

MongoDB Introduction & What You Need to Know about NoSQL Databases

Kim Greene

kim@kimgreene.com

507-216-5632

Skype/Twitter: iSeriesDomino

Copyright Kim Greene Consulting, Inc. All rights reserved worldwide.

1

KIM GREENE
CONSULTING, INC.

Kim Greene - Introduction

- Owner of an IT consulting company
 - Kim Greene Consulting, Inc.
 - www.kimgreene.com
- Started my career at IBM, left and launched my own business ... 18 years ago
- Focus areas:
 - IBM collaboration software portfolio
 - MongoDB
- Customers are worldwide and in multiple industries
- Blog: www.dominodiva.com
- Twitter: [iSeriesDomino](#)



Copyright Kim Greene Consulting, Inc. All rights reserved worldwide.

2

Agenda

- NoSQL databases
- What is MongoDB
- MongoDB basics
 - Components of MongoDB
 - Flavors of MongoDB
 - High availability
 - Scalability
 - Security
 - Interacting with MongoDB
 - Developer friendly
 - Indexes and search

Agenda

- How MongoDB compares to RDBMS
- Why and how customers are using MongoDB
- Where to find more information

NoSQL Databases

Copyright Kim Greene Consulting, Inc. All rights reserved worldwide.

5

KIM GREENE
CONSULTING, INC.

NoSQL

- A database system that is:
 - Non-relational
 - Distributed
 - Open-source
 - Horizontally scalable
- NoSQL as ...
 - “SQL”
 - Not Only SQL
 - Can allow SQL-like query languages to be used

Copyright Kim Greene Consulting, Inc. All rights reserved worldwide.

6

NoSQL

- Provide mechanism for storage and retrieval of data that has less constrained consistency requirements
- Typically used in big data and real-time web applications
- “May” support SQL-like query languages

NoSQL Advantages

- Scalability
- Schema flexibility
 - Sparse and semi-structured data
- Lower cost

NoSQL Disadvantages

- Less robust query capabilities
- Eventual consistency
- Lack of standardization
- Inadequate access control concerns

NoSQL Use Cases

- Massive data volumes
- Excessive query loads
- Changing schema design
- RDBMS distributed or scalability issues

Why NoSQL?

- Velocity and nature of data used/generated over Web is growing exponentially
 - Ex: social media, data has no specific structure boundary
- Unstructured data is a challenge for RDBMS
 - Overhead of joins and maintaining relationships doesn't bode well for fast CRUD operations

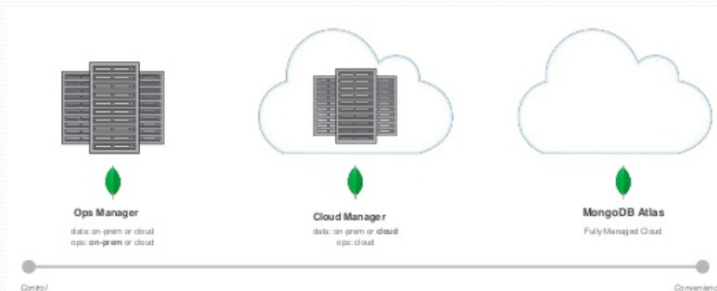
What is MongoDB?

What is MongoDB?

- MongoDB is a free and open-source cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schemas.
- Source: Wikipedia

More About MongoDB

- Open source
 - Source and packages available at [mongodb.com/download](https://www.mongodb.com/download)
 - Affero GPL license
- MongoDB deployment
 - Designed to be deployed on-premises or in the cloud



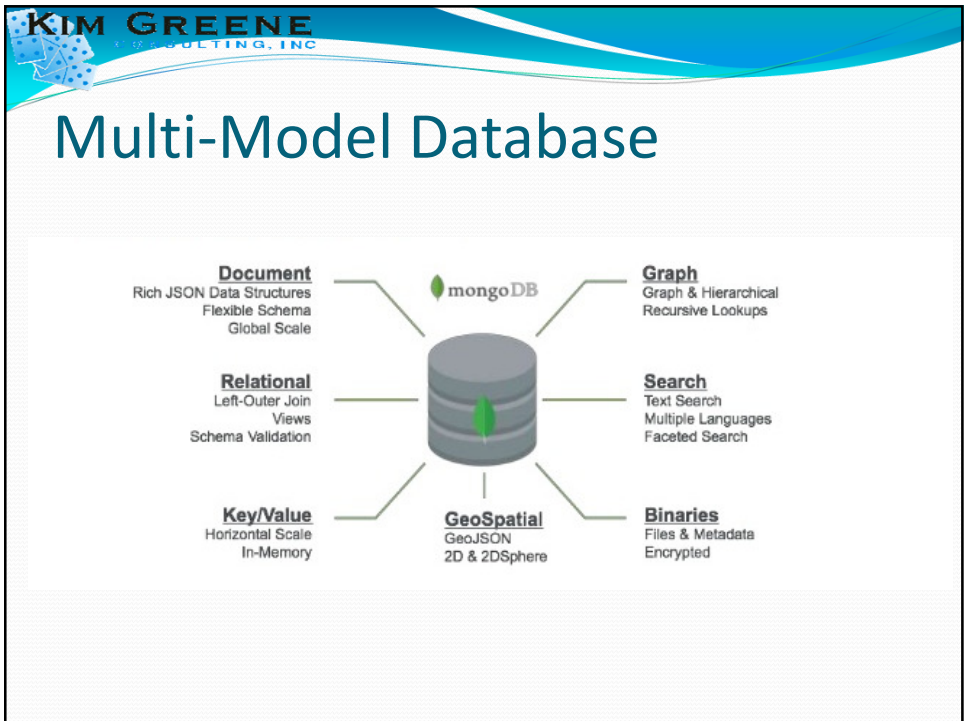
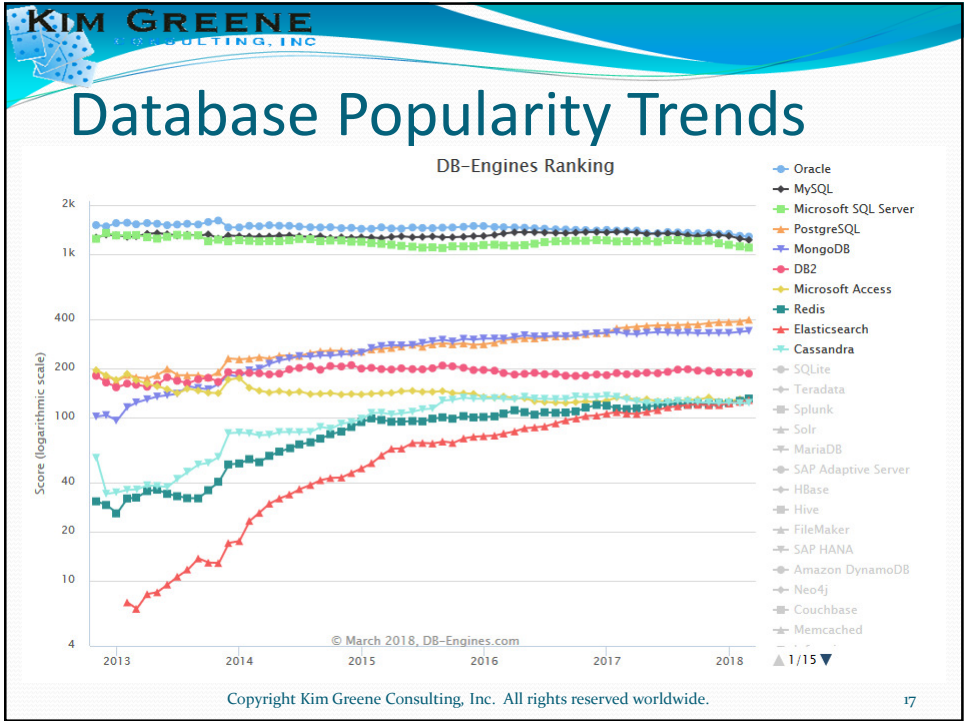
NoSQL

- Standard NoSQL
 - Non SQL
 - Non relational
- MongoDB
 - Not only SQL
 - Maintains foundation of relational
- Most commonly used in these scenarios
 - Big data
 - Real-time web applications
- More flexible than relational databases

5th Most Popular Database

Rank	DBMS	Model	Score
1	Oracle	Relational DBMS	1290
2	MySQL	Relational DBMS	1229
3	Microsoft SQL	Relational DBMS	1105
4	PostgreSQL	Relational DBMS	399
5	MongoDB	Document store	341
6	DB2	Relational DBMS	187
7	Microsoft Access	Relational DBMS	132
8	Redis	Key-value store	131
9	Elasticsearch	Search engine	129
10	Cassandra	Wide column store	124

- Source: DB-engines database popularity rankings: March 2018



4 Types of Databases

- WiredTiger
 - Most commonly used database type, the default
- Encrypted
 - For highly sensitive data
- In-memory
 - For performance critical data
- MMAPv1
 - Original database type, kept for compatibility reasons

Document Oriented

- Store all information for an object in a single instance
 - No spanning tables for all information
 - No more normalizing data
- Each stored object can be different
 - Not all documents need to contain the same data
- Handles semi-structured, unstructured, and polymorphic data

Document Oriented

DOCUMENTS ARE RICH DATA STRUCTURES



Document Oriented

- Documents can support polymorphic data

```
{
  "user": "Anna",
  "email" : "anna@gmail.com"
}

{
  "user": "Jon",
  "email" : [
    "jon@gmail.com",
    "jon@yahoo.com" ]
}
```

JSON-Like

- BSON
 - Serialization of JSON data in a quick to move format
 - Ensures can replicate quickly and efficiently

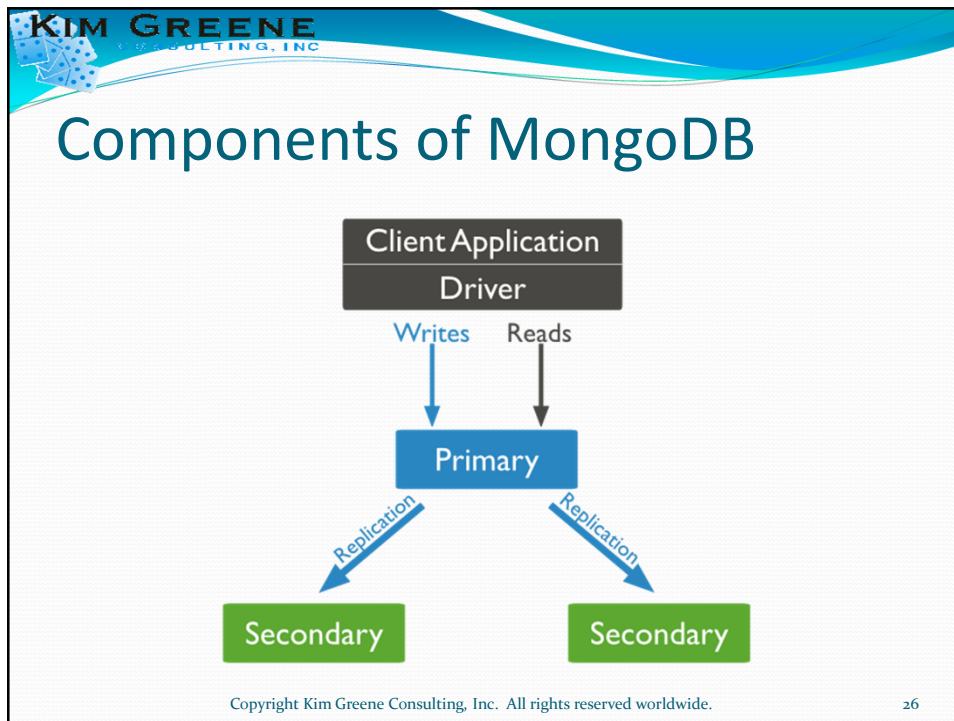
BSON {
01010100
11101011
10101110
01010101
}

MongoDB Basics

MongoDB Basics: Components of MongoDB

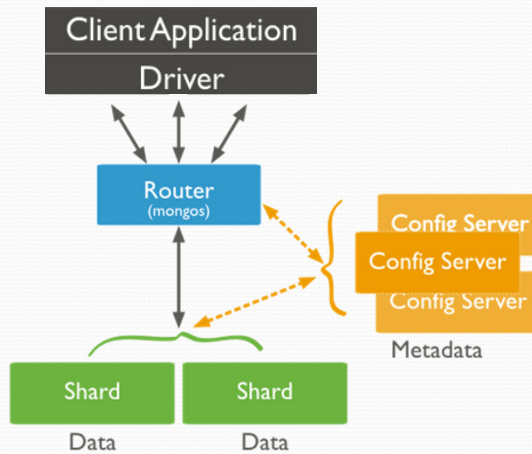
Copyright Kim Greene Consulting, Inc. All rights reserved worldwide.

25



26

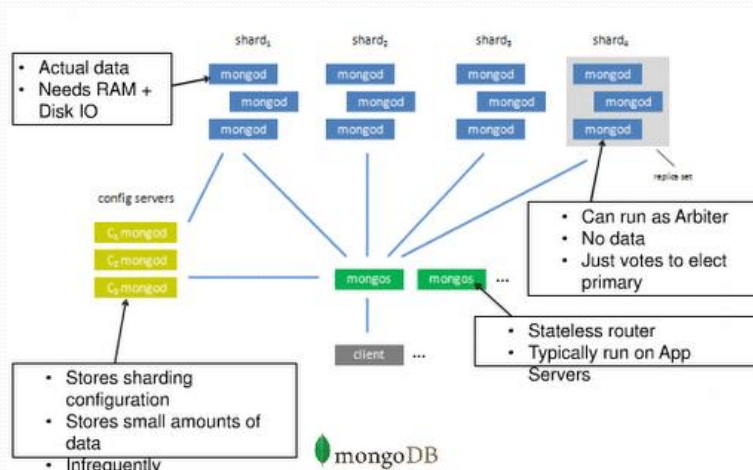
Components of MongoDB



Copyright Kim Greene Consulting, Inc. All rights reserved worldwide.

27

Components of MongoDB



Copyright Kim Greene Consulting, Inc. All rights reserved worldwide.

28

MongoDB Basics: Flavors of MongoDB

Copyright Kim Greene Consulting, Inc. All rights reserved worldwide.

29

KIM GREENE
CONSULTING, INC.

Flavors of MongoDB

- Community Edition
 - Available on Linux, OSX, Windows
- Enterprise Advanced
 - Provides more storage engine options:
 - In-memory
 - Encrypted
 - Advanced security features:
 - LDAP and Kerberos for authentication
 - Auditing capabilities
 - Available on Linux, Windows
- Atlas
 - Database as a service

Copyright Kim Greene Consulting, Inc. All rights reserved worldwide.

30

MongoDB Basics: High Availability

Copyright Kim Greene Consulting, Inc. All rights reserved worldwide.

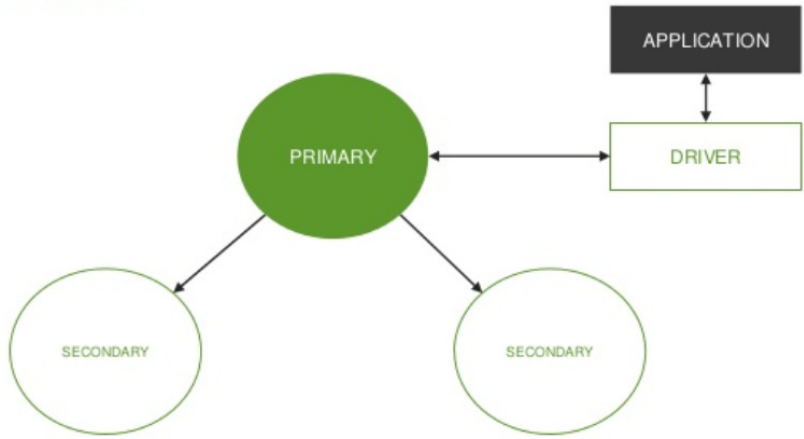
31

KIM GREENE
CONSULTING, INC.

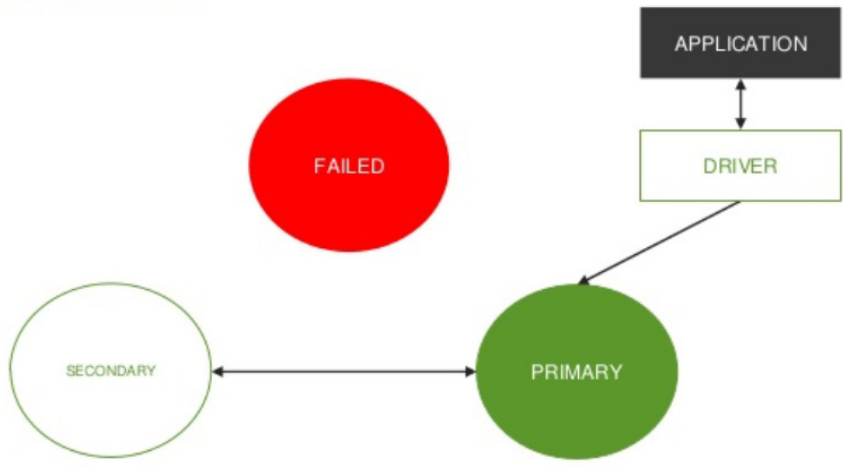
Replica Sets

- Group of MongoDB processes that maintain the same set of data
 - 3 replica sets is the standard for MongoDB
- Provide high availability and redundancy
 - Failover is fully automated, no administrator intervention required
 - Self-healing shard
- Optimize read operations

Replica Sets



Election of New Primary



MongoDB Basics: Scalability

Copyright Kim Greene Consulting, Inc. All rights reserved worldwide.

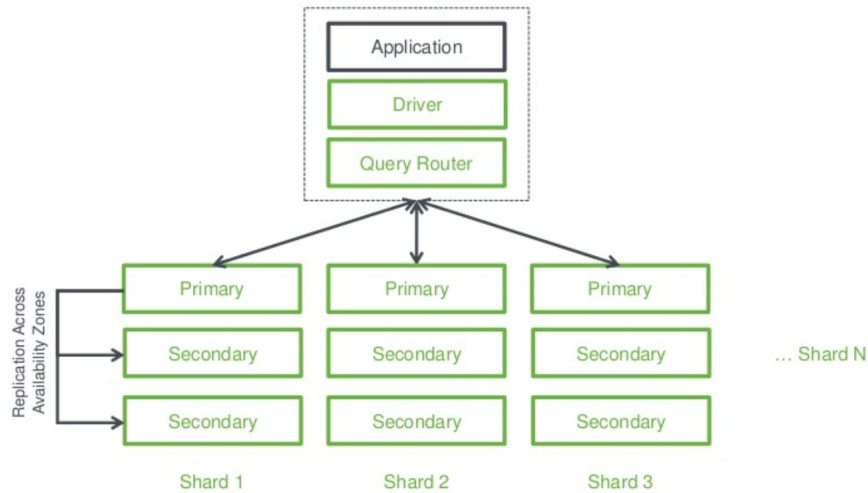
35

KIM GREENE
CONSULTING, INC.

Sharding

- Place a portion of data on certain servers
- Use with
 - Very large data sets
 - High throughput demands
 - Needs for geo location of data

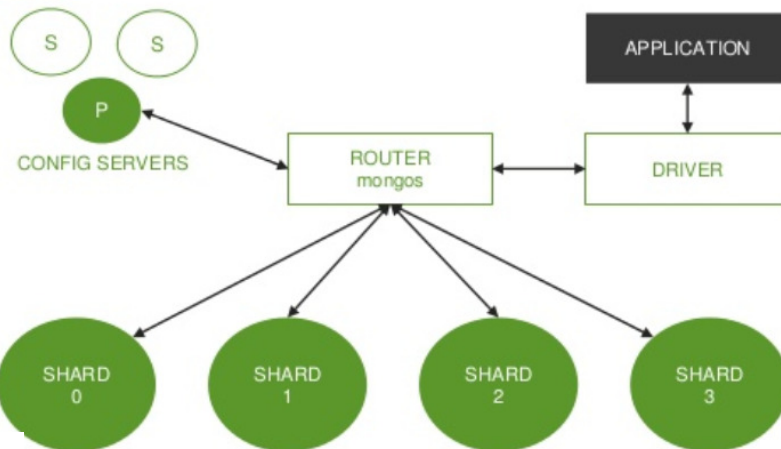
High Availability & Scalability



Copyright Kim Greene Consulting, Inc. All rights reserved worldwide.

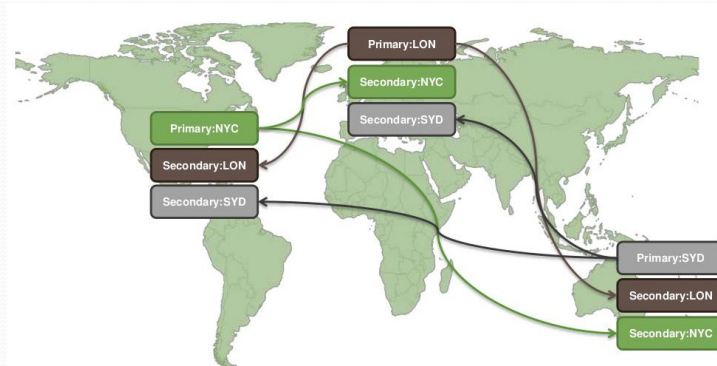
37

Sharding



Sharding

- Distribute data across cluster based on query patterns or data locality
 - Global development / local writes

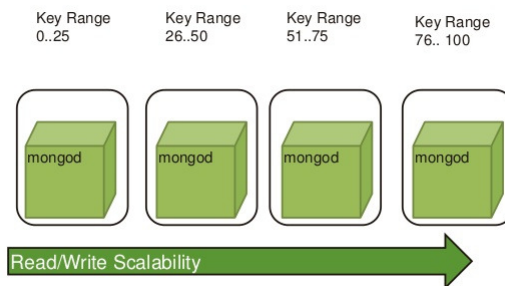


Sharding

- Types of sharding:
 - Range
 - Hash
 - Zone

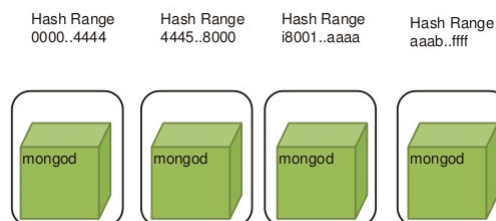
Range Sharding

- Divides data into ranges based on shard key values
- Efficient queries when reading documents in a contiguous range
- Can have poor read and write performance with poor shard key range selection



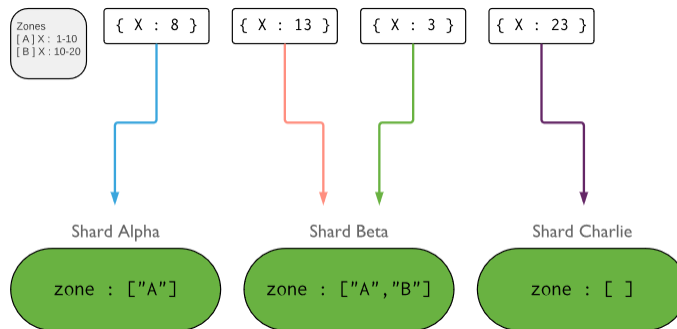
Hash Sharding

- More even data distribution
- Can impact performance of range-based queries



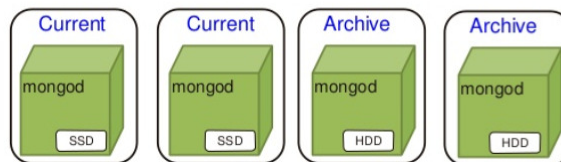
Zone Sharding

- Used to improve locality of data
 - By geographic region
 - By hardware configuration for tiered storage-architectures
 - By application feature



Sharding Allows for Tiered Storage

- Save hardware costs
 - Put frequently access data on fast servers



MongoDB Basics: Security

Copyright Kim Greene Consulting, Inc. All rights reserved worldwide.

45

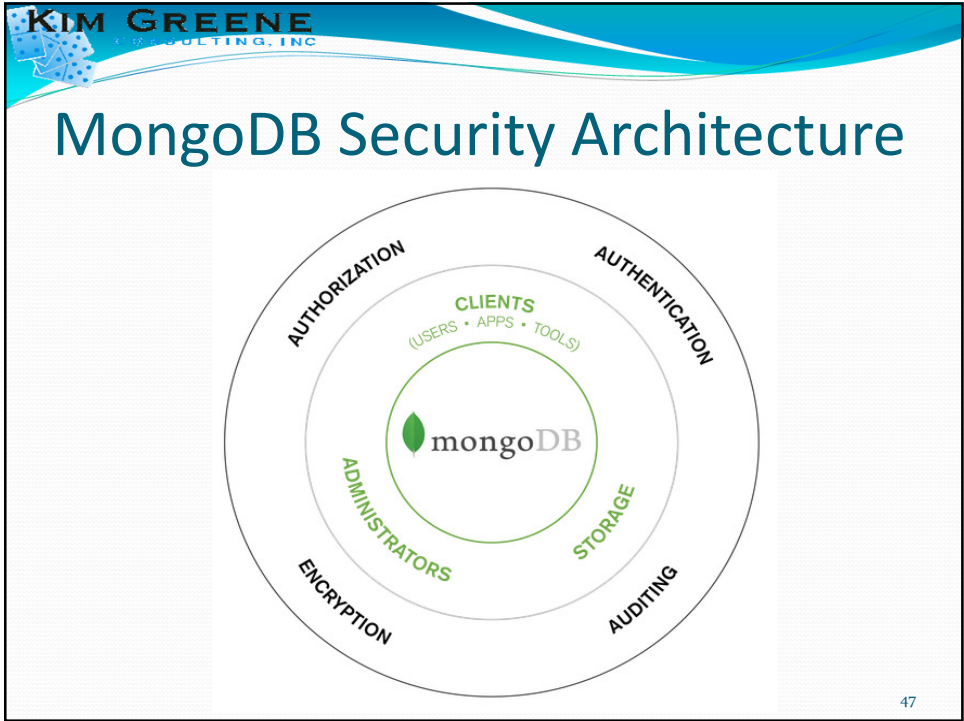
KIM GREENE
CONSULTING, INC.

Security

- Database security model needs to offer:
 - Control of read and write access to data
 - Protection of integrity and confidentiality of data stored
 - Control of modifications to database system configuration
 - Privilege levels for different user types, administrators, applications, ...
 - Auditing of sensitive operations
 - Stable and secure operation in potentially hostile environment

Copyright Kim Greene Consulting, Inc. All rights reserved worldwide.

46



MongoDB Security

Authentication vs Authorization

<p>Verifies the identity of a user.</p> <p>Answers the question: Who are you?</p>	<p>Verifies the privileges of a user.</p> <p>Answers the question: What do you have access to?</p>
---	--

Copyright Kim Greene Consulting, Inc. All rights reserved worldwide.

MongoDB Security

Authentication Mechanisms

Client/User Auth

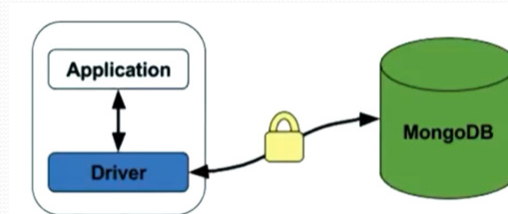
SCRAM-SHA-1
MONGODB-CR
X.509
LDAP
Kerberos

Internal Auth

Keyfile (SCRAM-SHA-1)
X.509

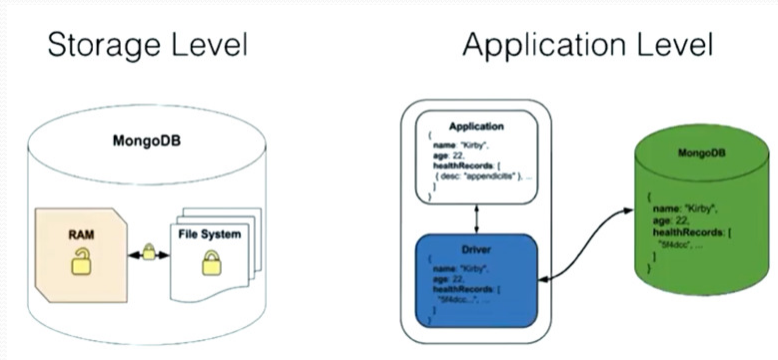
MongoDB Security - Encryption

- Transport encryption
 - Encrypt data over network traffic between the client and the server



MongoDB Security - Encryption

- Encryption at rest
 - Encrypt data stored on disk



Copyright Kim Greene Consulting, Inc. All rights reserved worldwide.

51

MongoDB Security - Auditing

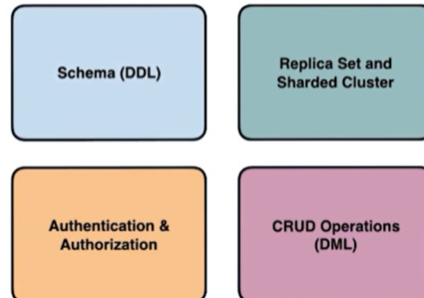
- Requires MongoDB Enterprise
- Allows for:
 - Added accountability
 - Investigation of suspicious activity
 - Monitor database activities

Copyright Kim Greene Consulting, Inc. All rights reserved worldwide.

52

MongoDB Security - Auditing

Auditing Capabilities

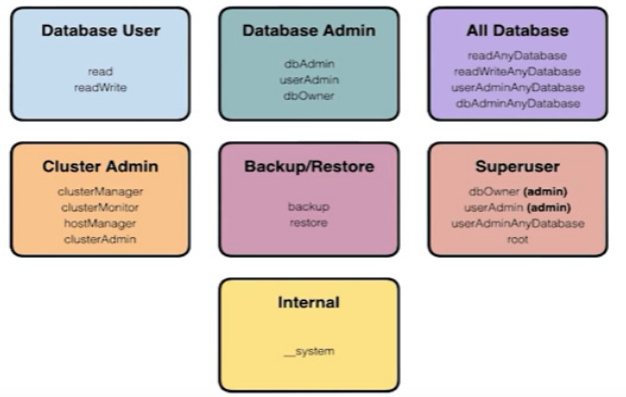


MongoDB Security - Roles

- **Roles** are groups of **privileges, actions** over **resources**, that are granted to **users** over a given **namespace (database)**.
 - Actions = all operations of commands
 - Resources = what actions are performed on
 - Privilege = action a user executes against a resource

MongoDB Security - Roles

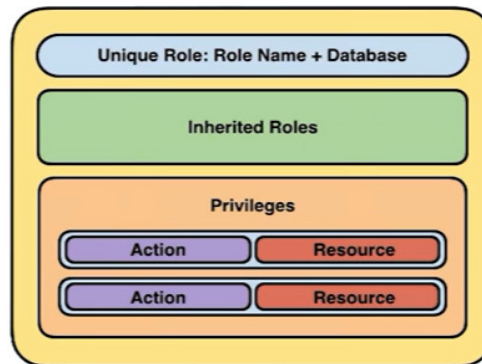
- Built-in roles



MongoDB Security - Roles

- User-defined roles

- Used when system defined roles don't provide level of access required



MongoDB Basics: Interacting with MongoDB

Copyright Kim Greene Consulting, Inc. All rights reserved worldwide.

57

KIM GREENE
CONSULTING, INC.

Interacting with the Database

- mongo shell
 - Interactive JavaScript shell
- MongoDB Compass
 - GUI
 - Modify docs, create validation rules, optimize query performance
 - Build and execute queries with results viewed both graphically and as a set of JSON documents
- MongoDB Professional
 - The full monty

MongoDB Basics: Developer Friendly

Copyright Kim Greene Consulting, Inc. All rights reserved worldwide.

59

KIM GREENE
CONSULTING, INC.

Developer Friendly

- Query model is implemented as methods or functions within the API of a programming language
- No need to write in a separate language like SQL
- MongoDB Drivers
 - APIs that expose methods to operate with MongoDB
 - Handle communication and pooling with server
 - Two types of drivers:
 - Officially supported drivers
 - Community supported drivers
 - 36 categories of drivers

Copyright Kim Greene Consulting, Inc. All rights reserved worldwide.

60





Developer Friendly

- Provides native drivers for popular programming languages plus over 30 community-developed drivers



Developer Friendly

Documents make developers more productive

 <p>Natural: map to how entities are represented in the real world, and in app code</p>	 <p>Flexible: adapt structure at any time, without expensive schema migrations</p>
 <p>Versatile: general purpose, to model and query data any way you like</p>	 <p>Fast: single structure for data access, simpler code, and efficient scale-out</p>

MongoDB Basics: Indexes and Search

Copyright Kim Greene Consulting, Inc. All rights reserved worldwide.

63



Indexes

- Fundamental requirement for performance
- Ensures quick and efficient access to data
- Prevent collection scans

Types of Secondary Indexes

- Unique
- Compound
- Array
- Time to Live (TTL)
- Geospatial
- Partial
- Sparse
- Text search

Types of Secondary Indexes

- Unique
 - Rejects insertion of new documents or the update of a document with an existing value for the field it is built over
- Compound
 - Useful for queries that specify multiple predicates
 - Example: Find customers based on last name, first name, and city of residence
 - Can reduce the need for single field indexes as any leading field in a compound index can be used

Types of Secondary Indexes

- Array
 - For fields that contain an array, each array value is stored as a separate index entry
- Time to Live (TTL)
 - Specify a period of time after which the data is automatically deleted from the database

Types of Secondary Indexes

- Geospatial
 - Allow MongoDB to optimize queries for documents that contain points or a polygon that are closest to a given point or line; that are within a circle, rectangle, or polygon; or that intersect with a circle, rectangle, or polygon
- Partial
 - Use to include only documents that meet specific conditions

Types of Secondary Indexes

- Sparse
 - Contain entries for documents that contain a specified field
 - Allow for smaller, more efficient indexes when fields are not present in all documents
- Text search
 - Specialized index for text search that uses advanced, language-specific linguistic rules for stemming, tokenization, case sensitivity and stop words

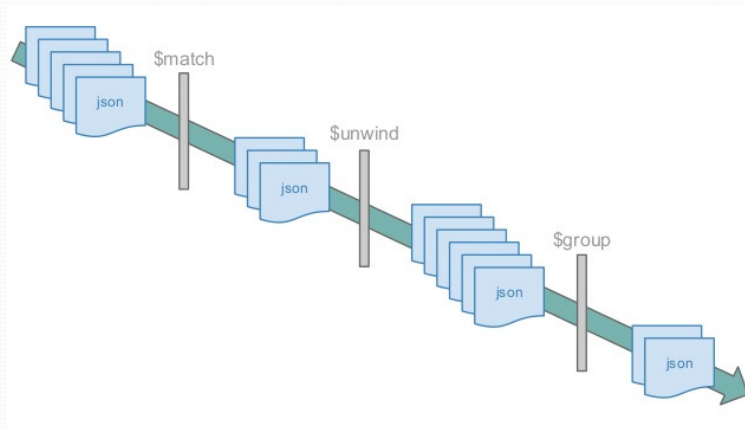
Expressive Query Language

Rich queries	Find Kim's cat Find all cats in Quebec ages 5 to 12
Geo spatial	Find all cats in the 10 mile radius of Alexandria, VA
Text search	Find all black cats Find all cats with tiger stripes
Aggregation	Calculate how much cat food it takes to feed the cats in Sydney
Map reduce	What is the population of cats on each continent over the past 50 years

Aggregation Pipeline

- Replaces find in certain scenarios
- Improves performance significantly
 - Moves processing from the client side to the server
 - Saves CPU and bandwidth
- Reduce the amount of data transmitted
- Consists of stages
 - Documents pass through each stage, output of each stage passes to next stage

Aggregation Pipeline



How MongoDB Compares to Relational Databases

Copyright Kim Greene Consulting, Inc. All rights reserved worldwide.

73



Relational Databases

- Relational databases
 - Designed for all purposes
 - Strong consistency, concurrency, recovery (ACID)
 - Mathematical background
 - Standard Query Language (SQL)

Copyright Kim Greene Consulting, Inc. All rights reserved worldwide.

74

Relational Databases

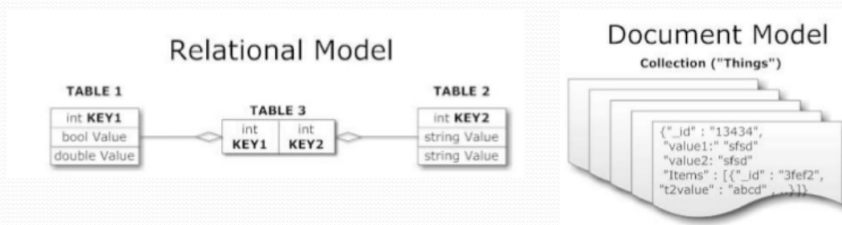
- Not built for distributed applications
 - Joins are expensive
 - Hard to scale horizontally
 - Impedance mismatch occurs
 - Expensive
 - Product cost, hardware, maintenance

Schema-Less Datamodel

- RDBMS limitations
 - Can't add record which doesn't fit schema
 - Need to add NULLS to unused items in row
 - Datatype limitations, i.e. can't add string to an integer field
 - Can't add multiple items in a field
 - Need to create separate tables
 - Primary-key, foreign key, joins, normalization required

Schema-Less Datamodel

- NoSQL
 - No schema to consider
 - No unused cells
 - No datatype limitations
 - All items are gathered in an aggregate (document)

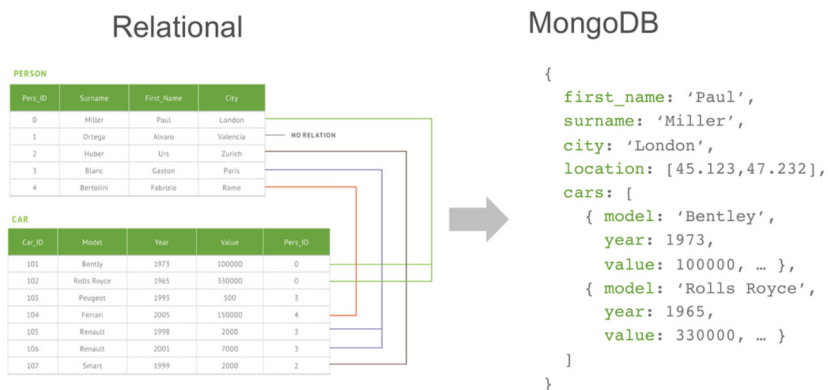


Copyright Kim Greene Consulting, Inc. All rights reserved worldwide.

77

MongoDB vs. RDBMS

DOCUMENT DATA MODEL



Copyright Kim Greene Consulting, Inc. All rights reserved worldwide.

78

ACID vs BASE

- RDBMS systems = ACID
 - Atomicity
 - All or nothing
 - A change should work or fail as a whole
 - Consistency
 - At end of transaction, all data is left in a consistent state
 - Isolation
 - Modifications of data performed by a transaction must be independent of another transaction
 - Durability
 - In event of a failure, database can fully recover
 - Once user/application has been notified of success, transaction will persist and not be undone

ACID vs BASE

- NoSQL = BASE
 - Basically Available
 - If a single node fails, part of the data may not be available, but the entire data layer stays operational
 - Soft state
 - State of system may change over time, due to eventual consistency model
 - Eventual consistency
 - Updates will eventually ripple through all servers, given adequate time

MongoDB and ACID

- Supported today at the document level
 - Just not writing to multiple documents at the same time
- Support for multi-document ACID transactions is coming in version 4.0

RDBMS vs NoSQL

- RDBMS
 - When data validity is required
 - When need to support dynamic queries
- NoSQL
 - When it's more important to have fast data than guaranteed 100% up-to-date data
 - When need to scale based on changing requirements

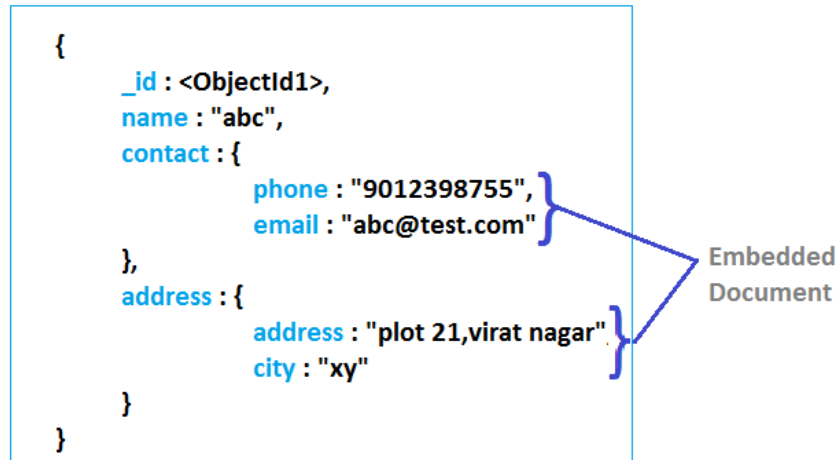
MongoDB and RDBMS

- RDBMS
 - Structures data into tables and rows
- MongoDB
 - Structures data into collections of JSON documents
 - Composed of a set of fields which are themselves key-value pairs

MongoDB vs. Relational

Relational Database	MongoDB
Database	Database
Table	Collection
Index	Index
Row	Document
Column	Field
Join	Embedding & Linking

Embedding



Referencing

- Link to other documents when:
 - One to many relationships
 - Need to access parts of data stand-alone

```
{
  name : 'left-handed smoke shifter',
  manufacturer : 'Acme Corp',
  catalog_number: 1234,
  parts : [ // array of references to Part documents
    ObjectID('AAAA'), // reference to the #4 grommet above
    ObjectID('F17C'), // reference to a different Part
    ObjectID('D2AA'),
    // etc
  ]
}
```

Data Validation

- RDBMS
 - Validation done in the database
- Most NoSQL databases
 - Push enforcement of data validation controls to application code
- MongoDB
 - Provides data validation within the database
 - Developers can enforce checks on document structure, data types, date ranges, presence of mandatory fields, ...

Developer Experience

- Relational
 - Query language separate language, typically SQL
- Most NoSQL
 - Limited to simple key-value operations, no complex queries
- MongoDB
 - Query model implemented as methods or functions within the API of a specific programming language
 - Supports complex queries, aggregations and secondary indexes
 - Sharding is automatic and built into the database, developers don't have to build sharding logic in application

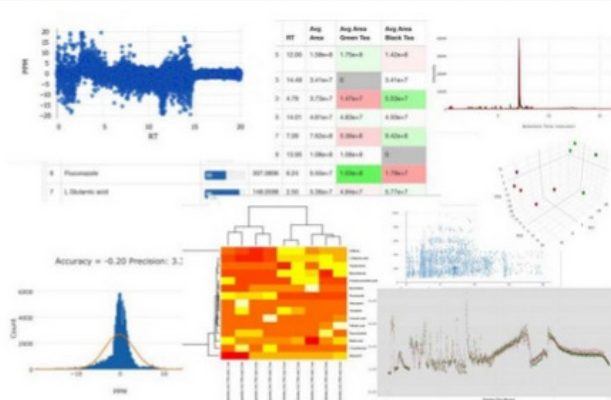
Why MongoDB vs. Relational?

- Applications needing to work with huge data volumes or new and rapidly changing data types
- Need to roll out code changes very quickly
- Applications delivered as services
- Companies moving to scale-out architecture and open source software

Why and How Customers are Using MongoDB

ThermoFischer

- Scientific applications generate humongous data



ThermoFischer

- Why MongoDB was chosen
 - Performance and scalability
 - Reliability
 - Developer productivity
 - Cost effective
 - Runs anywhere
 - Rich set of features
 - Achievement of legal and regulatory approval

Inserting Data: MongoDB vs. MySQL

- Inserting 1,615 chemical compound records into two parent-child tables
- Turned off foreign keys during insert and used string builder to create bulk insert SQL statement in MySQL

Database	Milliseconds	Lines of code
MySQL not optimized	147,600 (2.5 minutes)	21
MySQL optimized	410	40
MongoDB	68	1

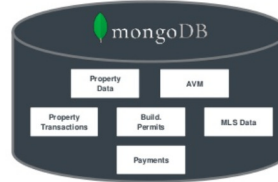
CoreLogic

	RENTAL PROPERTIES	REAL ESTATE	MORTGAGE & CAPITAL MARKETS	INSURANCE	GOVERNMENT
Consumer Experience	Rents an Apartment	Decides to Buy a House	Needs Financing or Refinancing	Needs Insurance & Makes Claims	Expects Regulatory Protection
Clients	Property Managers, Property Owners	Realtors, Property Information Services, Contractors	Lenders, Servicers, Capital Markets, GSEs	Insurance Carriers, Re-Insurance	Government, Regulators
CoreLogic Solutions	Underwriting				
	Risk Management				
	Valuations				
	Market Intelligence				
CoreLogic Reach	1 of 3 Rental Properties	70% of Real Estate Agents	3 Out of Every 4 Loans	70% of Homeowner Insurance Policies	Almost Every Housing Regulator

CoreLogic

MongoDB

Repository for Multiple Data Sets



- ◆ A new addition – not entire data warehouse
- ◆ Data for every real estate property in the US
 - Location, address, zip, city, county, characteristics, owner...
 - Sale transactions history, loans, payments
- ◆ Computation results
 - Estimated values
 - Confidence scores
 - Statistical distributions

```
{ "streetName": "THAYER POND",  
  "streetSuffix": "DR",  
  "unitNumber": "6",  
  "zip5": "01537",  
  "zip9": "1140",  
},  
"avm": {  
  "estimatedAet": 87373.0,  
  "estimatedStDev": 14.0,  
  "score": 76.0,  
  "valuationDate": ISODate("2017-05-04T00:00:00.000+0000")  
},  
"characteristics": {  
  "aboveGroundSqFt": 0.0,  
  "airConditioning": {  
    "value": "Y"  
  },  
  "assessmentInfo": {  
    "assessedLand": 0.0,  
    "assessedYear": ISODate("2017-01-01T00:00:00.000+0000"),  
    "improvedValue": 78900.0,  
    "totalAssessedValue": 78900.0  
  },  
  "basementFinishedArea": 0.0,  
  "basementUnfinishedArea": 0.0,  
  "baths": {  
    "calculated": 4
```

CoreLogic

- Efficient support for geospatial information

- Store latitude, longitude and other geo data
- Specialized geospatial index for fast search
 - Search by location or area (polygon, circle)
- Geospatial operators used:
 - \$geoWithin
 - \$GeoIntersects
 - \$near



Telefonica

Telefonica achieved almost 4x faster development with 1/2 the team – and better performance



Business requirement: personalization engine built on consolidated repository of customer data

Developed with Oracle

- Development team: 20
- Development time: 15 months
- 3 iterations
- Low performance and did not scale

Requirements changed

- 40% more data
- Reload entire data warehouse (22M customers) daily
- Oracle could not meet demand

Rebuilt with MongoDB

- 1/2 the development team
- <1/2 the development time
- Latency reduced 10x
- Storage costs decreased 67%

Ticketek

Ticketek relies on MongoDB for flexibility and scale in the core of its business



"When tickets to the most anticipated concerts go on sale, we need to make sure that we're providing the optimal experience to the hundreds of thousands of people trying to purchase them."

-Matt Cudworth, CTO

28M tickets for 20,000 events every year

- Tickets sold through an e-commerce platform built on **MongoDB Atlas**, a fully-managed Database-as-a-Service
- Real-time integration, analytics, and dashboards provide insights into ticket sales and trends

eHarmony



eHarmony

- Goals of redesign
 - Simplified communication flow
 - Realtime messaging system
 - Extend system to support richer content types
 - Giphy
 - Video/Photos
 - Stickers
 - Support various custom business requirements
 - Improved user experience

Where to Find More Information

Copyright Kim Greene Consulting, Inc. All rights reserved worldwide.

101

KIM GREENE
CONSULTING, INC.

Where to Find More Information

- MongoDB University
 - university.mongodb.com
- YouTube tutorials
 - youtube.com/mongodb
- Advocacy Hub
 - advocacy.mongodb.com

KIM GREENE
CONSULTING, INC.

Questions?



Copyright Kim Greene Consulting, Inc. All rights reserved worldwide. 103

KIM GREENE
CONSULTING, INC.

Contact Information

 www.linkedin.com/in/kimgreeneconsulting

 @iSeriesDomino

 kim@kimgreene.com

104