

# Syllabus

- ▶ **Introduction to MongoDB:** definition of MongoDB: what is a document database, what is JSON, How does it look, cross-platform, scalable, flexible.
- ▶ An overview of MongoDB architecture, introduction to mongo dB basics, core concepts and vocabulary, mongo dB databases
- ▶ Mongo dB collections, mongo dB documents, introduction to mongo dB shell, types of storage engines, locks

# MongoDB

- ▶ MongoDB is one of the top most popular modern databases which is now widely used for building modern day applications. It is different from traditional RDBMS (Relational Database Management Systems)
- ▶ It is categorized as NoSQL database because of the flexibility of schema which it provides.

# SQL vs. MongoDB

- ▶ SQL has databases, tables, rows, columns
- ▶ Mongo has databases, collections, documents, fields
- ▶ Both have primary keys, indexes
- ▶ Collection structures are not enforced heavily
  - ▶ Inserts automatically create schemas

# Interacting with MongoDB

- ▶ Multiple databases within MongoDB
  - ▶ Switch databases
    - ▶ use `use dbName`
  - ▶ New databases will be stored after an insert
- ▶ Create collection
  - ▶ `db.createCollection("collectionName")`
  - ▶ Not necessary, collections are implicitly created on insert

# BSON

- ▶ MongoDB uses BSON very heavily
  - ▶ Binary JSON
  - ▶ Like JSON with a binary serialization method
  - ▶ Has extensions so that it can represent data types that JSON cannot
- ▶ Used to represent documents, provide input to queries

# Selects/queries

- ▶ In MongoDB, querying typically consists of providing an appropriately crafted BSON
  - ▶ `SELECT * FROM collectionName`
    - ▶ `db.collectionName.find()`
  - ▶ `SELECT * FROM collectionName WHERE field = value`
    - ▶ `db.collectionName.find( {field: value} )`
  - ▶ `SELECT * FROM collectionName WHERE field > 5`
    - ▶ `db.collectionName.find( {field: { $gt: 5 } } )`
- ▶ Other functions that take a query argument have queries that are formatted this way

# Interacting with MongoDB

- ▶ Insert

- ▶ `db.collectionName.insert( {queryBSON} )`

- ▶ Update

- ▶ `db.collectionName.update( {queryBSON}, {updateBSON}, {optionBSON} )`

- ▶ updateBSON

- ▶ Set field to 5: `{ $set: {field: 5} }`

- ▶ Increment field by 1 `{ $inc: {field: 1} }`

- ▶ optionBSON

- ▶ Options that determine whether or not to create new documents, update more than one document, write concerns

# Interacting with MongoDB

## ► Delete

- `db.collectionName.remove( {queryBSON} )`



# What is Document Database

- ▶ The document database is a type of non-relational database that is designed to store documents in JSON-like format.

# What is JSON?

- ▶ JSON Stands for JavaScript Object Notation and it is one of the widely used open standard file formats which uses human-readable text to store and transmit data.
- ▶ It is a lightweight format for data transportation and is often used between the client and server architecture in software development.
- ▶ JSON is very easy to understand and usually, it is "self-describing".

# How JSON LOOKS?

```
{  
  "Students": [  
    {"firstName": "Siya", "lastName": "Sharma",  
     "Location": "India"},  
    {"firstName": "Ron", "lastName": "Smith",  
     "Location": "USA"}, {"firstName": "Bash",  
     "lastName": "Tao", "Location": "Philippines"}],  
  "Teachers": [{"firstName": "Dheeraj",  
               "lastName": "Sareen",  
               "Location": "India"}, {"firstName": "David",  
               "lastName": "Baker", "Location": "Canada"}]  
}
```

# EXAMPLE

- ▶ JSON data is specified in the name and value pairs
- ▶ Each data is separated by commas in JSON.
- ▶ The square brackets in JSON are used to hold an array of object(s)

# EXAMPLE

- ▶ The curly braces in JSON are used to hold object(s)
- ▶ It is highly flexible and easy to understand as you write JSON in a human-understandable format.
- ▶ The document-based databases make the lives of developers easier as they allow them to store data in the same document-model format as developers use JSON while coding. It is easy for them to relate the name and value pairs while referring to the data entered in these databases

# CROSS PLATFORM

- ▶ MongoDB is a cross-platform database, which means it can run on various operating systems and on various computer architectures.
- ▶ It also supports various computer architectures that are machine-based and hardware-specific and depends on the processors and hardware plus operating systems that run that machine

# PLATFORMS SUPPORT MONGO DB

- ▶ Amazon Linux 2
- ▶ Debian 9 and 10
- ▶ RHEL/CentOS 6, 7, and 8
- ▶ SLES 12 and 15
- ▶ Ubuntu LTS 16.04 and 18.04
- ▶ Windows Server 2016 and 2019

# SCALABLE

- ▶ MongoDB is extremely scalable, suppose we are developing an application that is presently not widely used and has limited usage but we believe that in the coming years, the application will have a huge increase in new users and there will be a multi-fold increase in the traffic.
- ▶ MongoDB is the first choice of Fortune 100 enterprises as well as startups. These organizations rely on MongoDB for their operations. The industry has seen a dramatic increase in the deployments of MongoDB, ranging from a single server to multi servers and clusters of MongoDB database



# MONGODB MATRICS OF SCALABILITY

- ▶ MongoDB offers scalability for three different metrics:
- ▶ In terms of cluster
- ▶ In terms of performance
- ▶ *Scalability* in terms of data

# FLEXIBILITY

- ▶ MongoDB doesn't heavily enforce schema but uses dynamic schema, and hence, is also referred to as a schema-less database. This makes it a highly flexible database.
- ▶ Schema is the database structure in **Relational Database Management System (RDBMS)** where you define it before using the database in the application

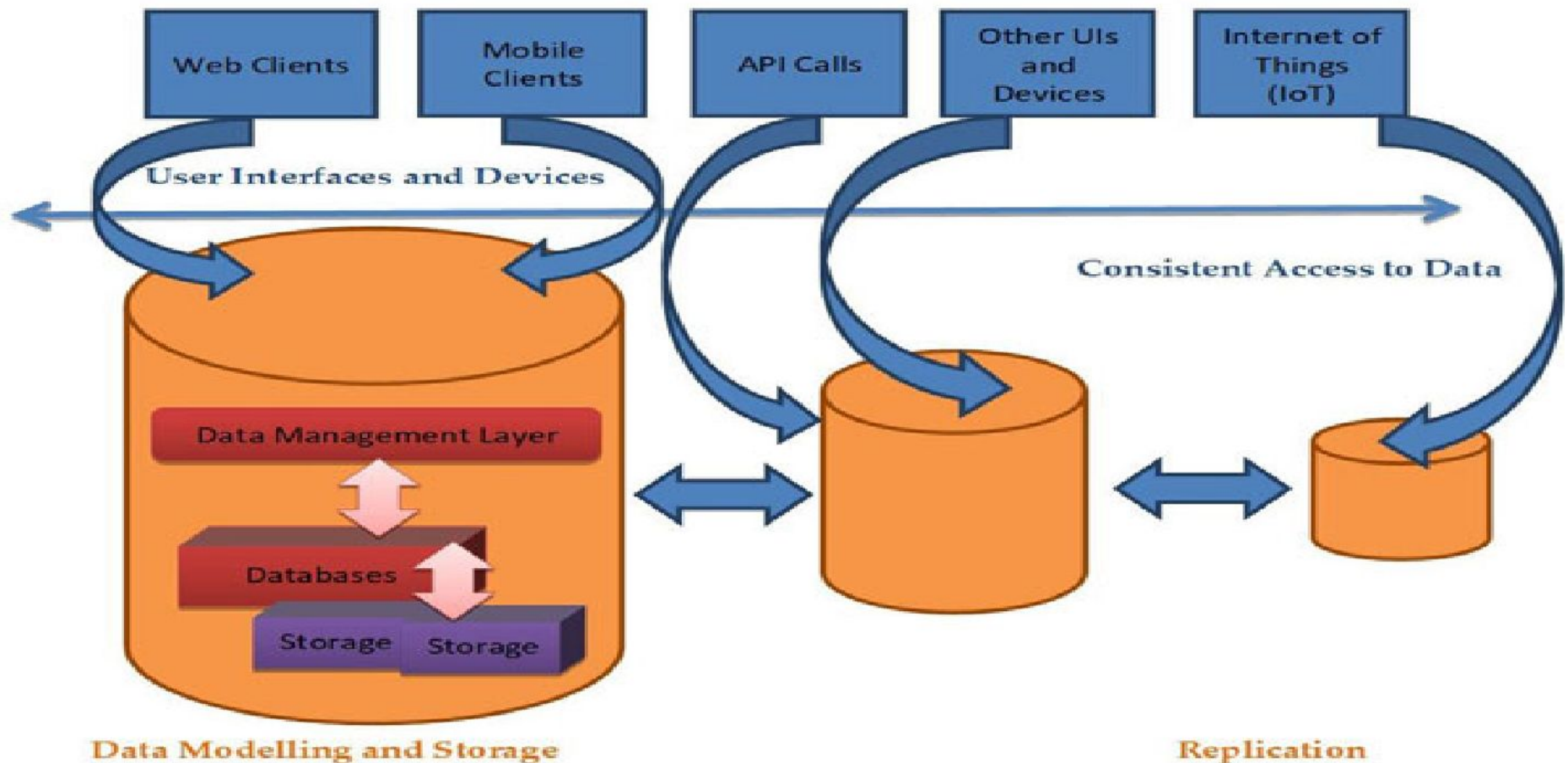
# ADVANTAGES OF FLEXIBILITY

- ▶ MongoDB is flexible in terms of its data structure.
- ▶ MongoDB relies on the JSON-like data structure which is very flexible in itself. So when we need to add any new data in MongoDB which has or more fields and where each data is dynamic
- ▶ We can easily add these records to our database in MongoDB without any issues as it won't restrict us and we are not bound to a specific data structure to be added in the database.

# NOSQL Database

- ▶ The NoSQL databases, sometimes also called non-SQL, non-relational, or not only SQL databases are those that have a different mechanism to store and retrieve data other than the tabular relations which are used in relational databases.

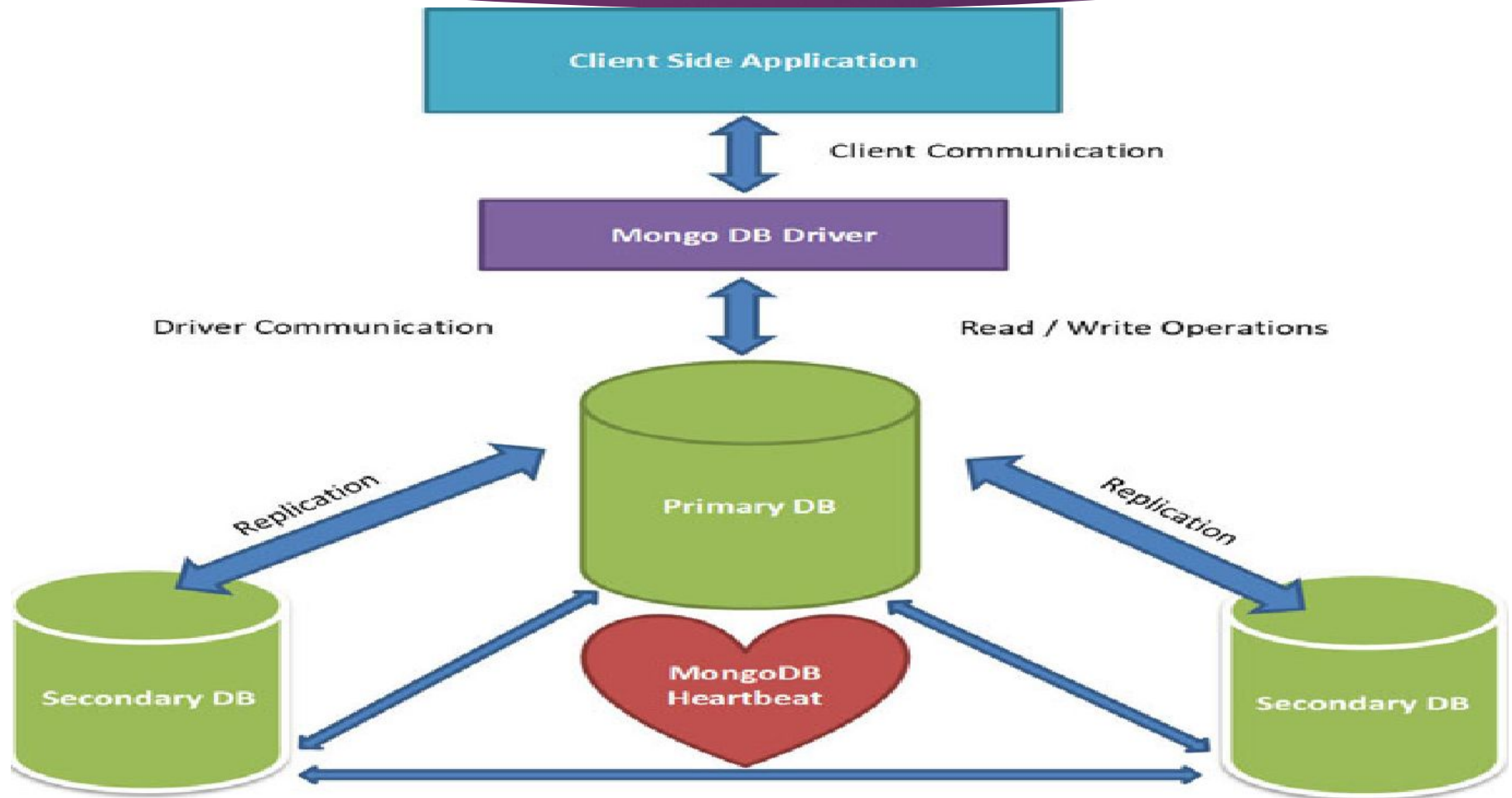
# AN OVERVIEW OF NOSQL ARCHITECTURE



# AN OVERVIEW OF NOSQL ARCHITECTURE

- ▶ The important feature of NoSQL databases with respect to their architecture is that it provides nested, hierarchical structures in data entities.
- ▶ These hierarchical data structures can be easily described with the JSON and other formats used by the NoSQL databases.
- ▶ These structures also closely match with the data structures used in the programming languages.

# Mongo DB Architecture



# Mongo DB Architecture

- ▶ The client-side application communicates with the MongoDB database with the help of MongoDB drivers.
- ▶ For any read and write operations, the MongoDB drivers play an important role to communicate with the MongoDB database.
- ▶ The MongoDB drivers depend on the programming language and help applications for various CRUD (Create, Read, Update, and Delete) and other operations with respect to the MongoDB database.
- ▶ The primary and secondary DBs ensure the high availability of data and the frequent synchronization provides eventually consistent data records.



# Mongo DB Platform

MongoDB Data Platform

MongoDB  
provides Best  
way to work  
with Data

MongoDB  
provides best  
way for Data  
Availability

MongoDB  
provides best  
way for Data  
Portability

1

2

3

# KEY FEATURES

- ▶ **Best way:** MongoDB provides an easy, fast, and flexible way to work with data. It also provides a wide range of data types and expressive queries that are easy to understand.
- ▶ **Best availability:** MongoDB provides the best way for data availability. With MongoDB, you can place and scale your data on any device as well as at any geographic location whenever you need it.
- ▶ **Best portability:** MongoDB runs the same everywhere, whether it is a local server or cloud. MongoDB is also available as a service throughout the world with major cloud platforms supporting it.

# KEY DIFFERENCES FROM OTHER DATABASES

SQL database	NoSQL database (MongoDB)
Relational database	Non-relational database
Supports SQL query language	Supports JSON query language
Table based	A collection based and key-value pair
Row-based	Document-based
Column-based	Field-based
Support for foreign key	No support for foreign key
Support for triggers	No Support for triggers
Contains a predefined schema	Contains a dynamic schema
Not fit for hierarchical data storage	Best fit for hierarchical data storage
Vertically scalable - increases RAM	Horizontally scalable - adds more servers
Emphasizes ACID properties (Atomicity, Consistency, Isolation, and Durability)	Emphasizes the CAP theorem (Consistency, Availability, and Partition tolerance)

# NOSQL Databases

- ▶ The NoSQL databases comprise a wide variety of database-related technologies to meet the dynamic demands of modern applications.
- ▶ Relational databases are not designed to meet the demands of these applications and development environments. So, the developers feel stuck at some point in the development cycle, whether during the decision on the architecture of the application or during the scaling of the application at a later stage of the deployment cycle.

# Types of NOSQL Databases

- ▶ There are four major types of NoSQL database management systems. These are classified as follows.
- ▶ Key-value paired databases
- ▶ Column-oriented databases
- ▶ Document-oriented databases
- ▶ Graph databases

# KEY VALUE PAIRED DATABASE

- ▶ These are the simplest NoSQL databases. Here, the data is stored as a name (or key) with its value. Some key-value paired databases also have the feature to give the data type such as `int` or `float`.
- ▶ Examples of key-value paired databases are:
- ▶ Berkeley DB
- ▶ Redis

# Column Oriented Database

- ▶ These types of NoSQL databases allow you to store the columns of data instead of rows.
- ▶ These are very effective in handling the queries of large data sets.
- ▶ Examples of column-oriented databases are:
  - ▶ Cassandra
  - ▶ HBase

# Document Database

- ▶ These databases pair each key with a data structure called to document and these documents then contain key-value pairs, key-array pairs, and nested documents.
- ▶ Examples of document databases are:
- ▶ MongoDB
- ▶ Couchbase



# Graph Databases

- ▶ These databases are useful in storing data that are inter-connected as nodes just like a graph.
- ▶ These databases add an extra layer of highlighting the relationship among the documents.
- ▶ Examples of document databases are:
  - ▶ OrientDB
  - ▶ Neo4j

# Introduction to Mongo DB Core Concepts and Vocabulary

## ► **Document Database**

- MongoDB documents are similar to JSON objects and these documents can contain arrays, other documents, or an array of documents.

## ► **Collections**

- The MongoDB documents are stored in collections. So you can consider this as a table with respect to the relational database.

# Introduction to Mongo DB Core Concepts and Vocabulary

- ▶ **Support for rich query language**
- ▶ MongoDB uses rich query language to support the CRUD (Create, Read, Update, and Delete) operations. It also supports the queries related to data aggregation, search operations, as well as geospatial queries.
- ▶ **Support multiple storage engines**
- ▶ MongoDB supports multiple storage engines like: Wired Tiger storage engine
- ▶ In-memory storage engine MongoDB provides a pluggable storage engine, API, that lets the third parties develop their own engine type for MongoDB

# Mongo DB Terminology

- ▶ The following are the basic MongoDB specific terms
- ▶ **MongoDB Database:** A single MongoDB server consists of various databases where each database is a physical container of collections.
- ▶ **MongoDB Collection:** A collection in MongoDB is equivalent to a database table of the SQL-based databases and it exists within a single database. It includes a group of MongoDB documents.

# Mongo DB Terminology

- ▶ **MongoDB Document:** A document can be defined as an instance of a MongoDB collection. It includes a set of key-value pairs.
- ▶ All the documents include a dynamic schema which means the documents that comprise of the same collection do not need to have the same set of fields and structure.

# Mongo DB Verses NOSQL

SQL Terms/Concepts	MongoDB Terms/Concepts
database	database
table	collection
row	document or BSON (Binary JSON) document
column	field
index	index
table joins	embedded documents
primary key	primary key
Specify any unique column or column combination as the primary key.	The primary key is automatically set to the <code>_id</code> field.