

Data Management for Interactive Applications

NoSQL Document Database





► **Provide KV & document database for web & mobile apps**

Flexible, schemaless JSON data model; 100% open source

► **Only NoSQL vendor with mobile database + sync**

Recognized as most innovative NoSQL vendor

► **Mission critical deployments at large enterprises & internet companies**

> 400 customers; >10,000 paid production nodes deployed; worldwide customer base



**High Availability
Cache**



Key Value



Document



Mobile device

Market Adoption

Internet Companies



amADEUS, ORBITZ Sabre



Aol. adscale criteo. DG

Rakuten skyscanner xo group/



[> 400 customers; > 10,000 production nodes deployed]

Enterprises



NoSQL Database Considerations



Consistent High Performance

Always awesome experience for your application users



Always On 24x7x365

The sun never sets on the Internet, your application needs the database to always serve data



Easy Scalability

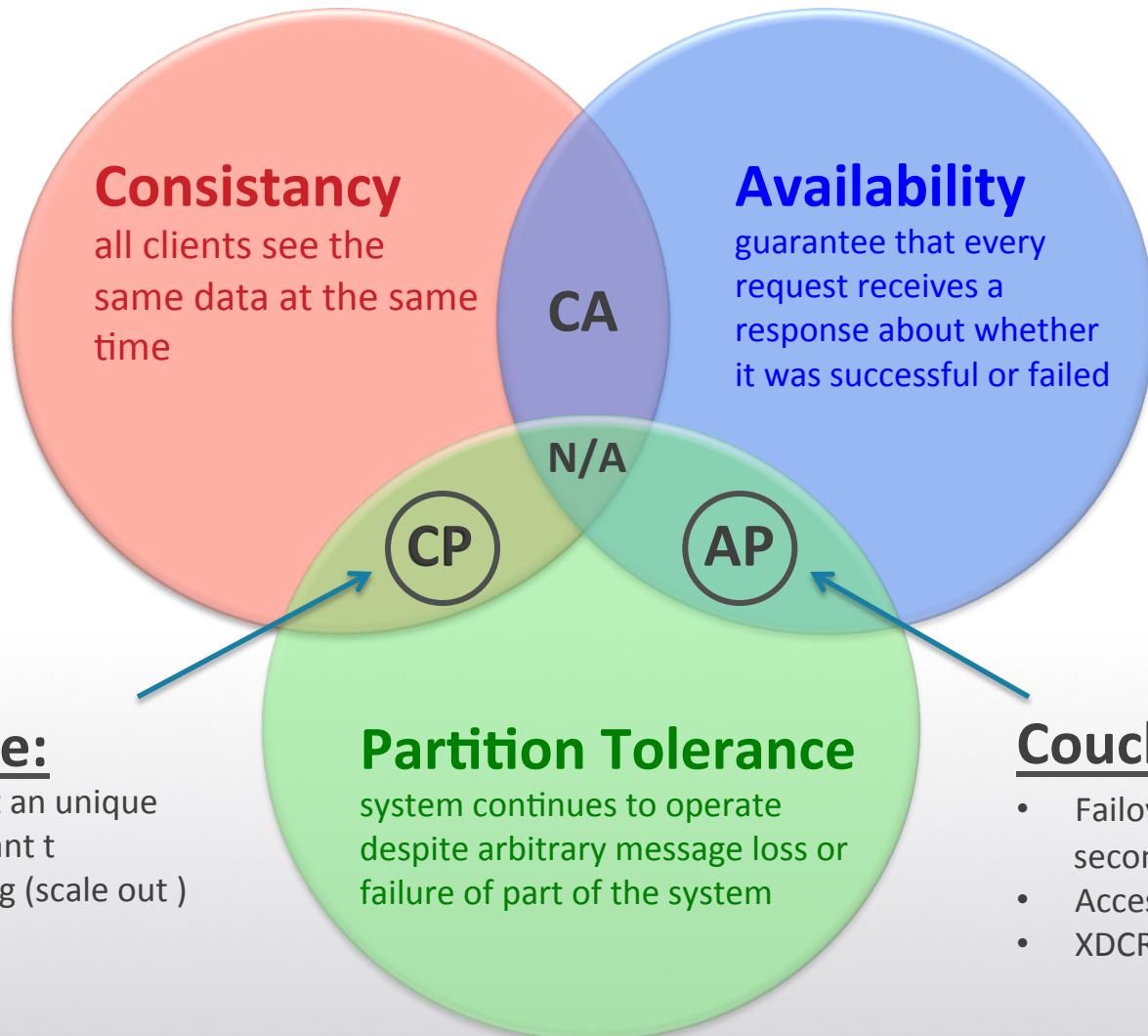
Grow cluster without application changes, without downtime when needed



Flexible Data Model

Keep developers productive and allow fast and easy addition of new features

Couchbase and CAP Theorem





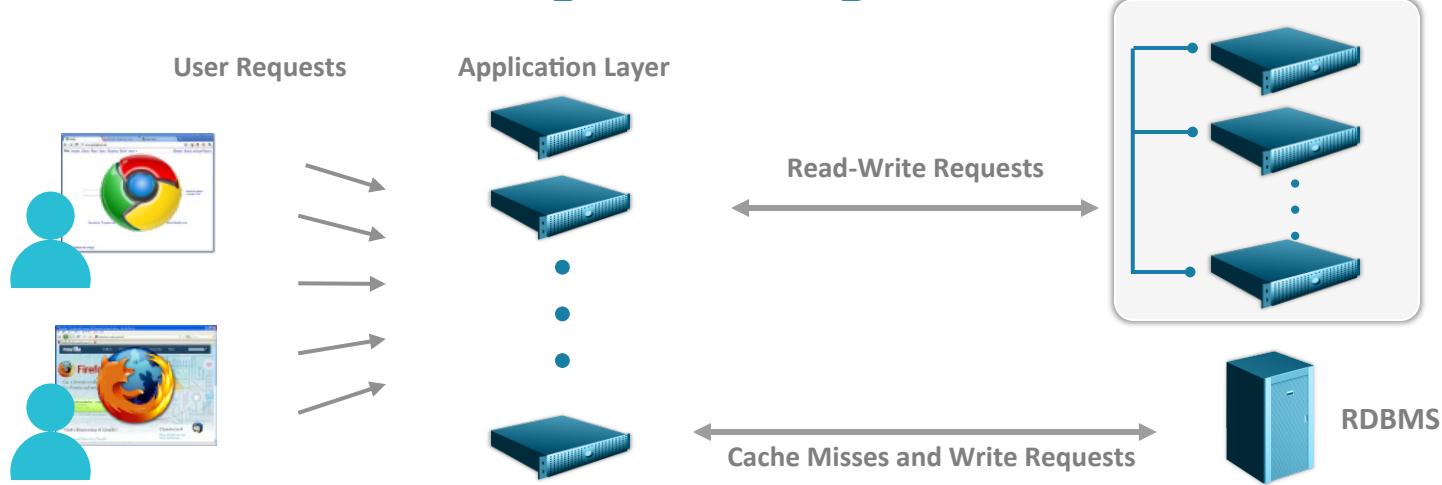
Use Cases

A look into some specific use cases

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

High-Availability Caching

[Use Case 1]



Data cached in Couchbase?

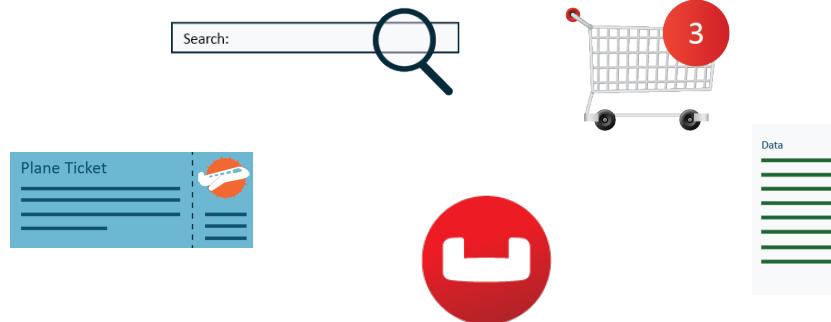
- Application objects
- Popular search query results
- Session information
- Heavily accessed web landing pages

Application characteristic

- Speed up RDBMS
- Consistently low response times for document / key lookups
- High-availability 24x7x365
- Replacement for entire caching tier

Session Store

[Use Case 2]



Data stored in Couchbase?

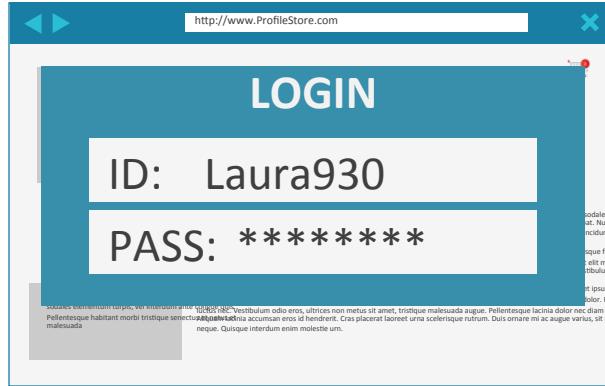
- Session values or Cookies (stored as key-value pairs)
- Examples include: items in a shopping cart, flights selected, search results, etc.

Application characteristic

- Extremely fast access to session data using unique session ID
- Easy scalability to handle fast growing number of users and user-generated data
- Always-on functionality for global user base

Globally Distributed User Profile Store

[Use Case 4]



Data stored in Couchbase?

- User profile with unique ID
- User setting / preferences
- User's network
- User application state

Application characteristic

- Extremely fast access to individual profiles
- Always online system as multiple applications access user profiles
- Flexibility to add and update user attributes
- Easy scalability to handle fast growing number of users

Realtime Computation Engine

[Use Case 6]



Data Collection Layer

Event Broker (dispatch, filter...)

NoSQL Database



Couchbase Distributed Cache

Calculation Processor

Calculation Processor

Calculation Processor

Calculation Processor

RealTime Computation Engine

Application characteristics

Scoring, profiling...

Fraud Detection...

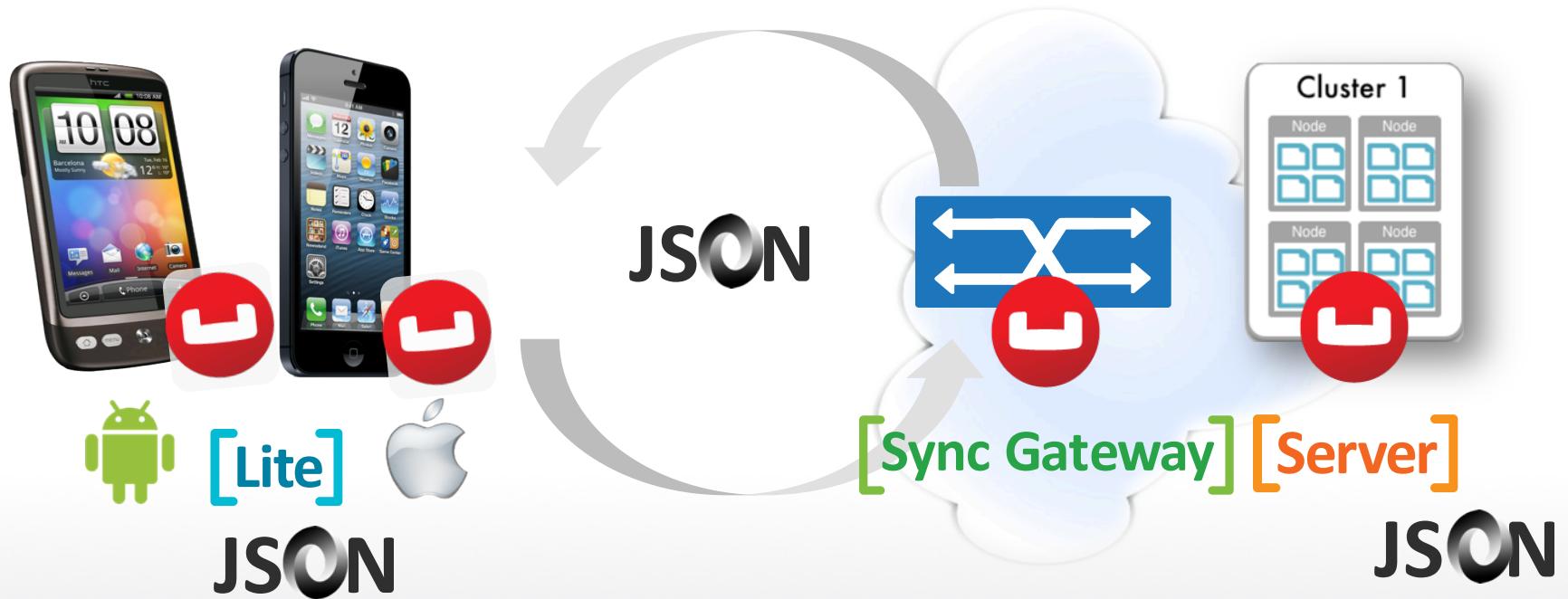
Recommendation Engine

RealTime Metrics

User / System Events

NoSQL Mobile Database

[Use Case 8]



- **JSON on the device**
 - Developers increasingly prefer NoSQL database
- **JSON on the wire**
 - No need for data transformation
- **JSON in the cloud**
 - Flexible data model
 - High performance
 - Easy scalability



Customer Case Studies



- **User profile and device mgmt store**
 - #1 over-the-top communication platform
 - 200M+ users worldwide
- **Wholesale replacement of mongoDB**
- **Chose Couchbase for ability to scalability and performance**
- **Deployed Couchbase in production since August**
- **70+ Server nodes**
- **Trillions of requests per-day with peaks of 120k operations per second, 80/20 reads to writes**
- **Reference customer**



User Profile, Ad Targeting & Real-Time Analytics



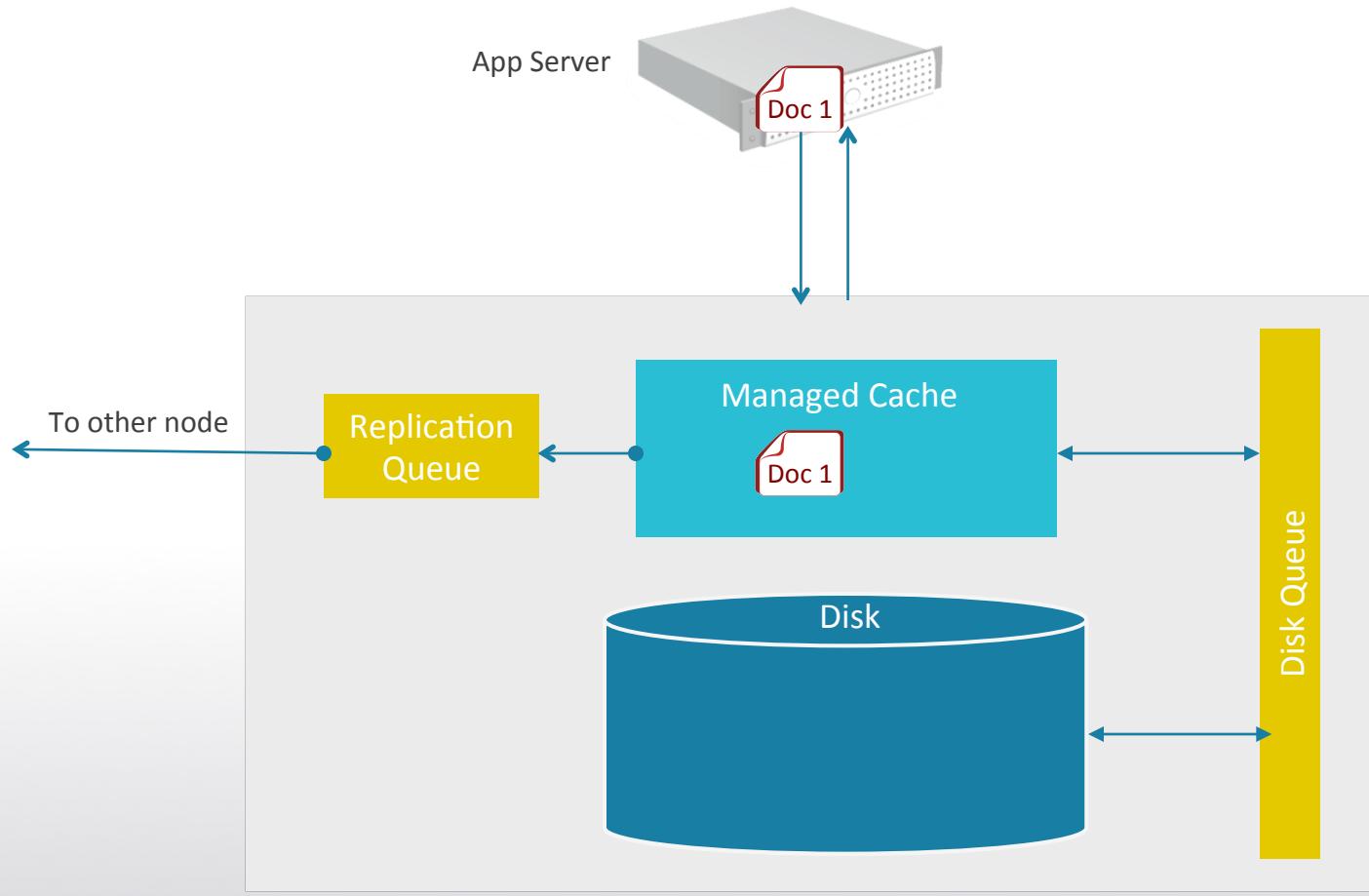
- **Company**
 - Global Leader in Online Payments
 - 132m Active Accounts, 193 Markets, 25 Currencies
- **Scalability and Performance Requirements**
 - 300m to 1bn documents with 3 Tb to 10TB
 - Billions of requests and sub 200ms response times access to JSON documents
- Read/write mix 50/50 with 5ms latency
- **Existing Database Infrastructure**
 - Multiple Tiers – Separate caching and durable store
 - MySQL, Oracle, Terracotta, Coherence
- **Pain**
 - Real-Time Access to Identity Mapping – eBay ID, PayPal ID, Social ID, 3rd Party ID, Email
- Performance – Ad needs to be served in 200ms
- Cost – Multiple tiers for caching and durability
- Highly Available – Across large clusters and across data centers
- **Couchbase Benefits**
 - Performance – Reduced latency with 5ms access times
 - Cost – Consolidation of database and cache layers
 - Cross Data Center Availability



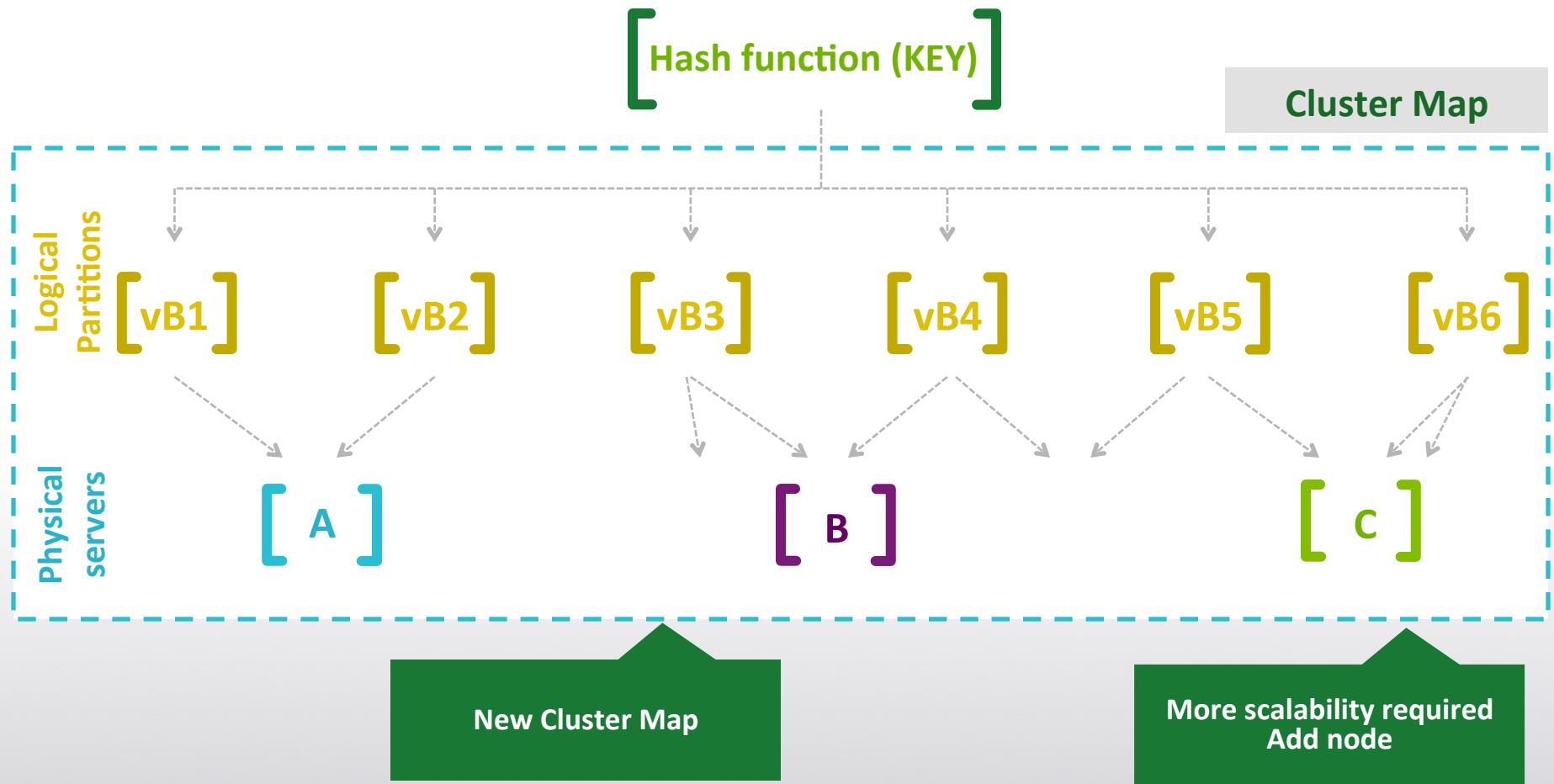
Couchbase solution

“The basics”

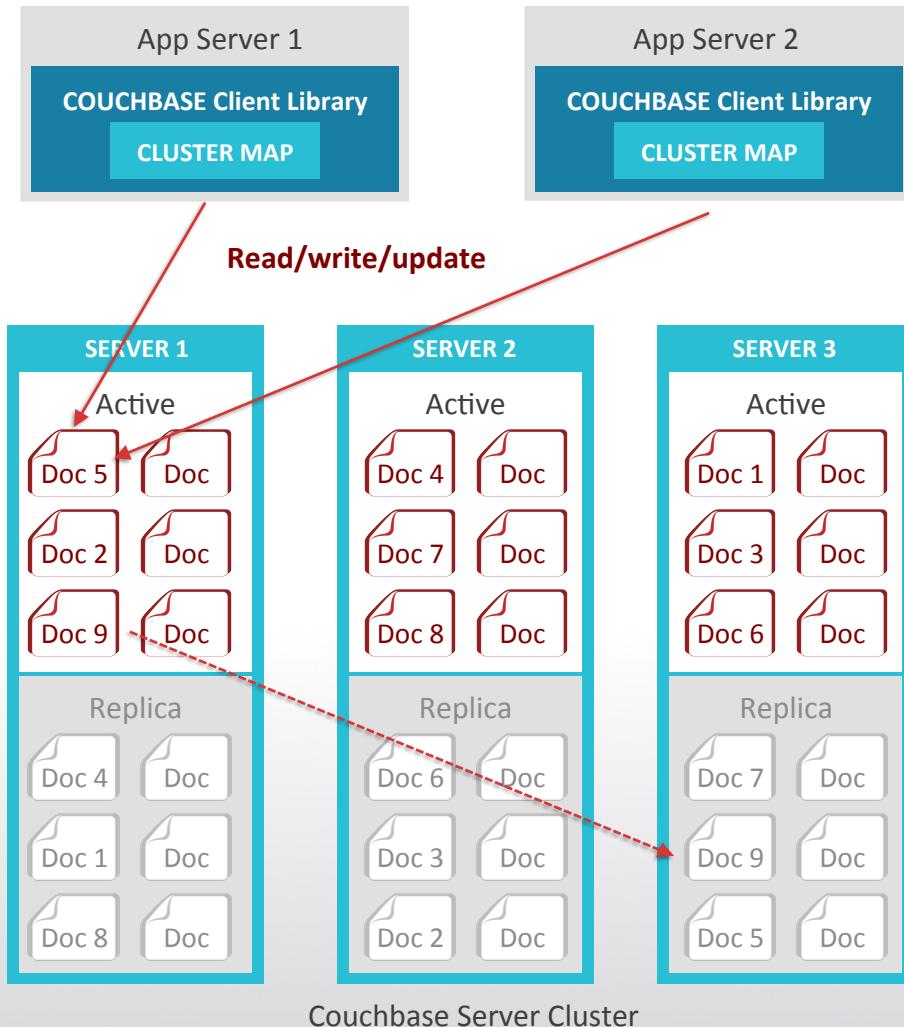
Single node – Couchbase Write Operation



Auto Sharding and Cluster Map



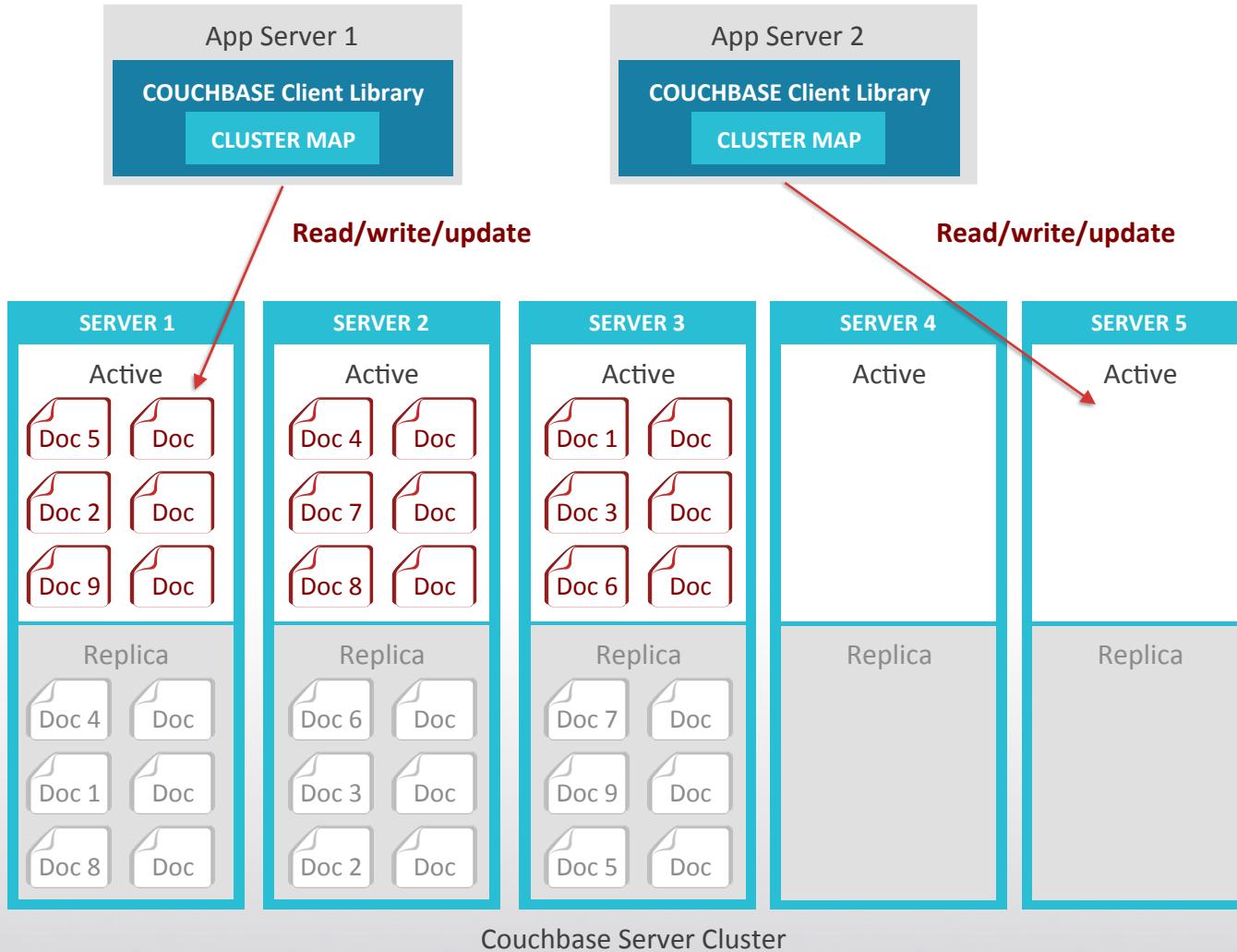
Basic Operation



- 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

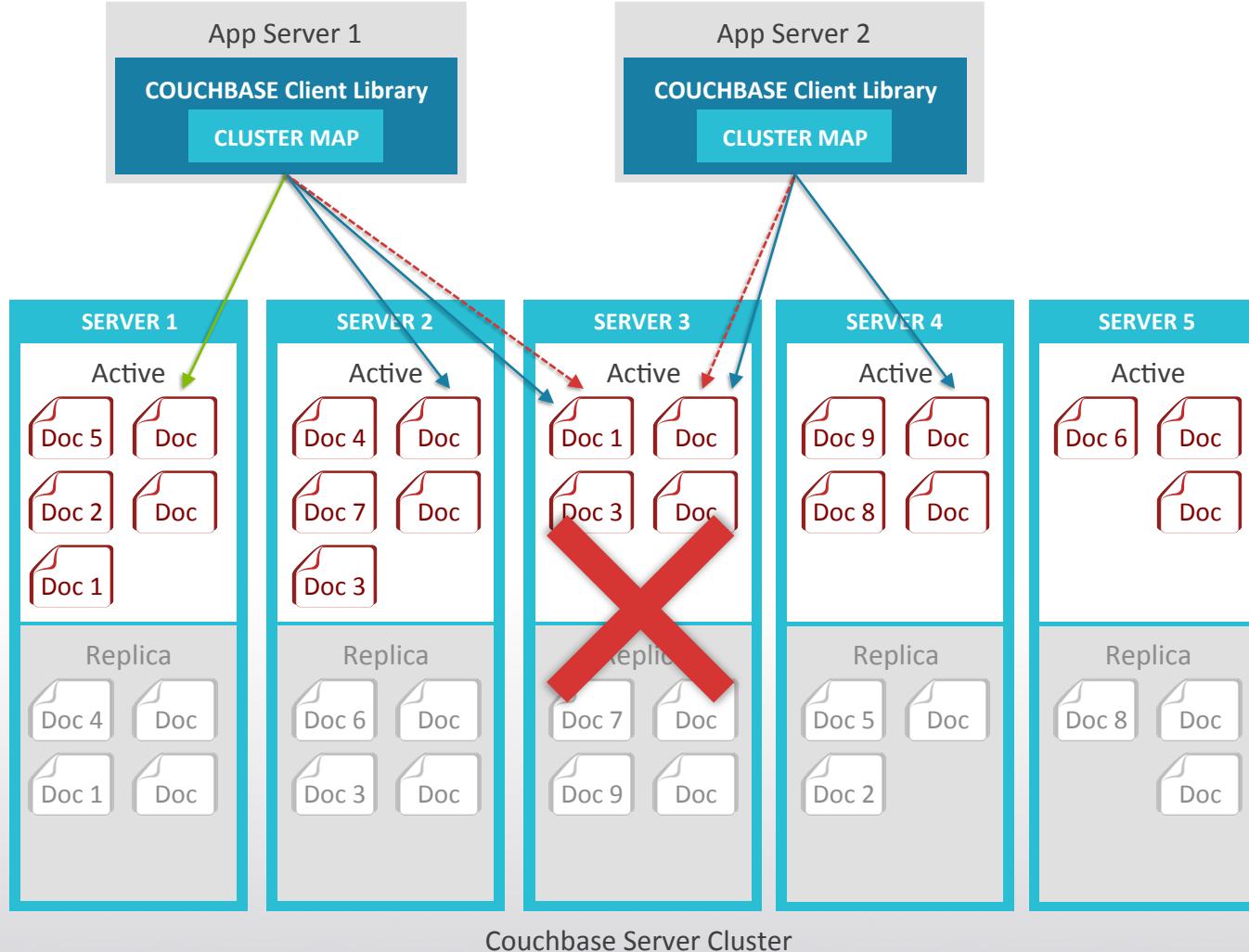


Add Nodes to Cluster



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

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



Couchbase

Query Language N1QL (Nickel)

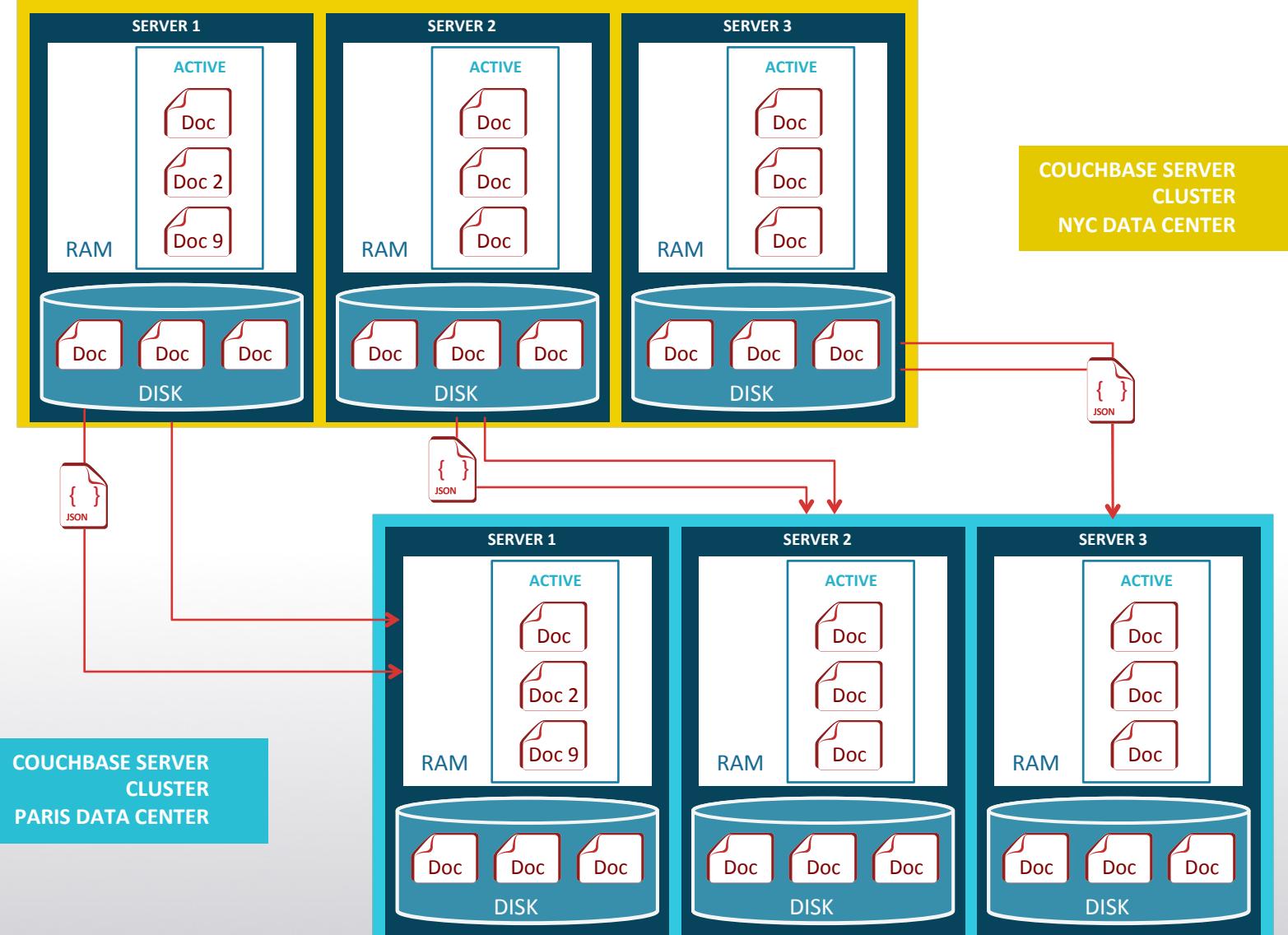
- SQL like query language
Select... from... where... group by...
having
- New Index method building:
 - Fastest index construction
 - Query performance (x 50)
- Query response formatted as JSON document based on extracted fields
- Driver ODBC /JDBC

```
SELECT product.name, round(avg(reviews.rating),1) as avg_rating
FROM reviews join product key reviews.productId
GROUP BY product
ORDER BY avg(reviews.rating) DESC LIMIT 5
```

```
{
  "resultset": [
    {
      "avg_rating": 4.5,
      "name": "Waring Pro IC70 Professional Stainless Steel Large-Capacity Ice Crusher"
    },
    {
      "avg_rating": 4.2,
      "name": "Racor PHL-1R Pro HeavyLift 4-by-4-Foot Cable-Lifted Storage Rack"
    },
    {
      "avg_rating": 4.2,
      "name": "Marineland Rite-Size Cartridge E, 4-Pack"
    }
  ]
}
```

<http://query.pub.couchbase.com/tutorial/#index>

Cross Data Center Replication (XDCR)



Enterprise vs. Community Edition

	Enterprise Edition Start Here	Community Edition
Cost	<ul style="list-style-type: none">• Free for pre-production	<ul style="list-style-type: none">• Free
Production Readiness	<p>Production Ready</p> <ul style="list-style-type: none">• Take open source snapshots• Put through rigorous QA process, fix bugs	<p>Unknown Production Readiness</p> <ul style="list-style-type: none">• Built with latest open source• Undefined quality
Features Differences	<p>Enterprise Focused Proprietary Features</p> <ul style="list-style-type: none">• Rack/Zone Awareness• Encryption support for XDCR	<ul style="list-style-type: none">• None
Technical Support	<p>Professional Support</p> <ul style="list-style-type: none">• Defined SLAs from the experts• Immediate hot bug fixes	<p>Community/Forum Support</p> <ul style="list-style-type: none">• No SLA, unknown response times• No bugs fixes assured in a timely manner
Best Practice Expertise	<p>Broad Best Practice Expertise</p> <ul style="list-style-type: none">• From working with 100s of customers	<p>Expertise of community members</p> <ul style="list-style-type: none">• Based on limited individual expertise
Release Support	<p>Long-Tail Release Support</p> <ul style="list-style-type: none">• Including hot bug fixes	<p>No Support for Old Releases</p> <ul style="list-style-type: none">• Must support old releases yourself
License Type	<p>Commercial License</p> <ul style="list-style-type: none">• Std license terms, SLA, indemnity	<p>Simple “As is” license</p>
Intended Use	<p>Production Deployments</p>	<p>Non-Production Use, Simple Use Cases</p>