



WELCOME

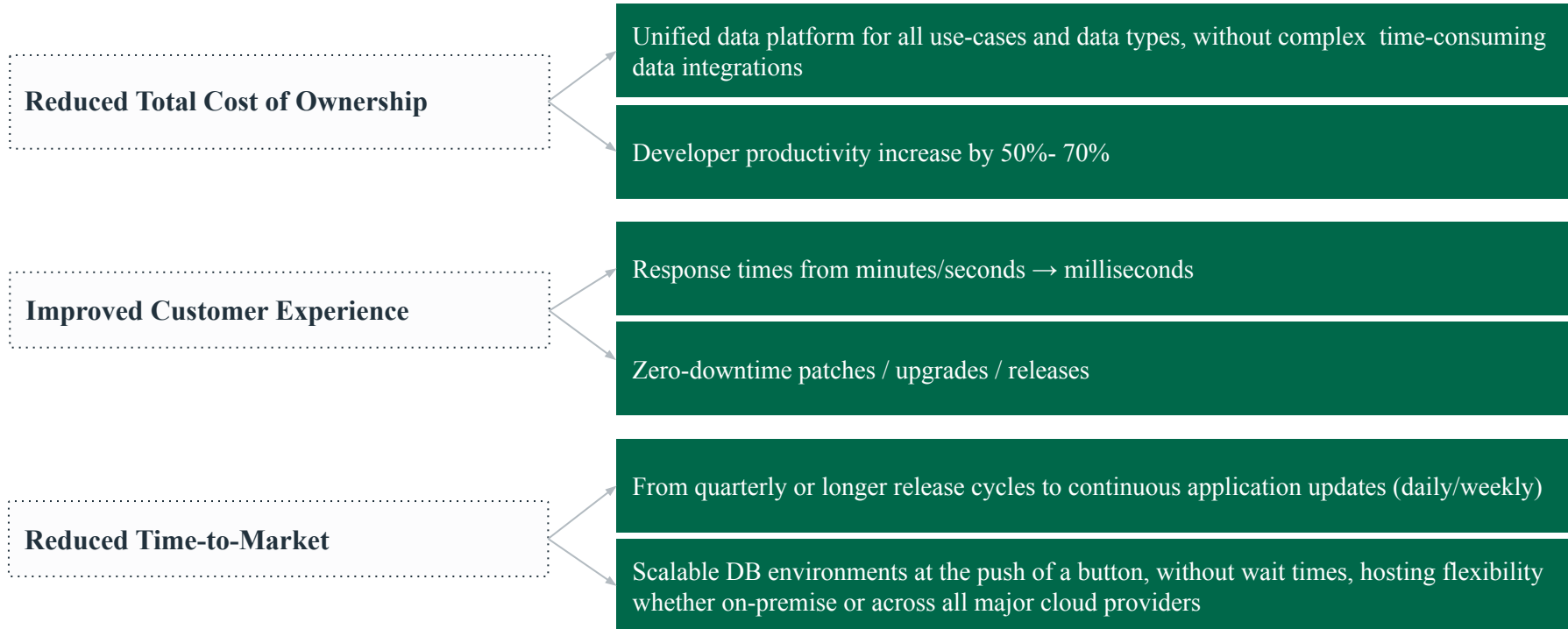
US Bank + MongoDB

2023 BAW Partnership

June 22th, 2023

RDBMS vs. MongoDB

Business Value



MongoDB Value Proposition

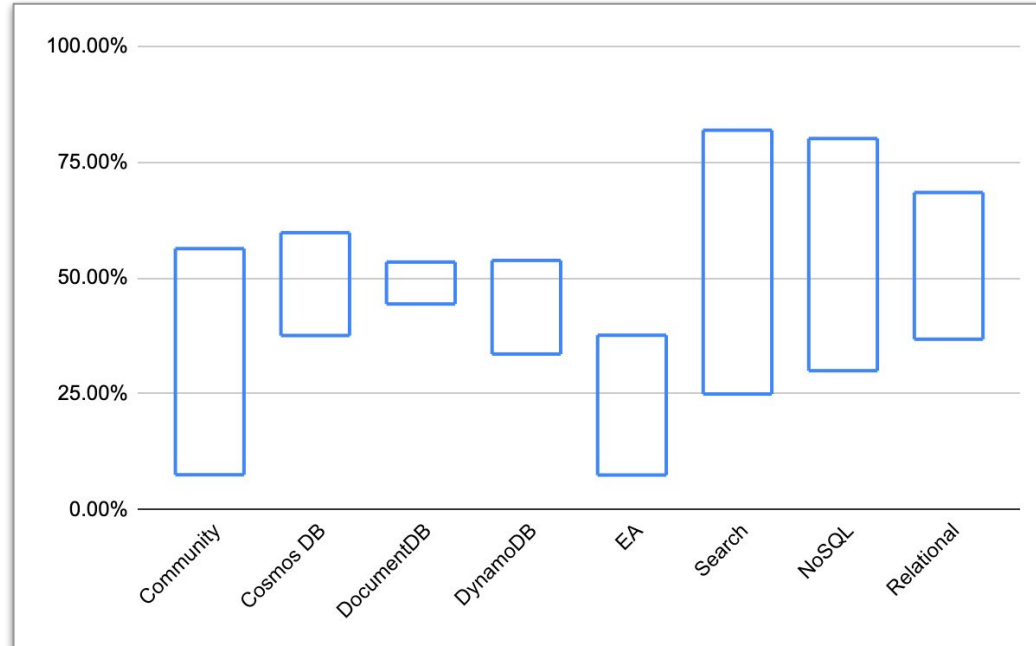


	Cost	Effort	Performance
RDBMS	TCO savings of 55%-75%	50% Effort reduction 70% increase in DevOps Productivity	Up to 20x better Response Times
NoSQL	DB cost savings of 30% - 3x	20% Effort reduction 40% increase in DevOps Productivity	13x faster Ingest Rates at same cost
MongoDB Community	TCO savings of 10% - 30%	12x Effort reduction 80% DevOps Productivity	Up to 50% improvement through automated Performance Advisor
Heterogeneous Tech-Stacks	Depends on the Technology Stack	Depends on the Technology Stack	Depends on the Technology Stack

All numbers are taken from implemented customer projects and vary depending on the amount of data and type of requirements.



MongoDB Savings Ranges



Every year MongoDB **Business Value Consulting** builds hundreds of business cases collaboratively with our customers

This chart shows the **range of savings** predicted across every business case completed in the last full financial year

Results are categorised by the technology being **replaced** or considered as an **alternative** solution to MongoDB

The future state for the majority of these business cases is **MongoDB Atlas**, a cloud managed service

	Community	Cosmos DB	DocumentDB	DynamoDB	EA	Search	NoSQL	Relational
n=	38	17	5	4	47	5	6	7
Average TCO	31%	47%	47%	44%	20%	46%	48%	52%

Complex legacy Oracle & Microsoft SQL estate of the RBS Investment Bank modernizes with MongoDB to build a Data as a Service “Data Fabric”

£10M+

in license, hardware, &
operational cost avoidance

34%

increase in
developer efficiency

284:1

storage savings (cost per TB)

14,105

legacy servers can be
decommissioned

“We’re replacing hundreds of separate products with a platform called Data Fabric, which will help reduce cost significantly and dramatically increase the speed at which we can deploy new capabilities.”

—Ross McEwan
CEO @ RBS

Comparative TCO Analysis:

Current State vs. At-Scale MS SQL+Redis



Category	Description	Annual OPEX Cost
Infrastructure	<ul style="list-style-type: none"> On-premise infrastructure (HW/OS/Mgmt tools, FTE inclusive) (est. avg. \$8K OPEX cost per VM per year) SQL Server: 12 servers/blades (Prod: 4, Perf: 4, Dev/Int: 4) Redis: Assume 1 256GB RAM VM for 20 shards (12.5GB RAM) (non-prod only) 	\$104,000
Database	<ul style="list-style-type: none"> MS SQL Server - est. avg 12 servers/blades, 4+ cores (est. \$10K per core per year) 	\$480,000
Cache	<ul style="list-style-type: none"> Redis in-memory cache to improve performance (currently small scale non-prod only: 20 shards - assume Dev) 	\$100,000
Labor	<ul style="list-style-type: none"> MS SQL DBAs: est. 2 FTE @ \$250K annual fully burdened rate) Redis - est. 3 FTE @ \$180K annual fully burdened rate) Developer & DBA effort: est. 20% of 4 FTE (@\$180K/yr) (App tier, persistence layer, caching layer, queuing layer) Staffing includes provisioning, maintenance, administration, DB patching and upgrades, incident management, QA/integration testing, data design development and performance management. 	\$1,184,000
Opportunity Cost (Missed Opportunities) and/or Additional Value with MongoDB	<p><u>Opportunity Cost (Missed Opportunities of Status Quo):</u></p> <ul style="list-style-type: none"> Performance and scale limits: serves up to 1,000 offers at once Data inflexibility: Unable to target by demographic/geography Missed revenue growth opportunity by inability to scale to higher customer volumes Stakeholder dissatisfaction (Providers: lack of scale/targeting; customers: lack of targeted/valued offers) 	tbd
Annual Total Cost of Ownership (TCO) - Status Quo		\$1,868,000

Description	Annual OPEX Cost
<ul style="list-style-type: none"> On-premise infrastructure (HW/SW/Mgmt tools, FTE inclusive) (est. avg. \$8K OPEX cost per VM per year) SQL Server: 52 VMs (conservative assumption 6x Prod/Perf growth factor) Redis: Prod + Perf: Requires 6 VMs per environment (224 shards) (6x growth factor) Redis: Dev + Int: Assume 1 VM per environment (similar to current state non-prod) 	\$528,000
<ul style="list-style-type: none"> MS SQL Server - est. 6x growth for Prod/Perf VMs 24 Prod, 24 Perf, 2 Integration, 2 Dev VMs, 4+ cores (est. \$10K per core per year) 	\$2,080,000
<ul style="list-style-type: none"> Redis in-memory cache scaled up to handle 20B offers (estimated cost) Prod: 224 shards @ 12.5GB RAM (112 shards * 2 data centers) (6x growth factor) Total cost of all environments shown (estimate) 	\$5,000,000
<ul style="list-style-type: none"> MS SQL DBAs: est. 4 FTE @ \$250K annual fully burdened rate) Redis - est. 6 FTE @ \$180K annual fully burdened rate) Developer & DBA effort (MS SQL, Redis scaled): est. 20% of 8 FTE (App tier, persistence layer, caching layer, queuing layer) Activities include provisioning, maintenance, administration, DB patching and upgrades, incident management, QA/integration, testing, data design development, performance mgmt. 	\$2,368,000
<p><u>Opportunity Cost (Missed Opportunities with Option 1):</u></p> <ul style="list-style-type: none"> Higher TCO compared to MongoDB option (see TCO difference) Continued data inflexibility (Redis does not solve this problem) Continued development and operations complexity due to complex and rigid tech stack architecture, multiple technologies Continued challenges to improve targeting 	tbd
Annual TCO - Self-Managed Redis / MS SQL Server at Scale	\$9,976,000

Comparative TCO Analysis:

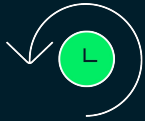
MongoDB



Category	Description	Annual OPEX Cost
Infrastructure	<ul style="list-style-type: none">On-premise infrastructure (HW/SW/Mgmt tools, FTE inclusive) (est. avg. \$4K OPEX cost per VM per year)MongoDB: Prod + Perf: 14 shards * 5 nodes (70 total) per environment;MongoDB: Cert: 2 shards * 5 nodes (10 total).MongoDB: Dev: 2 shards * 3 nodes (6 total).	\$624,000
Database	<ul style="list-style-type: none">MongoDB Enterprise Advanced subscriptionEst. 39 shards of 5 med. size server nodes (replication across 3 data centers (Never Down))	\$3,931,502
Cache	<ul style="list-style-type: none">Not required - high-performance cache capabilities included with MongoDB	n/a
Labor	<ul style="list-style-type: none">MongoDB: est. 3 FTE @ \$180K annual fully burdened rate)Developer & DBA effort with MongoDB: est. 10% of 4 FTE (App tier + unified persistence, caching, queuing layers (MDB))<i>Activities include provisioning, maintenance, administration, DB patching and upgrades, incident management, QA/integration, testing, data design development, performance mgmt.</i>	\$612,000
Opportunity Cost (Missed Opportunities) and/or Additional Value with MongoDB	Additional Value with MongoDB: <ul style="list-style-type: none">Maximize data flexibility and scalability to achieve business volume scaling and targeted offer goals (advanced targeting by demographics, geo-location, customer searchable)Revenue and operating margin impact with improved targetingLowest TCO compared to Redis/MS SQL (see TCO difference)Simplified architecture reduces development and operations complexity with unified platformIncreased agility with faster development, integrations/testing and time-to-market	
Annual TCO - Unified MongoDB Architecture at Scale		\$5,167,502



The costs of running on legacy systems



Being late to market

Sprawling tech landscape and slow development process block new features



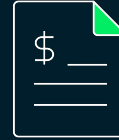
Poor user experience

Legacy systems are unable to meet availability and scale that customers demand



Missed opportunities

Modern apps create semi & unstructured data that is a poor fit for rigid, tabular data models

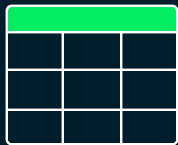


Lower technology ROI

Expensive hardware, punitive licensing, intrusive audits increases TCO



So why can't I just migrate my data from my relational database to MongoDB and call it a day?



You can, but MongoDB **isn't** designed for relational schemas.

It will be less performant and more expensive without changes.



Working with data in MongoDB is fundamentally different.

Data is organized **by access patterns** and how apps will generate/use data.



Moving to MongoDB is a **modernization** effort.

The entire architecture needs to be redesigned (i.e: breaking up monoliths, treating data as a service, etc.)



What is the result?

From this it was deduced that the current MongoDB schema was likely a fully Normalized schema. Subsequently, it was confirmed that "the schema was lifted originally and untouched from the original RDBMS schema", with no attempt to follow MongoDB Schema Best Practices to denormalize as much as possible.

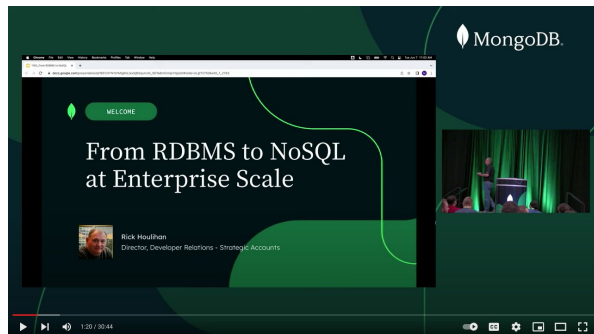
NOTE

THERE ARE NO SHORT-TERM SOLUTIONS OR WORKAROUNDS THAT WILL FULLY OFFSET THIS INEFFICIENT SCHEMA. OF THE ITEMS CURRENTLY BEING CONSIDERED:

- BETTER INDEXING: THERE ARE NO OTHER INDEXES THAT CAN BE BROUGHT TO BEAR
- INCREASED MEMORY: WILL LIKELY NOT MAKE A DIFFERENCE AT THIS TIME
- FASTER DISK WITH LOWER LATENCY: WHILE THIS COULD MAKE SOME SHORT-TERM DIFFERENCE, IT IS NOT AN ADEQUATE LONG-TERM SOLUTION

MongoDB DevRel Practice: Rick Houlihan

Rick Houlihan leads the developer relations team for strategic accounts at MongoDB. Prior to joining MongoDB, Rick was the Worldwide Technical Leader for NoSQL Services at AWS where he ran the BlackBelt team for Amazon Retail during the migration of **3,000 Oracle server instances** from relational database to NoSQL, educated **25,000 developers globally** and moved **12,000 workloads** off of Oracle.



Rick currently operates as the technical leader for the global strategic account team where he is responsible for building the market for MongoDB technology by facilitating the creation of similar Center of Excellence capabilities within our customer organizations through training, evangelism, and direct design consultation activities, working with our customers such as **Vodafone, Verizon, AT&T, Wells Fargo, Morgan Stanley** and **JPMorgan Chase**.



Our Ask

In order for our team to make the most of everyone's time during your design review session, we ask that you share a little more detail around your team's project or application.

Please review the questions below and share additional details where necessary:

What is the Project or Application you are working on or considering MongoDB for? This can also be your application or project name.

In preparation for the design review session please share the following items:

- Entity Relationship Diagram describing your logical data model
- List of common access patterns or queries
- Information regarding scale and velocity of those patterns

