



WELCOME

# MongoDB

## SI Associate Certification

Ramakrishna Kappagantula  
Partner Solutions Architect  
[ramakrishna.kappagantula@mongodb.com](mailto:ramakrishna.kappagantula@mongodb.com)

Rashmi Nayak  
Partner Solutions Architect  
[rashmi.nayak@mongodb.com](mailto:rashmi.nayak@mongodb.com)

Aicha Sarr  
Solutions Architect  
[aicha.sarr@mongodb.com](mailto:aicha.sarr@mongodb.com)



WELCOME

# MongoDB SI Certification

The MongoDB SI Certification program is a free, instructor lead or self-paced certification initiative that helps associates & architects deepen their expertise so as to expand business opportunities and help customers modernize to the cloud.



## Learning Path

Step 1

SI Associate

Developer/DBA/Architect/Leads  
(Any personas)

[Register here](#)

Step 2

Dev

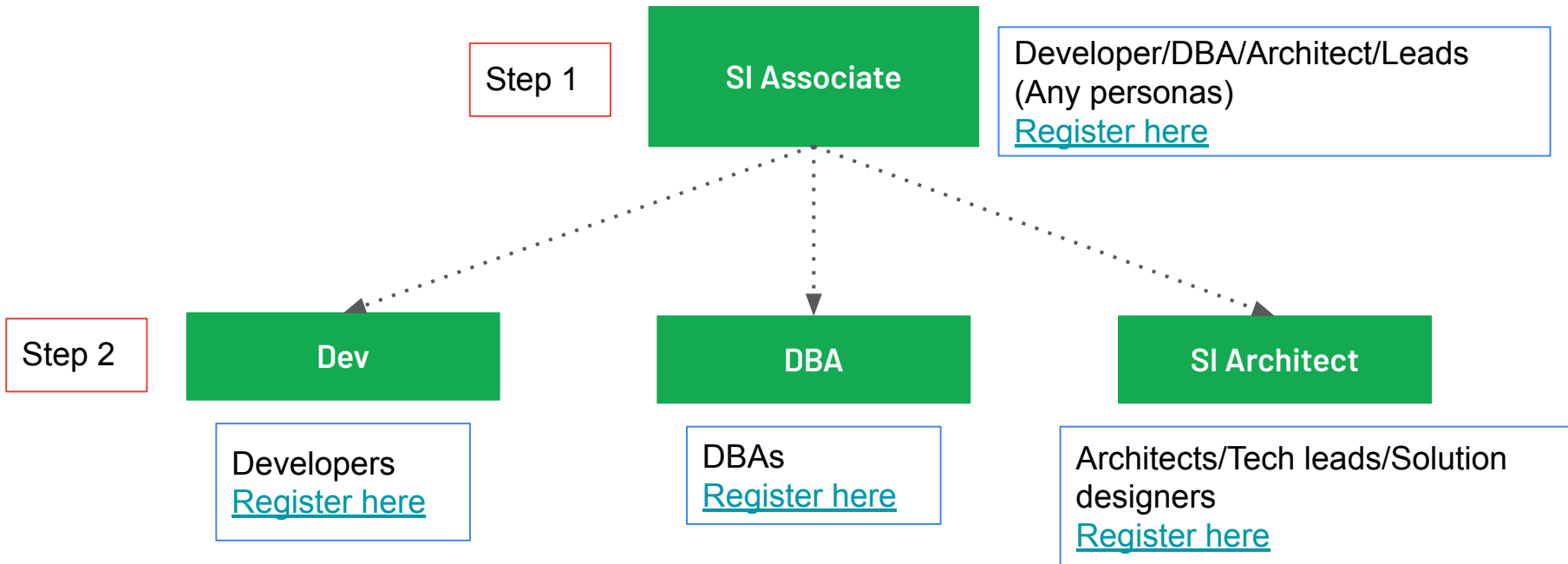
DBA

SI Architect

Developers  
[Register here](#)

DBAs  
[Register here](#)

Architects/Tech leads/Solution  
designers  
[Register here](#)

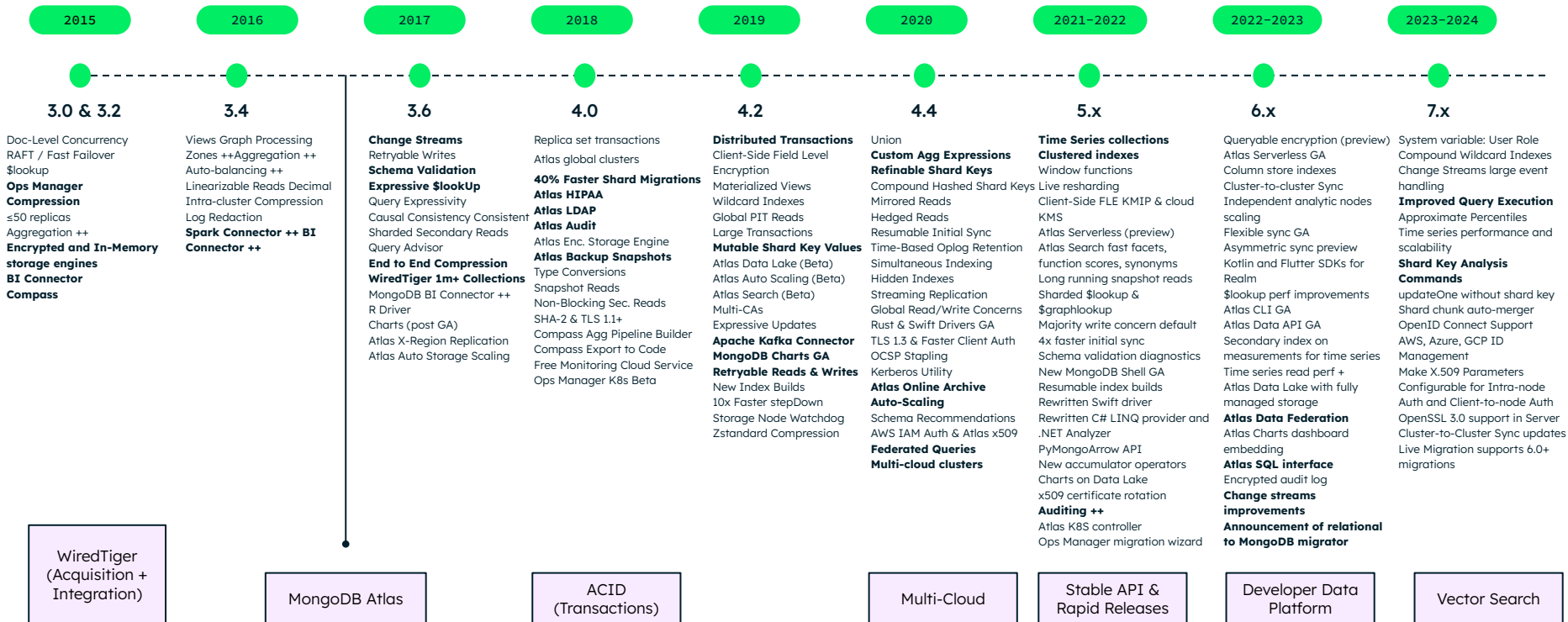




# MongoDB Value Proposition

# MongoDB features in a nutshell

STACK OVERFLOW SURVEY | MOST WANTED DATABASE

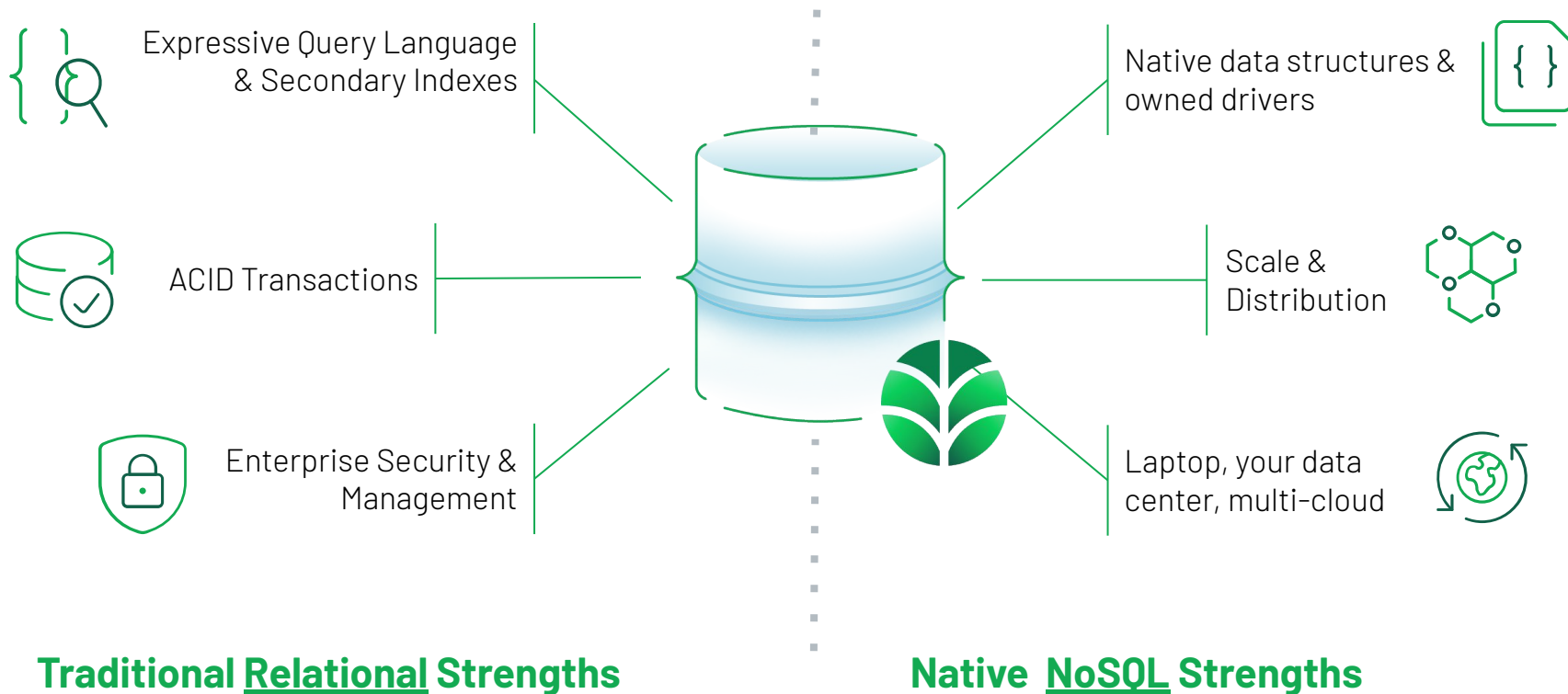


# MongoDB features in a nutshell



# MongoDB is **Mission Critical** and **General Purpose DB**

The ultimate combination of all data paradigms



# Have you ever thought of MongoDB can be used in **Mission critical** Applications??

## Payments



## Trading



## E-commerce/ Supply chain



## Oracle OFL



... and many  
more

## Mainframe OFL



Great Customer Experience

YoY Revenue Growth

Lower TCO

Innovation

Time to Market

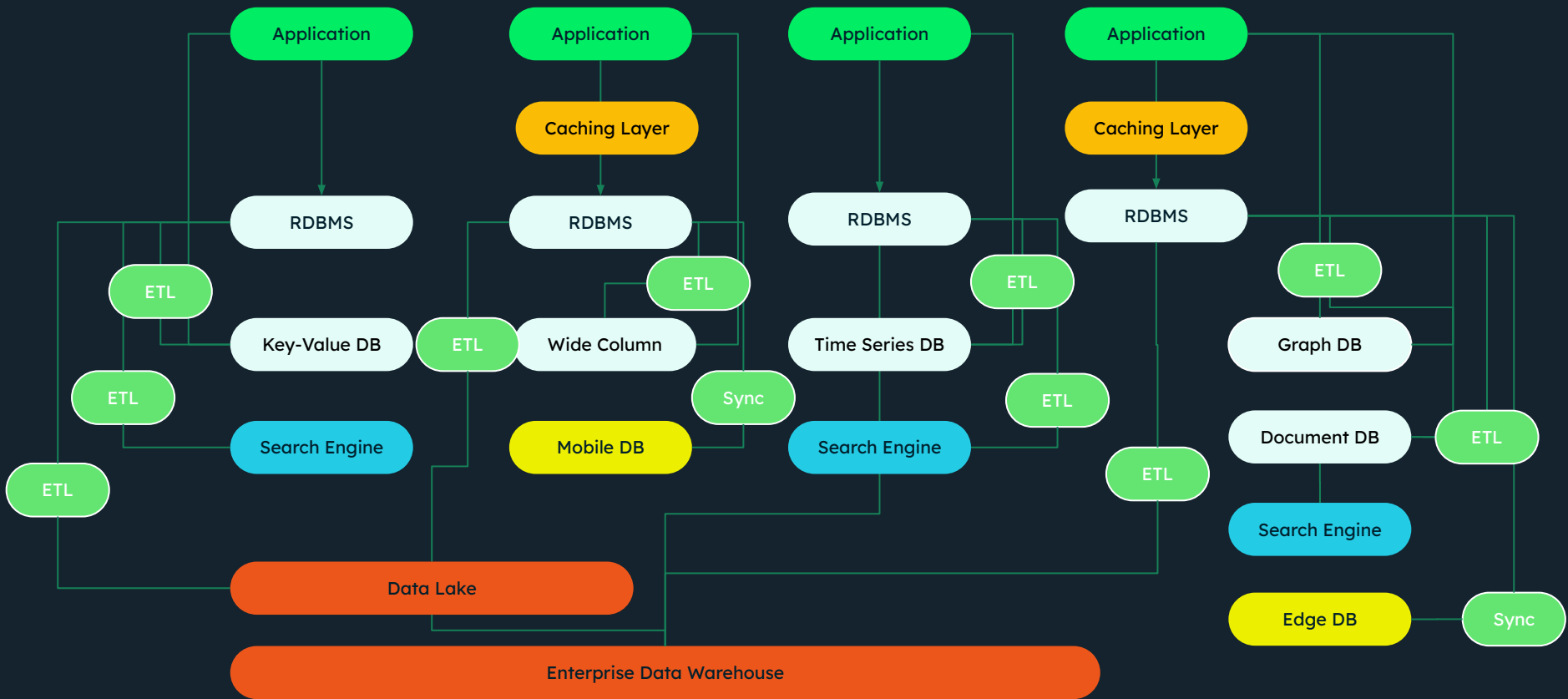
Increased Agility &  
Scalability

Improved Performance

Improvements in Service  
Delivery







Fragmented  
developer  
experience

Multiple  
operational and  
security models to  
rationalize

Significant data  
integration effort  
required

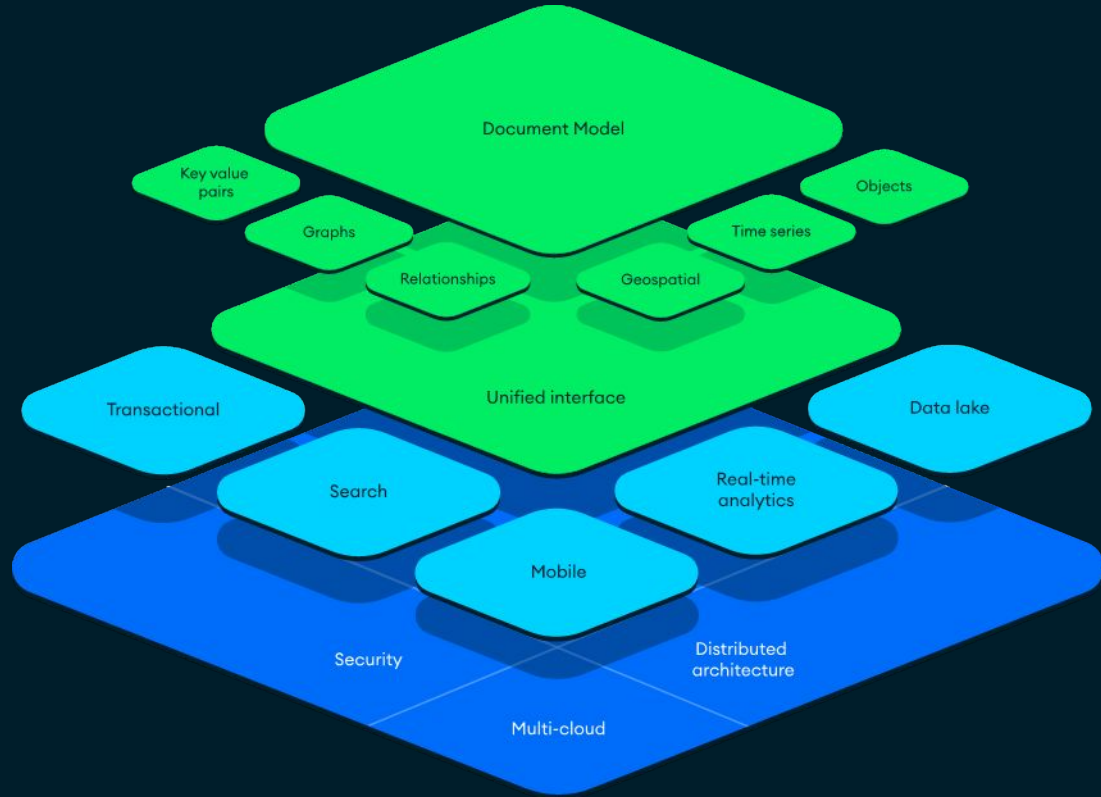
Unnecessary data  
duplication

This data architecture complexity creates a  
tax on innovation — a **Data & Innovation  
Recurring Tax (DIRT)**.



Accelerate innovation with

One interface.  
For any application.  
Anywhere.





# MongoDB replica sets

## Replica Set—2 to 50 copies

### Self-healing

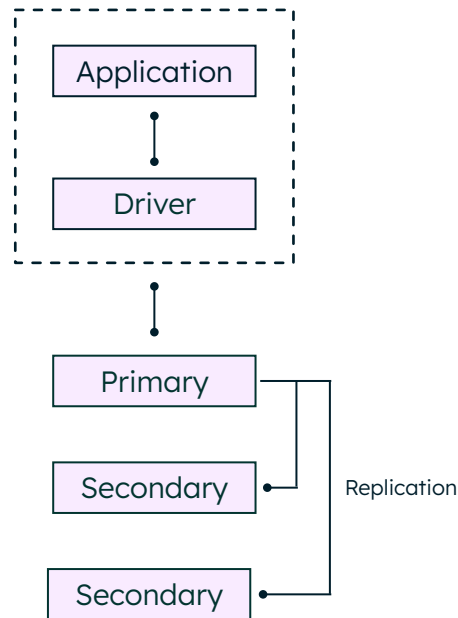
- Typical failover in 5 seconds or less
- Retryable reads and writes to catch temporary exceptions

### Data center aware, tunable durability, and consistency

### Addresses availability considerations:

- High Availability
- Disaster Recovery
- Maintenance

### Workload Isolation: operational & analytics



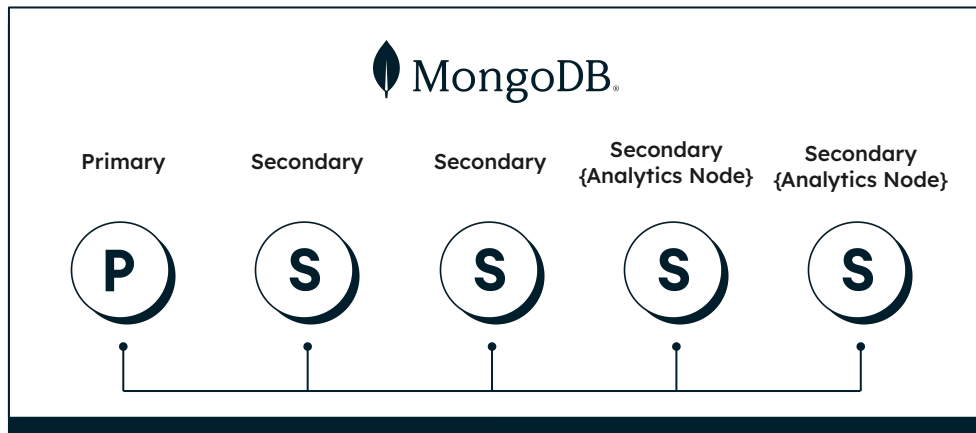


# MongoDB workload isolation

## Transactional Applications



Rich MongoDB Query API and distributed architecture allows you to run both Transactions and Analytics on the same cluster with no resource contention



## Analytics Consumers





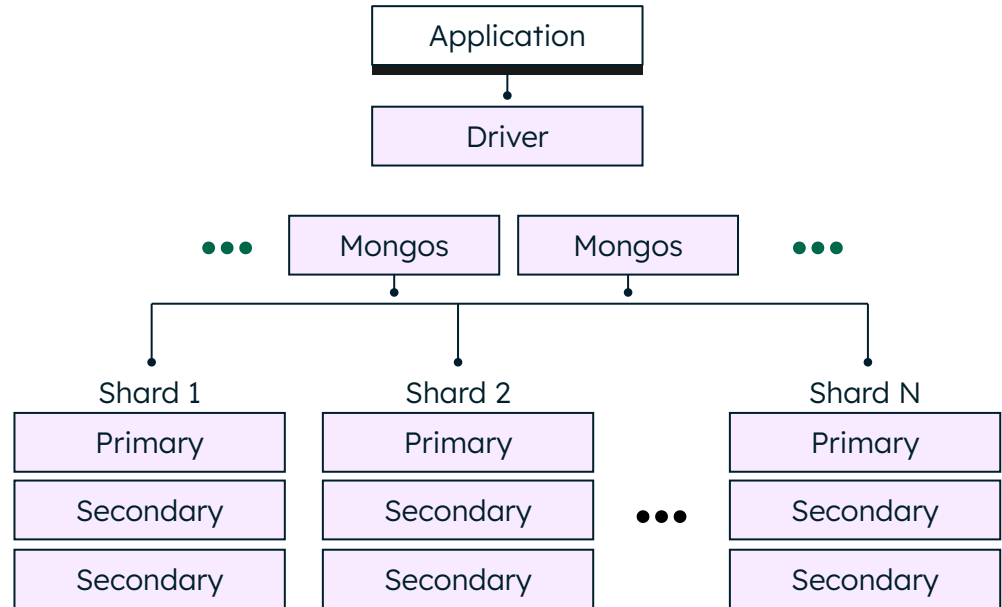
# Sharding architecture

## Horizontal scalability

Sharding

## High availability

Replica sets



# Démo

Atlas in action



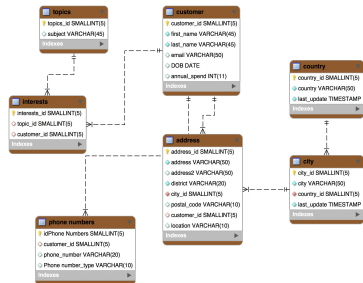


# The Document Model & MongoDB Query API





## INTUITIVE



## Tabular (Relational) Data Model

Related data split across multiple records and tables

# Contrasting data models

## Document Data Model

Related data contained in a single, rich document

```
{
  "_id" : ObjectId("5ad88534e3632e1a35a58d00"),
  "name" : {
    "first" : "John",
    "last" : "Doe" },
  "address" : [
    { "location" : "work",
      "address" : {
        "street" : "16 Hatfields",
        "city" : "London",
        "postal_code" : "SE1 8DJ"}},
    { "geo" : { "type" : "Point", "coord" : [
        51.5065752,-0.109081]}}],
  + {...}
],
"dob" : ISODate("1977-04-01T05:00:00Z"),
"retirement_fund" : NumberDecimal("1292815.75")
}
```

# Modeling Relationships:

- Nested Documents



## Nested Documents

```
{
  "_id" : ObjectId("5ad88534e3632e1a35a58d00"),
  "name" : {
    "first" : "John",
    "last" : "Doe"
  },
  "address" : [
    { "location" : "work",
      "address" : {
        "street" : "16 Hatfields",
        "city" : "London",
        "postal_code" : "SE1 8DJ"},
      "geo" : { "type" : "Point", "coord" : [
        51.5065752,-0.109081]}}},
    +   {...}
  ],
  "dob" : ISODate("1977-04-01T05:00:00Z"),
  "retirement_fund" : NumberDecimal("1292815.75")
}
```



## Nested Docs

# Modeling Relationships:

- Array of ancestors

### Array of Ancestors

```
{
  "_id" : ObjectId("5ad88534e3632e1a35a58d00"),
  "name" : "SMEG Toaster TSF01CREU white",
  "categories" : [
    "Toasters",
    "Breakfast",
    "Kitchen"
  ],
  "color" : "off white",
  "brand" : "SMEG"
}
```

# Modeling Relationships:

- Parent references

## Nested Documents

```
[
  { _id: "MongoDB", parent: "Databases" },
  { _id: "dbm", parent: "Databases" },
  { _id: "Databases", parent: "Programming" },
  { _id: "Languages", parent: "Programming" },
  { _id: "Programming", parent: "Books" },
  { _id: "Books", parent: null }
]
```

		Patterns												Related Examples
		Approximation	Attribute	Bucket	Computed	Document Versioning	Extended Reference	Outlier	Preallocated	Polymorphic	Schema Versioning	Subset	Tree	
Use Cases	Catalog	✓	✓		✓	✓	✓			✓	✓	✓	✓	Inventory Management
	Content Management		✓			✓				✓	✓	✓	✓	
	Internet Of Things	✓		✓	✓			✓	✓		✓			Log data, Time Series, Block Chain
	Mobile	✓			✓		✓	✓		✓	✓	✓		
	Personalization				✓	✓		✓			✓	✓		Point of sale, User Management
	Real-Time Analytics	✓		✓	✓		✓		✓		✓			Data Warehouse
	Single View		✓		✓	✓				✓	✓			

# Pattern in practice: Movies

---

```
{  
  title: "Star Wars",  
  director: "George Lucas",  
  ...  
  release_US: ISODate("1977-05-20T01:00:00+01:00"),  
  release_France: ISODate("1977-10-19T01:00:00+01:00"),  
  release_Italy: ISODate("1977-10-20T01:00:00+01:00"),  
  release_UK: ISODate("1977-12-27T01:00:00+01:00"),  
  ...  
}
```



```
{  
  title: "Star Wars",  
  director: "George Lucas",  
  ...  
  releases: [  
    {  
      location: "USA",  
      date: ISODate("1977-05-20T01:00:00+01:00")  
    },  
    {  
      location: "France",  
      date: ISODate("1977-10-19T01:00:00+01:00")  
    }  
  ]  
  ...  
}
```

```
{ "releases.location": 1, "releases.date": 1 }
```



# Attribute pattern

---

## Problem

Lots of similar fields

Want to search across many fields at once

Fields present in only a small subset of documents

## Solution

Break the field/value into a sub-document

```
{ "color": "blue", "size": "large" }
```

```
{ [ { "k": "color", "v": "blue" },
```

```
  { "k": "size", "v": "large" } ] }
```

## Use Case Examples

Characteristics of a product

Set of fields all having same value type

List of dates

## Benefits and Trade-offs

Easier to index

Allow for non-deterministic field names

Ability to qualify the relationship of the original field and value





# Data governance



## JSON Schema

Enforces strict schema structure over a complete collection for data governance & quality

- Builds on document validation introduced by restricting new content that can be added to a document
- Enforces presence, type, and values for document content, including nested array
- Simplifies application logic

## Tunable

Enforce document structure, log warnings, or allow complete schema flexibility

## Queryable

Identify all existing documents that do not comply



# Intuitive: client drivers



C



C++



C#



Go



Java



Node.js



Perl



PHP



Python



Ruby



Scala



Swift



Rust



R

- Common CRUD capabilities but idiomatic to each language
- Uniform HA & Failover capabilities across all



# Fully indexable

Fully featured secondary indexes—document optimized—extended beyond RDBMS experiences

## Index Types

### Primary Index

Every Collection has a primary key index

### Compound Index

Index against multiple keys in the document

### MultiKey Index

Index into arrays

### Wildcard Index

Auto-index all matching fields, sub-documents & arrays

### Text Indexes

Support for text searches. Atlas Search offers Lucene-based inverted indexes

### GeoSpatial Indexes

2d & 2dSphere indexes for spatial geometries

### Clustered Indexes

For time series collections, pre-sorted by timestamp for low latency queries

## Index Features

### TTL Indexes

Single Field indexes, when expired delete the document

### Unique Indexes

Ensures value is not duplicated

### Partial Indexes

Expression based indexes, allowing indexes on subsets of data

### Case Insensitive Indexes

Supports text search using case insensitive search

### Sparse Indexes

Only index documents which have the given field



# How to create an index

Create a standard index

```
db.collection.createIndex({"field":1})
```

//Query using that index

```
db.collection.find({"field": "foo"})
```

Create a compound index

```
db.collection.createIndex({"fieldA":1, "fieldB":1})
```

//Query using that index

```
db.collection.find({"fieldA": "foo", "fieldB": "bar"})
```

```
db.collection.find({"fieldA": "foo"})
```

//Query not using that index

```
db.collection.find({"fieldB": "bar"})
```

Create a unique index

```
db.collection.createIndex({"field":1, "unique": true})
```



# Aggregations

## Advanced data processing pipeline for transformations and analytics

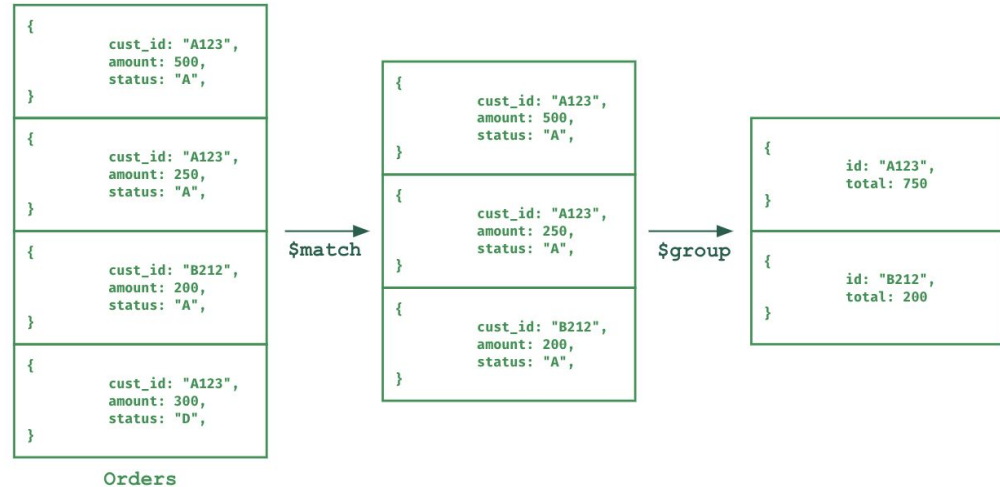
Multiple stages

Similar to a unix pipe

- Construct modular, composable processing pipelines

Rich Expressions

## Example Aggregation Command on the Orders Collection:



```
$match stage • db.orders.aggregate( [
                { $match: { status: "A" } },
                { $group: { _id: "$cust_id", total: { $sum: "$amount" } } }
                ] )
$group stage •
```



# Native Time-Series Platform

# Time Series Collection



## What is it?

“Hands-free” schema optimization:  
Organizes the data within a flexible time series schema optimized for high storage efficiency and low I/O

## What's the benefit?

Higher developer productivity:  
Eliminates trial and error in tuning for performance and storage efficiency

Lower cost, higher read performance:  
Reduces storage footprint, lower I/O improves scalability



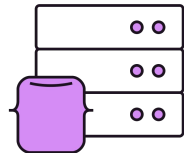
Run  
Anywhere



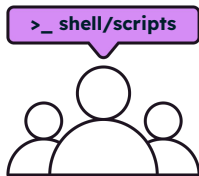
# MongoDB Community Edition



On-Prem



 MongoDB  
Community Edition



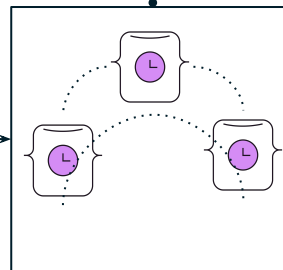
**No MongoDB Metrics / Alerts**  
(Oplog, Replication Window, Connections, ...)



**Backup**

**Mongodump**  
(very stressful to the operational workload)

**Automation**  
(Good to create a cluster, **difficult** for upgrades)



## Summary



Huge effort to create/maintain/update MongoDB Scripts



No MongoDB Metrics (e.g. if there is an issue, very hard to find the source)



No automatic backup, No Point-in-Time Restore, ...



**Recommended for exploration  
& poc development only**

# MongoDB provides a consistent developer experience everywhere



 MongoDB®  
Enterprise Advanced

On-premises

Private Cloud

Hybrid Cloud

Public Cloud

Managed Cloud



 MongoDB®  
Atlas



**Recommended for  
enterprise customers**



# Governance and security controls at every level

All MongoDB Atlas customer projects are deployed into their own VPC for network isolation. Private network peering is available for databases on all three clouds.

Business Trust Needs	Security Features	
Organization access	RBAC, multi-factor authentication, federated authentication, programmatic API keys	
Network isolation	Dedicated virtual private clouds for every project	
Network security	IP Access Lists, Peering connections, Private endpoints, configurable temporary access	
Database authentication	SCRAM, X.509, LDAPS, AWS IAM, configurable temporary access	
Database authorization	RBAC, read-only views, field-level redaction	
Database auditing	Admin, DML, DDL, DCL, role-based	
Database encryption	<b>In-Flight:</b> TLS 1.2+ <b>At-Rest:</b> Encrypted hardware, volume and database storage engine (AES-256)	
	<b>In-Use:</b> Client-Side Field Level Encryption <b>Key Management:</b> Cloud KMS	<b>Cloud agnostic:</b> Use AWS KMS, Azure Key Vault, or Google Cloud KMS regardless of underlying cloud provider





# Compliance





# Exam Time

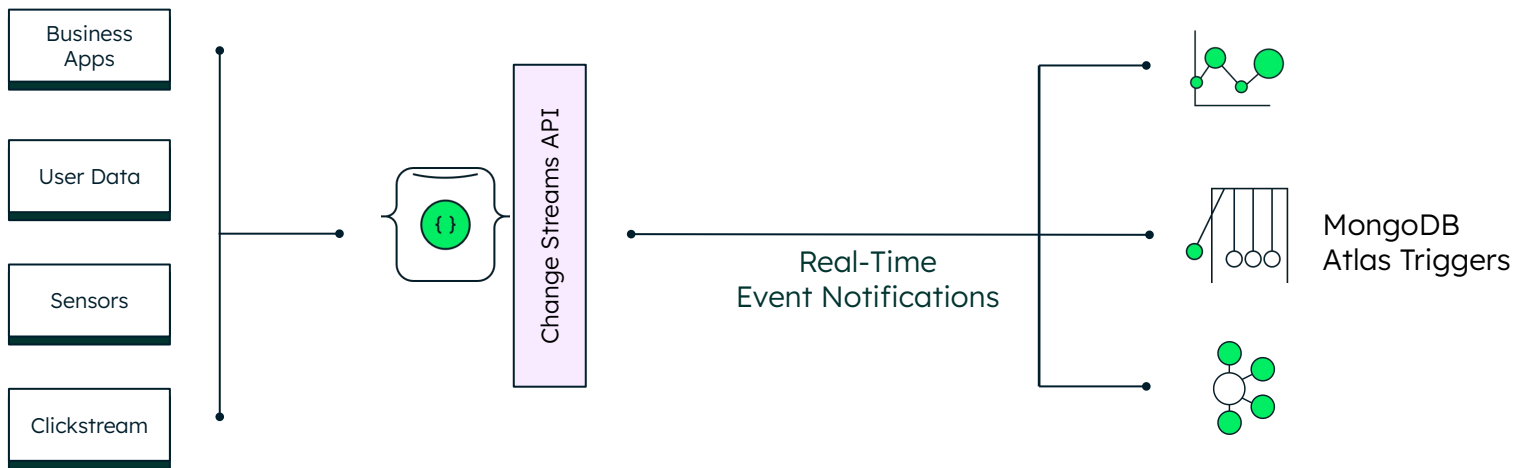
<https://learn.mongodb.com/pages/si-associate-certification-program>



**Thank You for your time**



# MongoDB change streams



Enabling developers to build  
reactive, real-time services

# Multi-cloud clusters on MongoDB Atlas

## Industry first and unique

Gives you unparalleled flexibility when it comes to where your data is stored and what cloud services you can use with MongoDB

- Take advantage of best-of-breed technology across multiple clouds
- Seamlessly migrate your cluster from one cloud to another
- Improve high availability with cross-cloud redundancy
- Reach more users by distributing your database across more regions



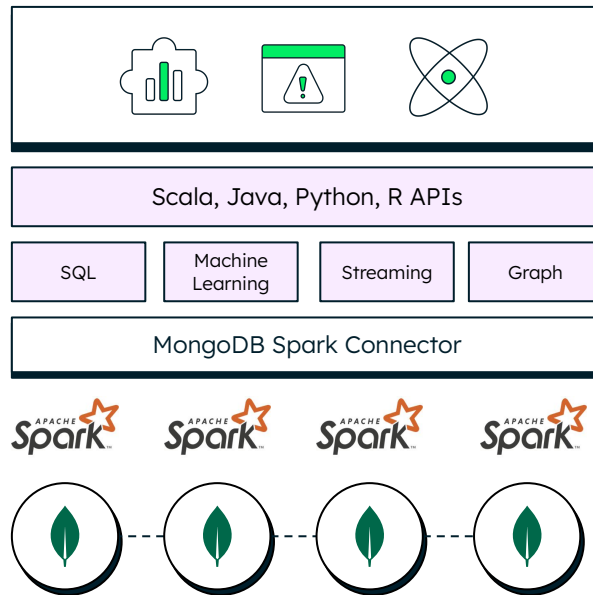
Multiple clouds simultaneously





# MongoDB Spark Connector

- Massively parallel processing, machine learning and streaming at scale
- Process data “in place” avoiding the latency of ETL
- Aggregation pre-filtering with secondary indexing to only select data that’s required
- Read from secondaries isolate analytics workload from business-critical operations
- Shard aware for data locality



**cloudera**

 **databricks**

 **Azure Databricks**