



Ahbaid Gaffoor



# Overview



- Introduction
- Installation
- Documents
- Basics
- Locking
- Execution Plans
- Replication
- Sharding
- Monitoring

# Introduction



- JSON Documents
- Schema / Database
- Table / Collection
- Row / Document
- Schemaless Design
- Some SQL functionality
- Locking

# Installation



- <https://www.mongodb.org/downloads>
- Install
  - “tar -zxvf mongodb-\*.tgz”
- Startup
  - “mkdir ./db1”
  - “./bin/mongod --dbpath ./db1 --smallfiles”
- Connect: “./bin/mongo”
- Default Port 27001

# Documents



## Relational Model:

SELLER_ID	NAME	ADDR	PHONE
100	Tom	1 Brick Ln, San Diego, CA	903-257-6778
200	Tina	12 Mead Ave., Dallas, TX	972-671-3214

SELLER_ID	SHIPPING_METHOD
100	FedEx 2-Day Express
100	UPS Next Day
200	USPS Standard Mail

## JSON Model:

```
{
  _id:"100", name:"Tom",
  addr:"1 Brick Ln., San Diego CA",
  phone: "903-257-6778",
  shipping_methods: [
    { shipping_method:"FedEx 2-Day Express" },
    { shipping_method:"UPS Next Day" }
  ]
}
```

# Basics



- Javascript
- “\_id”
- CRUD Operations
  - Create: `db.foo.insert`
  - Read:
    - `db.foo.find`
    - `it`
    - `db.foo.findOne`
  - Update: `db.foo.update`
  - Delete: `db.foo.remove`
  - Drop: `db.foo.drop`

# Locking



- Storage Engines
  - MMAPv1
  - WiredTiger
- Global, Database, Collection Locks
- Document Locks

# Execution Plans



- `db.<collection>.explain().find()`

```
> db.zips.explain().find({state:'AL'})
{
  "queryPlanner" : {
    "plannerVersion" : 1,
    "namespace" : "zips.zips",
    "indexFilterSet" : false,
    "parsedQuery" : {
      "state" : {
        "$eq" : "AL"
      }
    },
    "winningPlan" : {
      "stage" : "COLLSCAN",
      "filter" : {
        "state" : {
          "$eq" : "AL"
        }
      },
      "direction" : "forward"
    },
    "rejectedPlans" : [ ]
  },
}
```

```
"serverInfo" : {
  "host" : "Ahbaid's-MacBook-Air.local",
  "port" : 27017,
  "version" : "3.0.1",
  "gitVersion" :
"534b5a3f9d10f00cd27737fbc951032248b5952"
},
"ok" : 1
}
```



# Replication



- Primary
- Secondaries
- Database Failover
  - Majority
  - Votes
  - Priority

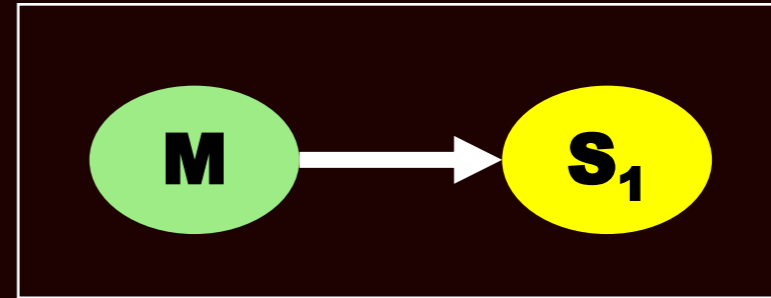


Fig 1.

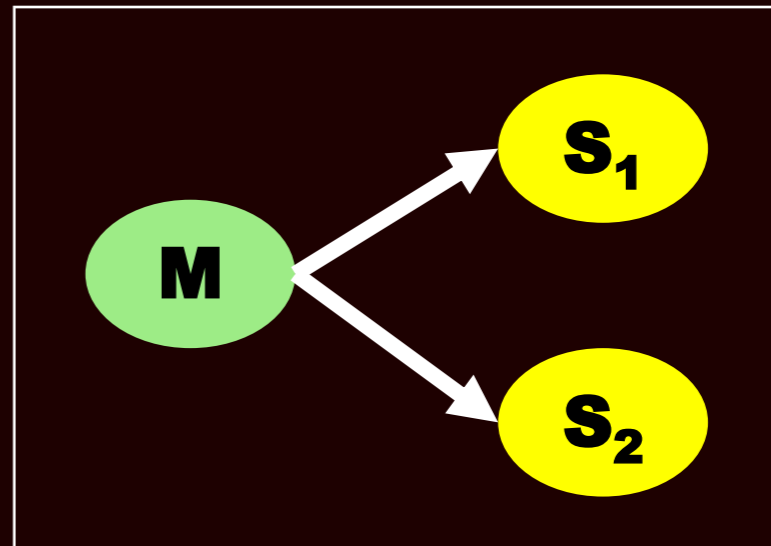


Fig 2.

# Replication



- Arbiters
  - Majority
  - Voters

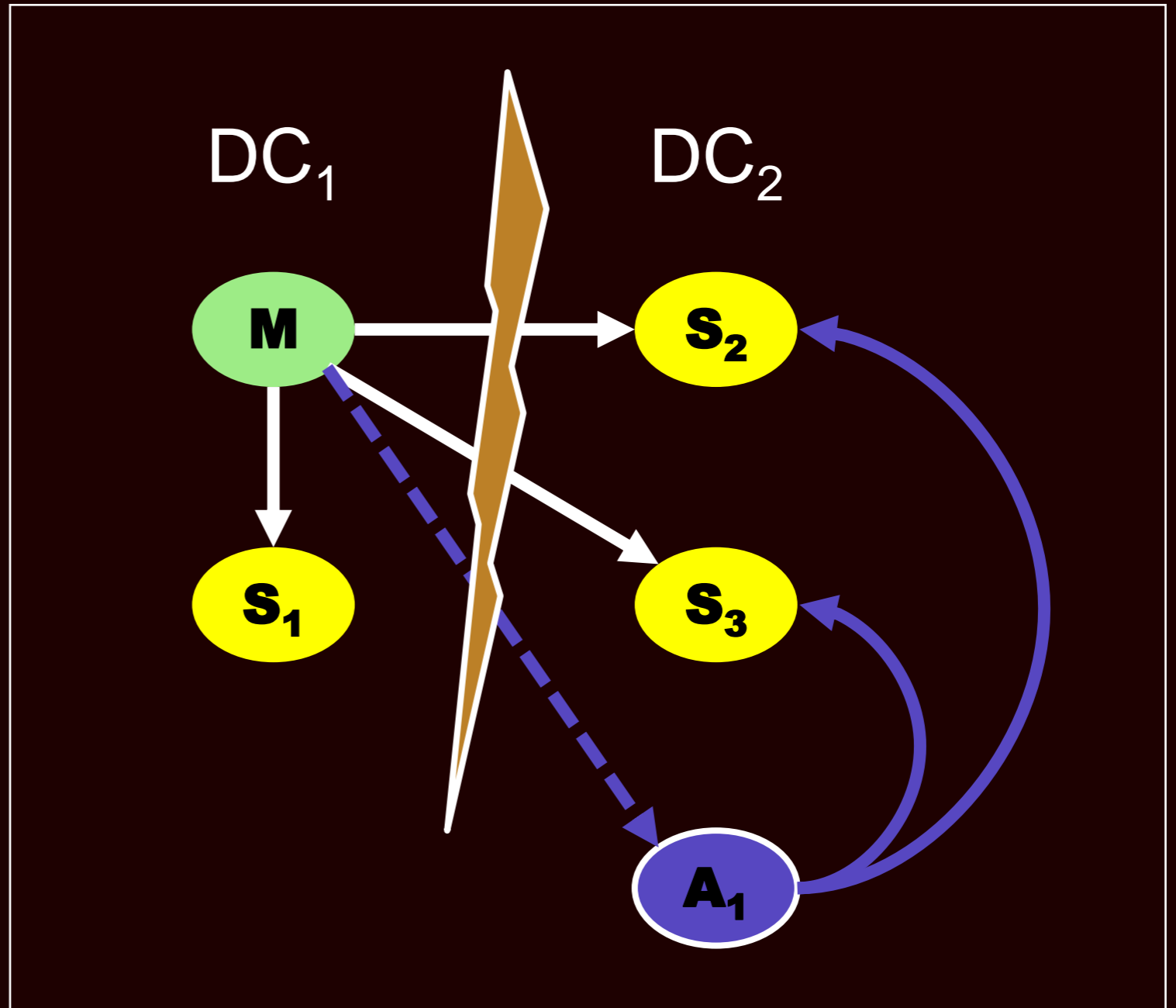


Fig 3.

# Replication



```
# demo.sh
# Create logs directory
mkdir -p logs

# Start mongod instances on ports 27017, 27018 & 27019
for p in `seq 27017 27019`; do

    # Create a data directory for each instance
    mkdir -p data/db-$p

    # Start a mongod process on each port using each data directory
    mongod --replSet demo \
        --logpath ./logs/$p.log --logappend \
        --dbpath ./data/db-$p \
        --port $p \
        --smallfiles --oplogSize 64 \
        --fork

done

# Connect and initialize the replicaSet
mongo --port 27017 < demo.js
```

# Replication



```
/* demo.js */
cfg = { _id: "demo", members:[
  { _id : 0, host : "localhost:27017"},
  { _id : 1, host : "localhost:27018"},
  { _id : 2, host : "localhost:27019"} ]
};

rs.initiate(cfg)

rs.status
```



```
demo:PRIMARY> rs.status()
{
  "set" : "demo",
  "date" : ISODate("2015-05-21T06:14:52.789Z"),
  "myState" : 1,
  "members" : [
    {
      "_id" : 0,
      "name" : "localhost:27017",
      "health" : 1,
      "state" : 1,
      "stateStr" : "PRIMARY",
      "uptime" : 89,
      "optime" : Timestamp(1432188803, 1),
      "optimeDate" : ISODate("2015-05-21T06:13:23Z"),
      "electionTime" : Timestamp(1432188807, 1),
      "electionDate" : ISODate("2015-05-21T06:13:27Z"),
      "configVersion" : 1,
      "self" : true
    },
    {
      "_id" : 1,
      "name" : "localhost:27018",
      "health" : 1,
      "state" : 2,
      "stateStr" : "SECONDARY",
      "uptime" : 89,
      "optime" : Timestamp(1432188803, 1),
      "optimeDate" : ISODate("2015-05-21T06:13:23Z"),
      "lastHeartbeat" : ISODate("2015-05-21T06:14:51.807Z"),
      "lastHeartbeatRecv" : ISODate("2015-05-21T06:14:51.806Z"),
      "pingMs" : 0,
      "configVersion" : 1
    },
    {
      "_id" : 2,
      "name" : "localhost:27019",
      "health" : 1,
      "state" : 2,
      "stateStr" : "SECONDARY",
      "uptime" : 89,
      "optime" : Timestamp(1432188803, 1),
      "optimeDate" : ISODate("2015-05-21T06:13:23Z"),
      "lastHeartbeat" : ISODate("2015-05-21T06:14:51.806Z"),
      "lastHeartbeatRecv" : ISODate("2015-05-21T06:14:51.806Z"),
      "pingMs" : 0,
      "configVersion" : 1
    }
  ],
  "ok" : 1
}
```

# Replication



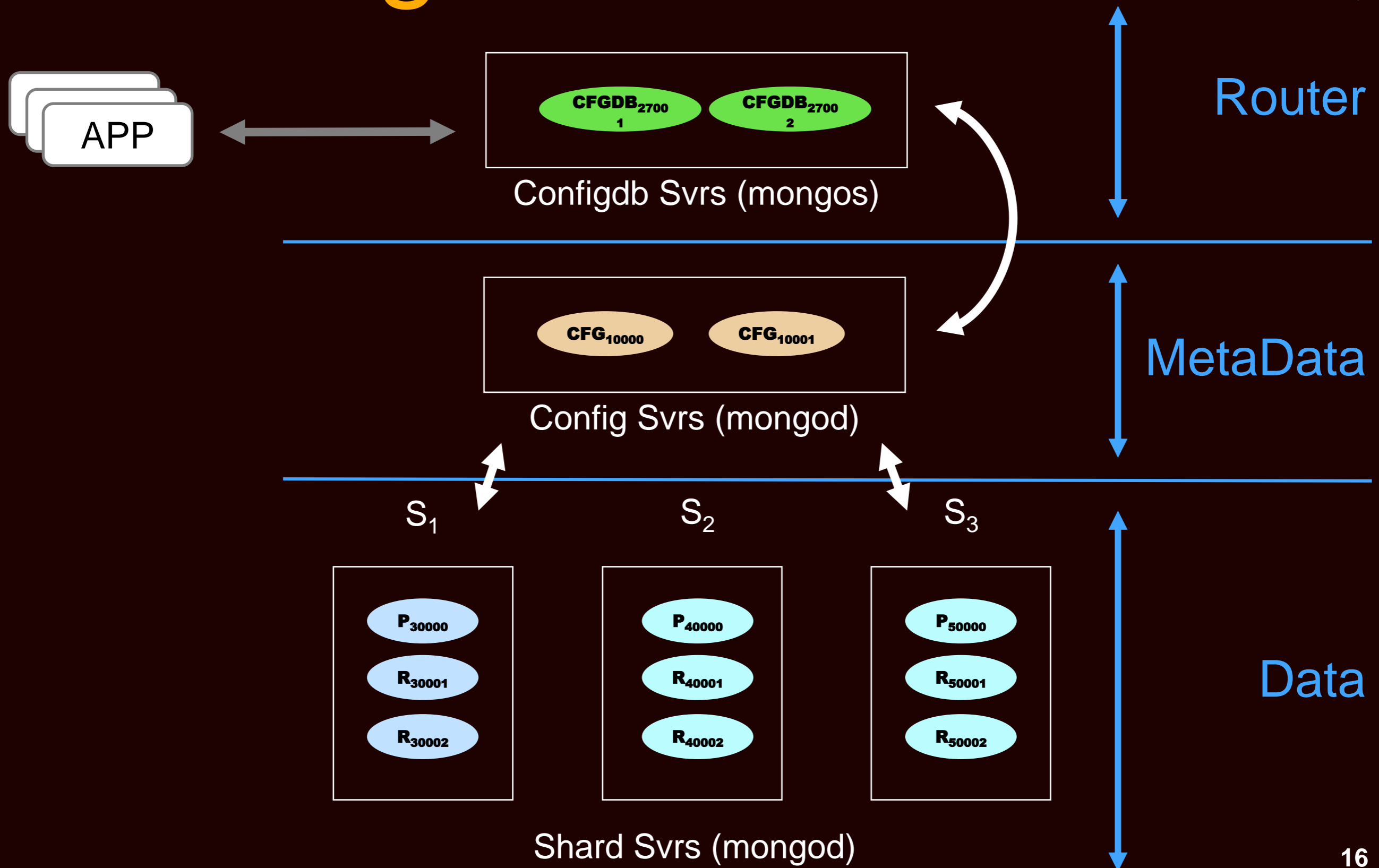
- rs commands
- local db
- oplog.rs collection

# Sharding



- Scaling a db
- Sharding
- Shard Keys
- Components
  - Config Servers: `mongod --configsvr`
  - Shard Servers: `mongod --shardsvr --replSet X`
  - `mongos --configdb`

# Sharding





# Monitoring

- mongotop
- mongostat



# Resources



- Online:
  - <http://university.mongodb.com>
  - M102 - DBA
  - M101p - Python
  - [docs.mongodb.com](http://docs.mongodb.com)
  - <http://www.w3schools.com/js>
- Books
  - MongoDB Cookbook
  - MongoDB The Definitive Guide



Thank You

agaffoor@[ebay.com](https://www.ebay.com) / [ahbaid@gmail.com](mailto:ahbaid@gmail.com)