MongoDB Engine Map – Complete README

A complete, structured, production-ready map of all MongoDB internal systems and commands. Use this as a GitHub README to understand MongoDB from developer to DBA level.

📌Overview

MongoDB exposes two layers of control: - **High-level CRUD API** – used by backend developers - **Low-level db.runCommand API** – used for database engine management

This document maps every MongoDB command to the internal subsystem it belongs to.

🖨️MongoDB Subsystem Map

MongoDB internally has **9 subsystems**: 1. CRUD Layer 2. Query Engine 3. Indexing System 4. Storage Engine 5. Replication System 6. Sharding System 7. Transaction Engine 8. Security & Access Control 9. Monitoring & Diagnostics

Each section below lists the important commands.

✂️CRUD Layer (Application API)

Commands for application-level operations: - insertOne / insertMany - find / findOne - findOneAndUpdate - updateOne / updateMany - deleteOne / deleteMany - replaceOne

Low-level equivalents: - `insert` - `find` - `update` - `delete`

✒️Query Engine (Read Pipeline)

Controls how MongoDB plans and executes queries.

Query Planning & Stats - `explain` - `collStats` - `dbStats`

Cursor Controls - `getMore` - `killCursors`

Aggregation Engine - `aggregate` - `distinct` - `count`

🛏️Indexing System

Commands for index creation, removal, and inspection.

- `createIndexes`
- `listIndexes`
- `dropIndexes`
- `validate`

🌂Storage Engine (WiredTiger)

These commands interact with database kernel and hardware-level behaviors.

- `serverStatus`
- `hostInfo`
- `getParameter`
- `setParameter`
- `fsync`
- `logRotate`
- `replSetResizeOplog`

🌡️ Replication System (High Availability)

Commands for managing replica set clusters.

- `replSetGetStatus`
- `replSetGetConfig`
- `replSetInitiate`
- `replSetReconfig`
- `replSetStepDown`
- `replSetFreeze`
- `hello` / `isMaster`

🎒Sharding System (Distributed Clustering)

Commands used to manage horizontally scalable MongoDB clusters.

- `enableSharding`
- `shardCollection`
- `addShard`
- `removeShard`

- `balancerStart`
- `balancerStop`
- `getShardMap`
- `getShardVersion`

🎓Transaction Engine

Commands for ACID-compliant multi-document transactions.

- `startTransaction`
- `commitTransaction`
- `abortTransaction`
- `prepareTransaction`

🎩Security & Access Control

User, role, and privilege commands.

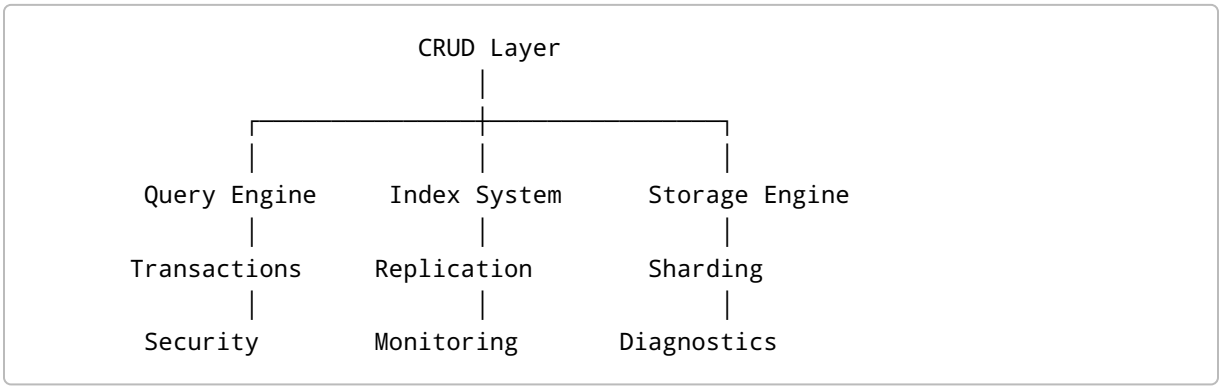
- `createUser`
- `dropUser`
- `updateUser`
- `usersInfo`
- `rolesInfo`
- `grantRolesToUser`
- `revokeRolesFromUser`

🎽Monitoring & Diagnostics

Commands for performance analysis, debugging, and incident response.

- `currentOp`
- `killOp`
- `profile`
- `serverStatus`
- `getLog`
- `top`
- `explain`

Visual Architecture



🕑Summary

- CRUD is for application code.
- `db.runCommand()` is for database engine control.
- This map helps you navigate MongoDB at developer, DevOps, and DBA level.

If you want, I can generate: - 📘A PDF version - 🛻A website/Docsify version - 📀A cheat-sheet version - 🧪A practical exercise workbook