



Couchbase

Architecture and Administration Basics

Workshop Day 1 - Introduction



1

Introduction Interactions vs Transactions



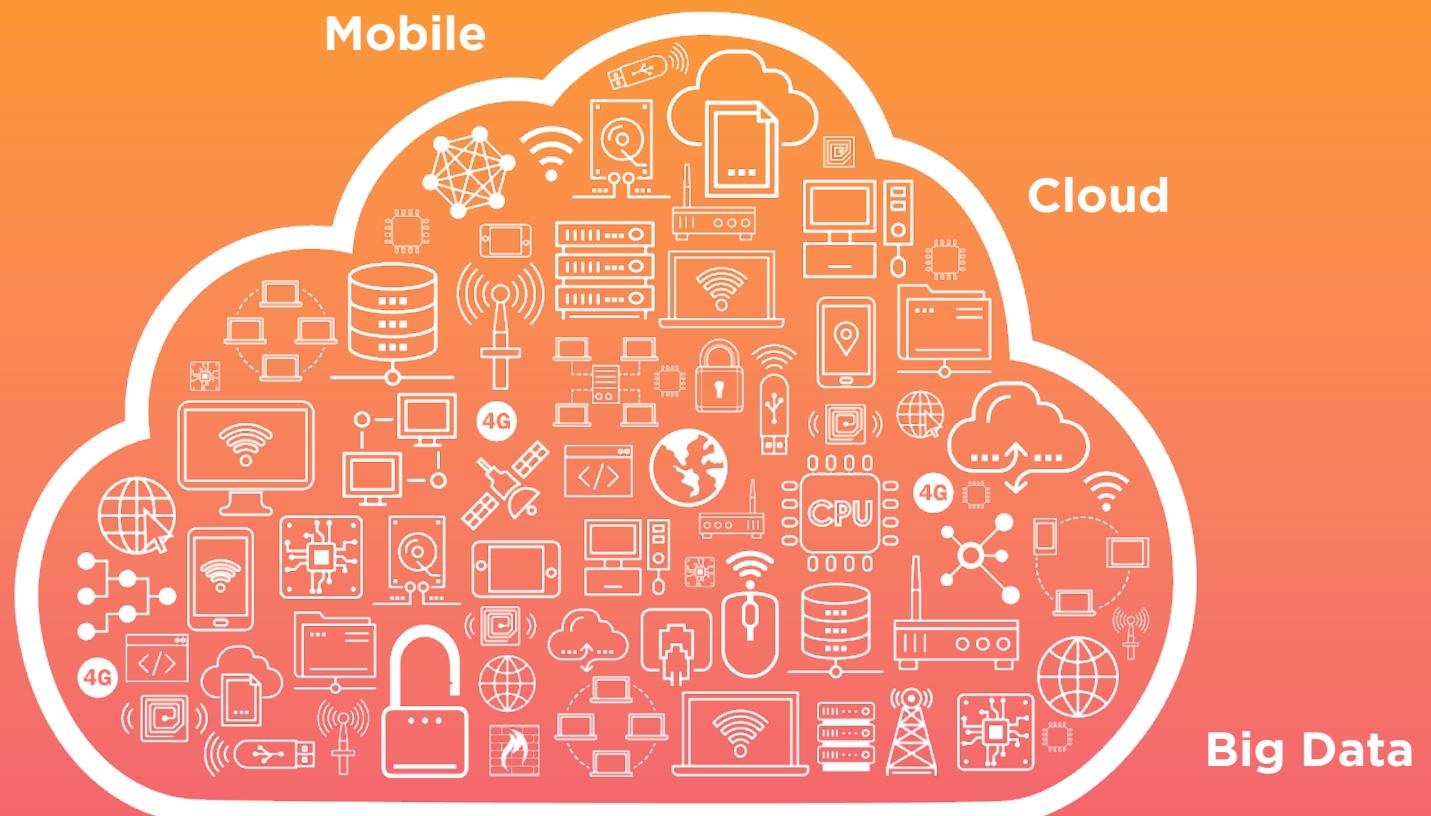
**CHANGE IS
HAPPENING:
THE WAY WE
ENGAGE HAS
EVOLVED**





Technology is Further Driving Digital Innovation

Internet of Things



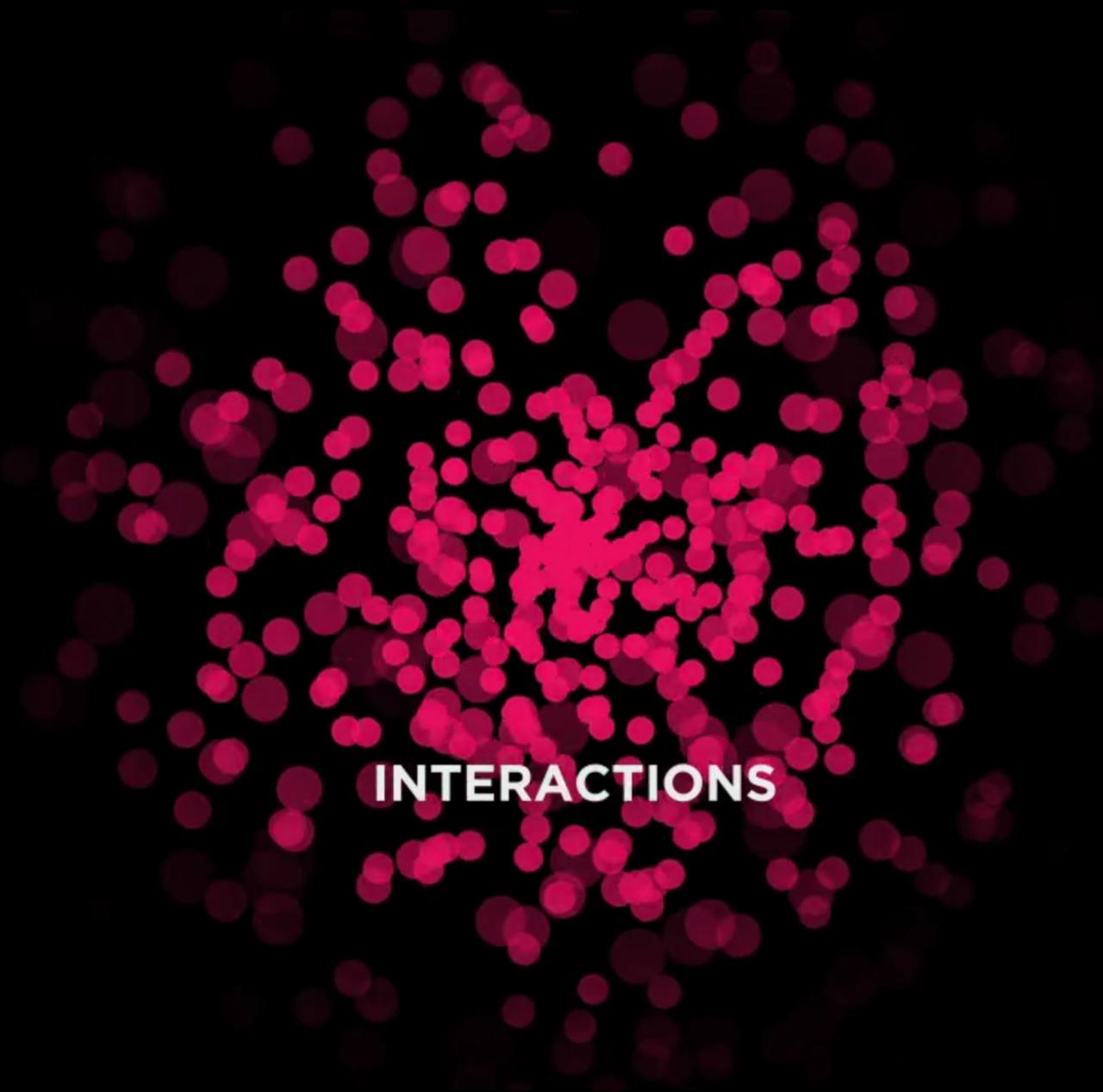


Customers now expect exceptional digital experiences

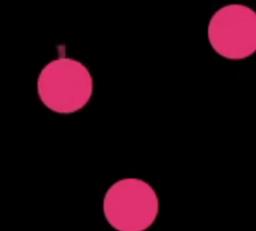




Customers spend
more time
interacting than
transacting...

A large, dense cluster of numerous small, semi-transparent red and pink circular dots, creating a glowing, organic shape that resembles a brain or a network. This cluster is positioned on the left side of the frame.

INTERACTIONS

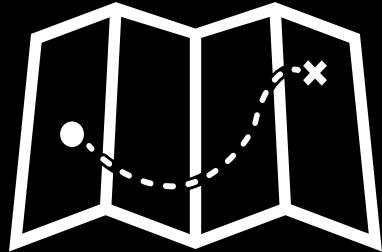
A smaller, more organized group of four red and pink circular dots arranged in a loose cluster, positioned on the right side of the frame.

TRANSACTIONS



INTERACTIONS

TRANSACTIONS



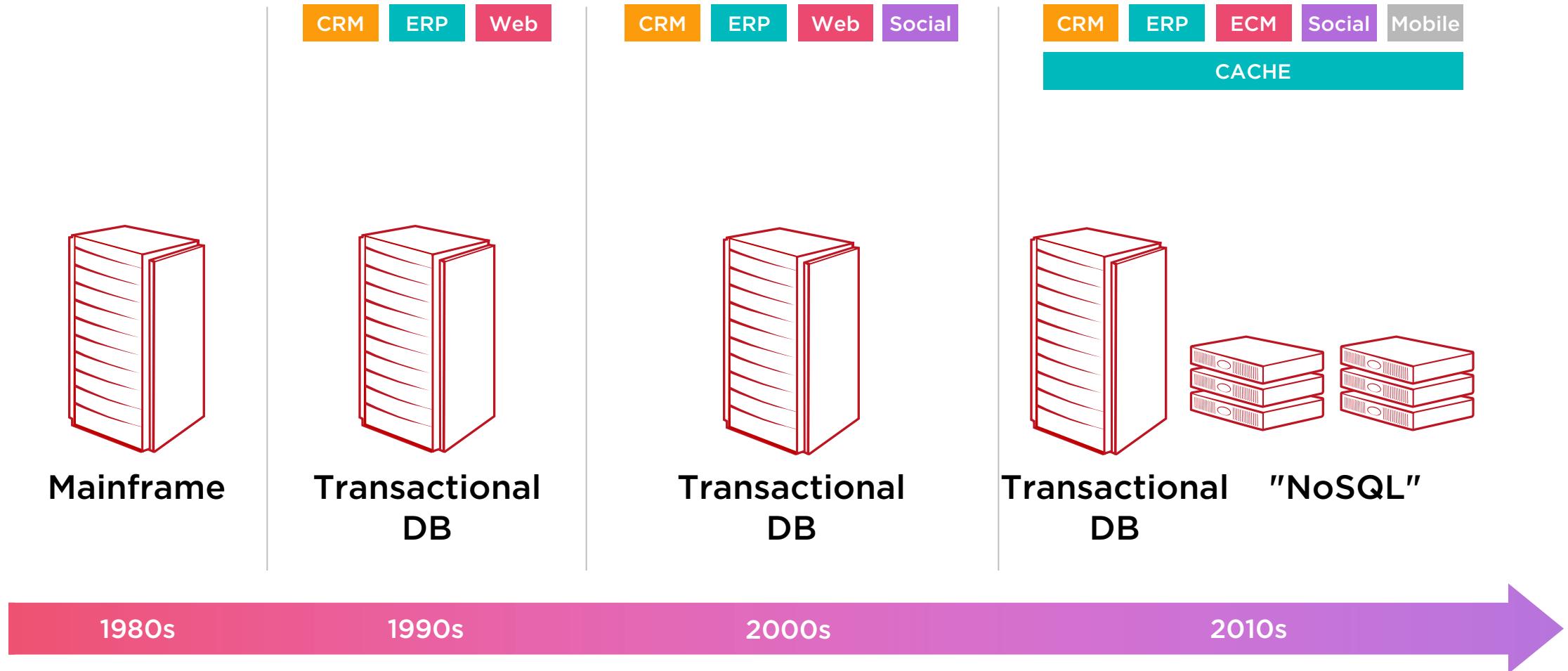
Look

Searching for a flight or hotel room
Comparing deals
Personalizing the experience



Book

The Beginnings of Digital Engagement



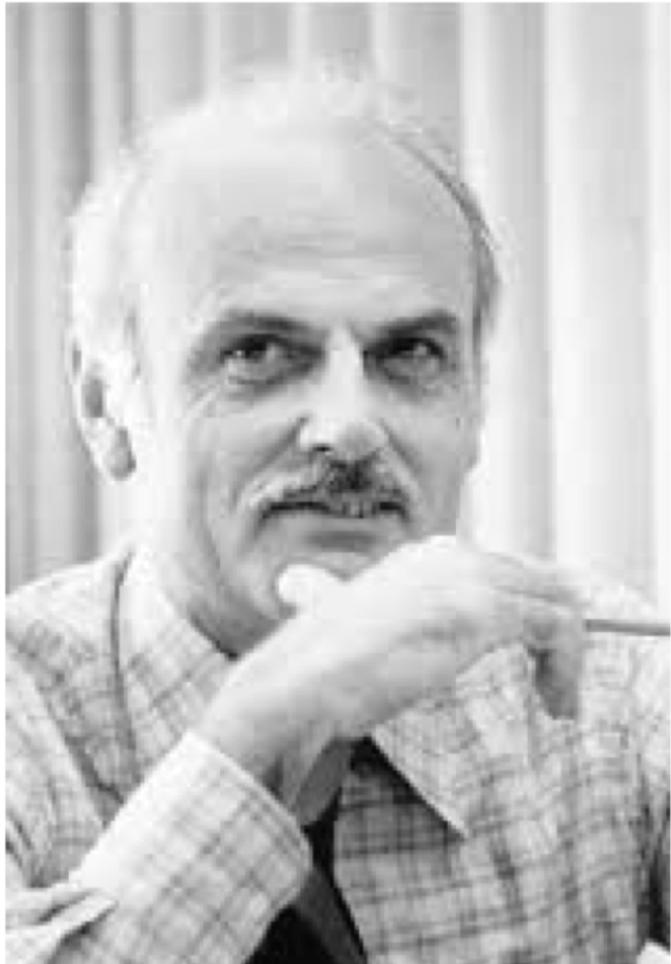


2

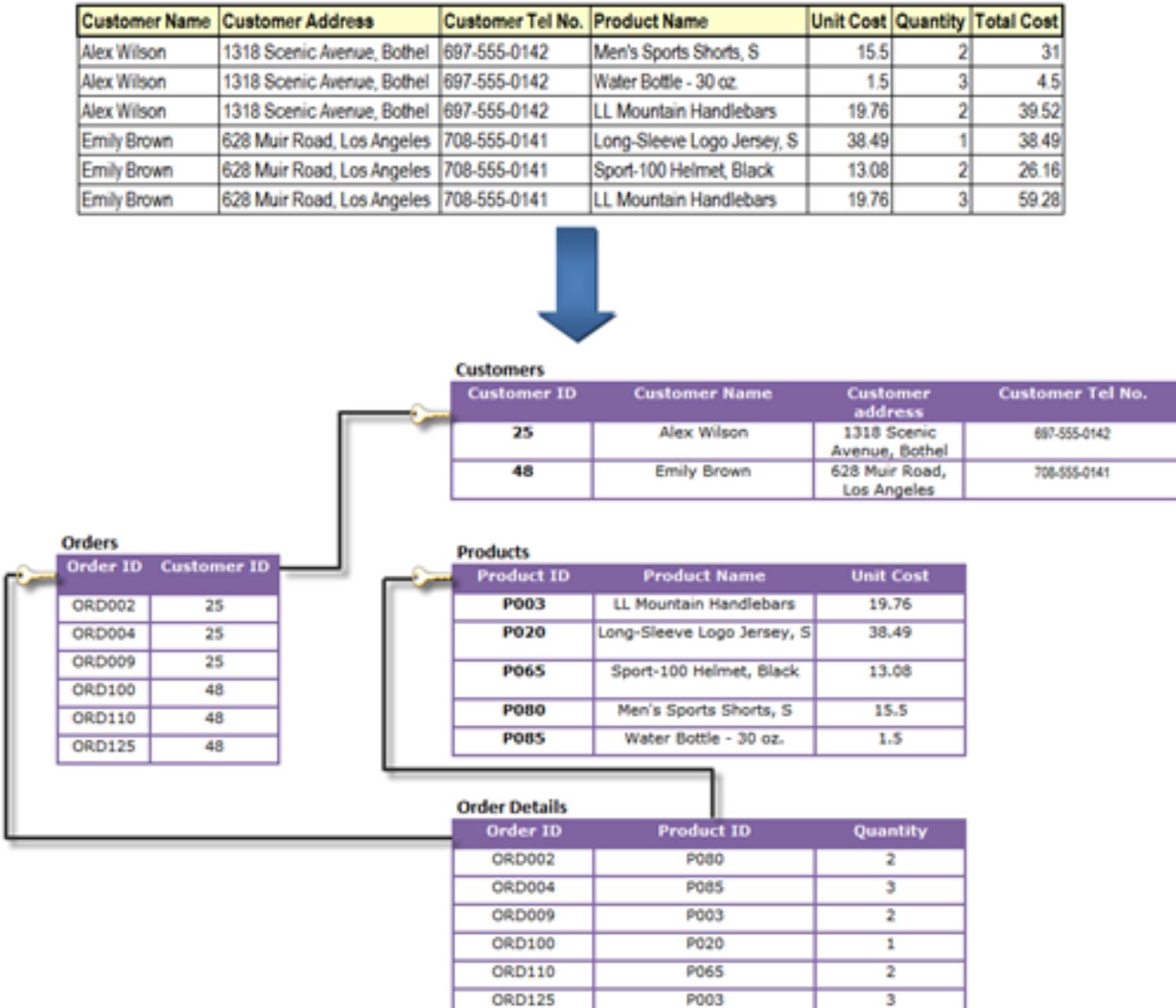
Introduction

From RDBMS to NoSQL

Who invented the RDBMS?



RDBMSs "normalize" the data



What was the key driver behind the RDBMS?



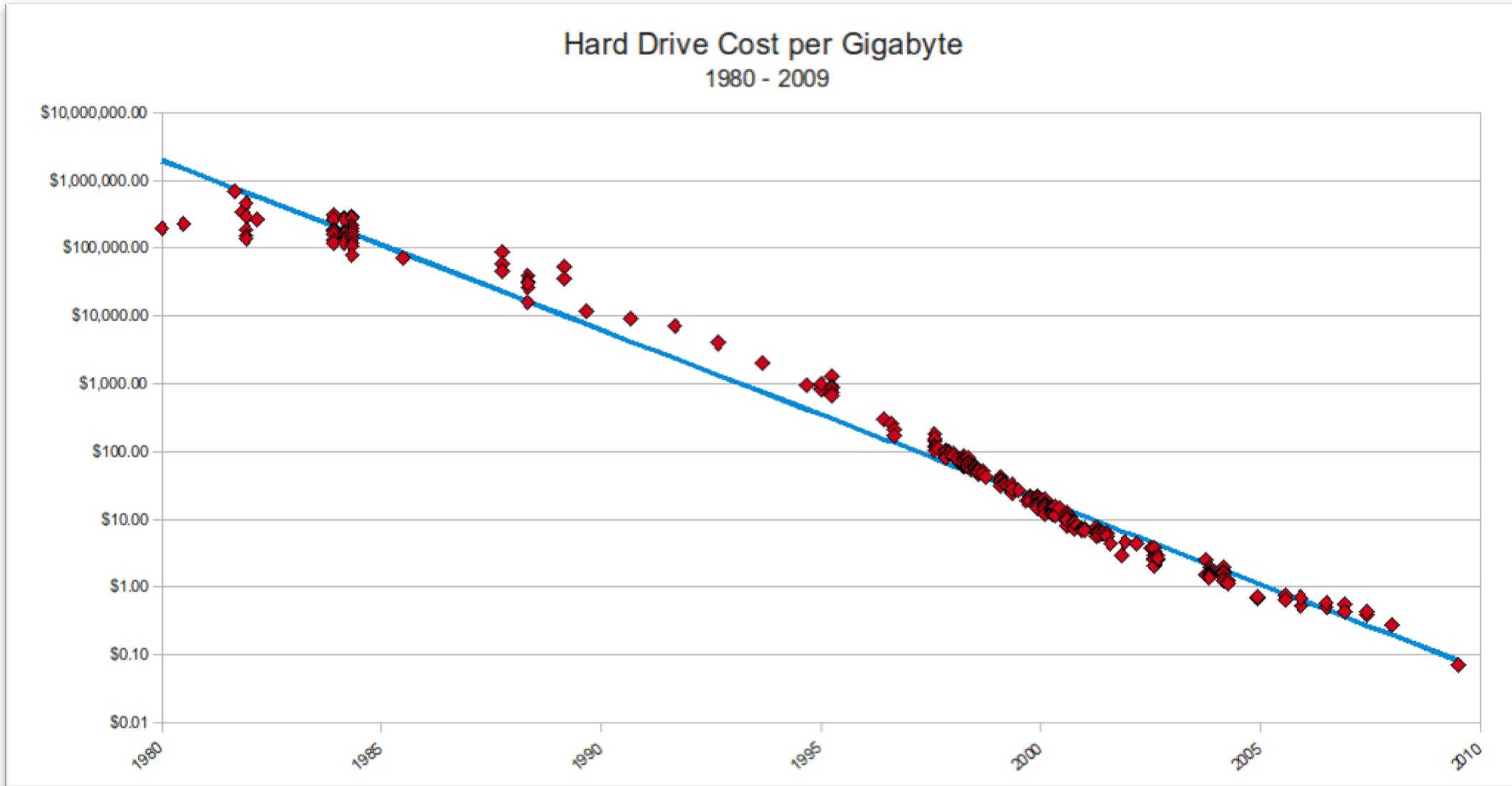
- The number “967” takes 10 bits to store
- The word “computer” takes 64 bits to store

(01100011 01101111 01101101 01110000
01110101 01110100 01100101 01110010)

How much was 1 Gig of Storage in 1980?



The Cost of Disk



IBM 3330 Direct Access Storage Facility



- 1.2 Gigs
- \$97,000 – \$142,200
- \$500 per mb in 2016

What does 1.2 Gigs cost today?



** Image may not exactly match product **



SEAGATE - BARRACUDA 1TB 7200RPM SERIAL ATA-300 (SATA-II) 32MB BUFFER 3.5 INCH HOT SWAPPABLE HARD DISK DRIVE (ST31000340NS). SYSTEM PULL. IN STOCK.

Part No. ST31000340NS

\$60.00

Quantity:

1

Add to cart

Add 1 Yr. Warranty \$15.00

Brand: SEAGATE

Request a Quote



SHOPPING
GUARANTEE

FREE

ID Protection Purchase Lowest Price

The Cost of 1 Gigabyte of Disk – Yesterday vs. Today



1980	2016
\$500,000	\$.06

Relational vs. NoSQL – The 6 Reasons



- **Joins are EXPENSIVE and don't scale well**
- RDBMSs are *complex* to scale and operate
- RDBMSs are *brittle*
- RDBMSs are not reliable enough
- Developing against an RDBMS is NOT agile
- RDBMSs are expensive to license



Quora

In summary, joins are "bad" because (1) they are inherently computationally expensive, (2) the unsophisticated implementations in typical big data environments make them even *more* expensive, and (3) the lack of effective parallelization for those implementations means that you can't increase throughput simply by adding more hardware.

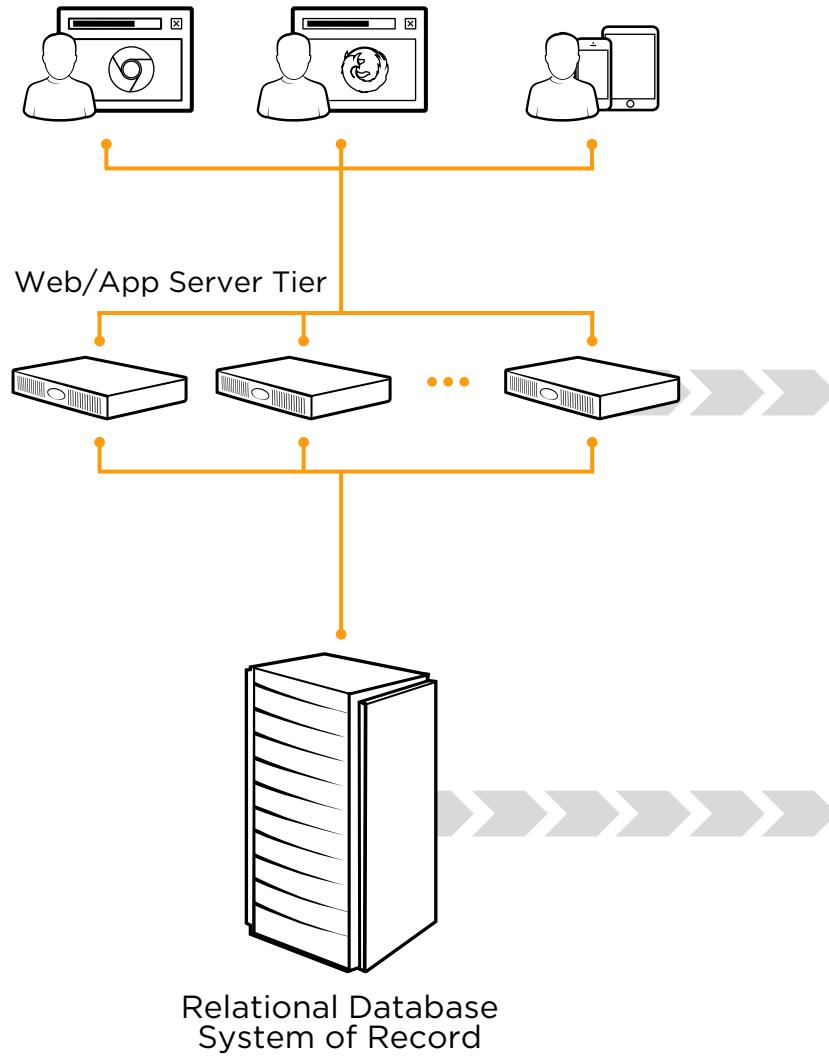
Relational vs. NoSQL – The 6 Reasons



- Joins are EXPENSIVE and don't scale well
- **RDBMSs are *complex* to scale and operate**
- RDBMSs are *brittle*
- RDBMSs are not reliable enough
- Developing against an RDBMS is NOT agile
- RDBMSs are expensive to license

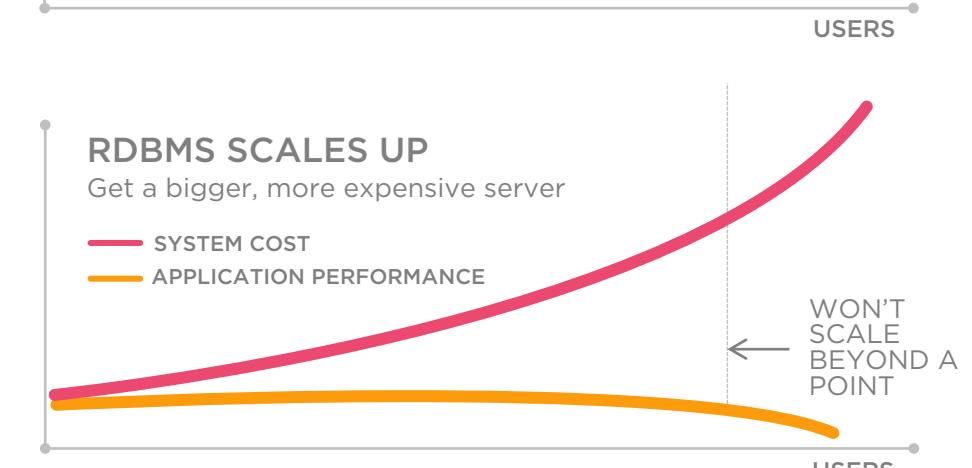


Relational Technology Scales Up And has Rigid Schemas



APPLICATION SCALES OUT

— SYSTEM COST
— APPLICATION PERFORMANCE



RDBMS SCALES UP

Get a bigger, more expensive server

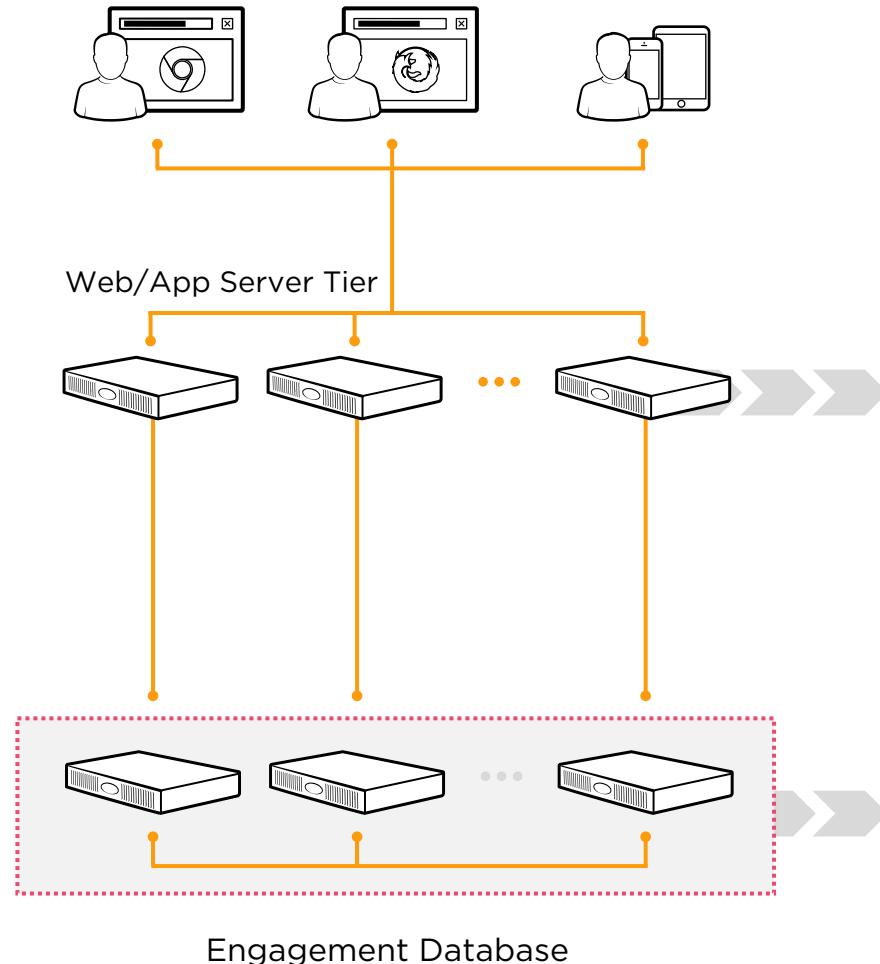
— SYSTEM COST
— APPLICATION PERFORMANCE

← WON'T SCALE BEYOND A POINT

USERS



A different approach is needed to power engaging experiences



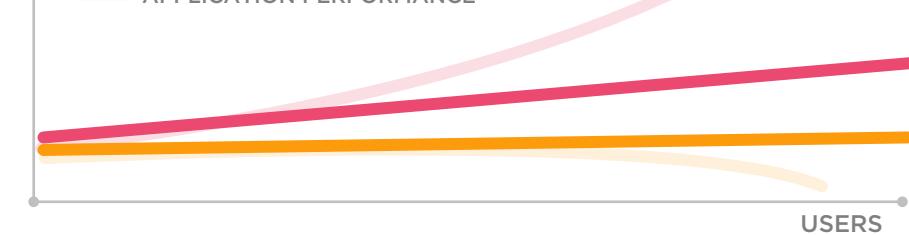
APPLICATION SCALES OUT

— SYSTEM COST
— APPLICATION PERFORMANCE

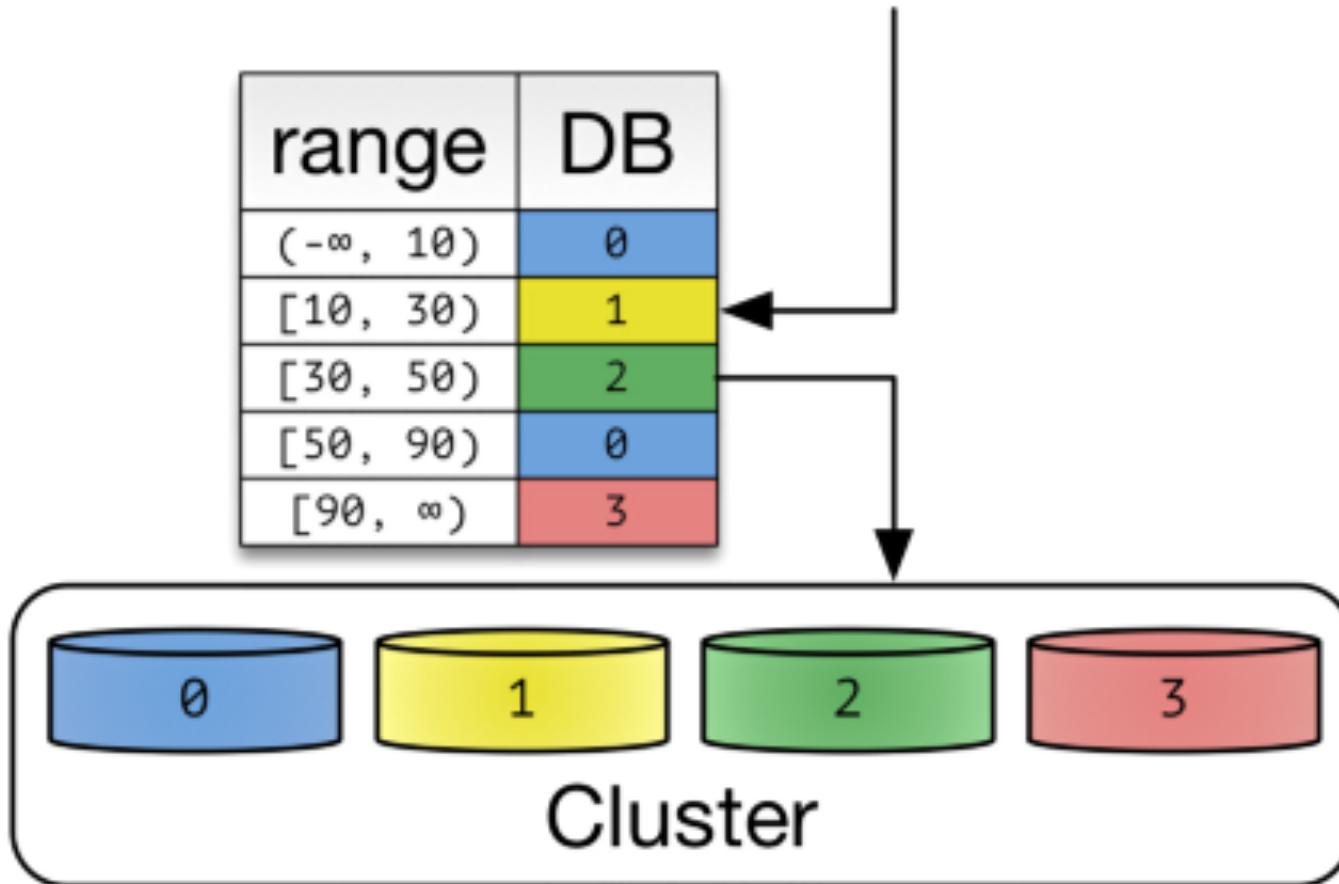


COUCHBASE SCALES OUT

— SYSTEM COST
— APPLICATION PERFORMANCE



The Problem With Sharding



- Disruptive
- Manual
- Inefficient

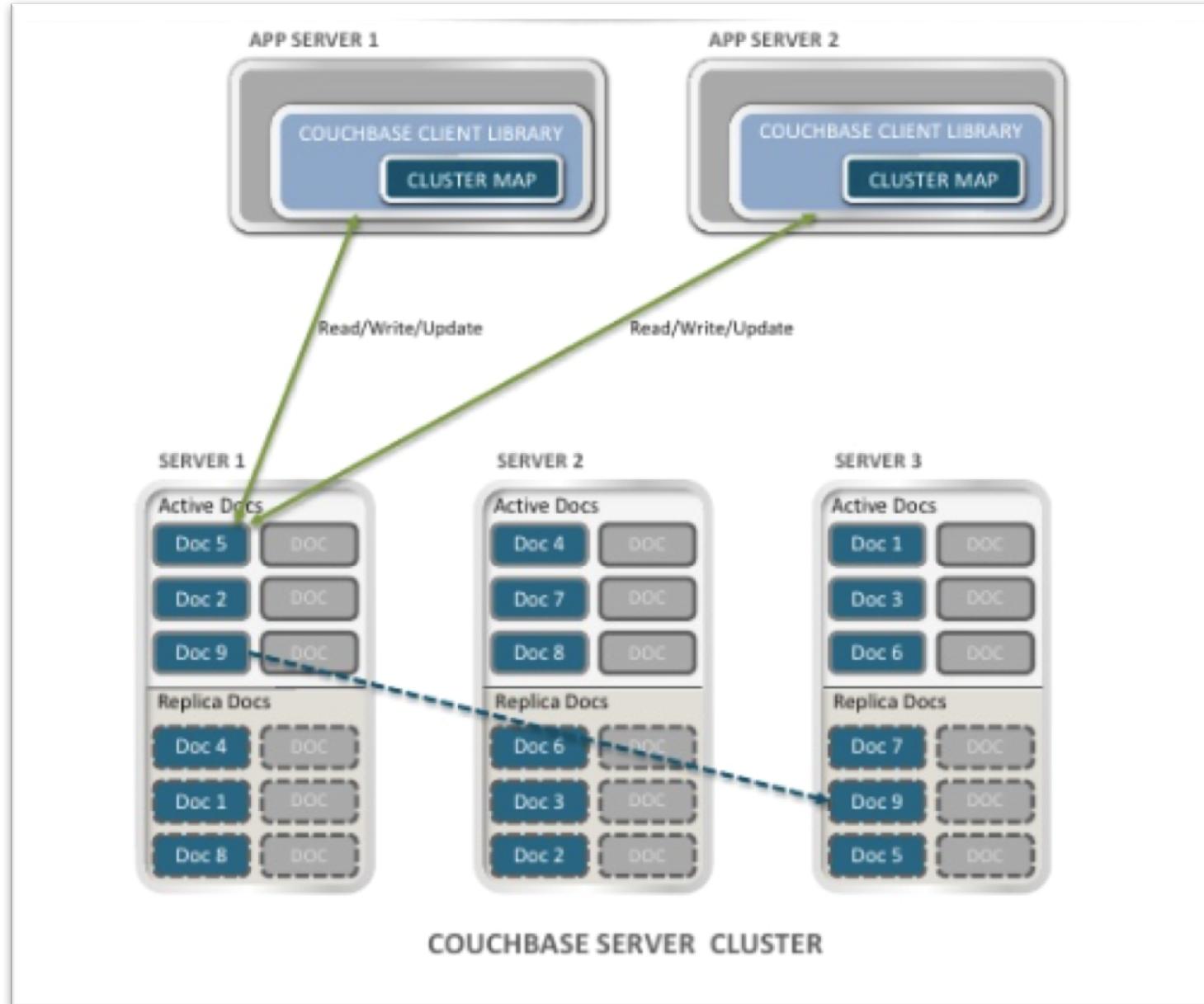
Sharding in Oracle 12c R2



“Sharding in Oracle 12c Release 2....is a feature aimed to the ‘top 5%’ of Oracle customers where price isn’t the issue but they want Oracle to scale to the size of clusters supported by Hadoop and NoSQL. Time will tell how well it’ll work and what it’ll cost.”



In Couchbase data is “sharded” by design



Relational vs. NoSQL – The 6 Reasons



- Joins are EXPENSIVE and don't scale well
- RDBMSs are *complex* to scale and operate
- **RDBMSs are strict with schema.**
- RDBMSs are not reliable enough
- Developing against an RDBMS is NOT agile
- RDBMSs are expensive to license

RDBMSs are strict with Schema

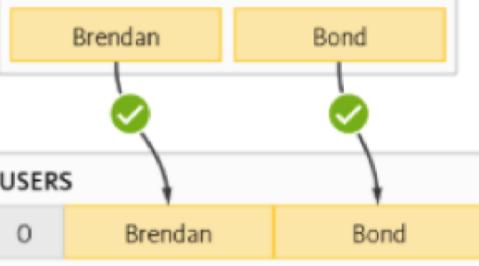


Iteration 1 — First, Last

Schema Utilized

USERS

ID	First	Last
----	-------	------

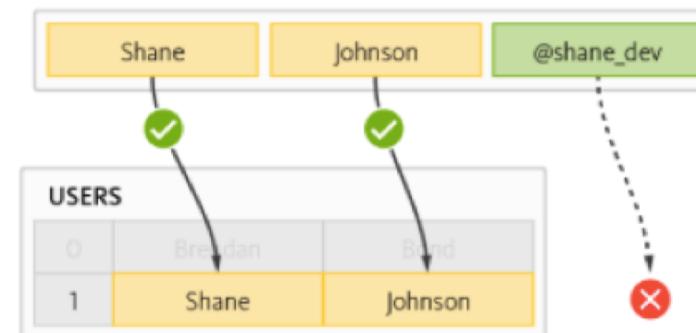


Iteration 2 — First, Last, Twitter

Schema Utilized

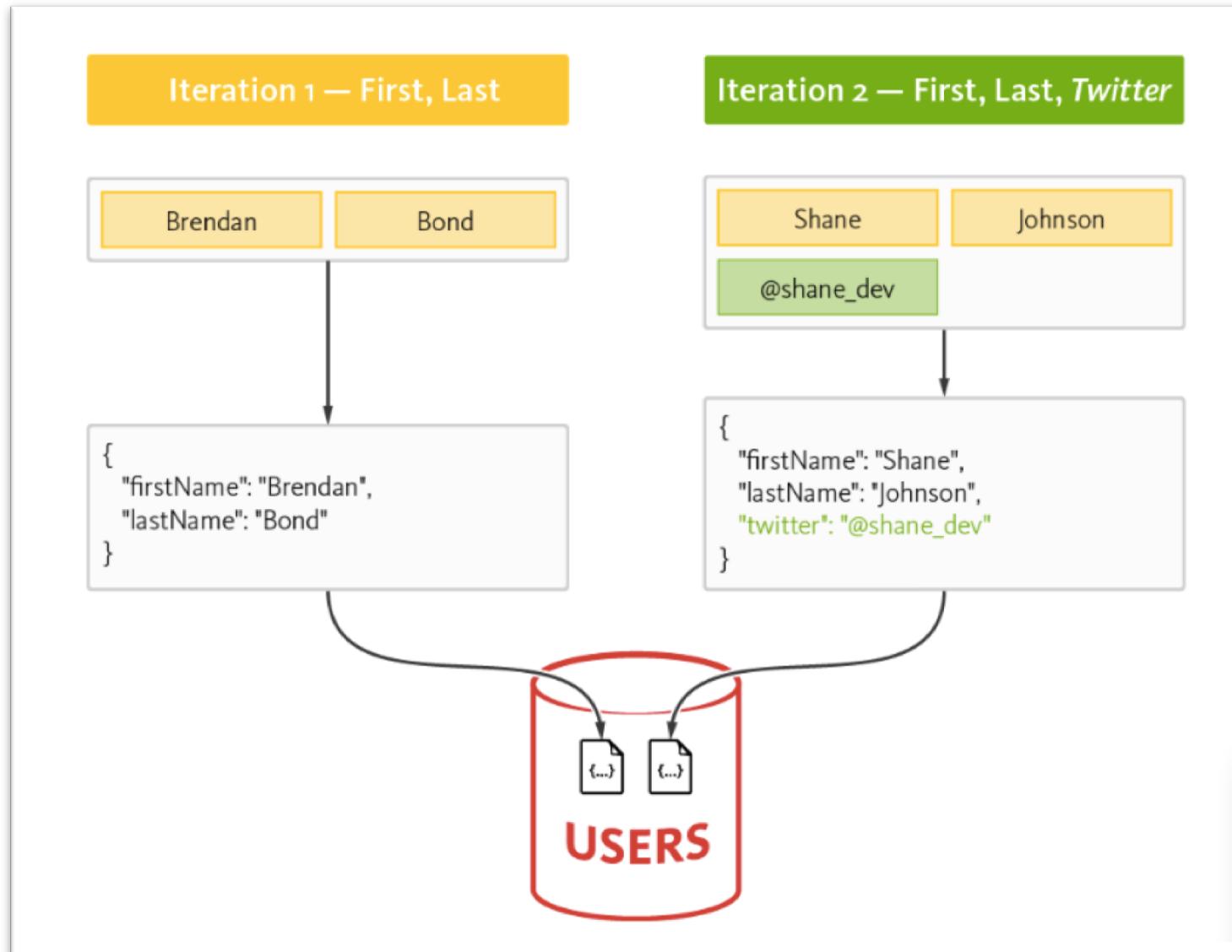
USERS

ID	First	Last
----	-------	------



- Disruptive
- Time-consuming
- Change impact on apps?

NoSQL offers Schema Flexibility

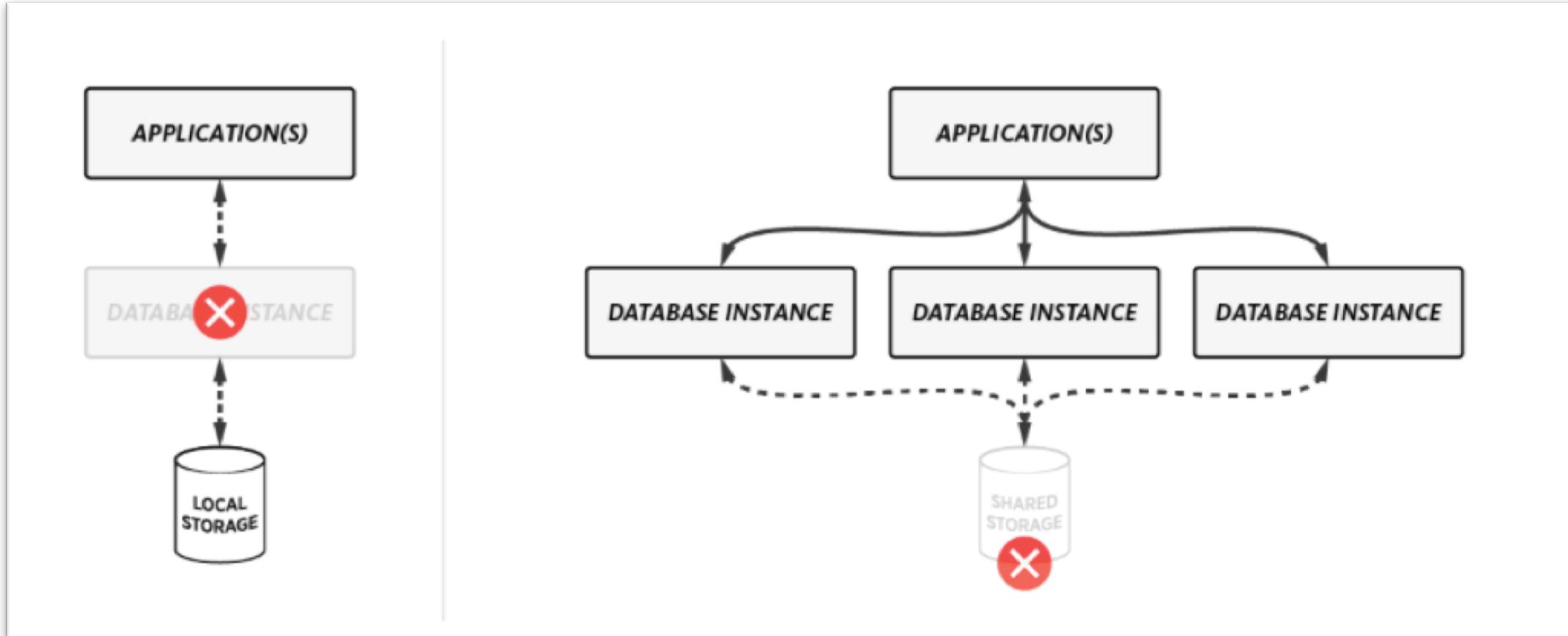


Relational vs. NoSQL – The 6 Reasons

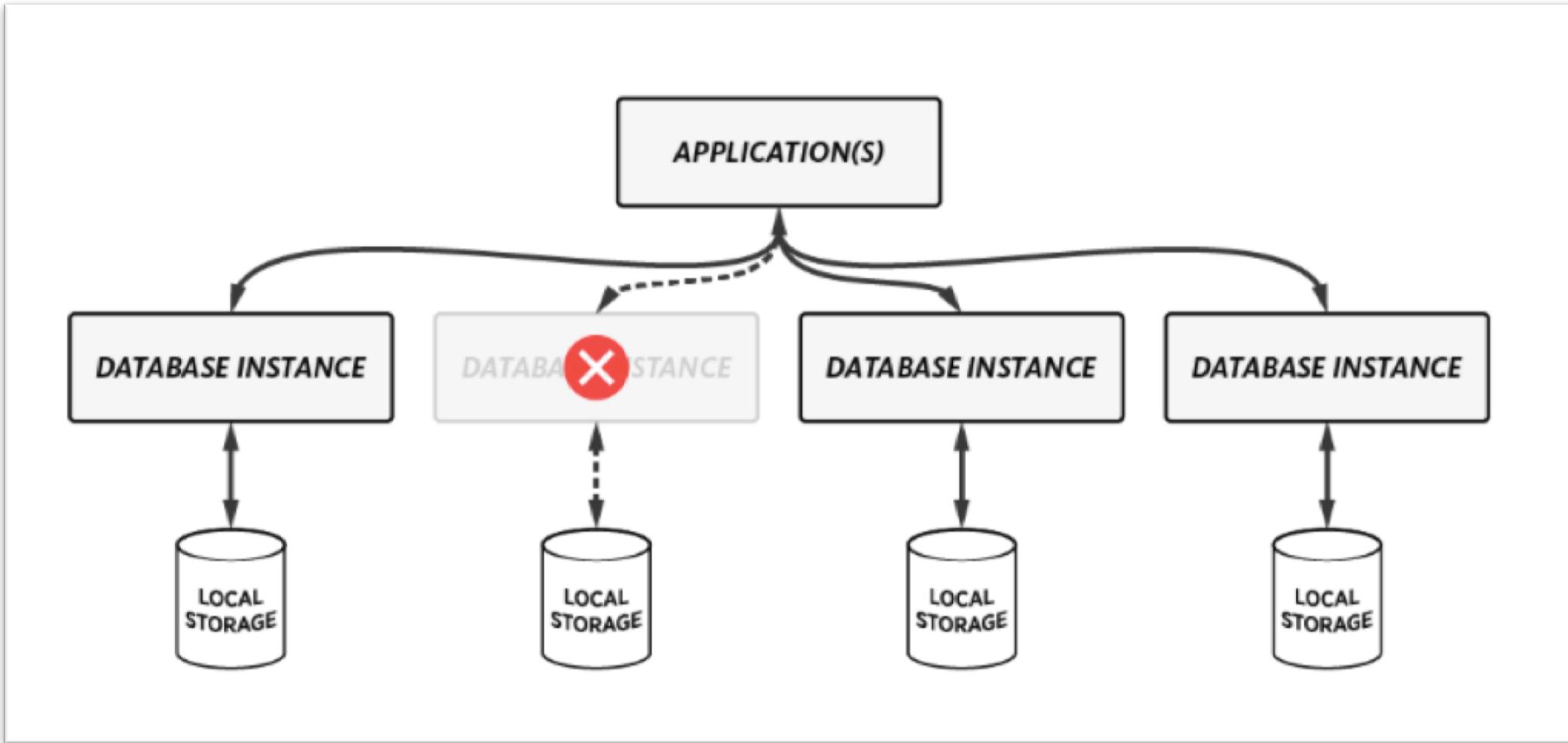


- Joins are EXPENSIVE and don't scale well
- RDBMSs are *complex* to scale and operate
- RDBMSs are *brittle*
- **RDBMSs are not reliable enough**
- Developing against an RDBMS is NOT agile
- RDBMSs are expensive to license

RDMBSs introduce single points of failure



NoSQL = RELIABLE

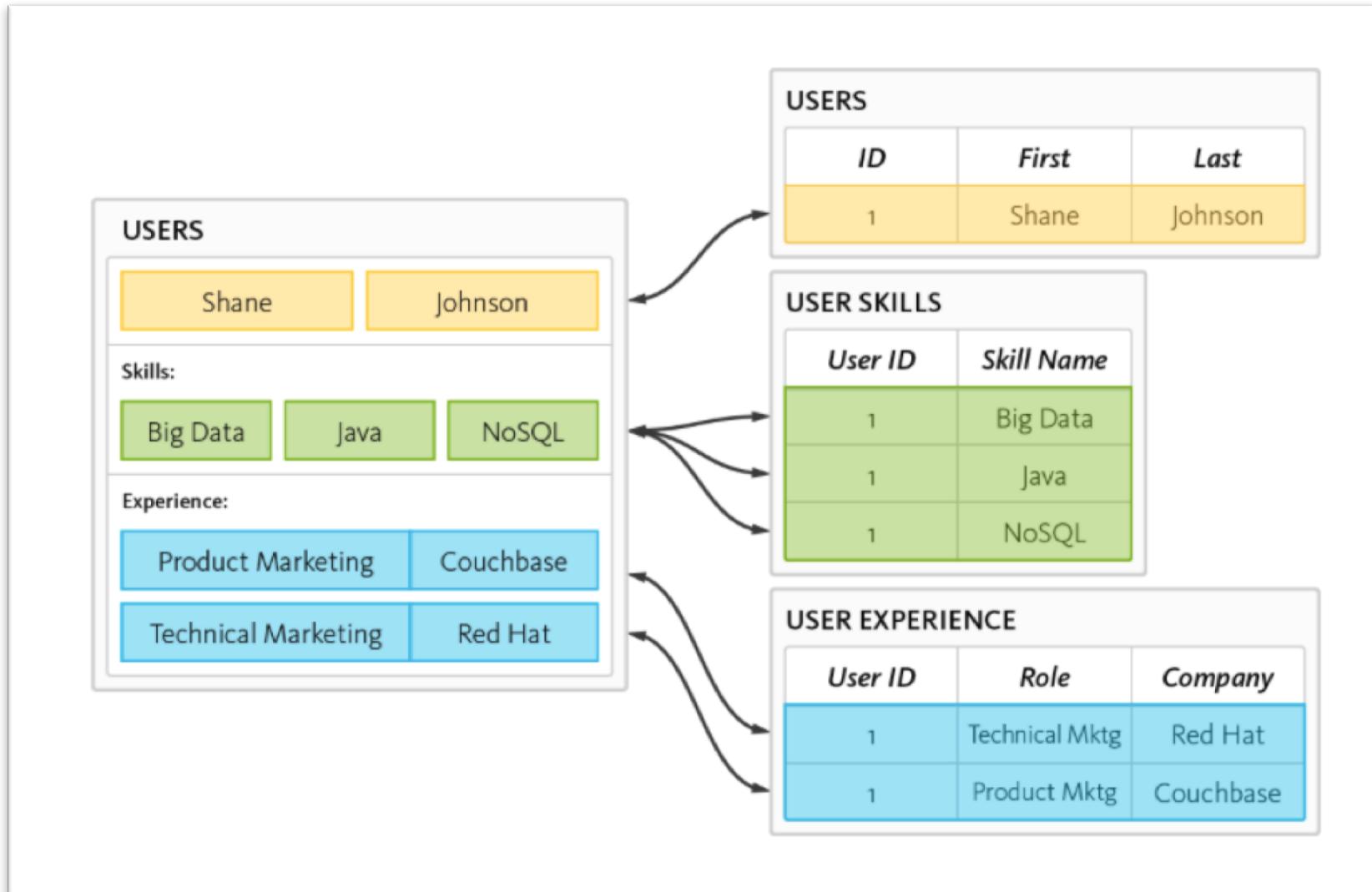


Relational vs. NoSQL – The 6 Reasons

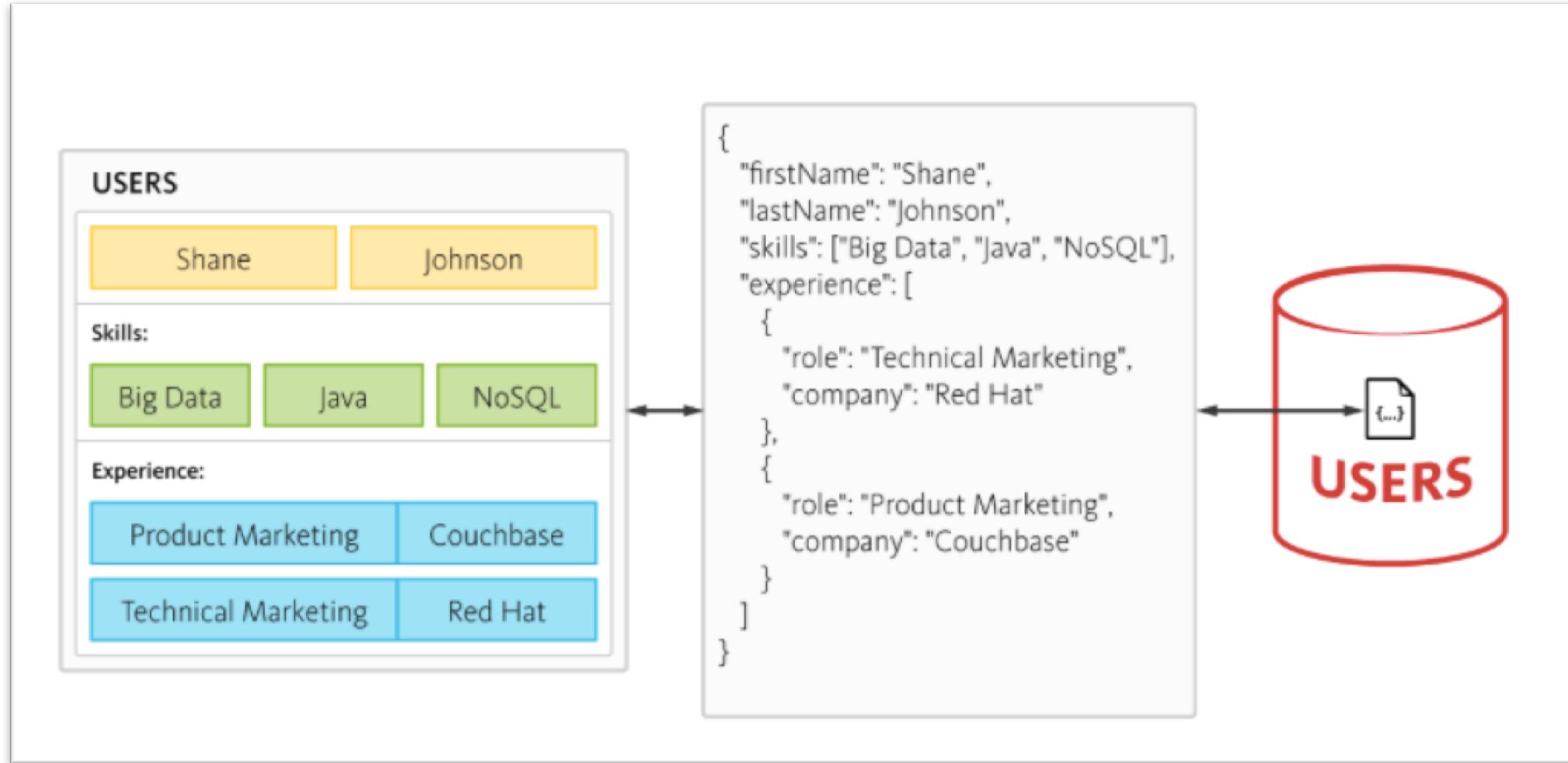


- Joins are **EXPENSIVE** and don't scale well
- RDBMSs are *complex* to scale and operate
- RDBMSs are *brittle*
- RDBMSs are not reliable enough
- **Developing against an RDBMS is NOT agile**
- RDBMSs are expensive to license

RDBMSs = Impedance Mismatch



NoSQL matches application with data



Relational vs. NoSQL – The 6 Reasons



- Joins are **EXPENSIVE** and don't scale well
- RDBMSs are *complex* to scale and operate
- RDBMSs are *brittle*
- RDBMSs are not reliable enough
- Developing against an RDBMS is NOT agile
- **RDBMSs are expensive to license**

Oracle vs. Couchbase Feature Comparison



Capability Provided	Oracle Option	Couchbase
Clustered DB deployments	Real App Clusters	<i>Included</i> in CE & EE
Standby DB	Data Guard & Partitioning	XDCR <i>included</i> in CE and EE
Encryption	Advanced Security	<i>Included</i> in EE
Role Based Access Control	Database Vault	<i>Included</i> in EE
High performance	In Memory/Cache	<i>Included</i> in CE and EE

Example – Oracle List Pricing



	Oracle Database			
	Named User Plus	Software Update License & Support	Processor License	Software Update License & Support
Database Products				
Oracle Database				
Standard Edition 2	350	77.00	17,500	3,850.00
Enterprise Edition	950	209.00	47,500	10,450.00
Personal Edition	460	101.20	-	-
Mobile Server	-	-	23,000	5,060.00
NoSQL Database Enterprise Edition	200	44	10,000	2,200.00
Enterprise Edition Options:				
Multitenant	350	77.00	17,500	3,850.00
Real Application Clusters	460	101.20	23,000	5,060.00
Real Application Clusters One Node	200	44.00	10,000	2,200.00
Active Data Guard	230	50.60	11,500	2,530.00
Partitioning	230	50.60	11,500	2,530.00
Real Application Testing	230	50.60	11,500	2,530.00
Advanced Compression	230	50.60	11,500	2,530.00
Advanced Security	300	66.00	15,000	3,300.00
Label Security	230	50.60	11,500	2,530.00
Database Vault	230	50.60	11,500	2,530.00
OLAP	460	101.20	23,000	5,060.00
Advanced Analytics	460	101.20	23,000	5,060.00
Spatial and Graph	350	77.00	17,500	3,850.00
TimesTen Application-Tier Database Cache	460	101.20	23,000	5,060.00
Database In-Memory	460	101.20	23,000	5,060.00
Retail Data Model	800	176.00	40,000	8,800.00
Communications Data Model	1,500	330.00	50,000	11,000.00
Airlines Data Model	800	176.00	40,000	8,800.00
Utilities Data Model	800	176.00	40,000	8,800.00
Database Enterprise Management				
Diagnostics Pack	150	33.00	7,500	1,650.00
Tuning Pack	100	22.00	5,000	1,100.00
Database Lifecycle Management Pack	240	52.80	12,000	2,640.00
Data Masking and Subsetting Pack	230	50.60	11,500	2,530.00
Cloud Management Pack for Oracle Database	150	33.00	7,500	1,650.00
Enterprise Edition				\$47,500
Real App Clusters				\$70,500
Data Guard & Part.				\$93,500
Advanced Security				\$108,500
Database Vault				\$120,000
In-memory/Cache				\$166,000

Example – Oracle List Pricing cont'd



	Oracle Database			
	Named User Plus	Software Update License & Support	Processor License	Software Update License & Support
Database Products				
Oracle Database				
Standard Edition 2	350	77.00	17,500	3,850.00
Enterprise Edition	950	209.00	47,500	10,450.00
Personal Edition	460	101.20	-	-
Mobile Server	-	-	23,000	5,060.00
NoSQL Database Enterprise Edition	200	44	10,000	2,200.00
Enterprise Edition Options:				
Multitenant	350	77.00	17,500	3,850.00
Real Application Clusters	460	101.20	23,000	5,060.00
Real Application Clusters One Node	200	44.00	10,000	2,200.00
Active Data Guard	230	50.60	11,500	2,530.00
Partitioning	230	50.60	11,500	2,530.00
Real Application Testing	230	50.60	11,500	2,530.00
Advanced Compression	230	50.60	11,500	2,530.00
Advanced Security	300	66.00	15,000	3,300.00
Label Security	230	50.60	11,500	2,530.00
Database Vault	230	50.60	11,500	2,530.00
OLAP	460	101.20	23,000	5,060.00
Advanced Analytics	460	101.20	23,000	5,060.00
Spatial and Graph	350	77.00	17,500	3,850.00
TimesTen Application-Tier Database Cache	460	101.20	23,000	5,060.00
Database In-Memory	460	101.20	23,000	5,060.00
Retail Data Model	800	176.00	40,000	8,800.00
Communications Data Model	1,500	330.00	50,000	11,000.00
Airlines Data Model	800	176.00	40,000	8,800.00
Utilities Data Model	800	176.00	40,000	8,800.00
Database Enterprise Management				
Diagnostics Pack	150	33.00	7,500	1,650.00
Tuning Pack	100	22.00	5,000	1,100.00
Database Lifecycle Management Pack	240	52.80	12,000	2,640.00
Data Masking and Subsetting Pack	230	50.60	11,500	2,530.00
Cloud Management Pack for Oracle Database	150	33.00	7,500	1,650.00

\$200,000



3

Introduction Engagement Database



ATTRIBUTES OF AN ENGAGEMENT DATABASE



Hello cloud,
hello world



Seamlessly
mobile



Built-in
smarts



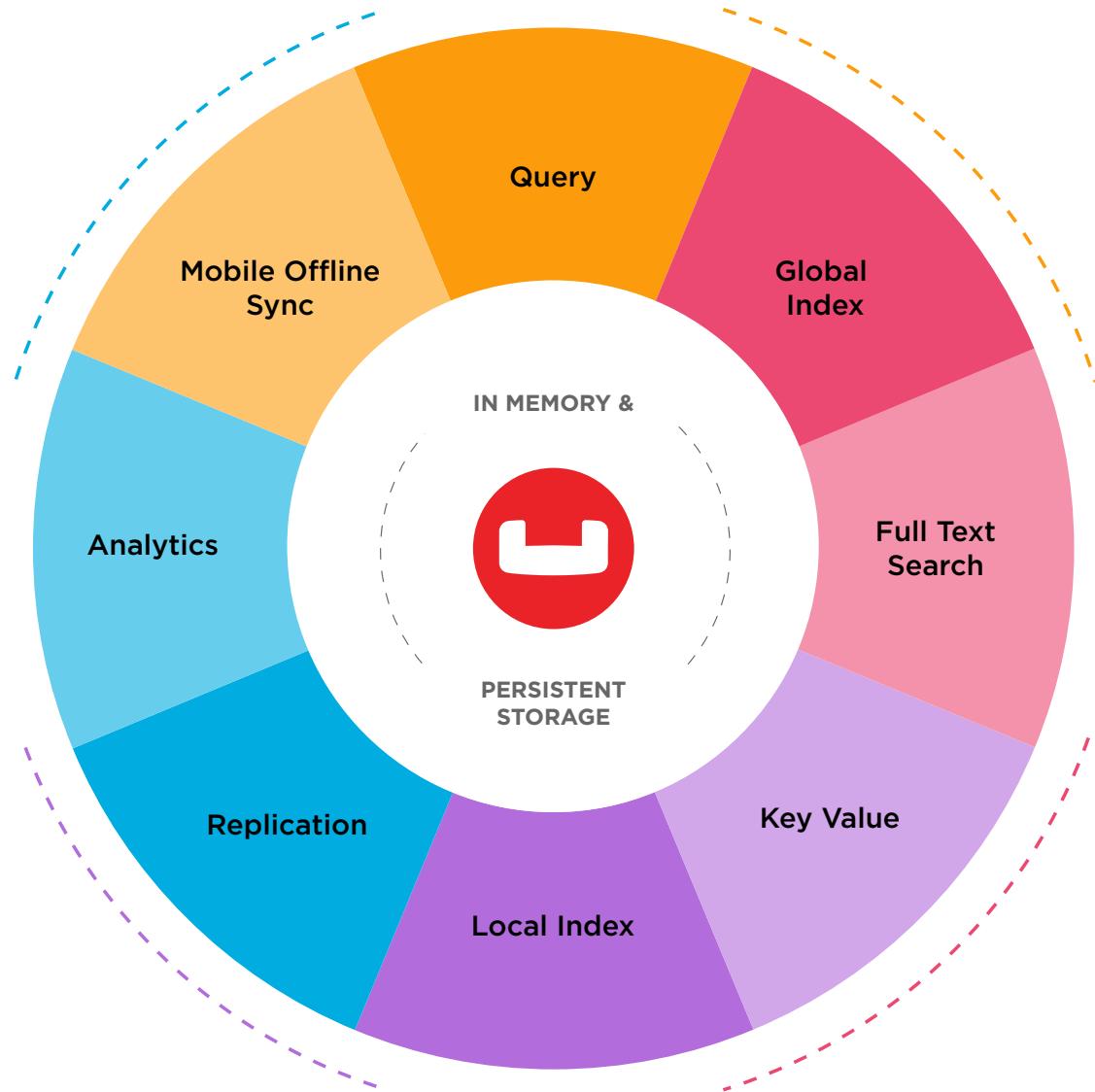
Built for change
- at scale



Always on,
always fast



Secure, secure,
secure







How does Couchbase fit with Big Data?

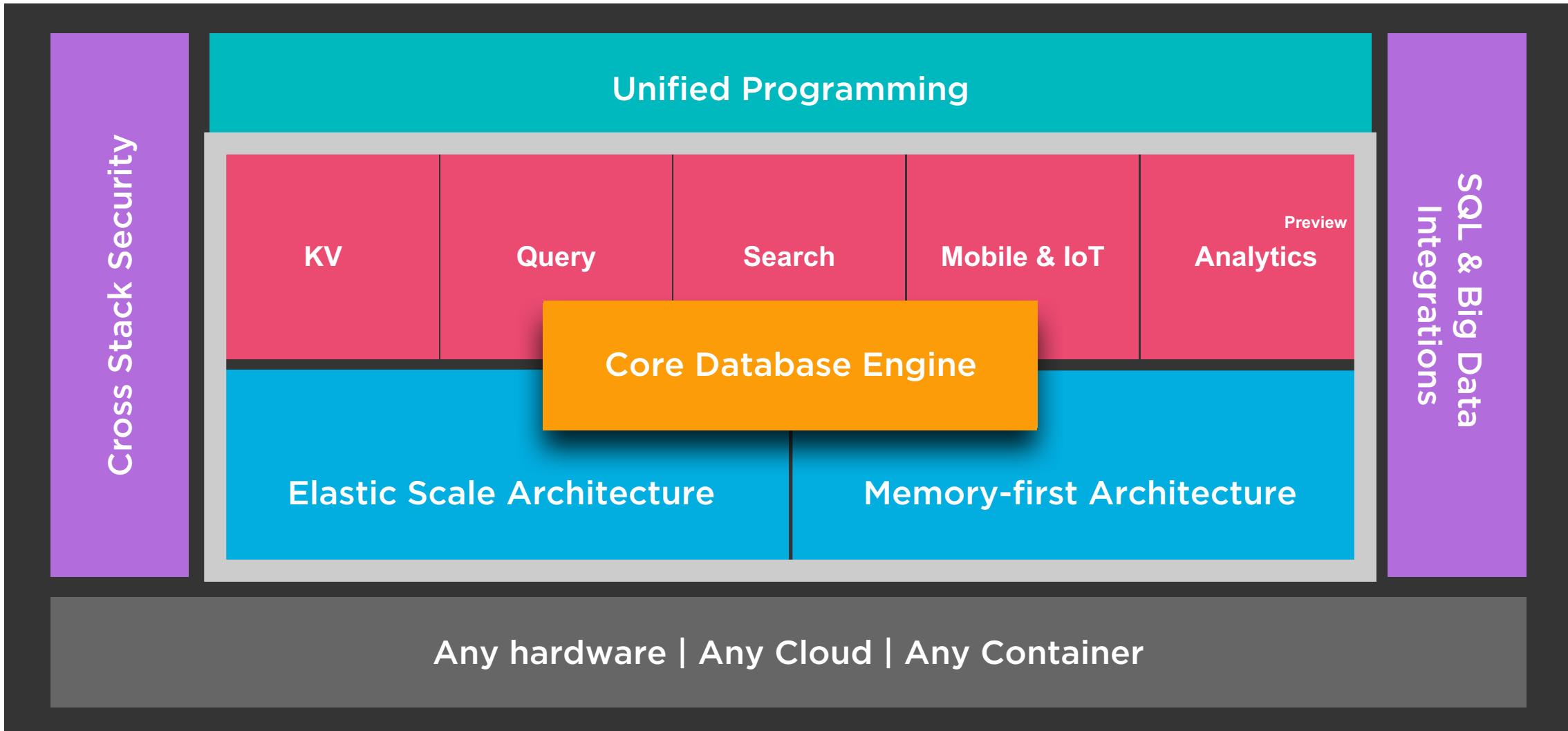
Couchbase and Hadoop are complementary.



EXAMPLES:

cloudera  Hortonworks  MAPR 

Couchbase Data Platform





Developing with Couchbase

Couchbase supports a wide range of frameworks, languages, platforms and infrastructure choices.

FRAMEWORKS

LINQ



.NET



MOBILE



LANGUAGES



php

PLATFORMS



MOBILE



INFRASTRUCTURE





4

Introduction Use Cases

Customers Across Every Industry Embracing Couchbase



Engagement Database Use cases



User Profile
Database



Session Database



Entitlement
Management



Operational
Dashboarding



Inventory &
Availability



Asset/Resource
Management



Catalog



Metadata



Customer 360



Field Service
Enablement



Device User Data
Management



Endpoint Data
Management





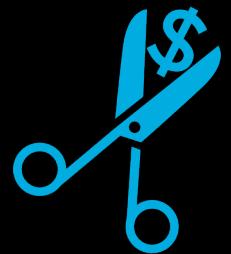
WHAT PROBLEMS ARE OUR CUSTOMERS SOLVING?



Improving customer experience & engagement



Faster innovation & time to market



Reducing infrastructure & operations costs

Improving Customer Experience

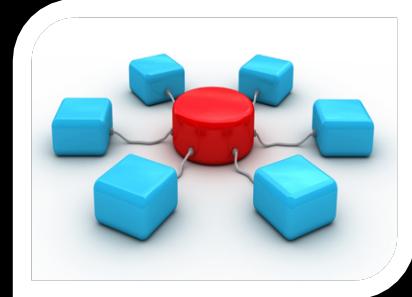


What our customers tell us about this challenge



“We are losing x-sell and up-sell opportunities because of poor user experience.”

Biz Tech



“We struggle to integrate the right data to give our customers and staff a complete view.”

IT Leaders



“Our legacy technology stack is holding us back from providing the optimal customer experience.”

Architects

Improving Customer Experience - X-Sell & Up-Sell



What our customers tell us about this challenge



“A more personalized experience comes at the cost of slower performance and increased backend complexity. **”**



“I would like my mobile app AND website to reflect a similar experience - in both performance and functionality. **”**



IMPROVING CUSTOMER
EXPERIENCE &
ENGAGEMENT



To improve user experience, eBay needed to be able to scale without expanding Oracle environment



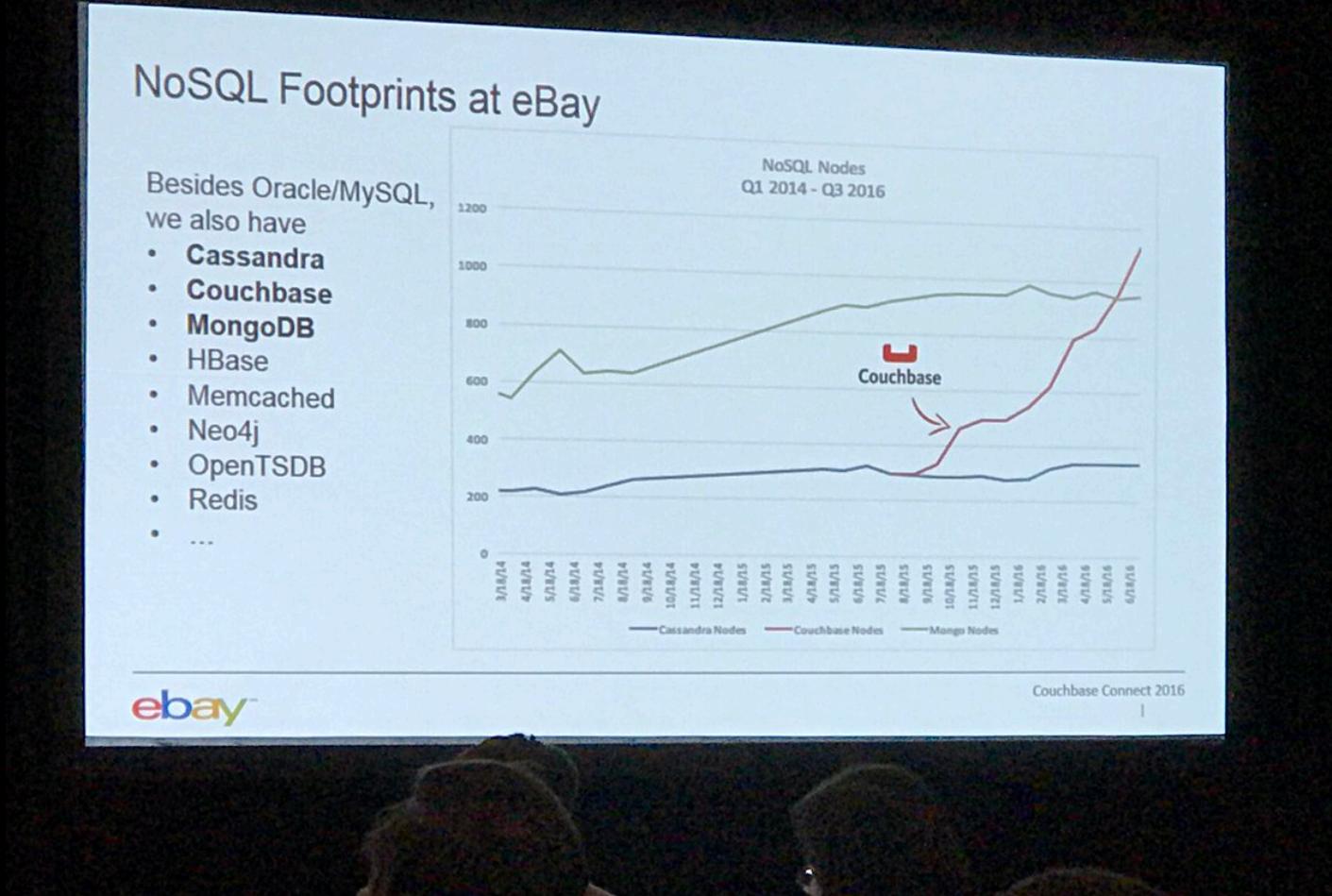
Requirements

- Oracle licensing, hardware, and support costs made scaling difficult
- ACID led to performance penalty for advanced features
- Needed to increase performance, provide high availability

Outcomes

- Linear scalability and very high throughput keep site responsive for users
- Flexible schema increases developer agility
- Active-active bi-directional **XDCR** keeps site online and available

Couchbase XDCR made the difference at eBay



- Strong Consistency – “read your own writes”
- Bidirectional replication across three data centers in the US
- Elastic scalability, add nodes to scale systems to meet SLAs
- Key feature when comparing against MongoDB and Cassandra

IMPROVING CUSTOMER
EXPERIENCE &
ENGAGEMENT

Couchbase Mobile stores semi-static data locally and syncs updates as they occur, providing a faster, better booking experience



Requirements

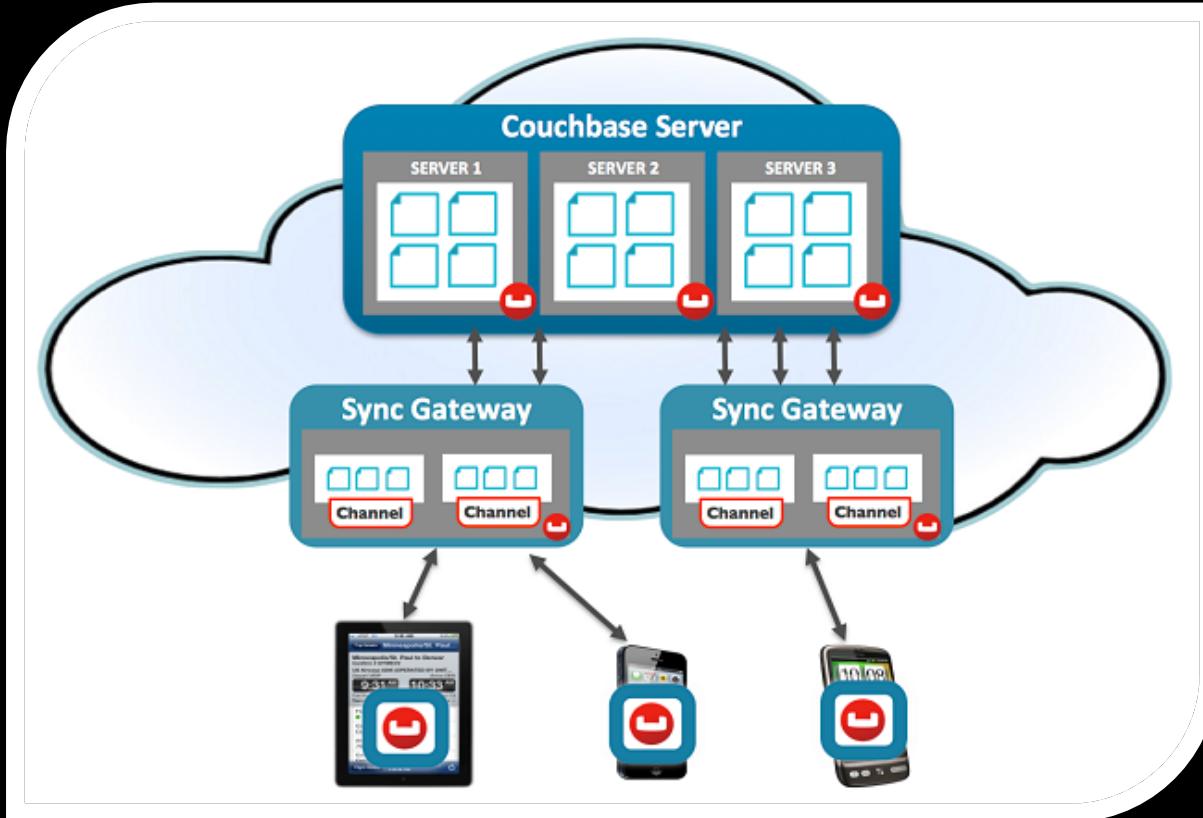
- Better booking experience for customers
- Improved management of cached data without mobile re-architecture
- Cross-platform support
- OOTB mobile sync

Outcomes

- Better experience for customers: vastly improved app ratings
- 60%+ faster booking process
- Reduced infrastructure cost: reduced network traffic for bookings 87%



Couchbase Sync Gateway at Ryanair

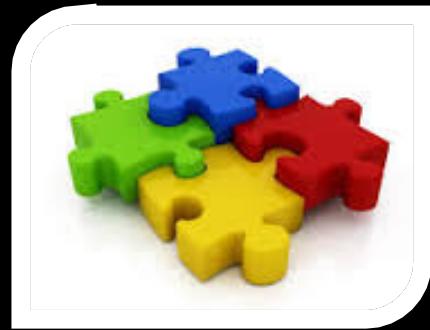


- Couchbase Lite is used as the on-device database for Ryanair's native iOS and Android applications
- Couchbase Sync-Gateway is used to push the latest revision of semi-static reference data used within the Ryanair application to all mobile clients

Improving Customer Experience - Data Integration



What our customers tell us about this challenge



“ We have *data in disparate systems* that we would like to *integrate* into our customer facing applications. ”



“ We would like to *aggregate data* into a *single customer view*, for internal and customer facing functions. ”

IMPROVING CUSTOMER
EXPERIENCE &
ENGAGEMENT

Fast access to credit information, vehicle registrations and valuations with Couchbase as standardized distributed caching layer



Requirements

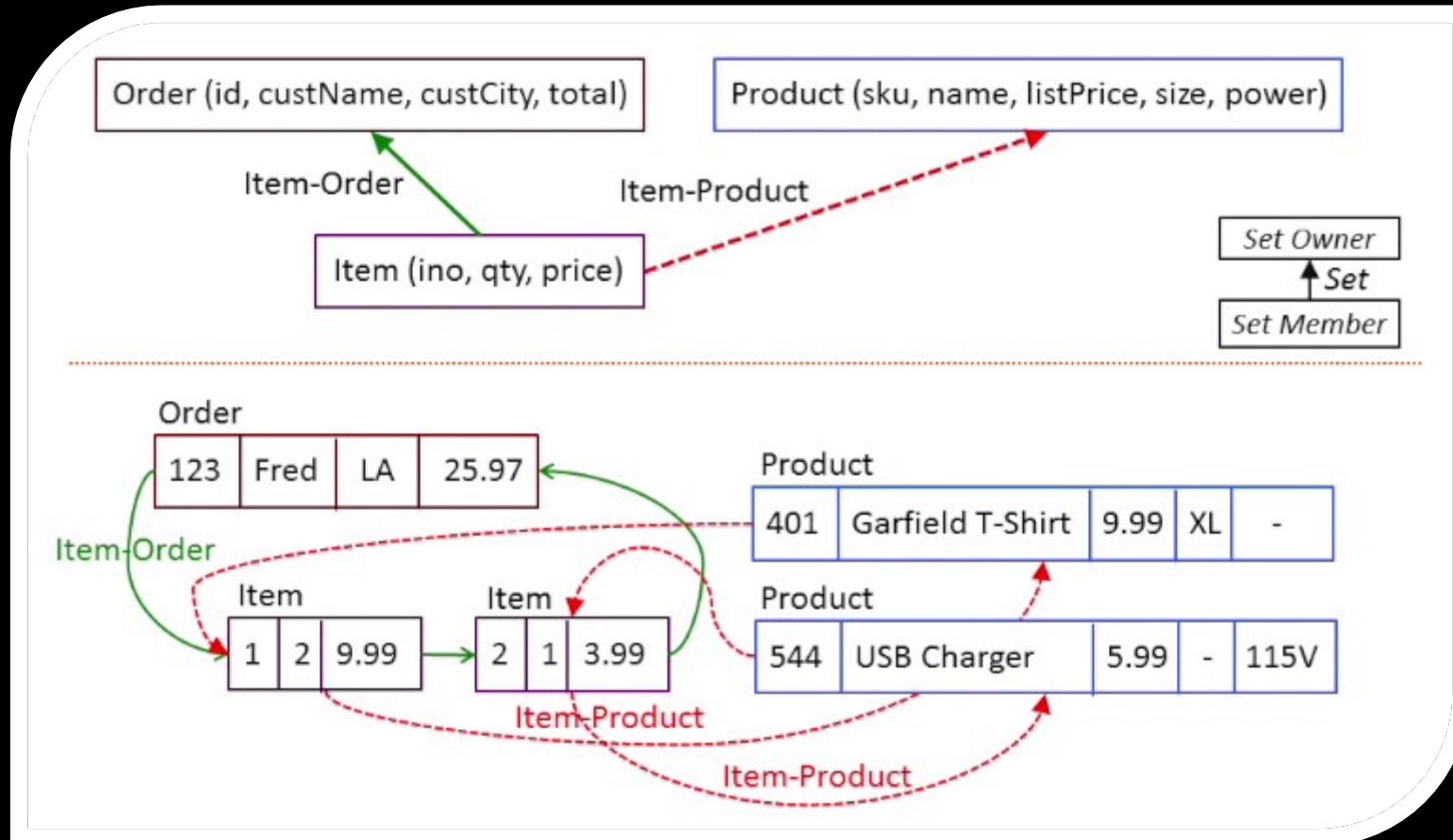
- Support massive increase in workload due to price comparison websites
- Affordable alternative to scaling mainframes
- Help new, rich-functionality applications come online quickly and easily

Outcomes

- Lowered costs AND increased performance:
- reduced calls to mainframe by 60%
- Improved dev agility with support for JSON



Couchbase JSON Support Critical at Experian



- JSON support meant apps could be built at a fraction of the time
- Avoided the dreaded “impedance mismatch”
- Support for any binary data type

Improving Customer Experience - Legacy Tech



What our customers tell us about this challenge



“*It takes a lot of money, time, and effort to get our legacy databases to perform at acceptable levels.* **”**



“*Predictable performance is a key issue. If there is a surge of user activity or if we add a new data set, we struggle to keep up.* **”**



FASTER INNOVATION &
TIME TO MARKET



Real-time intelligent data processing pipeline allows ingest of data from sensors/devices at scale and provides insight needed to take action



Requirements

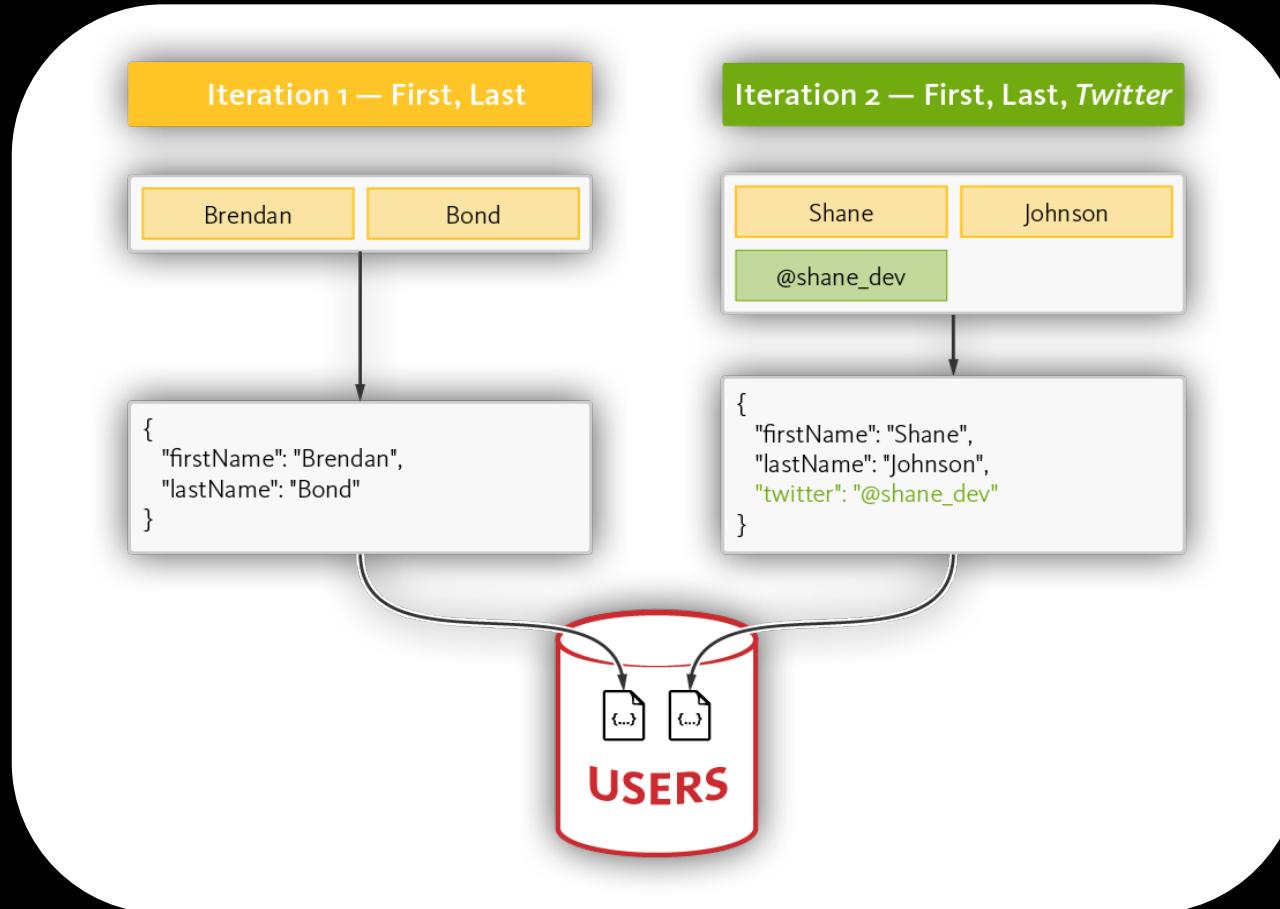
- Deliver new products and services easily
- Manage massive datasets and interact with devices, sometimes unconnected
- Capture new data types at high speed while providing near real-time analytics

Outcomes

- Performance at scale, auto sync device/cloud
- **Rapidly evolve schema** as requirements change
- Push-button scalability
- Five 9s HA and DR (5min/year)



Verizon Couldn't Waste Time on Schema Changes



- Schema flexibility was key in Verizon's comparison of Couchbase vs. Cassandra
- IoT use case drove many new and evolving datatypes that required total schema flexibility
- It was all about agility!



FASTER INNOVATION &
TIME TO MARKET



Dynamic product catalog service, including near-real-time stock updates, product catalog, and price, stock and customer order information



Requirements

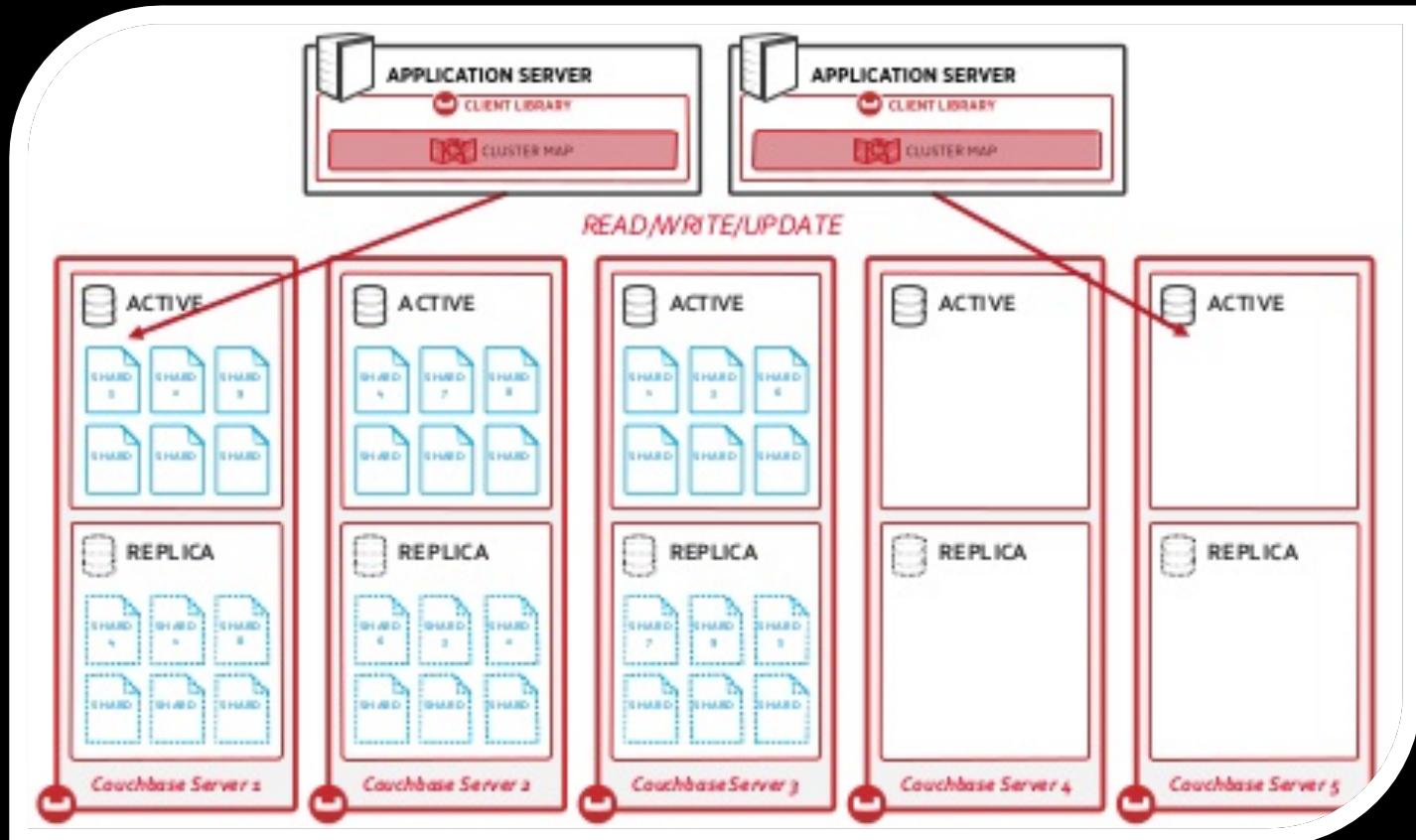
- Data stored in multiple relational databases
- Store, share and update product data for 10M items
- Support frequently changing data and multiple data structures

Outcomes

- Easily **scale** to 10m products and 35k requests per second
- Flexible schema support fast-changing SKUs
- Low-latency access for great user experiences



Tesco Needed Horizontal Scaling



- Critical to their Microservices-based approach
- Can't take any chances on Black Friday and during other spiky periods
- Ability to rapidly deploy compute without disruption or huge costs
- Critical in comparison vs. MongoDB



Couchbase acts as profile store, replacing traditional relational databases



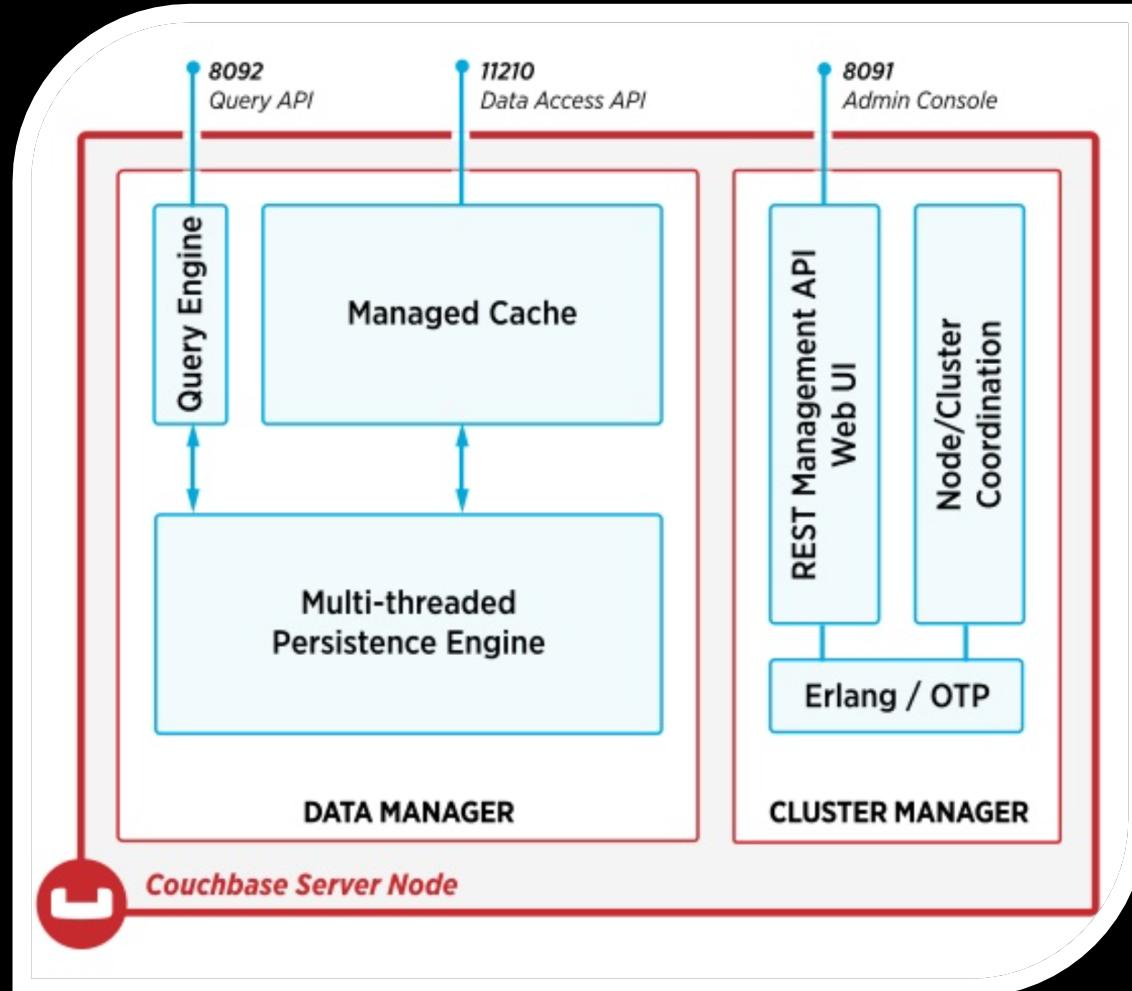
Requirements

- Growing data needs to be tracked with <ms latency and high throughput
- Relational systems unable to scale
- HA/DR solutions needed custom development

Outcomes

- Memory-first architecture for <1ms response times
- Complete HA/DR solution delivers 24x365 application uptime
- Neural networking algorithms

Couchbase's Single Node Architecture - Key for FICO



- Rapid time-to-market and zero failure tolerance required a solution with integrated caching
- No need to bolt on 3rd party cache to get desired low-latency
- Key decision criteria vs. other NoSQL when moving off Oracle



Couchbase is used for online pricing, catalogs, Daily Deals, customer profiles, mobile apps, and much more



Requirements

- Better management of personalized catalogs, 1.6 billion real-time rules
- Inability to scale MongoDB, DB2, and many other DBs easily/affordably
- Simplified replication, easy scalability

Outcomes

- Simplified management with **N1QL** and JSON, less tech sprawl
- Enhanced experience by boosting performance, scalability, reliability
- Improved responsiveness and availability



N1QL Absolutely Critical to Staple's Choice

The diagram illustrates the comparison between SQL and N1QL queries and their results.

SQL STATEMENT:

```
SELECT b.name, YEAR(a.year) AS year, a.name AS award
FROM awards a INNER JOIN books b
ON a.book_id = b.id
WHERE a.year > 1969
ORDER BY name, year, award
```

SQL RESULTS (ROWS):

name	year	award
Gateway	1978	Hugo
Gateway	1978	Nebula
Neuromancer	1984	Philip
Neuromancer	1985	Hugo
Neuromancer	1985	Nebula
-	-	-

N1QL STATEMENT:

```
SELECT b.name, DATE_PART_STR(a.year, "year") as year, a.
FROM awards a INNER JOIN books b
ON KEYS a.book_id
ORDER BY b.name, year, award
```

N1QL RESULTS (DOCUMENT):

```
{
  "results": [
    {"name": "Gateway", "year": "1978", "award": "Hugo"},
    {"name": "Gateway", "year": "1978", "award": "Nebula"},
    {"name": "Neuromancer", "year": "1984", "award": "Philip"},
    {"name": "Neuromancer", "year": "1985", "award": "Hugo"},
    {"name": "Neuromancer", "year": "1985", "award": "Nebula"}
  ]
}
```

- Heavy N1QL users
- CRUD is a quick and easy way for them to manage complex product catalogs
- Key in their decision when comparing against MongoDB (complex query API)

REDUCING
INFRASTRUCTURE &
OPERATIONS COSTS



Needed to move reservations off mainframe / away from Oracle to deploy new applications faster and more reliably



Requirements

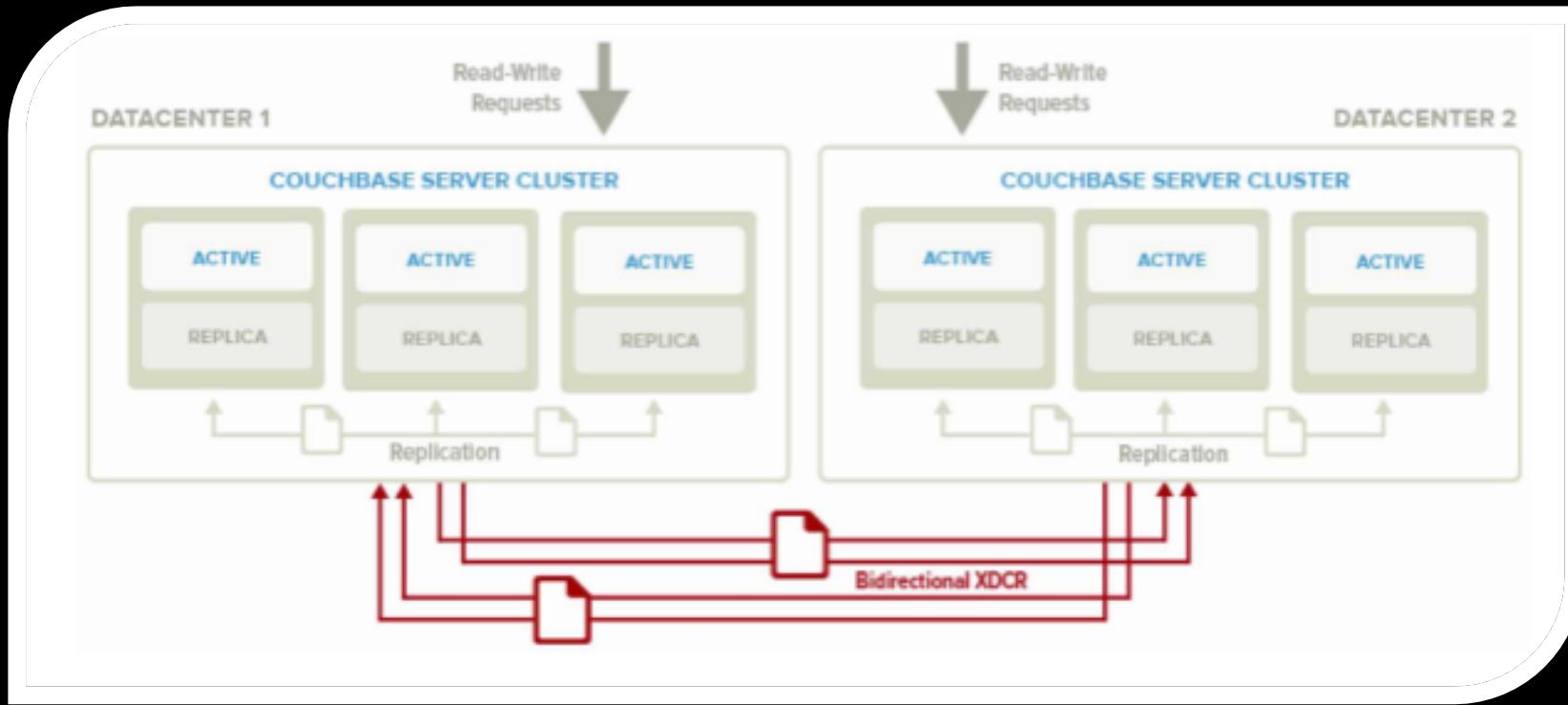
- Single central mainframe - outage could cost millions in lost booking revenue
- Needed to increase performance, distribute data
- Support cloud strategy with commodity hardware

Outcomes

- Scalable, flexible cloud-based model reduces dev costs, improves speed
- N1QL delivers customized hospitality experiences
- **Highest levels of availability** without extra add-ons, \$\$\$



Marriott needed “OOTB” High Availability



- No need for extra add-ons, code, or additional configuration to get HA/DR
- Bidirectional replication across global datacenters
- Elastic scalability, add nodes to scale systems to meet SLAs
- Key feature when comparing against other NoSQL

REDUCING INFRASTRUCTURE & OPERATIONS COSTS



As user base skyrocketed, more and more Couchbase was replacing Oracle. But CE nodes were proliferating too fast, causing H/W costs to soar.



Requirements

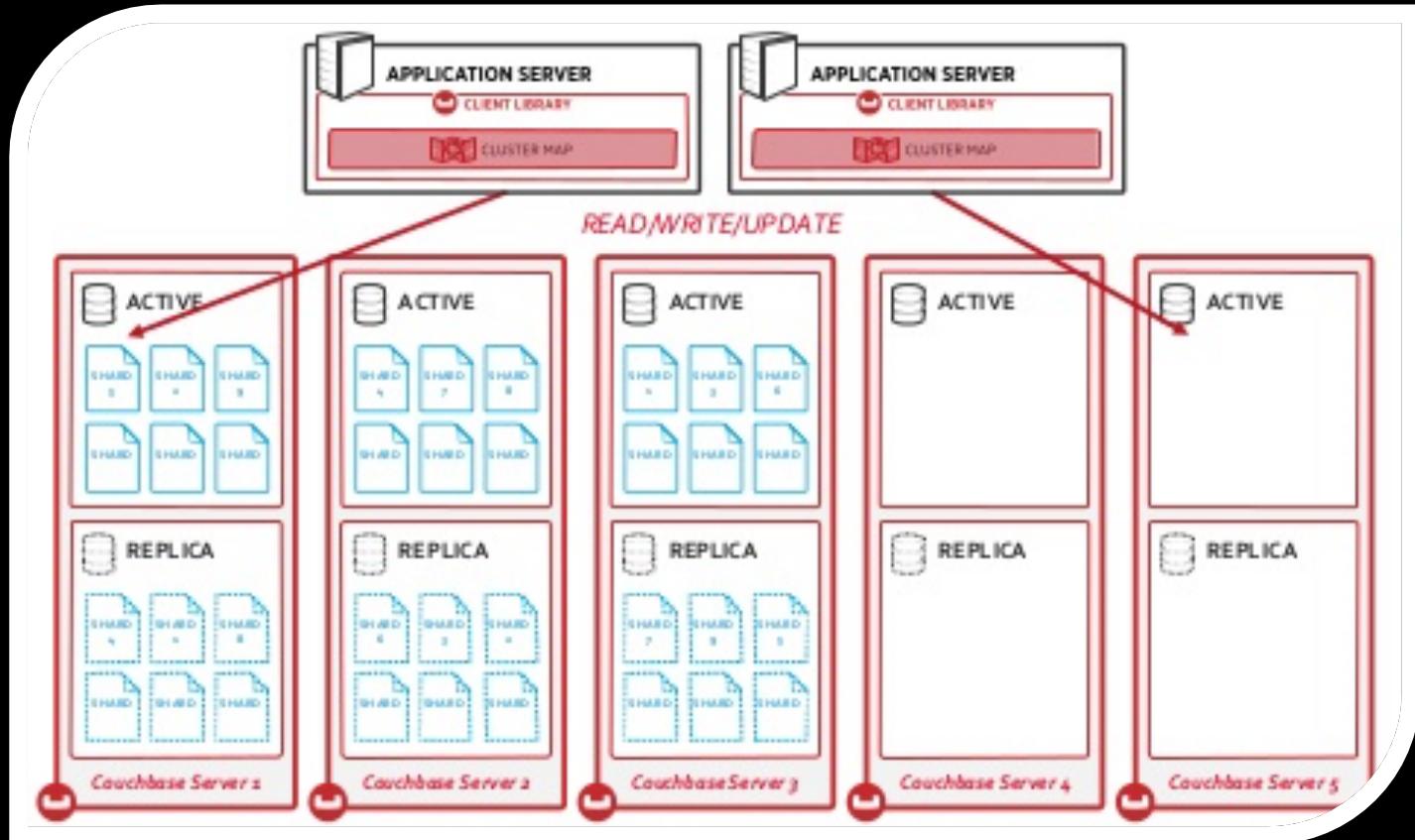
- Ability to scale reads
- Memcached caused manageability and reliability problems
- Needed to balance performance with eased load on ops team

Outcomes

- Tremendous performance at scale, <4ms latency for over 2.5 billion items
- 10+ million queries per second across all servers and clusters
- Reduced ops costs with simple **scaling**



LinkedIn Needed Easy, Horizontal Scaling



- Just add a node and do a "one-button" rebalance
- Dramatically simplified scale out compared to RDBMS and other NoSQL
- Ability to rapidly deploy compute without disruption or huge costs
- Upgraded from Couchbase CE after 2000 nodes deployed

REDUCING
INFRASTRUCTURE &
OPERATIONS COSTS

Couchbase supports United's crew management platform, streamlining and modernizing crew scheduling and resources



Requirements

- Identify truly scalable NoSQL solution
- Streamline and improve crew scheduling
- Support flexible schema and complex queries

Outcomes

- Achieved NoSQL Standardization
- Allow highly mobile workforce to resolve issues from the field
- Out of the box DR and protection for critical ops
- Cloud-enabled: can be run in AWS and Azure



United: “Mobile, XDCR, AND Horizontal Scaling!”

The screenshot shows a mobile application interface titled "Pairing D6700/02 - #5 Segments". The screen displays five completed activities with their respective details:

- Flight 440 IAH-JFK**: Report time 09/02/14 10:30, Gate # B22, Scheduled time 09/02/14 11:30, Arrival time 09/02/14 13:30, Dispatcher # +1 44319870. A "View flight" button is present.
- Shuttle**: Shuttle name Shuttle Chicago, Location Next to exit E2 outside, Pick-up time 09/02/14 14:00, Telephone +1 44319870.
- Hilton Hotel**: Reservation # 4455677, Telephone +1 44319870, Check-in 09/02/14 15:00, Check-out 09/04/14 11:30, Address 890 Adams St. Chicago, IL.
- Flight 670 JFK-IAH**: Report time 09/04/14 12:30, Gate # B22, Scheduled time 09/04/14 13:00, Arrival time 09/04/14 14:30, Dispatcher # +1 44319870. A "View flight" button is present.
- Layover**: Flight 344 IAH-LAX, Scheduled time 09/04/14 18:30, Arrival time 09/04/14 20:00.

A "UNITED" logo is at the bottom left. A black overlay at the bottom right contains the text "Enabling new tools for pilots".

- Enterprise Architect Team at UA did an extensive evaluation of all major NoSQL vendors
- Teams couldn't scale MongoDB and DynamoDB
- Couchbase is now the NoSQL standard at United



Couchbase Solves Problems!



Improving customer experience & engagement

1. Increasing personalization means sacrificing performance ([eBay](#))
2. Mobile apps lag web versions in features and usability ([Ryanair](#))
3. Difficulty integrating data from disparate systems into consumer apps ([Experian](#))
4. Difficulty aggregating data into a single view of customer ([Comcast](#))
5. Takes too much time, money and effort to get legacy DBs to perform at acceptable levels ([Nielsen](#))
6. It is difficult to ensure *predictable* performance ([Sky](#))



Faster innovation & time to market

1. Backend schema changes delay application dev cycles ([Verizon](#))
2. NoSQL implementations need to be mission critical ([DirecTV](#))
3. Requirements change faster than the backend database can keep up ([Tesco](#))
4. Mobile apps just take too long to develop and integrate with backend systems ([GE](#))
5. The latest and greatest database technology isn't being leveraged ([Fico](#))
6. Too many technology piece parts tends to stifle innovation ([Staples](#))



Reducing infrastructure & operations costs

1. The database layer is too costly and time consuming to scale effectively ([eBay](#))
2. Frequent mainframe access is too costly and slow ([Marriott](#))
3. NoSQL "sprawl" is leading to increased operational and H/W costs ([LinkedIn](#))
4. Scaling up certain NoSQL solutions is far to costly and time consuming ([Nuance](#))
5. It's a struggle to operationalize so many different database technologies ([BT](#))
6. It's hard to rationalize paying multiple DB vendors for marginally-differentiated technology ([United](#))



5

Introduction Couchbase - Who are we?

Couchbase, by the Numbers



300+
EMPLOYEES

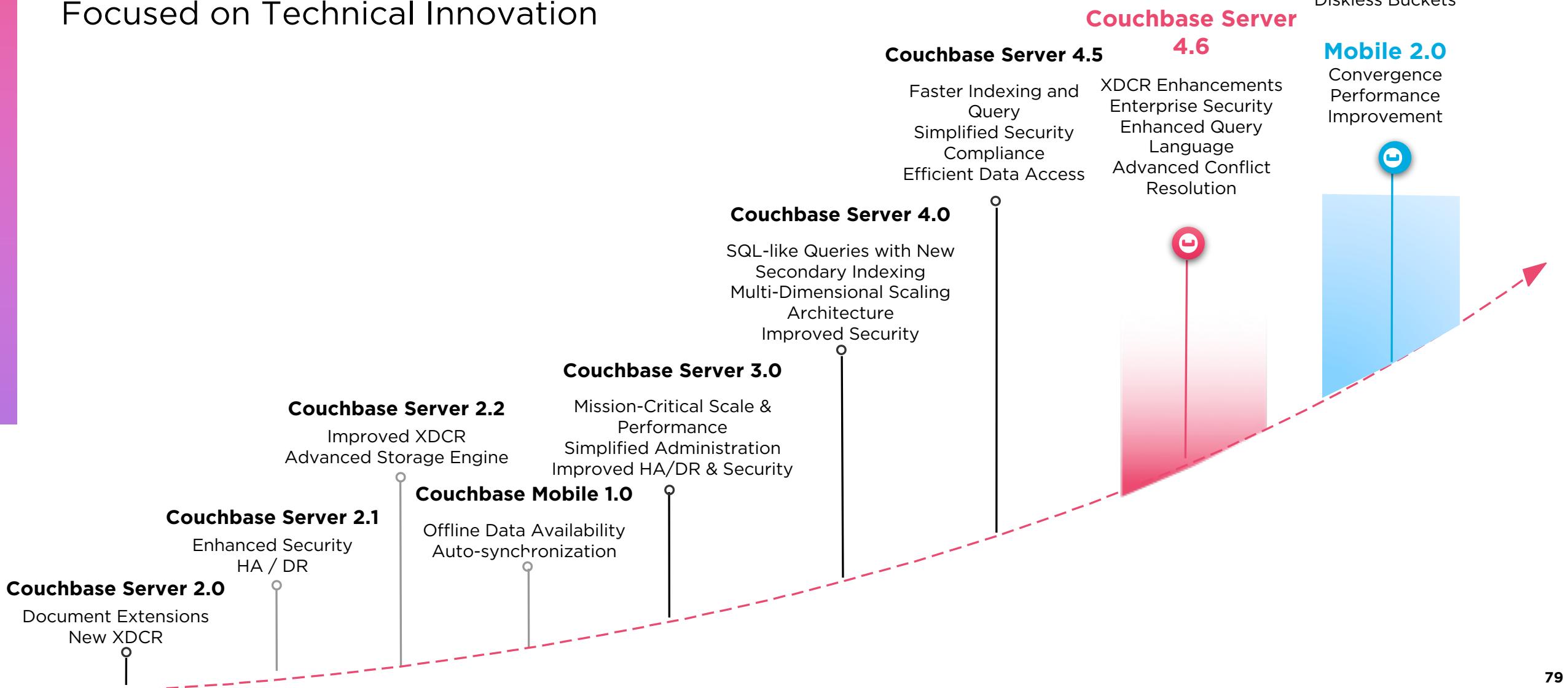
100%
OPEN SOURCE

500+
CUSTOMERS



The Journey thus Far

Focused on Technical Innovation





Performance

KV: 2 M ops/s/node. 60% > 4.5

Import/Export: 2X to 14X > 4.5

N1QL-GSI:

10X better than 4.5 on average
New YCSB-SOE workload
Pagination: 125X > MongoDB

XDCR:

45% better throughput than 4.5
50% lower latency than 4.5
2X faster replication over SSL

FTS (vs. Dev Preview)

Throughput 5x > 4.6 DP

Fast Failover:

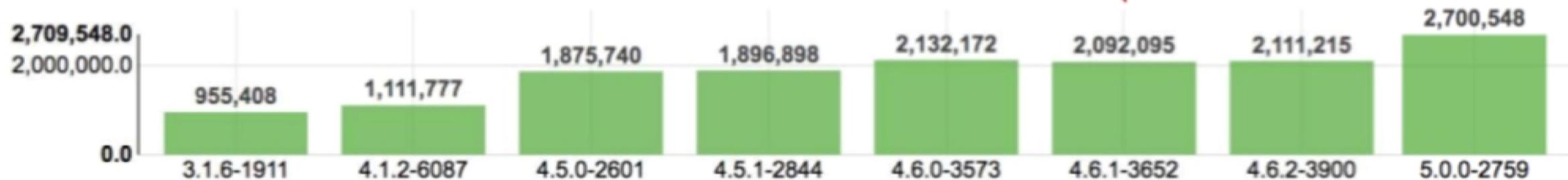
KPI: 10 seconds
Actual: ~7 seconds



Even at its CORE - Key-Value Keeps Getting Faster

Max ops/sec, cbc-pillowfight, 2 nodes, 50/50 R/W, 512B binary items

Cluster: ares OS: CentOS 7 CPU: E5-2630 v4 (40 vCPU) Memory: 64 GB Disk: SSD ←



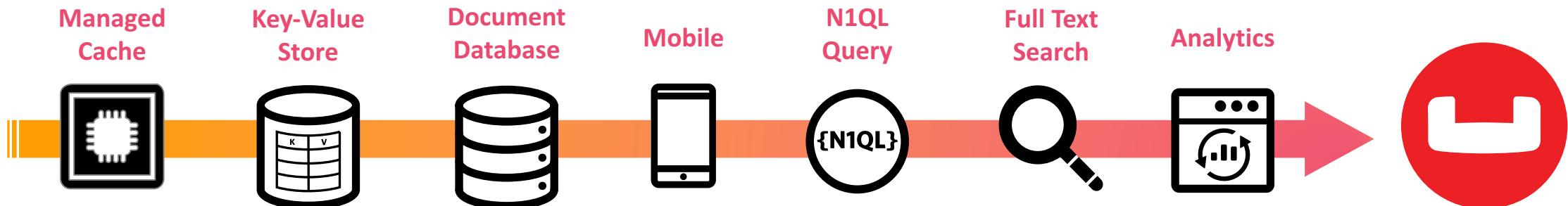
50/50 R/W with 0.5K Object Size

Core Principles Drove Platform Evolution



Started with Core Principles

- True auto sharding
- JSON-based flexible data model
- Memory-first Architecture
- Asynchronous approach to everything
- Scale workloads independently



Thank you



Couchbase