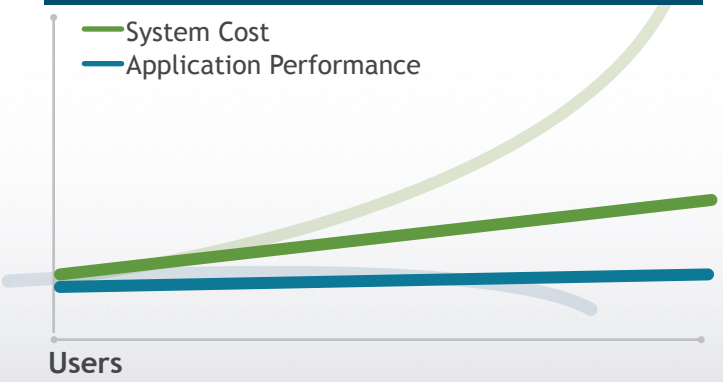
Couchbase Server Scales Out Like App Tier



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

Relational Target Market

Ideal for workloads requiring

(1) **ACID properties**

- Atomicity
- Consistency
- Isolation
- Durability

(2) **Ad Hoc Queries ****

(3) **High Security**

- Full authentication model
- Full auditing
- etc.

** Couchbase currently developing N1QL

Market

- Transaction Processing
  - Billing
  - Banking
- High Security
  - Financial market
  - Health industry
- Small number of users
  - Scale-out not required

NOSQL Market

Internet Companies

- Social Gaming
- Ad Networks
- Social Networks
- Online Business Services
- E-Commerce
- Online Media
- Content Management
- Cloud Services

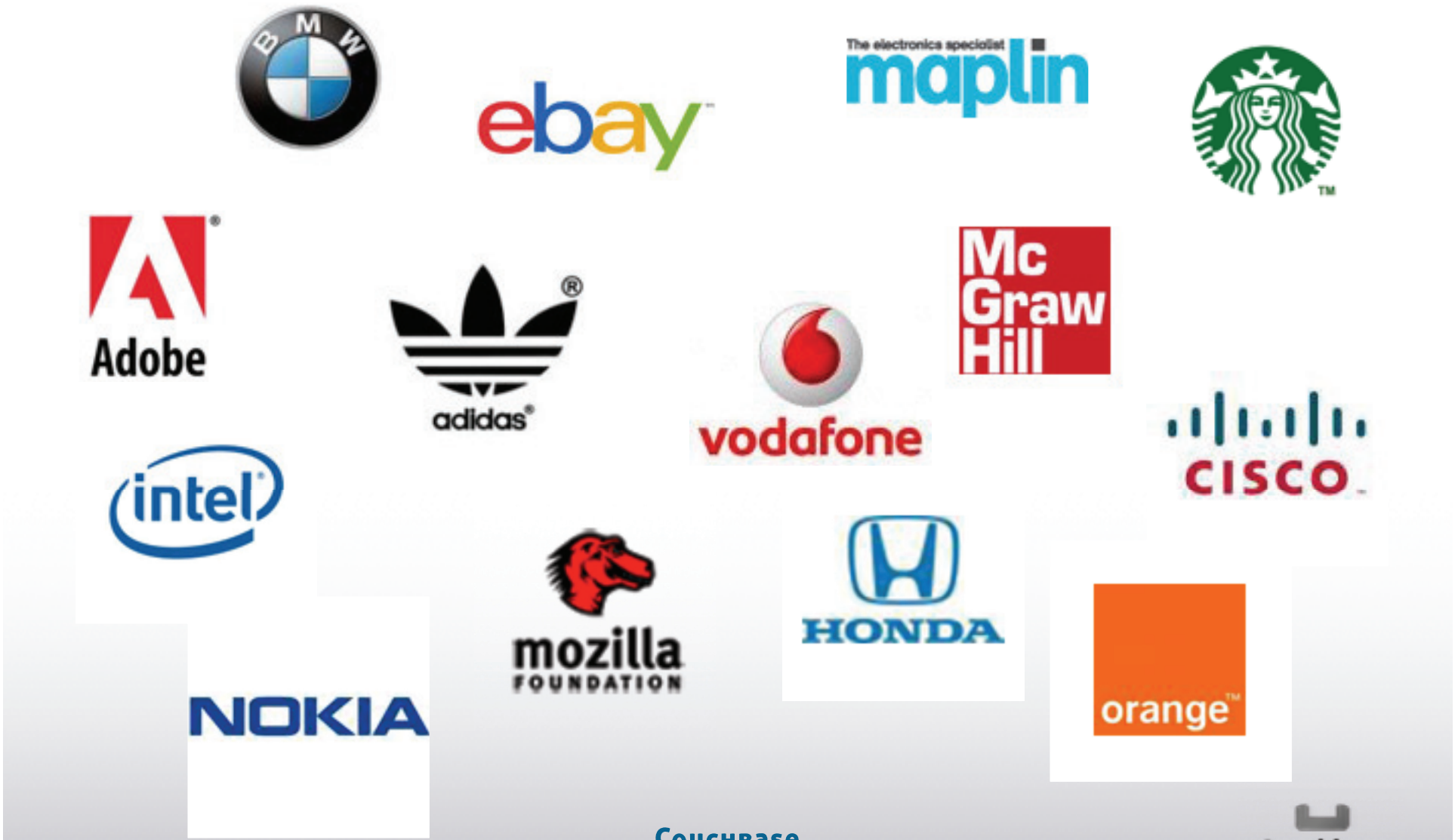
Enterprises

- Communications
- Retail
- Financial Services
- Health Care
- Automotive/Airline
- Agriculture
- Consumer Electronics
- Business Systems

NOSQL Use Cases

Web app or Use-case	Couchbase Solution
Content Store & Metadata System	Couchbase document store + Elastic Search
Social Game & Mobile App	Couchbase store game and player data
Ad Targeting	Couchbase stores user information for fast access
User Profile Store	Couchbase Server as a key-value store
Session Store	Couchbase Server as a key-value store
High Availability Caching Tier	Couchbase Server as a memcached tier replacement
Chat/Messaging Platform	Couchbase Server

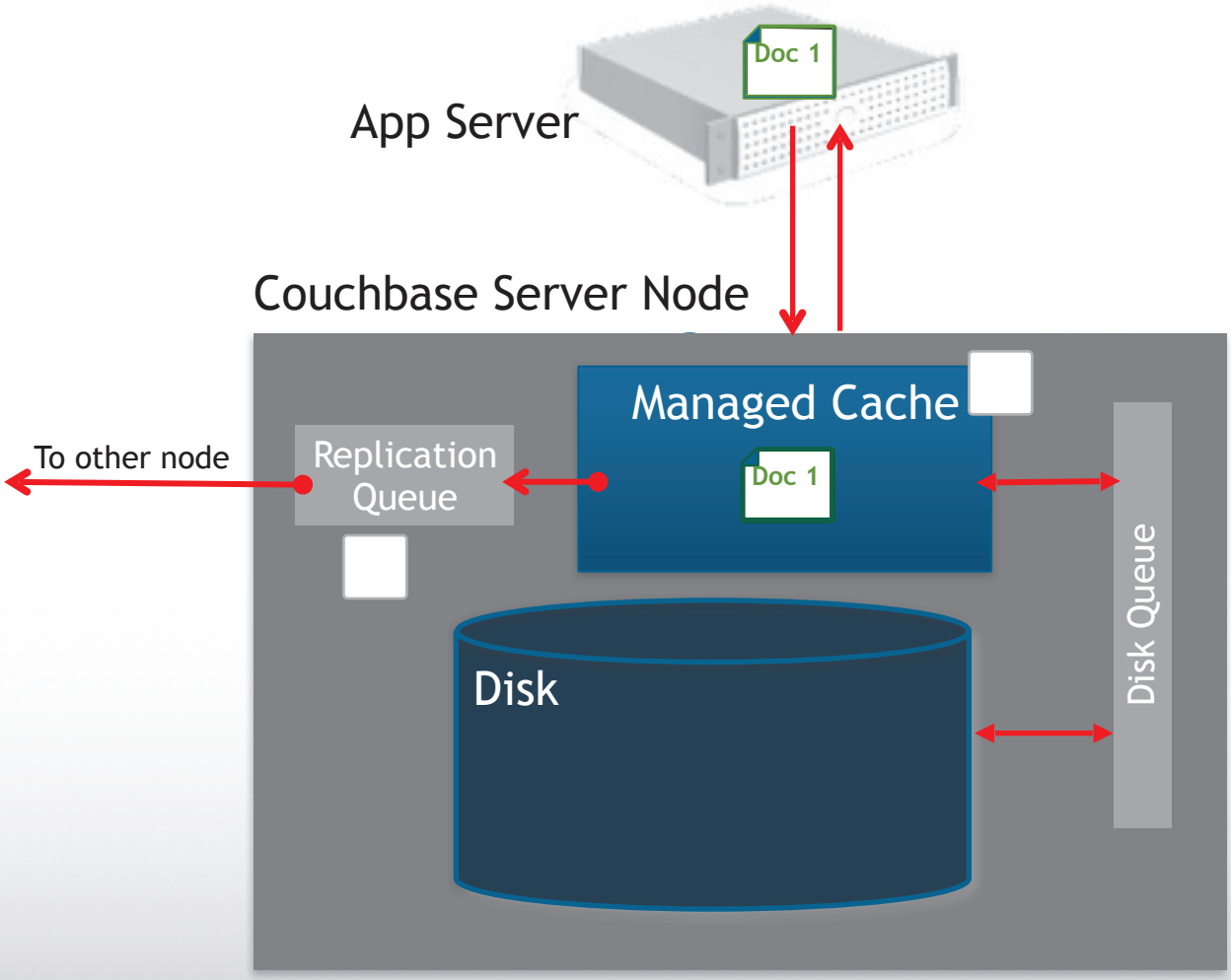
Some of Couchbase Customers



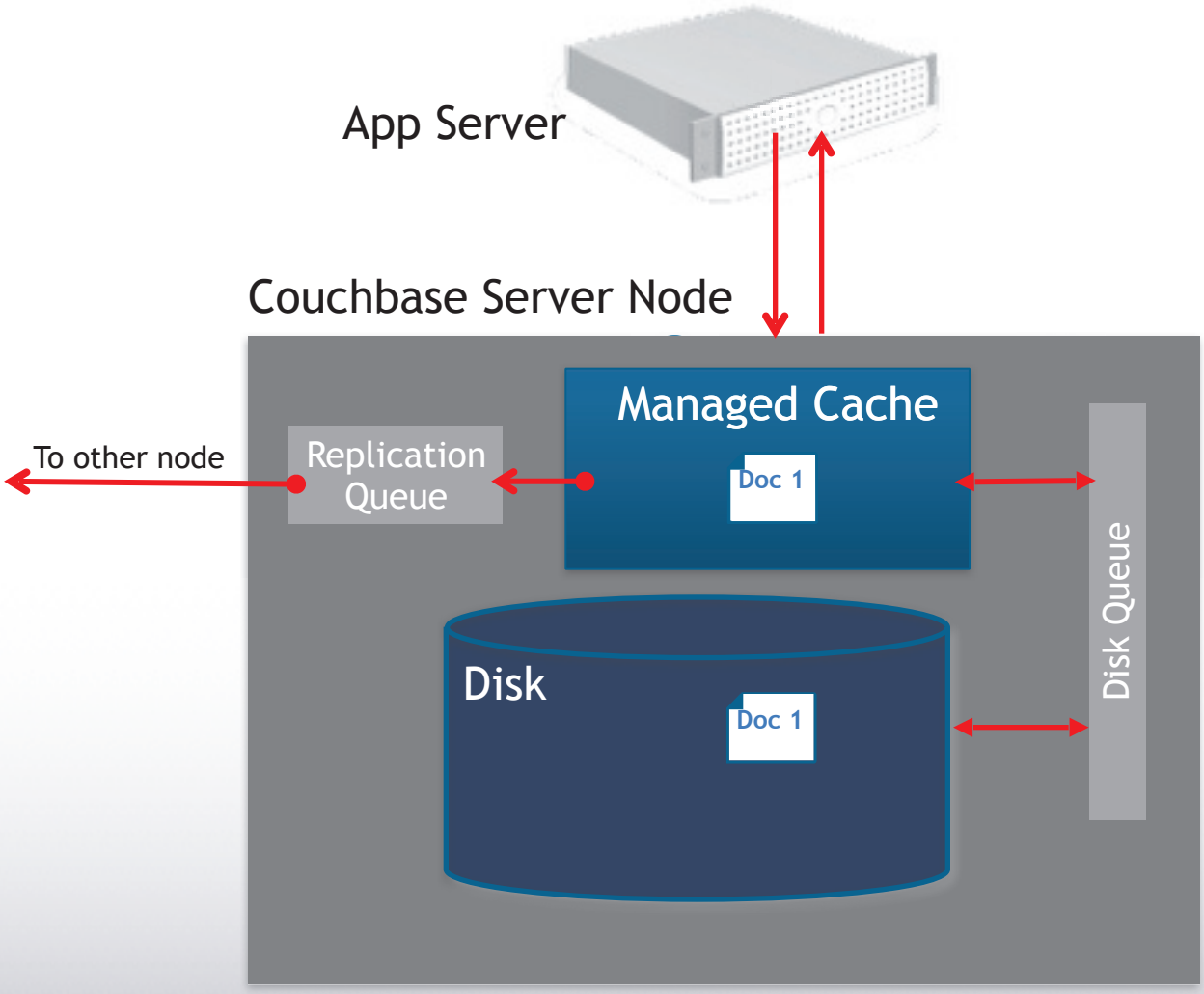
Couchbase

Couchbase Operations

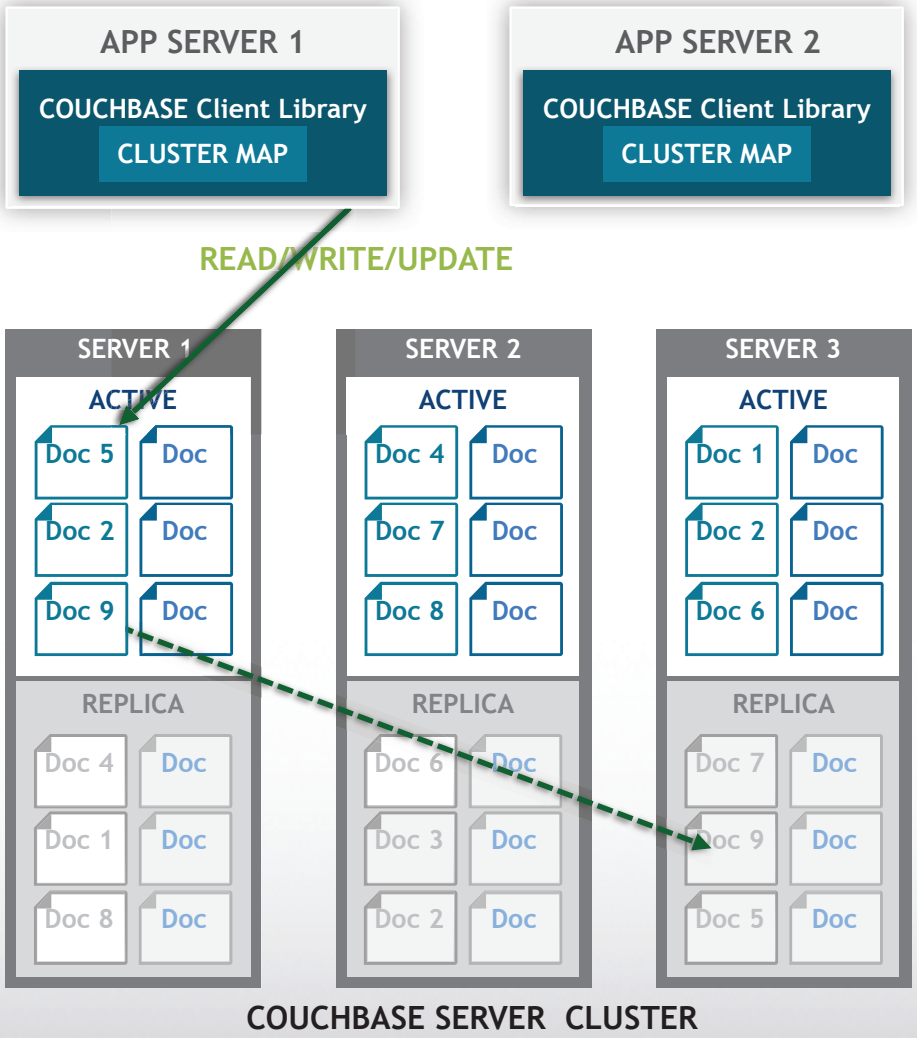
Single node - Couchbase Write Operation



Single node - Couchbase Read Operation



Basic Operation

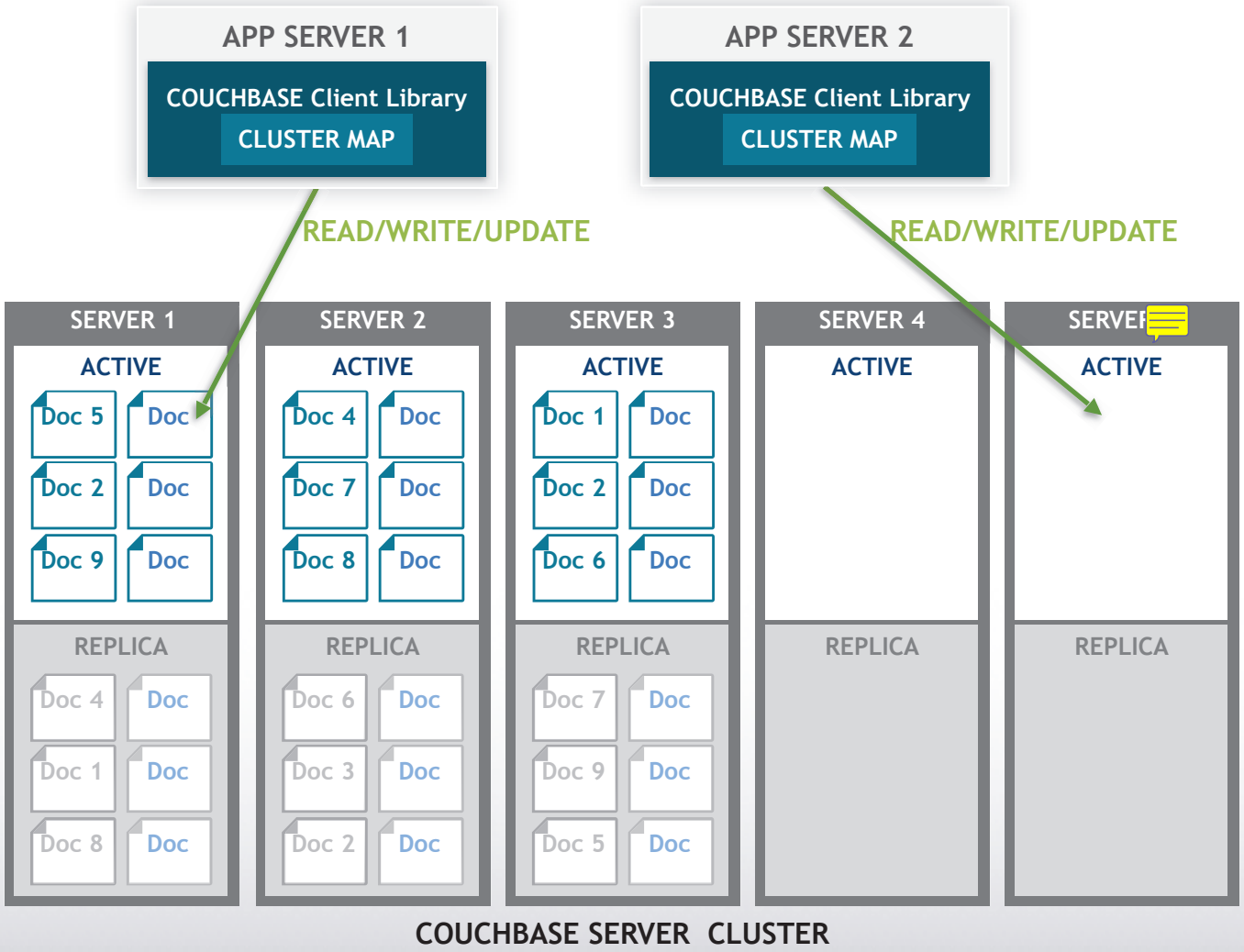


- Docs distributed evenly across servers
- Each server stores both active and replica docs
  Only one server active at a time
- Client library provides app with simple interface to database
- Cluster map provides map to which server doc is on
  App never needs to know
- App reads, writes, updates docs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

User Configured Replica Count = 1

Couchbase

Add Nodes to Cluster

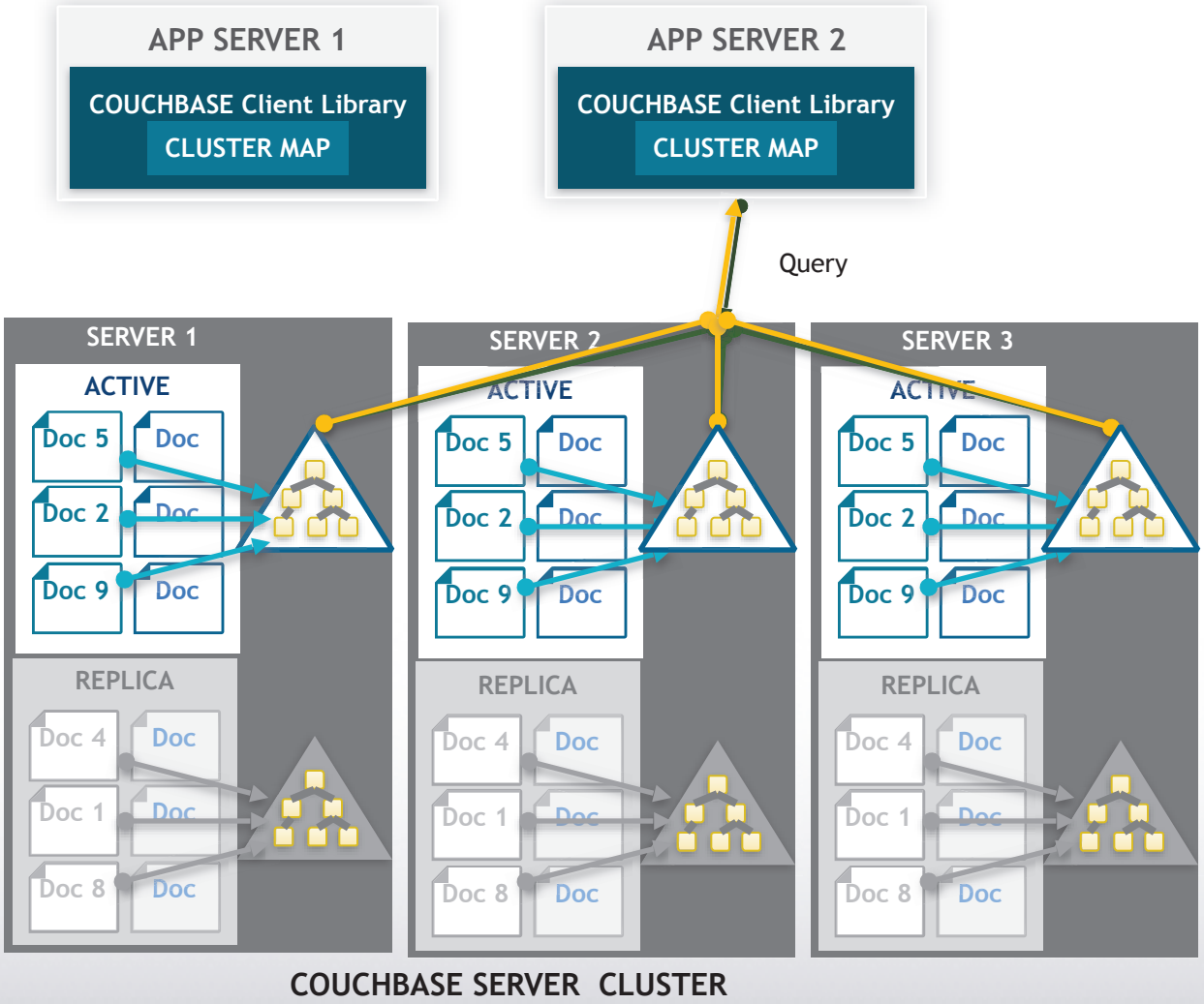


- Two servers added One-click operation
- Docs automatically rebalanced across cluster
  Even distribution of docs
  Minimum doc movement
- Cluster map updated
- App database calls now distributed over larger number of servers

User Configured Replica Count = 1

Couchbase

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

Couchbase