WELCOME

MongoDB Atlas

Developer Data Platform Workshop

David Hiltenbrand - Solutions Architect
Memphis, TN

Soheyl Rafi - Solutions Architect
Atlanta, GA

Ian Paeth - Enterprise Account Executive
Atlanta, GA

Agenda

- Introduction
- Delta + MongoDB Atlas
- MongoDB Fundamentals and the Developer Data Platform
- Q&A (Throughout)
- Break
- Hands-on Atlas Lab!

The evolution of MongoDB

Pre-2017 — 3.4

Doc-Level Concurrency
RAFT / Fast Failover
$lookup
Ops Manager
Compression
≤50 replicas
Aggregation ++
Encrypted and In-Memory storage engines
BI Connector
Compass
Views Graph Processing Zones ++Aggregation ++
Auto-balancing ++ Linearizable Reads Decimal Intra-cluster Compression Log Redaction Spark Connector ++ BI Connector ++

WiredTiger (Acquisition + Integration)

2017 — 3.6

Change Streams
Retryable Writes
Schema Validation Expressive
$lookUp
Query Expressivity
Causal Consistency Consistent
Sharded Secondary Reads
Query Advisor
End to End Compression
WiredTiger 1m+ Collections
MongoDB BI Connector ++
R Driver
Charts (post GA)
Atlas X-Region Replication Atlas Auto Storage Scaling

2018 — 4.0

Replica set transactions
Atlas global clusters
40% Faster Shard Migrations
Atlas HIPAA
Atlas LDAP
Atlas Audit
Atlas Enc. Storage Engine
Atlas Backup Snapshots
Type Conversions
Snapshot Reads
Non-Blocking Sec. Reads
SHA-2 & TLS 1.1+
Compass Agg Pipeline Builder
Compass Export to Code
Free Monitoring Cloud Service
Ops Manager K8s Beta

2019 — 4.2

Distributed Transactions
Client-Side Field Level Encryption
Materialized Views
Wildcard Indexes
Global PIT Reads
Large Transactions
Mutable Shard Key Values
Atlas Data Lake (Beta)
Atlas Auto Scaling (Beta)
Atlas Search (Beta)
Multi-CAs
Expressive Updates
Apache Kafka Connector
MongoDB Charts GA
Retryable Reads & Writes
New Index Builds
10x Faster stepDown
Storage Node Watchdog
Zstandard Compression

2020 — 4.4

Union
Custom Agg Expressions
Refinable Shard Keys
Compound Hashed Shard Keys
Mirrored Reads
Hedged Reads
Resumable Initial Sync
Time-Based Oplog Retention
Simultaneous Indexing
Hidden Indexes
Streaming Replication
Global Read/Write Concerns
Rust & Swift Drivers GA
TLS 1.3 & Faster Client Auth
OCSP Stapling
Kerberos Utility
Atlas Online Archive
Auto-Scaling
Schema Recommendations
AWS IAM Auth & Atlas x509
Federated Queries
Multi-cloud clusters

2021-2022 — 5.x

Time Series collections
Clustered indexes
Window functions
Live resharding
Client-Side FLE KMIP & cloud KMS
Atlas Serverless (preview)
Atlas Search fast facets, function scores, synonyms
Long running snapshot reads
Sharded $lookup & $graphlookup
Majority write concern default
4x faster initial sync
Schema validation diagnostics
New MongoDB Shell GA
Resumable index builds
Rewritten Swift driver
Rewritten C# LINQ provider and .NET Analyzer
PyMongoArrow API
New accumulator operators
Charts on Data Lake
x509 certificate rotation
Auditing ++
Atlas K8S controller
Ops Manager migration wizard

2022-2023 — 6.x

Queryable encryption (preview)
Atlas Serverless GA
Cluster-to-cluster sync
Independent analytic nodes scaling
Flexible sync GA
Asymmetric sync preview
Kotlin and Flutter SDKs for Realm
$lookup perf improvements
Atlas CLI GA
Atlas Data API GA
Secondary index on measurements for time series
Time series read perf +
Atlas Data Lake with fully managed storage
Atlas Data Federation
Atlas Charts dashboard embedding
Atlas SQL interface
Encrypted audit log
Change streams improvements
Relational Migrator (preview)

2023-2024 — 7.x

Vector Search (Preview)
Queryable Encryption GA
Stream Processing (Preview)
Relational Migrator GA
Compound Wildcard Index
Improved Query Execution
Approximate percentiles
Time series performance and scalability updates
Shard key analysis
updateOne w/out shard key
Shard chunk auto-merger
AWS, Azure, GCP ID management
Make X.509 Parameters configurable for Intra-node Auth and Client-to-node Auth
OpenSSL 3.0 support in Server
C2C sync updates
Live Migration supports 6.0+ migrations
PyMongo support

MongoDB Atlas

ACID (Transactions)

Multi-Cloud

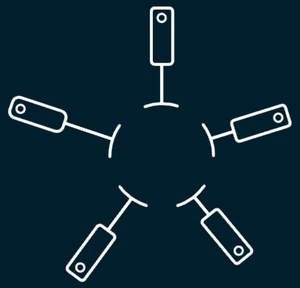
Stable API & Rapid Releases

Developer Data Platform

Why MongoDB

Cloud to Edge, Any Workload

Document Model and MongoDB Query API:
The fastest way to innovate



Multi-Cloud, Global Database:
Intelligently place data where you need it

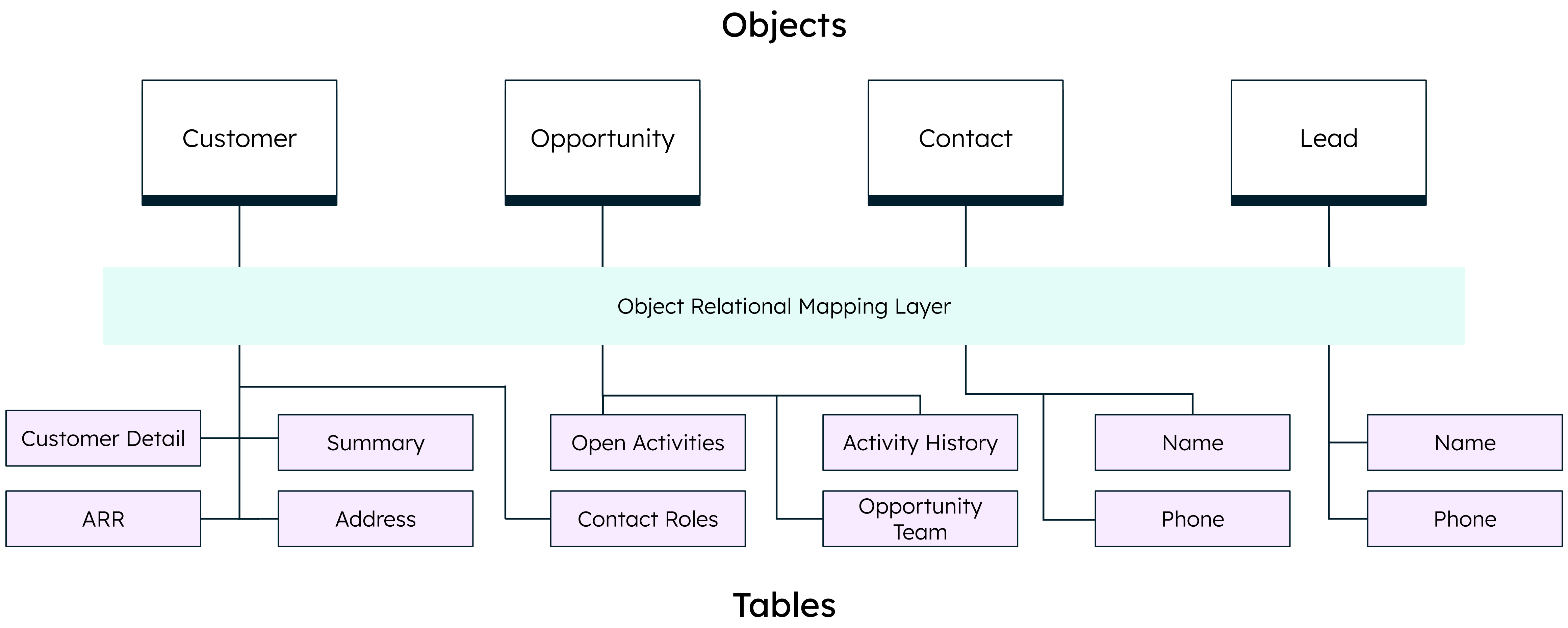
MongoDB Developer Data Platform:
Unified Experience and freedom to run anywhere

The Document Model and MongoDB Query API

The fastest way to innovate

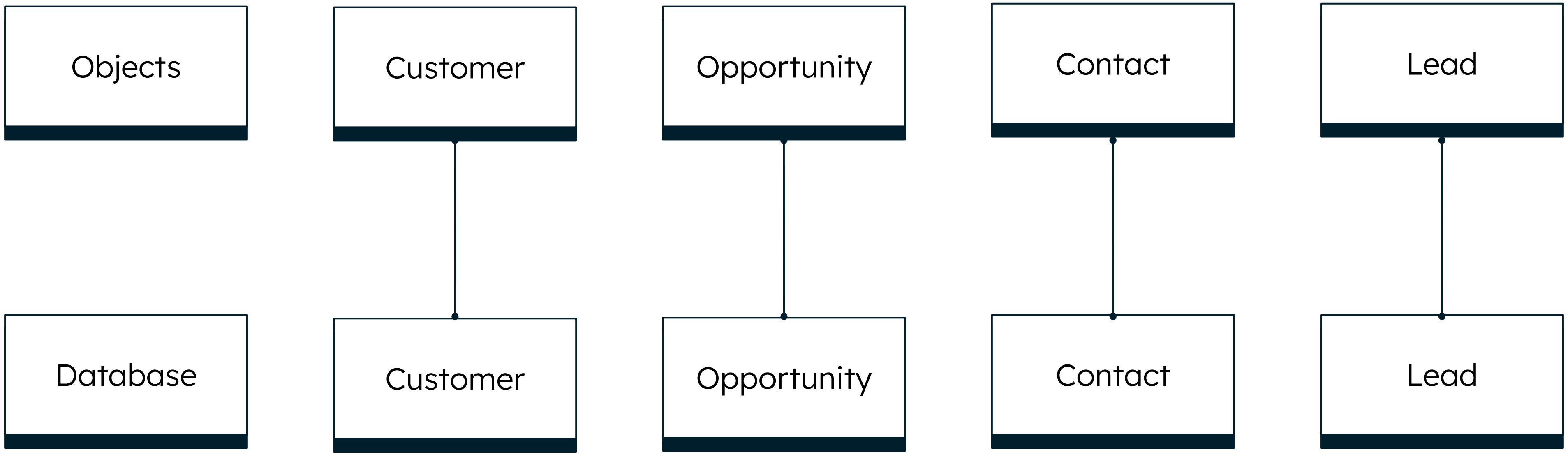
You probably have **thousands** of tables

Go from this...


Objects
Customer
Opportunity
Contact
Lead
Object Relational Mapping Layer
Customer Detail
Summary
Open Activities
Activity History
Name
Name
ARR
Address
Contact Roles
Opportunity Team
Phone
Phone
Tables

...to this

Store objects directly...

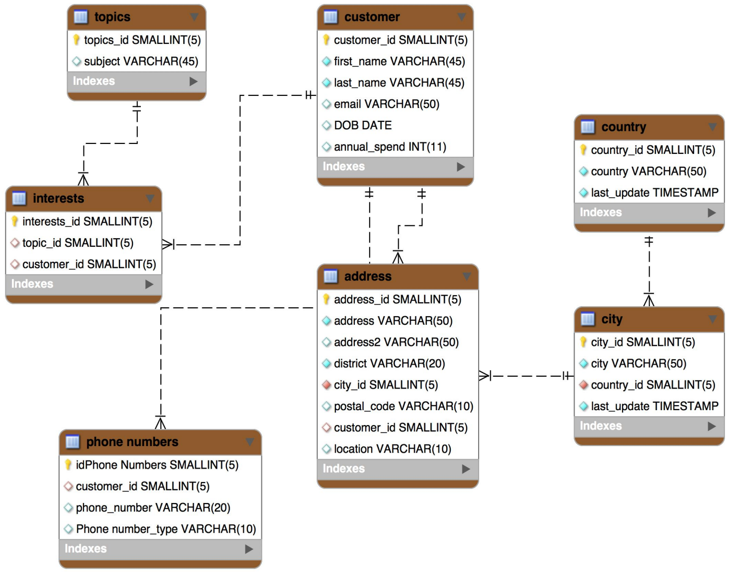


Some terminology

A comparison

RDBMS	MongoDB
Database	Database
Table	Collection
Parent—Child Tables	Nested Sub-Document or Array
Index	Index
Row	Document*
Column	Field
Join	Embedding, Linking, $lookup
View	Read-Only View On-Demand Materialized View
Multi-Record ACID Transaction	Multi-Document ACID Transaction

* Proper document schema design yields more entity data per document than found in a relational database row



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" : [
         -0.109081, 51.5065752]}},
+   {...}
  ],
  "dob" : ISODate("1977-04-01T05:00:00Z"),
  "retirement_fund" : NumberDecimal("1292815.75")
}
```

Naturally maps to objects in code

- Eliminates requirements to use ORMs
- Breaks down complex interdependencies between developer and DBAs teams

Represent data of any structure

- Polymorphic: each document can contain different fields
- Modify the schema at any time

Strongly typed for ease of processing

- Over 20 binary encoded JSON data types

Access by idiomatic drivers in all major programming languages

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

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" : [
         -0.109081, 51.5065752]}},
+   {...}
  ],
  "dob" : ISODate("1977-04-01T05:00:00Z"),
  "retirement_fund" : NumberDecimal("1292815.75")
}
```

Flexible: adapt to change

Add new fields dynamically at runtime

```
{
  "_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" : [
        -0.109081, 51.5065752]}},
+   {...}
  ],
     "dob" : ISODate("1977-04-01T05:00:00Z"),
     "retirement_fund" : NumberDecimal("1292815.75")

}
```

```
{
  "_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" : [
         -0.109081, 51.5065752]}},
+   {...}
  ],
  "phone" : [
    { "location" : "work",
      "number" : "+44-1234567890"},
+   {...}
  ],
  "dob" : ISODate("1977-04-01T05:00:00Z"),
  "retirement_fund" : NumberDecimal("1292815.75")
}
```

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

Model data any way you need

- JSON Documents
- Tabular
- Key-Value
- Text
- Geospatial
- Graph
- Time Series
- Events

Documents are universal

JSON documents are the modern standard in today's application stacks

Documents are universal

JSON documents are the modern standard in today's application stacks

Query data any way you need

Point
Range
Geospatial
Rich Search
Aggregations
JOINs & UNIONs
Graph Traversals

All wrapped in a single API, giving a consistent experience for any workload

MongoDB Query API: rich and expressive

Expressive queries	• Find anyone with phone # "1-212…" • Check if the person with number "555…" is on the "do not call" list
Geospatial	• Find the best offer for the customer at geo coordinates of 42nd St. and 6th Ave
Text search	• Find all tweets that mention the firm within the last 2 days
Aggregation	• Count and sort number of customers by city, compute min, max, and average spend
Native binary JSON support	• Add an additional phone number to Mark Smith's record without rewriting the document • Update just 2 phone numbers out of 10 • Sort on the modified date
JOIN ($lookup)	• Query for all San Francisco residences, lookup their transactions, and sum the amount by person
Graph queries ($graphLookup)	• Query for all people within 3 degrees of separation from Mark

MongoDB

```
{
    customer_id : 1,
    first_name : "Mark",
    last_name : "Smith",
    city : "San Francisco",
    phones: [
       {
          number : "1-212-777-1212",
          type : "work"
       },
       {
          number : "1-212-777-1213",
          type : "cell"
       }
    ]
}
```

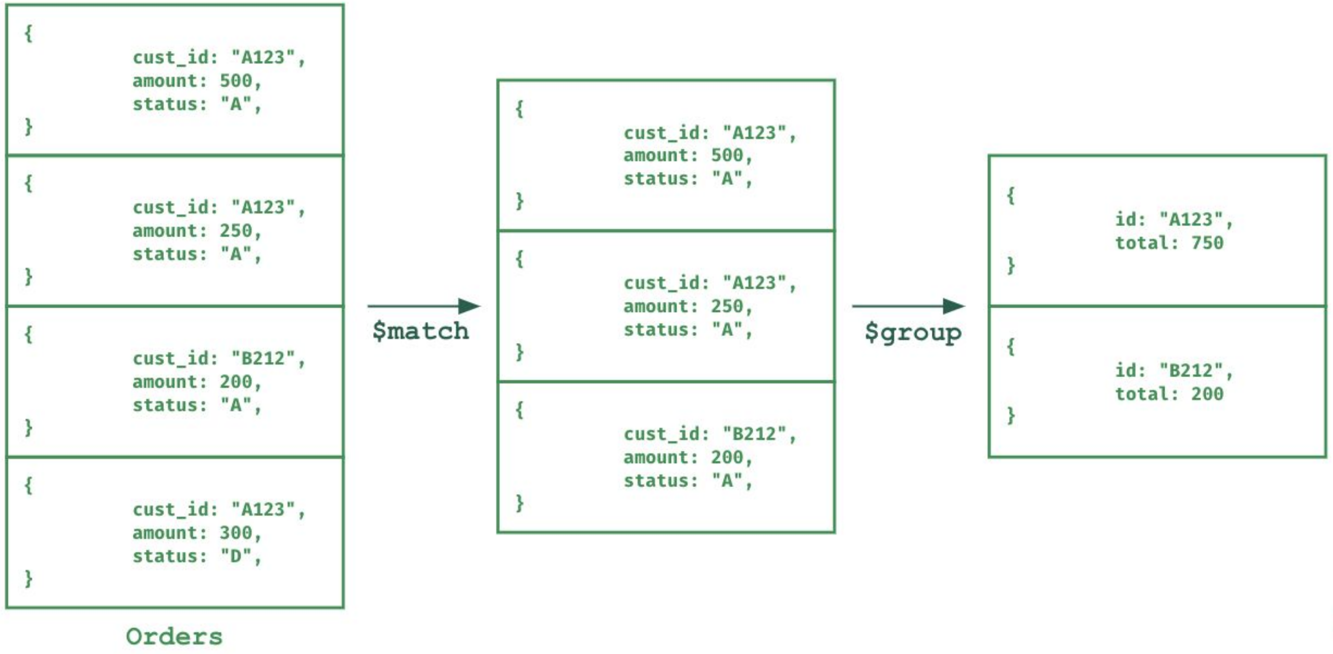
Aggregations

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
$group stage

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

Aggregation features

A feature rich framework for data transformation and Analytics

Pipeline Stages

- $match
- $group
- $facet
- $geoNear
- $graphLookup
- $lookup
- $merge
- $project
- $search
- $sort
- $setWindowFields
- $unionWith
- $unwind
- ...and more

Operators

- Mathematical

  $add, $abs, $subtract, $multiply, $divide, $log, $log10, $stdDevPop, $stdDevSam, $avg, $sqrt, $pow, $sum, $zip, $convert, $round, etc.

- Array

  $push, $reduce, $reverseArray, $addToSet, $arrayElemAt, $slice, etc.

- Conditionals

  $and, $or, $eq, $lt, $lte, $gt, $gte, $cmp, $cond, $switch, $in, etc.

- Temporal

  Window Functions

  $dateAdd, $dateDiff, $dateSubstract, $dateTrunc

  $dateFromParts, $dateToParts, $dateFromString, $dateToString, $dayOfMonth, $isoWeek, $minute, $month, $year, etc.

- String

  $toUpper, $toLower, $substr, $strcasecmp, $concat, $split, etc.

- Literals

  $exp, $let, $literal, $map, $type, etc.

- Regex

  $regexFnd, $regexMatch, etc

- Trigonometry

  $sin, $cos, $degreesToRadians, etc.

- Custom Aggregation Expressions

PRACTICAL MONGODB AGGREGATIONS EBOOK

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

Transactional guarantees

Transactional data guarantees

For many apps, single document transactions meet the majority of needs

```
  _id: 12345678
> name: Object
> address: Array
> phone: Array
  email: "john.doe.@mongodb.com"
  dob: 1966-07-30 01:00:00:000
˅ interests:Array
    0: "Cycling"
    1: "IoT"
```

Related data modeled in a single, rich document against which ACID guarantees are applied

ACID GUARANTEE

MongoDB multi-document ACID transactions

Multi-node transactional guarantees delivered at scale

- Multi-statement, familiar relational syntax
- Easy to add to any application
- Multiple documents in 1 or many collections and databases, across replica sets and sharded clusters

ACID guarantees

Snapshot isolation, all or nothing execution

Syntax (Python)

```
with client.start_session() as s:
    s.start_transaction()
    collection_one.insert_one(doc_one, session=s)
    collection_two.insert_one(doc_two, session=s)
    s.commit_transaction()
```

Natural for developers

- Idiomatic to the programming language
- Familiar to relational developers
- Simple

Syntax (Java)

```
try (ClientSession clientSession = client.startSession()) {
            clientSession.startTransaction();
            collection_one.insertOne(clientSession, docOne);
            collection_two.insertOne(clientSession, docTwo);
            clientSession.commitTransaction();
}
```

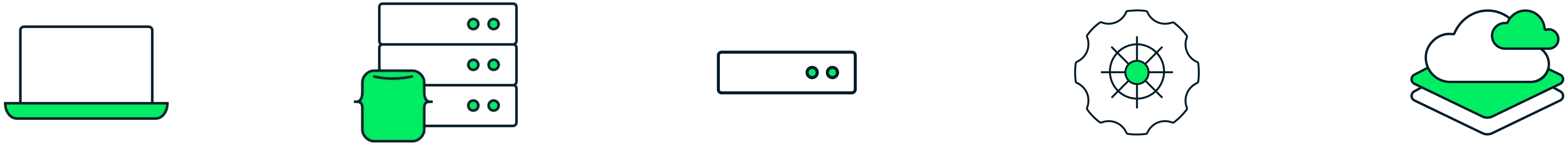
Natural for developers

- Idiomatic to the programming language
- Familiar to relational developers
- Simple

Multi-cloud global database

Freedom and flexibility

Broadest reach: private, hybrid, public



Laptop

Mainframe

Self-managed on-premises or in the cloud

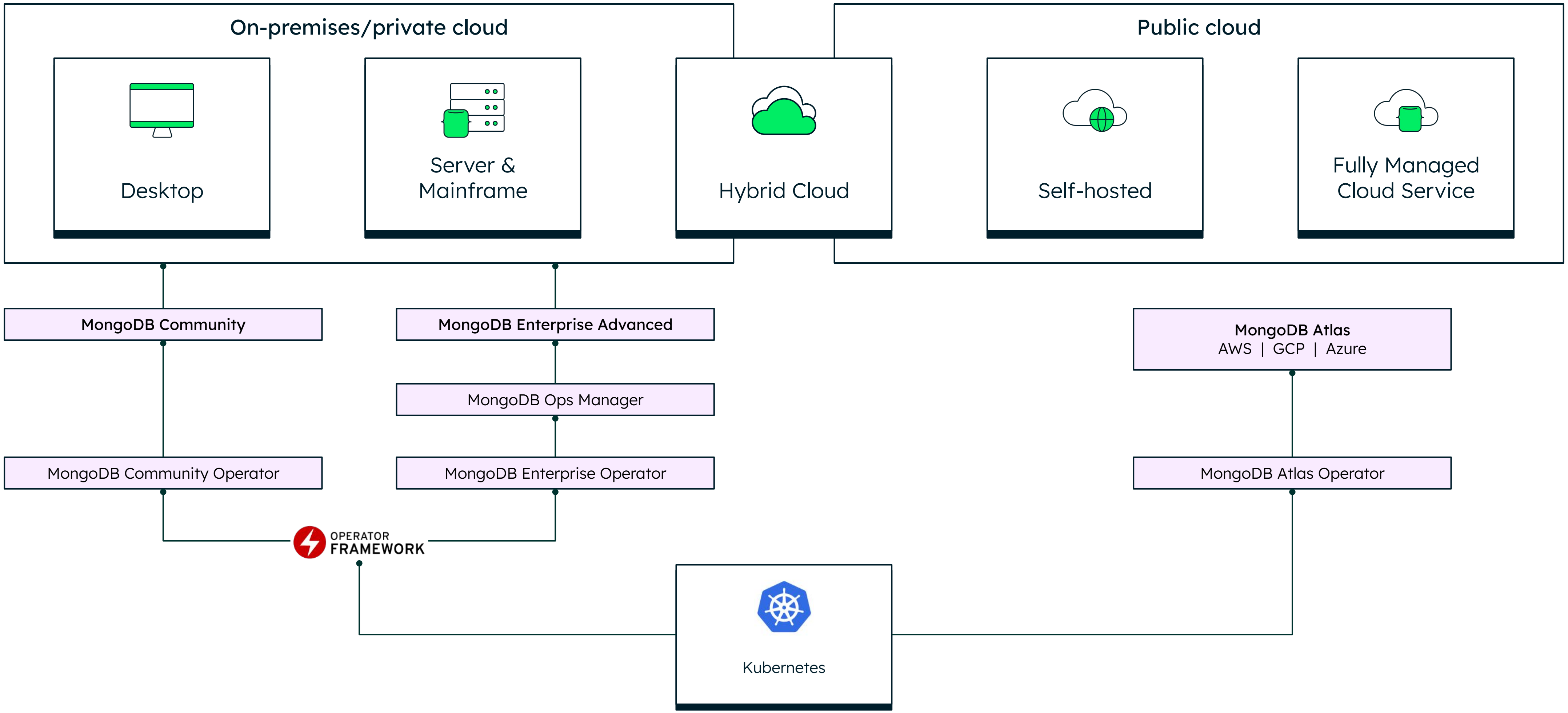
Managed in K8s and OpenShift

Database as a Service

Consistent developer experience

Same codebase, same APIs, same tools, wherever you run

Deployment on your terms



Challenges of scale and HA with tabular databases

Single Node, Vertically Scaled

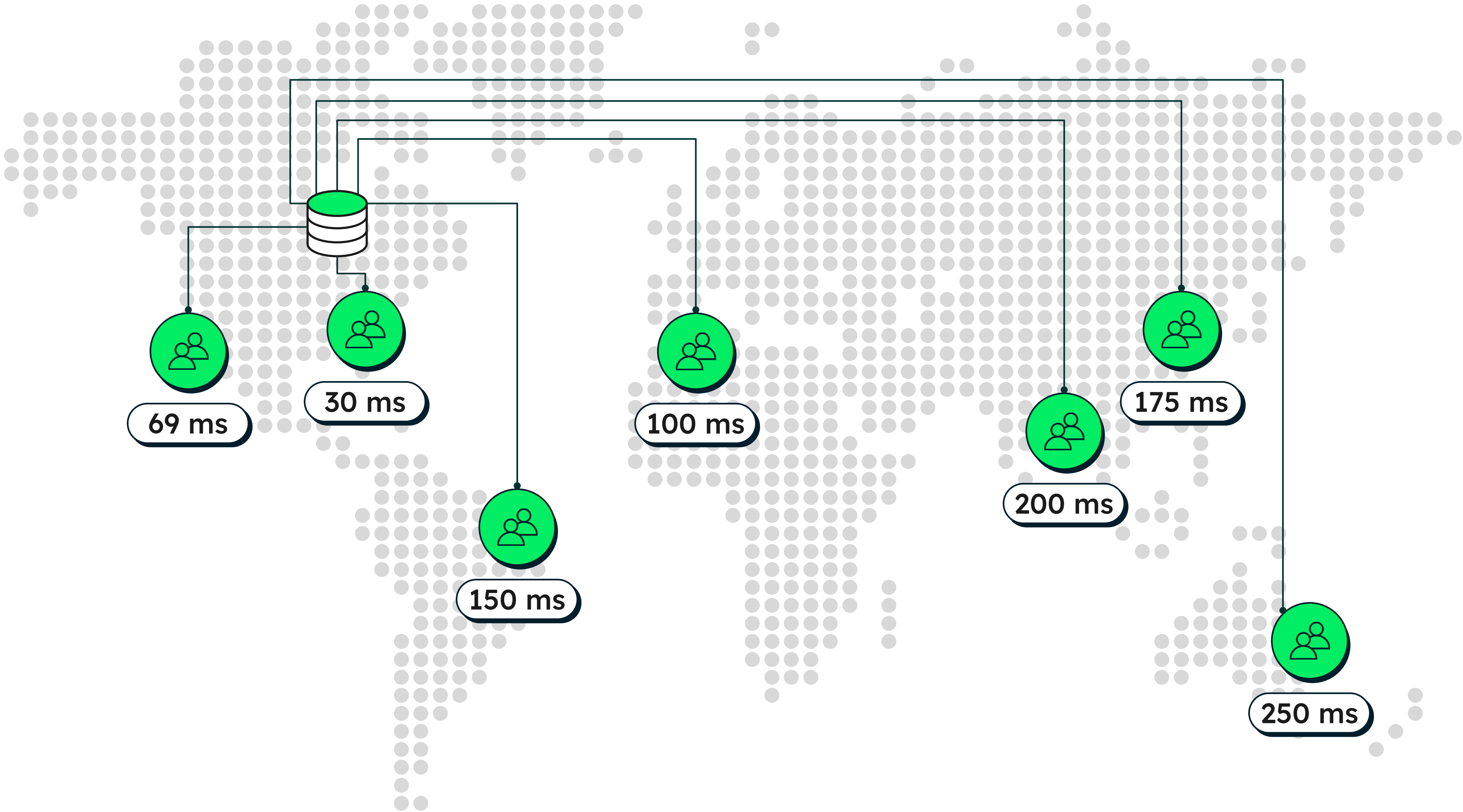
Scale-Out

- Developer pain: complex application-level sharding or external sharding frameworks
- Static: Limited elasticity
- Trade-Offs: Sacrifice key RDBMS functionality: cross-shard transactions, JOINs, referential integrity

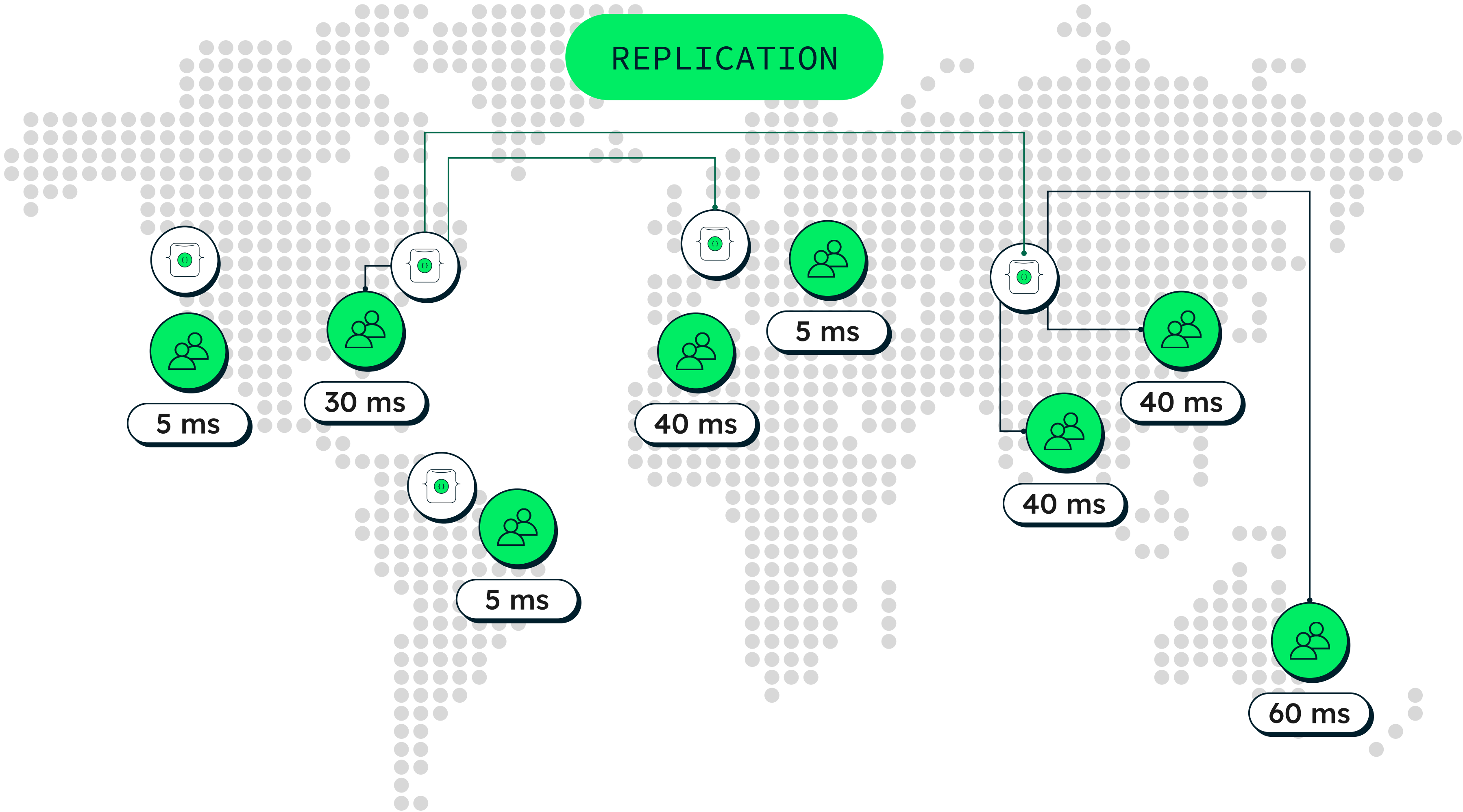
Availability

- Complex: requires integration of external cluster managers for failure detection and recovery
- Downtime: extended outages with multi-minute recovery times
- Developer pain: write complex exception-handling code to handle multi-minute node failovers

Serving global audiences with relational databases



Serving global audiences with MongoDB



MongoDB replica sets

Replica Set—2 to 50 copies

- Up to 7 voting nodes

Self-healing

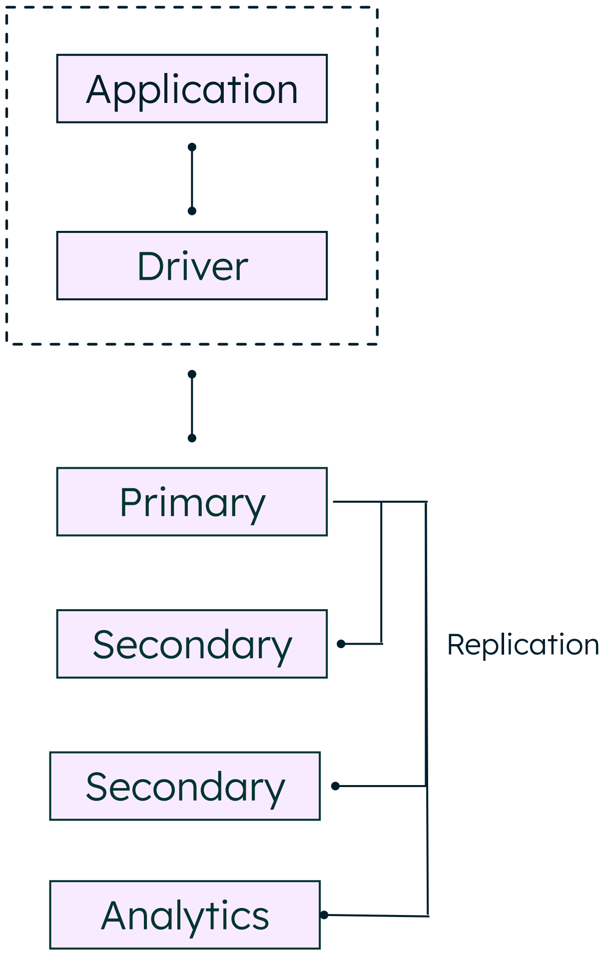
- Typical failover in 5 seconds or less
- Retryable reads and writes to catch temporary exceptions

Data center aware, tunable durability, and consistency

Addresses availability considerations:

- High Availability
- Disaster Recovery
- Maintenance

Workload Isolation: operational & analytics

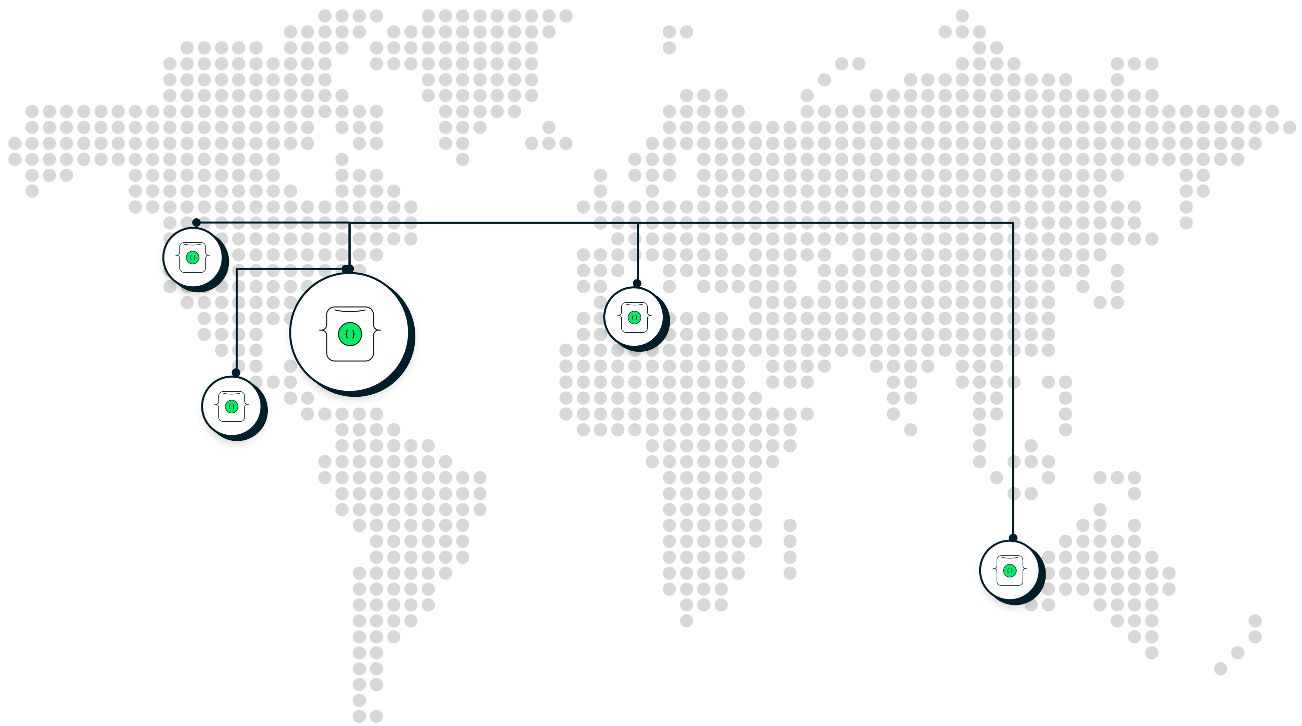


Cross-region replication

Ensure uptime in the unlikely event of a multi-zone or total region outage

Get improved performance for local reads by geographically distributing read-only replica set members that do not participate in the failover process

Available for Atlas deployments running on dedicated instances in Google Cloud, AWS, or Azure

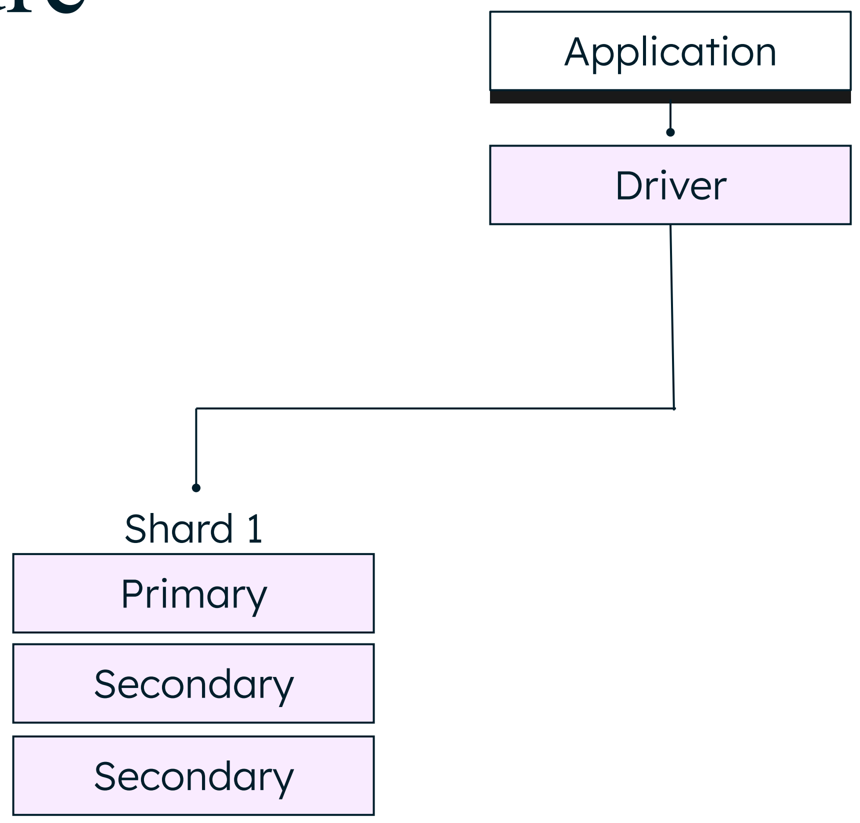


Cost-effective at any scale

Native-Sharding for horizontal scale-out

- Automatically scale beyond the constraints of a single node
- Application transparent
- Scale, refine, rebalance, and reshard data at any time
- Unlike NoSQL systems that randomly spray data across a cluster, MongoDB exposes multiple data distribution policies (hashed, ranged, zoned) to optimize for query patterns and locality

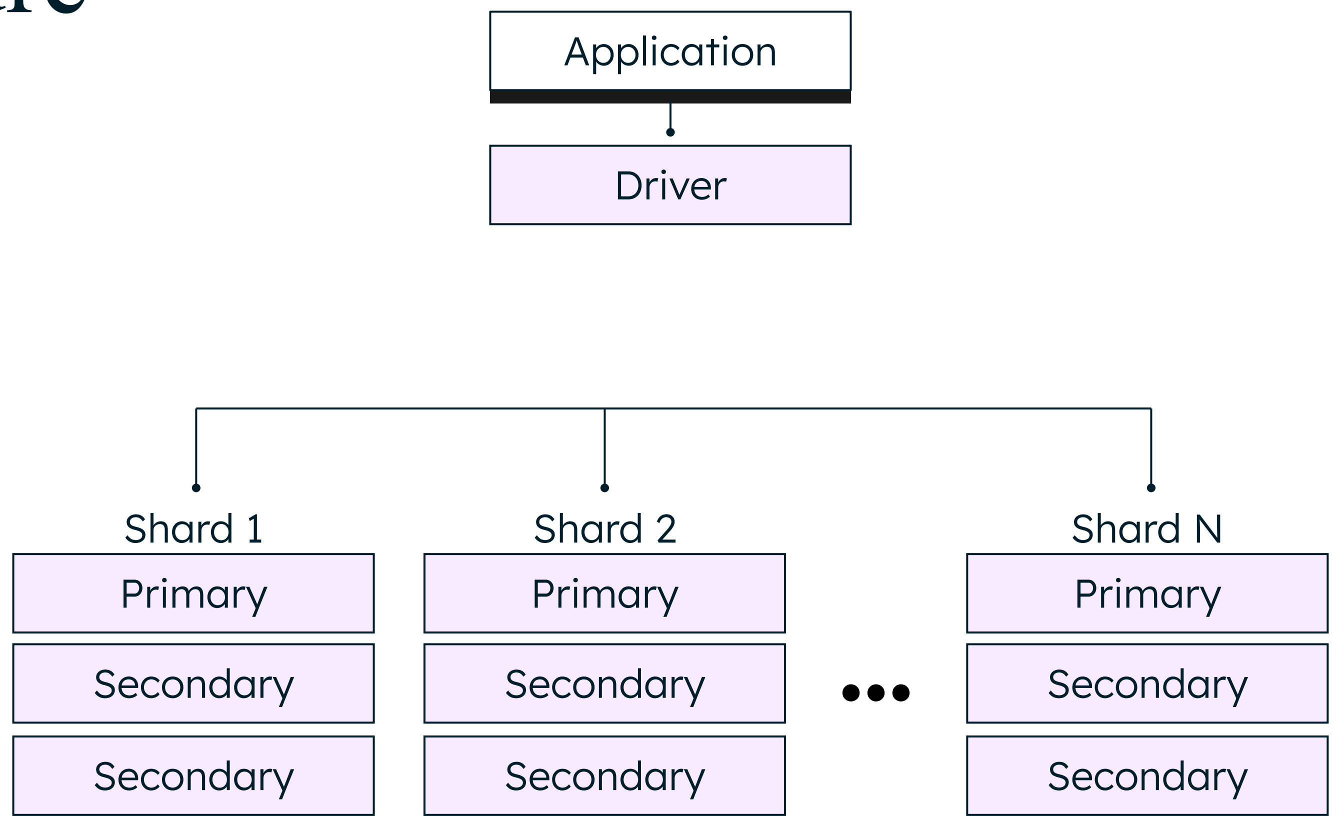
Sharding architecture



High availability

Replica sets

Sharding architecture



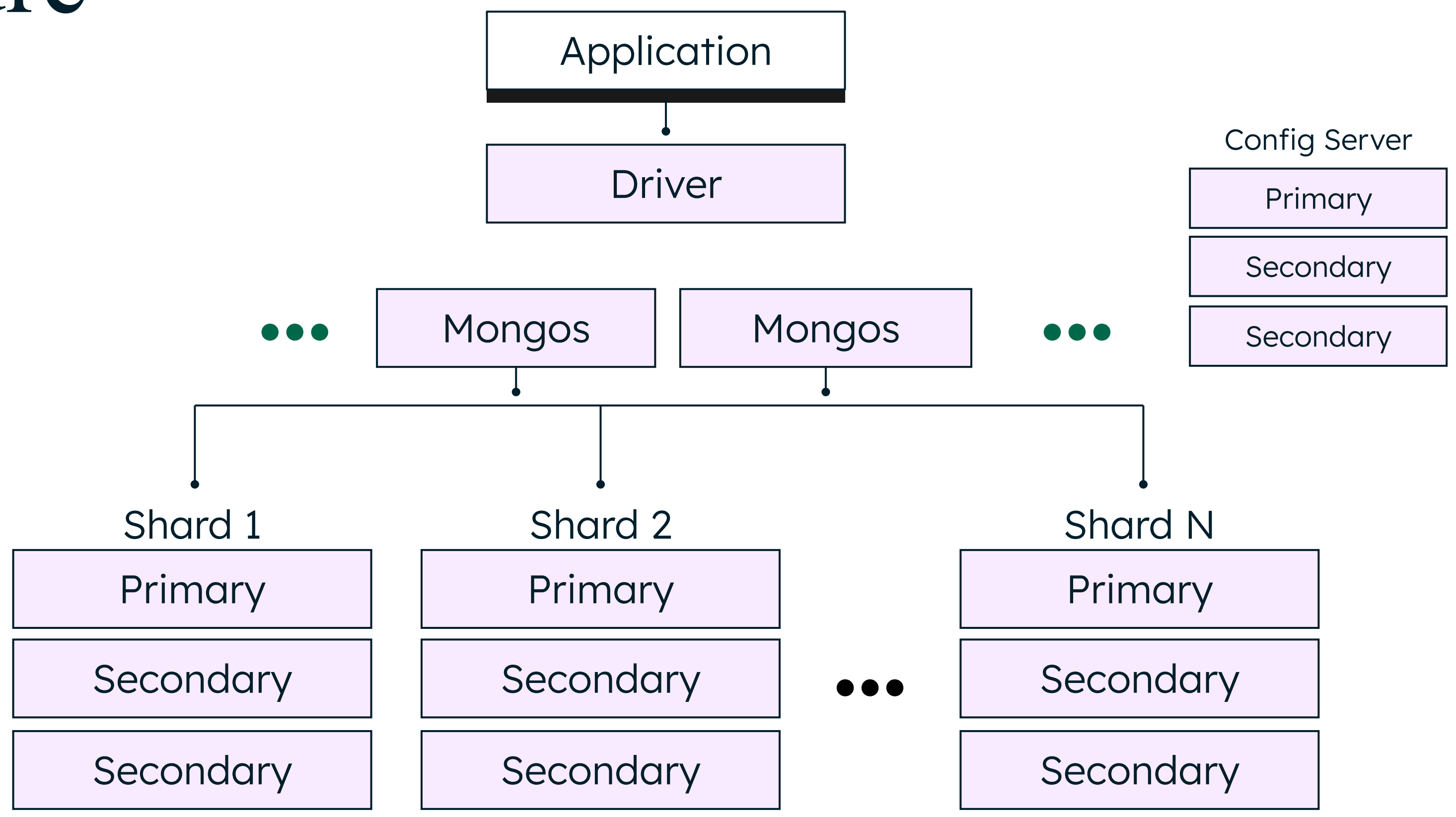
Horizontal scalability

Sharding

High availability

Replica sets

Sharding architecture



Horizontal scalability

Sharding

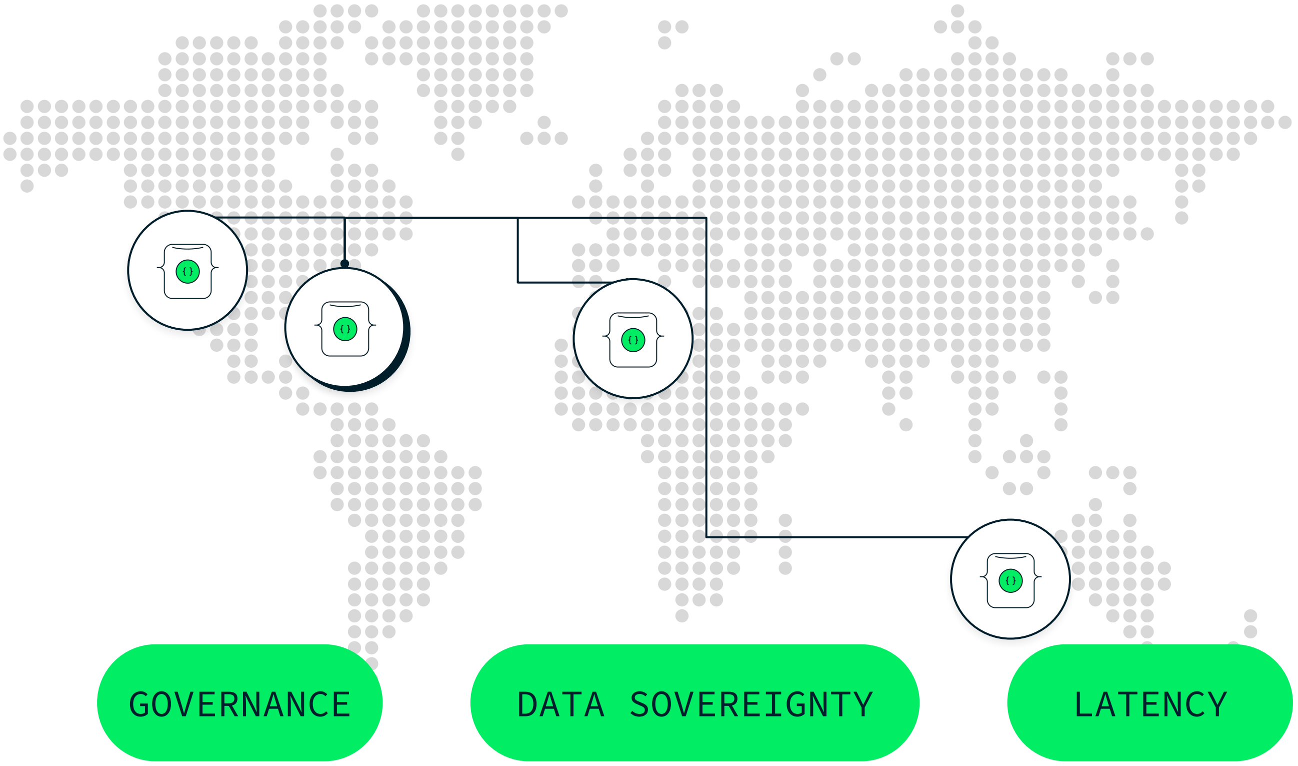
High availability

Replica sets

Global clusters for data residence

Intelligent Distribution via Zoned Sharding

- Policies to define data placement
- Name a server by region, tag your documents by region and MongoDB does the rest
- Documents automatically migrated as shard key ranges are modified
- Global queries across all data in all zones



Atlas Deployment Demo

Secure wherever you put your data

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
Authentication	SCRAM, X.509, LDAPS, AWS IAM, IP Access Lists
Authorization	RBAC, Read-Only Views, Field-Level Redaction
Auditing	Admin, DML, DDL, DCL, Role-based
Encryption	**In-Flight:** TLS 1.2+ **At-Rest:** Encrypted Hardware, Volume and Database Storage Engine (AES-256) **In-Use:** Client-Side Field Level Encryption, Queryable Encryption **Key Management:** Cloud Services (AWS KMS, Azure KV, Google Cloud KMS)

MongoDB Atlas
Developer Data Platform

Unified modern application experience

Atlas Database is fully managed MongoDB, available across all major public clouds

Fully managed database lifecycle; completely automated, elastic, and always-on

Featuring —

> 99.995% uptime SLA

> Secure defaults: TLS, encrypted storage, network isolation, and more

> Reduce costs with automated data tiering to queryable cloud object storage

> Fully managed backups with point-in-time recovery

Unmatched data distribution and mobility across 100+ regions on AWS, Azure, and Google Cloud

Featuring —

> Leverage the best of each cloud with multi-cloud clusters

> Cross-region, even cross-cloud failover

> Easily partition and distribute your data across regions for low-latency access, data residency guarantees with global clusters

Intelligent recommendations with built-in automation for resource and workload optimization

Featuring —

> Bi-directional auto-scale to match workload demand

> Intelligent recommendations for indexing and schema design

> Serverless deployment option now generally available

Integrated services that help developers build more seamlessly against their data in the cloud

Featuring —

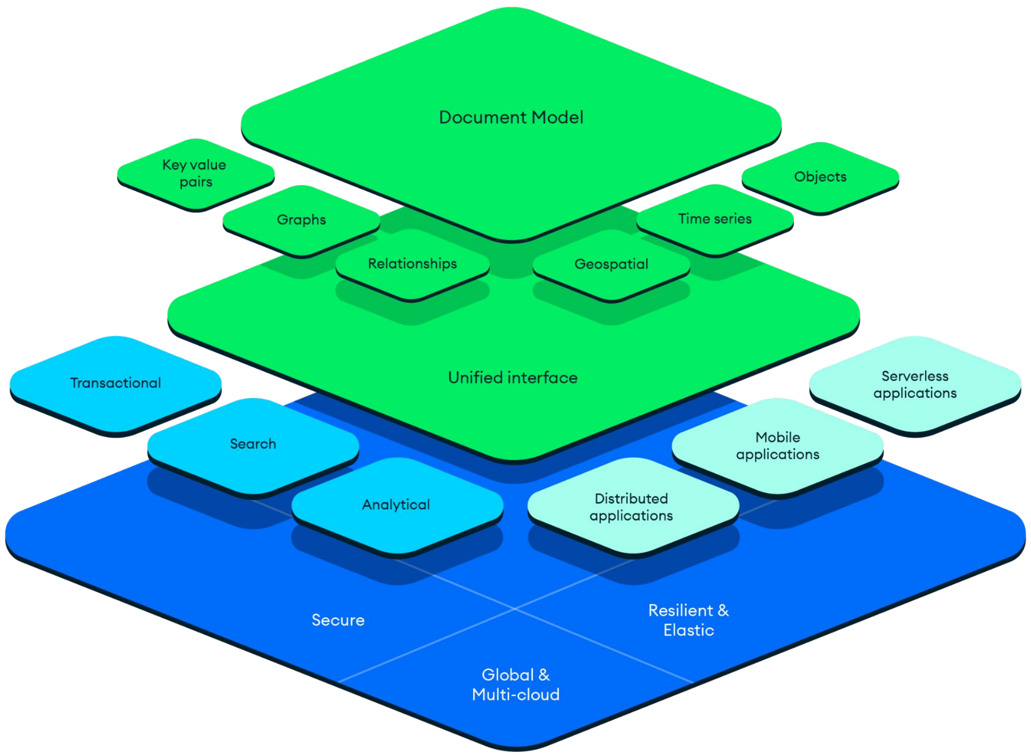
> GraphQL API that works with any standard GraphQL client

> Data API allows data read/write access over HTTPS

> Triggers and functions for easily building event-driven applications

> Easily define rules for user auth & data access permissions

MongoDB Atlas is a **developer data platform** that extends **beyond** fully managed MongoDB to address more of your data requirements for building modern applications



MongoDB Atlas is a developer data platform that supports a wide variety of data requirements

Atlas Search — Deliver better customer experiences with integrated full-text search capabilities

Featuring —

> Apache Lucene capabilities exposed via the unified MongoDB Query API: fuzzy search, synonyms, custom scoring, autocomplete, faceting, and more.

> Visual editor that enables users to build search indexes and queries in guided and easy-to-use interface

> Deliver rich search capabilities 30-50% faster

Atlas Device Sync — Expand possibilities on mobile with edge-to-cloud data synchronization

Featuring —

> Bi-directional data synchronization between embedded storage on devices (Realm) and Atlas Database

> Built in conflict resolution, networking code simplifies app architectures and the delivery of reactive mobile experiences

Atlas Data Federation — Federate queries across databases & between databases & cloud storage

Featuring —

> The ability to run a single query across multiple databases, S3, and HTTPS endpoints

> Analyze data stored as JSON, BSON, CSV, TSV, Avro, ORC, and Parquet without the complexity, cost of data transformation

> No infrastructure to setup or manage; pay only for the queries run

Atlas Charts — Easily create, share, and embed visualizations from data in MongoDB Atlas

Featuring —

> Zero setup or ETL required; easy-to-use drag & drop interface for building charts

> Data visualization built for rich document data model with support for nested objects & arrays

> Easily embed charts or dashboards using iFrames or embedding SDKs

MongoDB Atlas: Developer Data Platform

MongoDB Software

Cluster to cluster sync

Live migration

Self-Managed | On-Premise
Community or Enterprise
Advanced

- Flexible document data model
- Expressive Query API
- ACID transactions
- Time series collections
- Geospatial support
- Graph traversal
- Joins
- Change streams
- Client-side FLE | Queryable Encryption

Fully managed MongoDB

Atlas Database
Dedicated | Shared | Serverless

- Multi-Cloud
- Perf. Advisor
- Global clusters
- Online Archive

Search | Analytical

- Atlas Search
- Data Federation
- Analytics Nodes
- Charts
- Data Lake*
- SQL Interface

Workload Isolation

Application Services

- Data API
- Device Sync
- Realm
- GraphQL API
- User auth.
- Triggers & Functions
- Schemas & rules

Developer tooling: MongoDB Compass, VS Code Integration, MongoDB shell, idiomatic drivers in dozens of programming languages, CLI & Programmatic API for MongoDB Atlas

Integrations to broader application ecosystem: Kafka Connector, Spark Connector, Kubernetes Operator, Terraform Provider, and many more.

* in preview

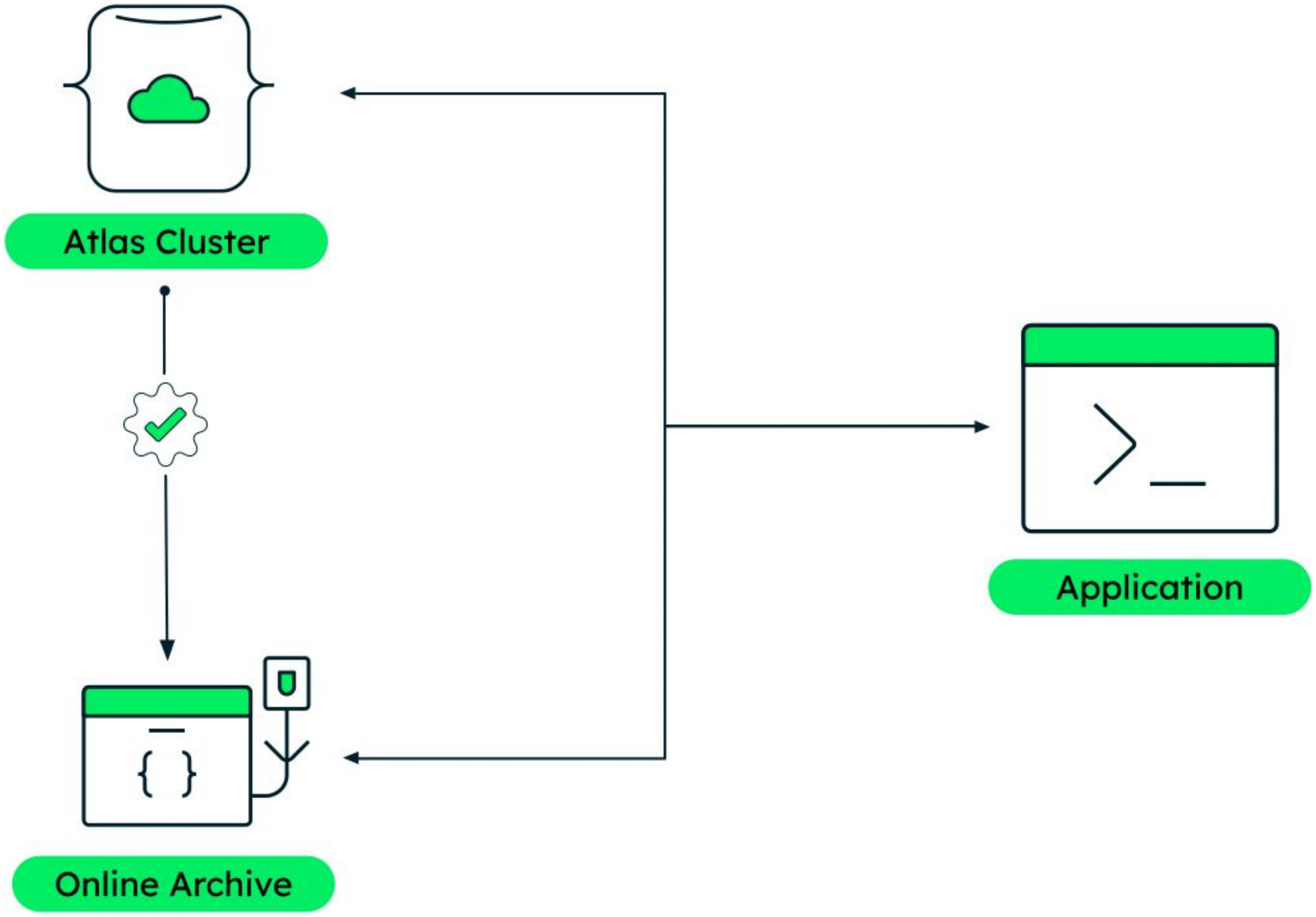
Continuing to invest in a cloud-first, data platform for building applications

	2018	2019	2020	2021	2022	2023
MongoDB version	4.0	4.2	4.4	5.x	6.x	7.x
	Multi-doc transactions Type conversions Non-block sec. reads	**Distributed transactions** **Client-side encryption** Global PIT reads Materialized views Wildcard indexes	**Refinable shard keys** **Hedged and mirrored reads** Union aggregation stage Custom agg expressions Compound hash shard keys	**Native time series support** **Live resharding** **Versioned API** Multi-Cloud FLE Majority write concern default	**Queryable encryp. preview** **Cluster-to-cluster sync** **Column store indexes** Time series ++ Change streams ++	**Queryable encryp. GA** **Improved query execution** **Filtered C2C sync** Change stream event size + Time series deletes
Cloud/tools releases	CONTINUOUS DELIVERY					
	mLab acquisition **Global clusters** Rich database auditing LDAP integration BYO encryption keys MongoDB Charts	**Realm acquisition** **Atlas Search** **Atlas Data Lake** Compute auto-scaling Analytics nodes Snapshot backups Query profiler AWS PrivateLink support	**Multi-cloud clusters** **Online Archive** Azure Private Link Multiple connection strings Cross-org billing Schema advice AWS IAM authentication GraphQL API	**Edge-to-cloud sync** **Serverless preview** Custom archiving rules Multi-cloud enhancements Kubernetes operator Search query builder UI	**Serverless GA** **Data Lake Storage** **Atlas CLI** Atlas Data Federation Flexible sync GA Data API GA Ind. analytics nodes scaling Atlas Search ++	**Vector Search preview** **Stream Processing preview** **Relational Migrator GA** Atlas SQL Interface GA Search nodes preview Atlas Live Migration ++ New drivers & integrations Search query analytics

Atlas Online Archive

- Automate data tiering in MongoDB Atlas to reduce storage costs while preserving easy access to historical data
- Set date-based or custom archival rules to automatically offload aged data to fully managed object storage for a more cost-effective data storage strategy
- Run federated queries from a unified endpoint to access data in an Atlas Cluster and its Online Archives on-demand

Archive data to fully managed, queryable storage



Why use Atlas Online Archive?

Automated Data Tiering

Automatically archive aged data to cheaper storage to balance cost and performance

Eliminate the need to manually migrate or delete data

Queryable Archives

Preserve the ability to query all your MongoDB data—live and historical

Query Atlas clusters and online archives together with a single connection string

Fully Managed

No need to create & configure separate cloud object storage

Create, update and pause archival rules using the Atlas UI or API

Summary

MongoDB uniquely delivers...

Fastest way to innovate	Freedom and flexibility	Unified experience for modern applications

Thank you!

Next Steps

- Technical Deep Dives
- MongoDB University
- MongoDB .local ATL