

mongoDB[®]

An introduction to MongoDB

SQL vs NoSQL

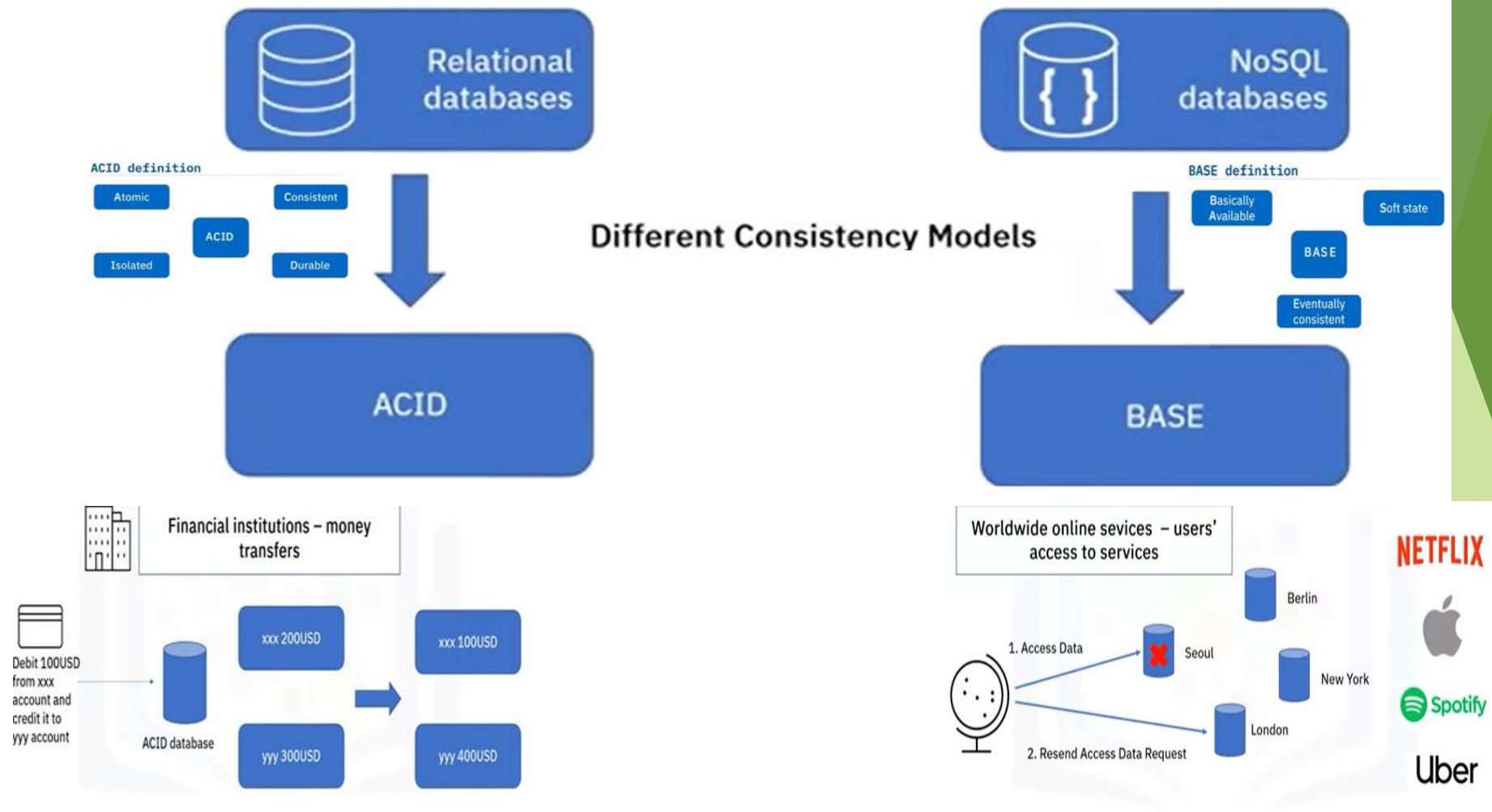


mongoDB®

- ✓ NoSQL (often interpreted as Not only SQL) database
- ✓ It provides a mechanism for storage and retrieval of data that is modeled in means other than the tabular relations used in relational databases.

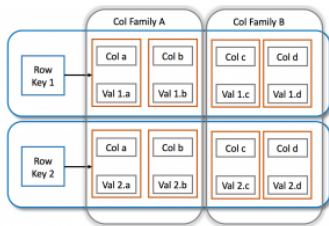
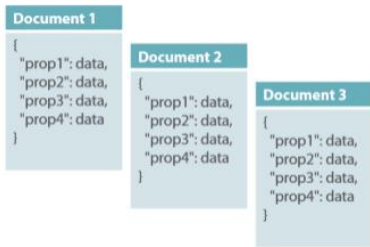
SQL	NoSQL
Relational Database Management System (RDBMS)	Non-relational or distributed database system.
These databases have fixed or static or predefined schema	They have dynamic schema
These databases are best suited for complex queries	These databases are not so good for complex queries
Vertically Scalable	Horizontally scalable
Follows ACID property	Follows BASE property

SQL vs NoSQL





NoSQL Types



Graph database



Document-oriented



Column family





mongoDB®

What is MongoDB?

- ✓ MongoDB is an **open source, document-oriented** database designed with both scalability and developer agility in mind.
- ✓ Instead of storing your data in **tables and rows** as you would with a relational database, in MongoDB you store **JSON-like documents** with **dynamic schemas**(schema-free, schema less).

```
{
  "_id" : ObjectId("5114e0bd42..."),
  "FirstName" : "John",
  "LastName" : "Doe",
  "Age" : 39,
  "Interests" : [ "Reading", "Mountain Biking"
]
  "Favorites": {
    "color": "Blue",
    "sport": "Soccer"
  }
}
```

MongoDB is Easy to Use

Relational

Person:

Pers_ID	Surname	First_Name	City
0	Miller	Paul	London
1	Ortega	Alvaro	Valencia
2	Huber	Urs	Zurich
3	Blanc	Gaston	Paris
4	Bertolini	Fabrizio	Rom

Car:

Car_ID	Model	Year	Value	Pers_ID
101	Bentley	1973	100000	0
102	Rolls Royce	1965	330000	0
103	Peugeot	1993	500	3
104	Ferrari	2005	150000	4
105	Renault	1998	2000	3
106	Renault	2001	7000	3
107	Smart	1999	2000	2

no relation



MongoDB Document

```
{  
  first_name: 'Paul',  
  surname: 'Miller'  
  city: 'London',  
  location: [45.123,47.232],  
  cars: [  
    { model: 'Bentley',  
      year: 1973,  
      value: 100000, ... },  
    { model: 'Rolls Royce',  
      year: 1965,  
      value: 330000, ... }  
  ]  
}
```



Scheme Free

MongoDB does not need any pre-defined data schema

Every document could have different data!

```
{name: "will",
eyes: "blue",
birthplace: "NY",
aliases: ["bill", "ben"],
loc: [32.7, 63.4],
boss: "ben"}
```

```
{name: "jeff",
eyes: "blue",
loc: [40.7, 73.4],
boss: "ben"}
```

```
{name: "brendan",
boss: "will"}
```

```
{name: "ben",
age: 25}
```

```
{name: "matt",
weight: 60,
height: 72,
loc: [44.6, 71.3]}
```

RDBMS vs MongoDB

RDBMS		MongoDB
Database	➡	Database
Table	➡	Collection
Row	➡	Document (JSON, BSON)
Column	➡	Field
Index	➡	Index
Join	➡	Embedded Document
Partition	➡	Shard

JSON Format

```
{
  "_id": 1,
  "name": { "first" : "John", "last" : "Backus" },
  "contribs": [ "Fortran", "ALGOL", "Backus-Naur Form", "FP" ],
  "awards": [
    {
      "award": "W.W. McDowell Award",
      "year": 1967,
      "by": "IEEE Computer Society"
    }, {
      "award": "Draper Prize",
      "year": 1993,
      "by": "National Academy of Engineering"
    }
  ]
}
```


BSON

```
{"hello": "world"} →  
\x16\x00\x00\x00      // total document size  
\x02                   // 0x02 = type String  
hello\x00              // field name  
\x06\x00\x00\x00world\x00 // field value  
\x00                   // 0x00 = type E00 ('end of object')
```

```
{"BSON": ["awesome", 5.05, 1986]} →  
\x31\x00\x00\x00  
  \x04BSON\x00  
  \x26\x00\x00\x00  
  \x02\x30\x00\x08\x00\x00\x00awesome\x00  
  \x01\x31\x00\x33\x33\x33\x33\x33\x14\x40  
  \x10\x32\x00\xc2\x07\x00\x00  
  \x00  
  \x00
```



Features Of MongoDB

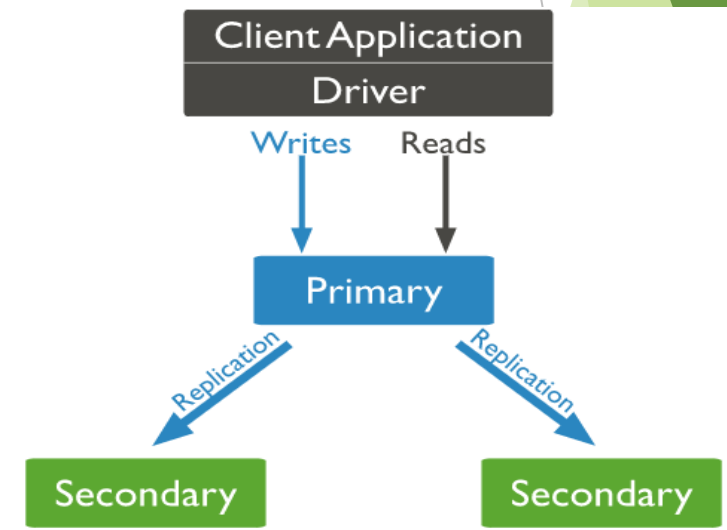
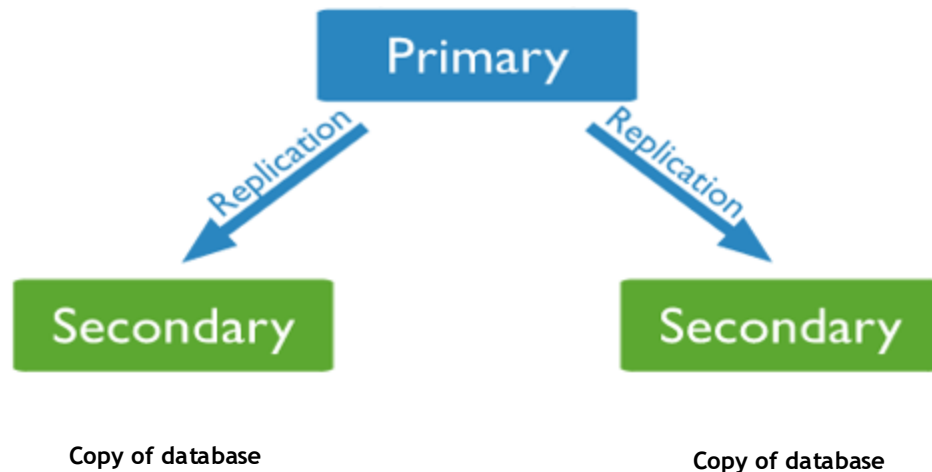
- Document-Oriented storage
- Full Index Support
- Replication & High Availability
- Auto-Sharding
- Aggregation
- MongoDB Atlas
- Various APIs
 - JavaScript, Python, Ruby, Perl, Java, Java, Scala, C#, C++, Haskell, Erlang
- Community



mongoDB®

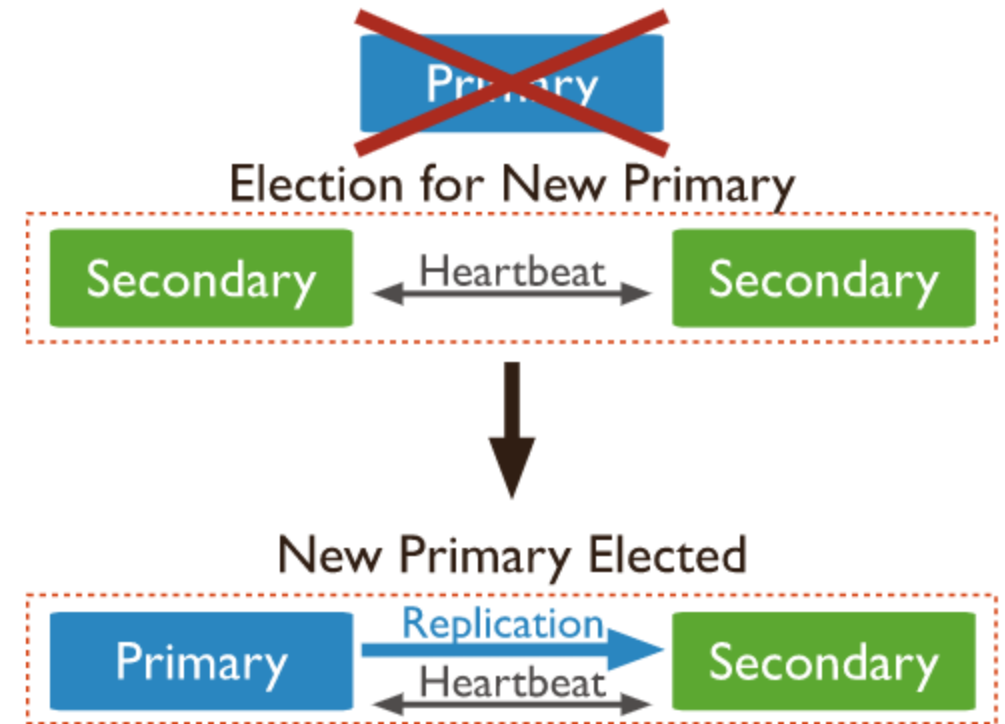
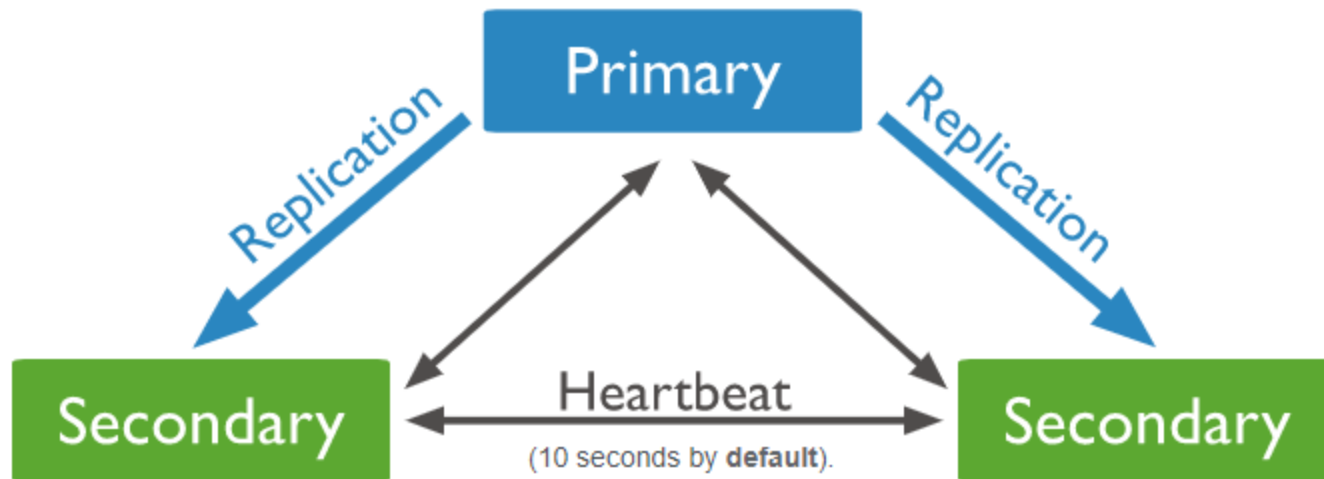
Replication

- Replication provides redundancy and increases data availability.
- With multiple copies of data on different database servers, replication provides a level of fault tolerance against the loss of a single database server.





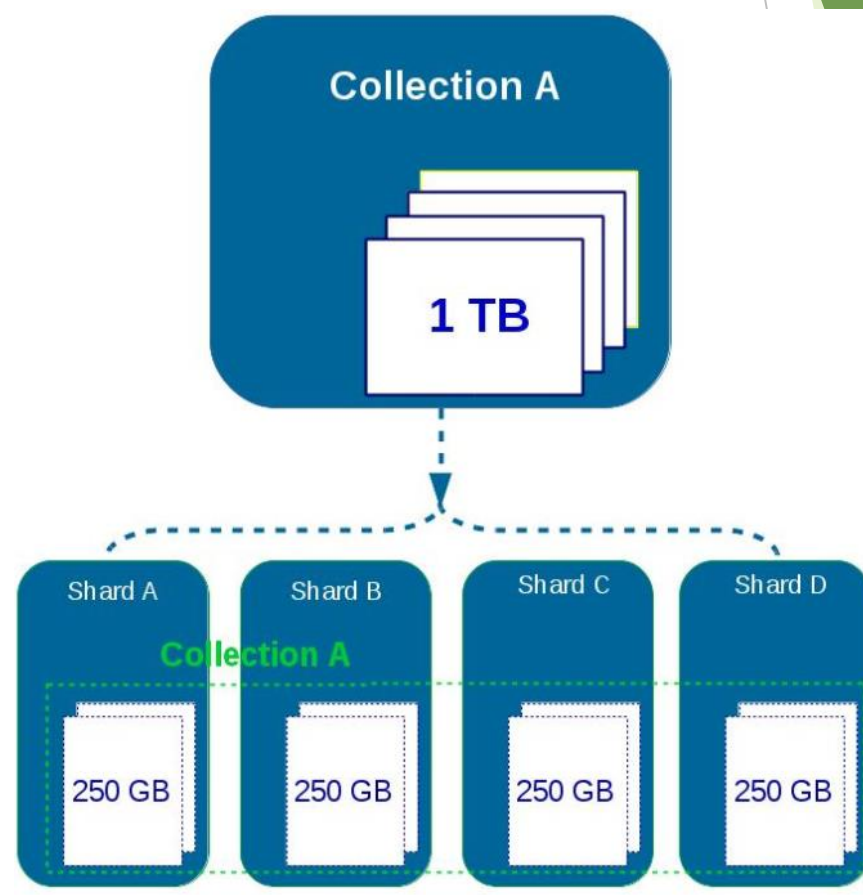
Replication





Sharding

- Sharding is a method for distributing data across multiple machines.
- MongoDB uses sharding to support deployments with very large data sets and **high throughput operations**.

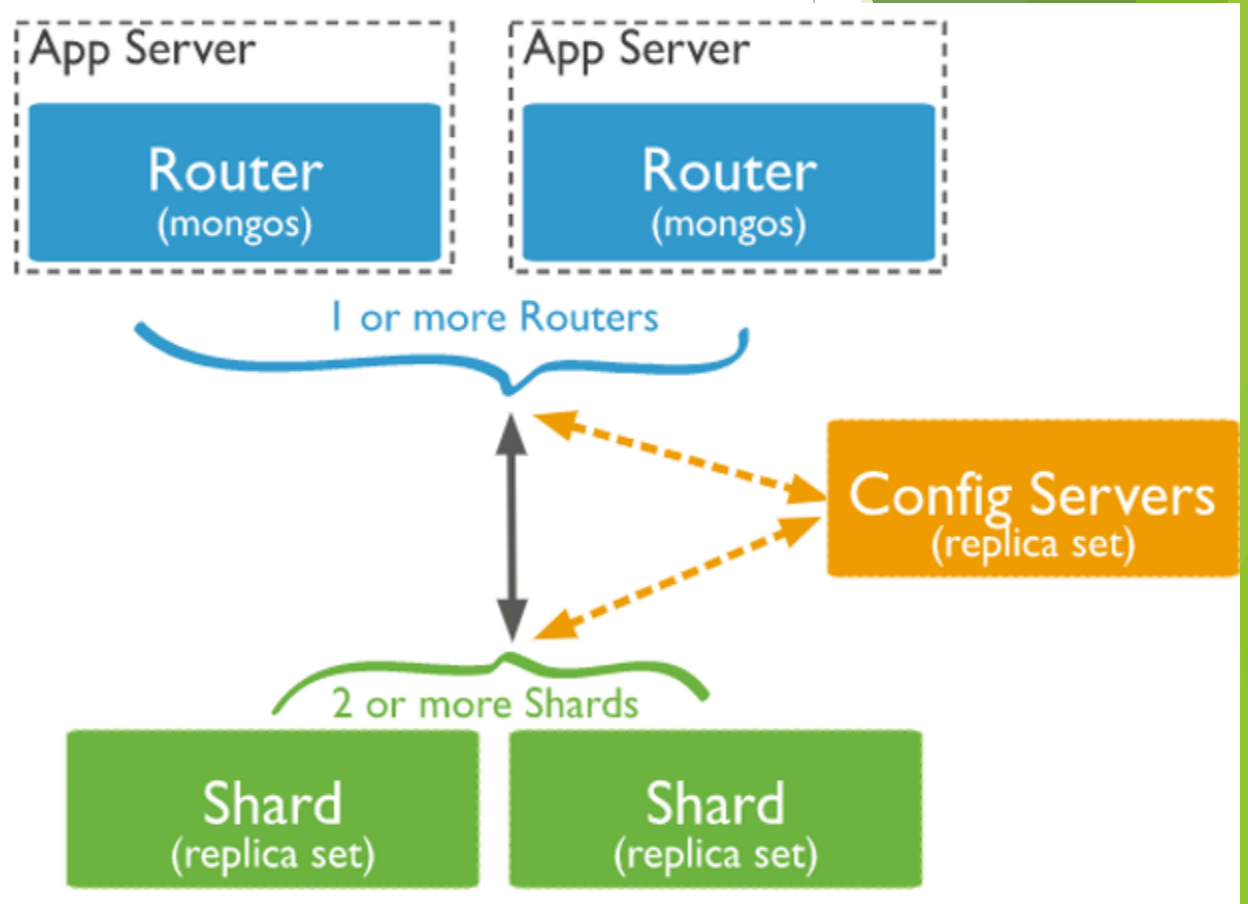




mongoDB®

Sharding Architecture

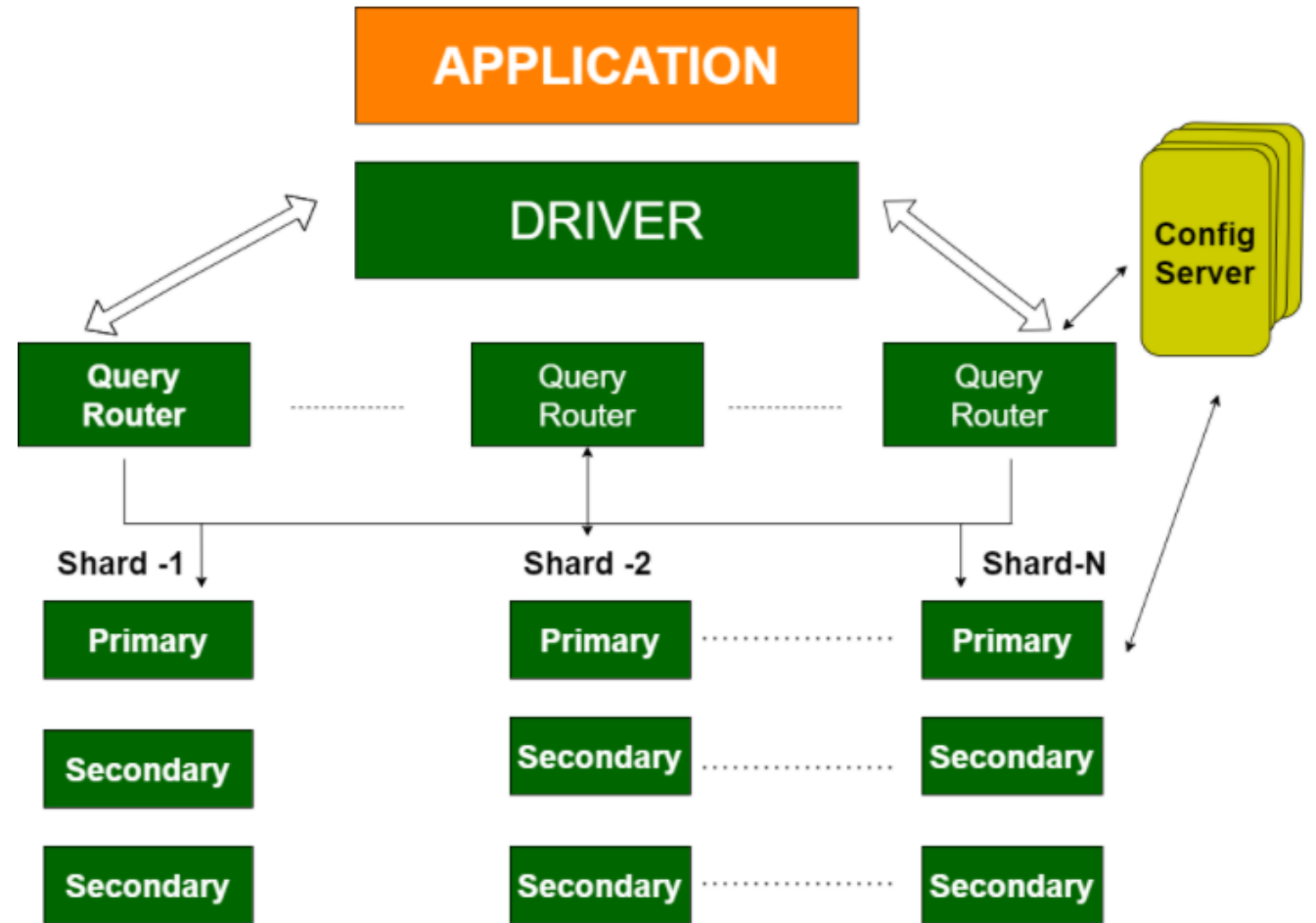
- **Shard** is a Mongo instance to handle a subset of original data.
- **Mongos** is a query router to shards.
- **Config Server** is a Mongo instance which stores metadata information and configuration details of cluster.



Sharding/Replication



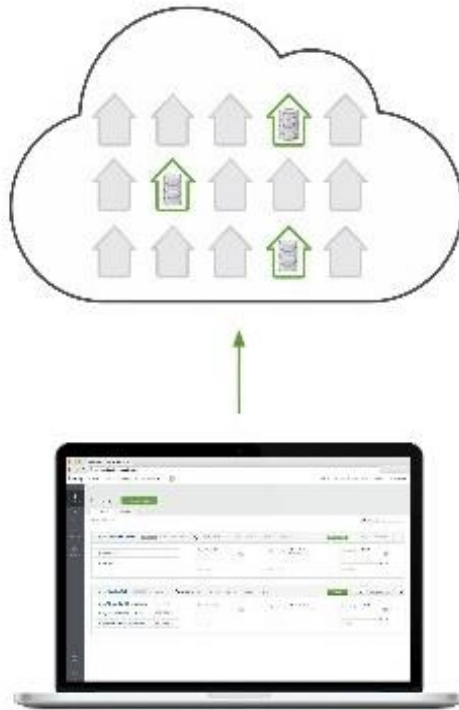
- Replication Split data sets across multiple data nodes for high availability.
- Sharding scale up/down horizontally when it is required for high throughput





MongoDB Atlas Benefits

Database as a service for MongoDB



Run for You

- Spin up a cluster in seconds
- Replicated & always-on deployments
- Fully elastic: scale out or up in a few clicks with zero downtime
- Automatic patches & simplified upgrades for the newest MongoDB features

Safe & Secure

- Authenticated & encrypted
- Continuous backup with point-in-time recovery
- Fine-grained monitoring & custom alerts

No Lock-In

- On-demand pricing model; billed by the hour
- Multi-cloud support (AWS available with others coming soon)
- Part of a suite of products & services designed for all phases of your app; migrate easily to different environments (private cloud, on-prem, etc) when needed



Thank You

For Your Attention