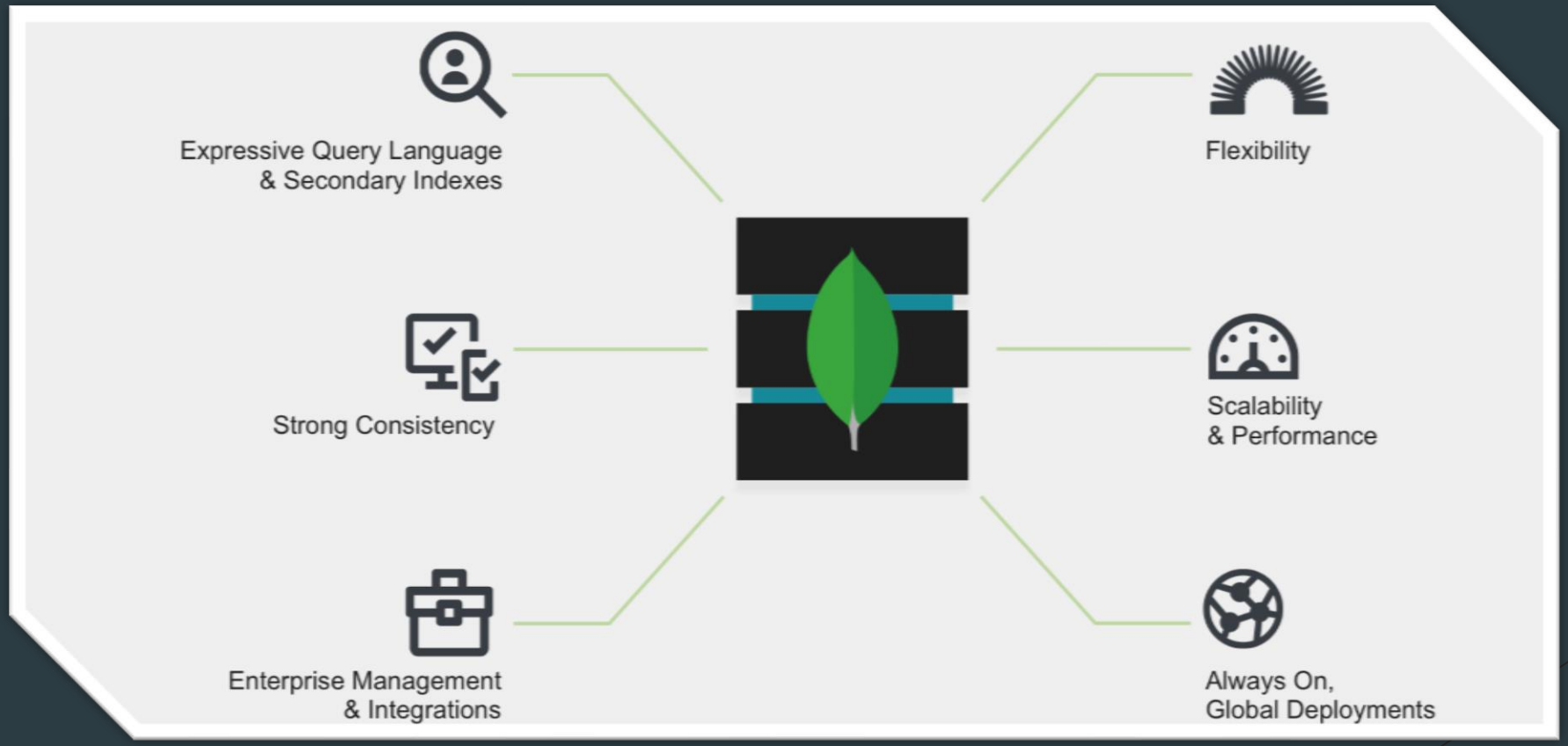
mongoDB®

Šta je MongoDB?

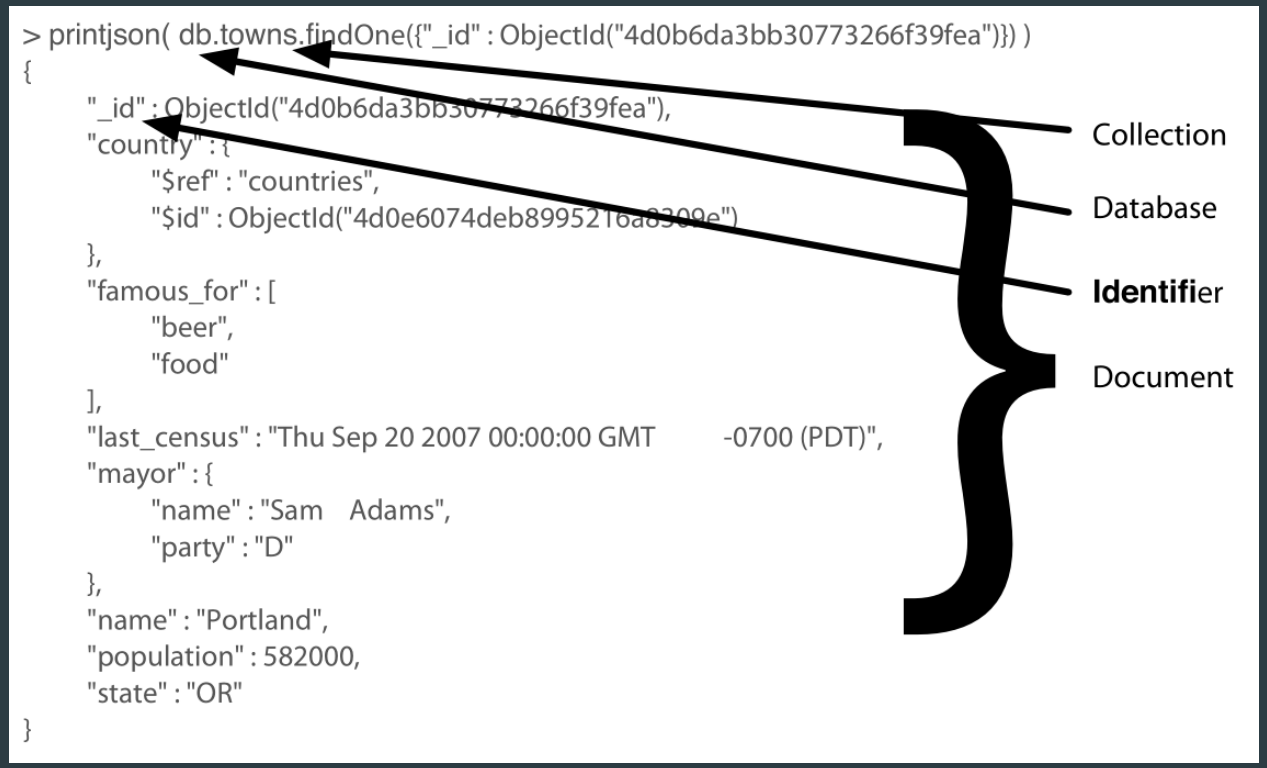
MongoDB je baza podataka namenjena za savremene aplikacije koja omogućava:

- Sisteme za ubrzanje pri konkurentnom korišćenju
- Smanjenje rizika kod kritičnih operacija
- Ubrzanje
- Izuzetno manje celokupne troškove

Nexus arhitektura



JSON



Mongo je JSON dokument baza (ipak su tehnički podaci smesteni u binarnu formu JSON-a, poznatiju kao BSON). Mongo dokument može biti posmatran kao red u tabeli relacione baze bez šeme, čije vrednosti mogu biti ugnježdene do proizvolje dubine.

Kreiranje, čitanje, ažuriranje i brisanje

- Kreiranje baze: $ mongo book
- Listanje baza: >show dbs
- Prebacivanje na bazu: >use (dbName)

Kreiranje, čitanje, ažuriranje i brisanje

- Kreiranje kolekcije:

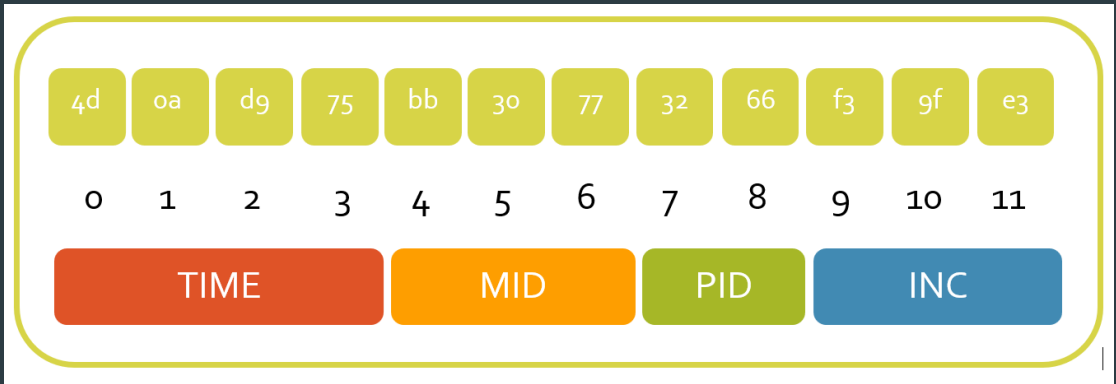
```
> db.towns.insert({
    name: "New York",
    population: 22200000,
    last_census: ISODate("2009-07-31"),
    famous_for: [ "statue of liberty", "food" ],
    mayor : {
        name : "Michael Bloomberg",
        party : "I"
    }
})
```

Kreiranje, čitanje, ažuriranje i brisanje

- Listanje kolekcija: show collections
- Pretraživanje kolekcija:

```
> db.towns.find()
{
    "_id" : ObjectId("4d0ad975bb30773266f39fe3"),
    "name" : "New York",
    "population": 22200000,
    "last_census": "Fri Jul 31 2009 00:00:00 GMT-0700 (PDT)",
    "famous_for" : [ "statue of liberty", "food" ],
    "mayor" : { "name" : "Michael Bloomberg", "party" : "I" }
}
```

Kreiranje, čitanje, ažuriranje i brisanje



- polje „_id“ (12 bajtova):
  - Vremenski žig
  - ID klientske mašine
  - ID klientskog procesa
  - Uvećavajući brojač

Kreiranje, čitanje, ažuriranje i brisanje

- Pretraga:
  - db.towns.find({ "_id" : ObjectId("4d0ada1fbb30773266f39fe4") })
  - db.towns.find({ _id : ObjectId("4d0ada1fbb30773266f39fe4") }, { name : 1 })
  - db.towns.find({ _id : ObjectId("4d0ada1fbb30773266f39fe4") }, { name : 0 })
  - db.towns.find(

    { name : /^P/, population : { $lt : 10000 } },

    { name : 1, population : 1 }

)

Kreiranje, čitanje, ažuriranje i brisanje

- Pretraga:

```
var population_range = {}
population_range['$lt'] = 1000000
population_range['$gt'] = 10000
db.towns.find(
    { name : /^P/, population : population_range },
    { name: 1 }
)
```

Kreiranje, čitanje, ažuriranje i brisanje

- Pretraga po nizu:
  - db.towns.find(
    { famous_for : 'food' },
    { _id : 0, name : 1, famous_for : 1 }
)
  - db.towns.find(
    { famous_for : /statue/ },
    { _id : 0, name : 1, famous_for : 1 }
)
  - db.towns.find(
    { famous_for : { $all : ['food', 'beer'] } },
    { _id : 0, name:1, famous_for:1 }
)
  - db.towns.find(
    { famous_for : { $nin : ['food', 'beer'] } },
    { _id : 0, name : 1, famous_for : 1 }
)

Kreiranje, čitanje, ažuriranje i brisanje

- Kopanje u dubinu:
  - db.towns.find(
    { 'mayor.party' : 'I' },
    { _id : 0, name : 1, mayor : 1 }
)
  - db.towns.find(
    { 'mayor.party' : { $exists : **false** } },
    { _id : 0, name : 1, mayor : 1 }
)

Kreiranje, čitanje, ažuriranje i brisanje

- Ažuriranje

```
db.towns.update(
    { _id :
ObjectId("4d0ada87bb30773266f39fe5")
},
    { $set : { "state" : "OR" } }
);
```

- Brisanje

```
var bad_bacon = {
    'exports.foods' : {
        $elemMatch : {
            name : 'bacon',
            tasty : false
        }
    }
}
db.countries.remove( bad_bacon )
```

Indeksi

- B-stabla
- Dvodimenzionalni geografski indeksi
- Sferični geografski indeksi

Indeksi

```
populatePhones = function(area,start,stop) {
    for(var i=start; i < stop; i++) {
        var country = 1 + ((Math.random() * 8) << 0);
        var num = (country * 1e10) + (area * 1e7) + i;
        db.phones.insert({
            _id: num,
            components: {
                country: country,
                area: area,
                prefix: (i * 1e-4) << 0,
                number: i
            },
            display: "+" + country + " " + area + "-" + i
        });
    }
}
populatePhones( 800, 5550000, 5650000 )
```

Indeksi

- Listanje indeksa

```
db.phones.getIndexes()
[
    {
        "v" : 2,
        "key" : {
            "_id" : 1
        },
        "name" : "_id_",
        "ns" : "book.phones"
    }
]
```

Indeksi

- Vreme izvršavanja

```
db.phones.find({display: "+1 800-5650001"}).explain("executionStats")

{...
"executionStages" : {
        "stage" : "FETCH",
        "nReturned" : 1,
        "executionTimeMillisEstimate" : 54,
        "works" : 2,
...}
```

Indeksi

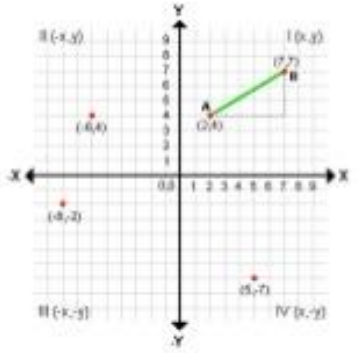
- Podizanje indeksa

```
db.phones.ensureIndex(
    { display : 1 },
    { unique : true, dropDups : true }
)
```

Indeksi

Surface Types

Flat



2d Indexes

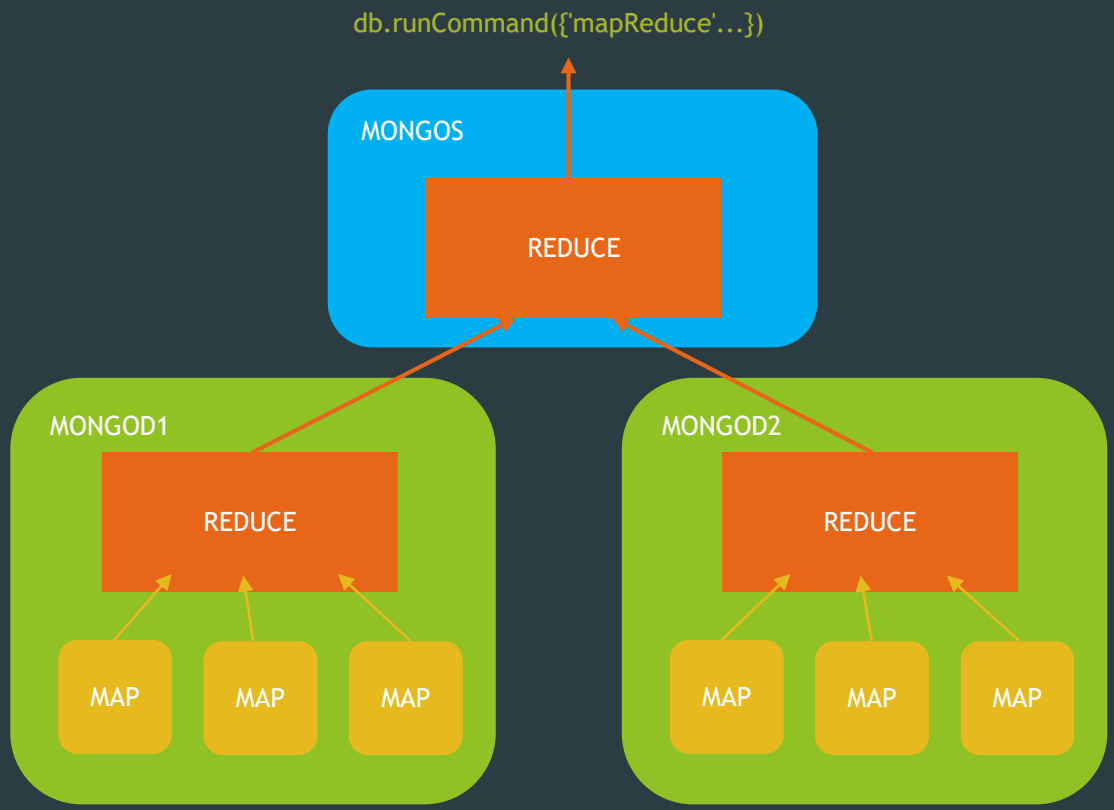
Spherical

2dsphere Indexes

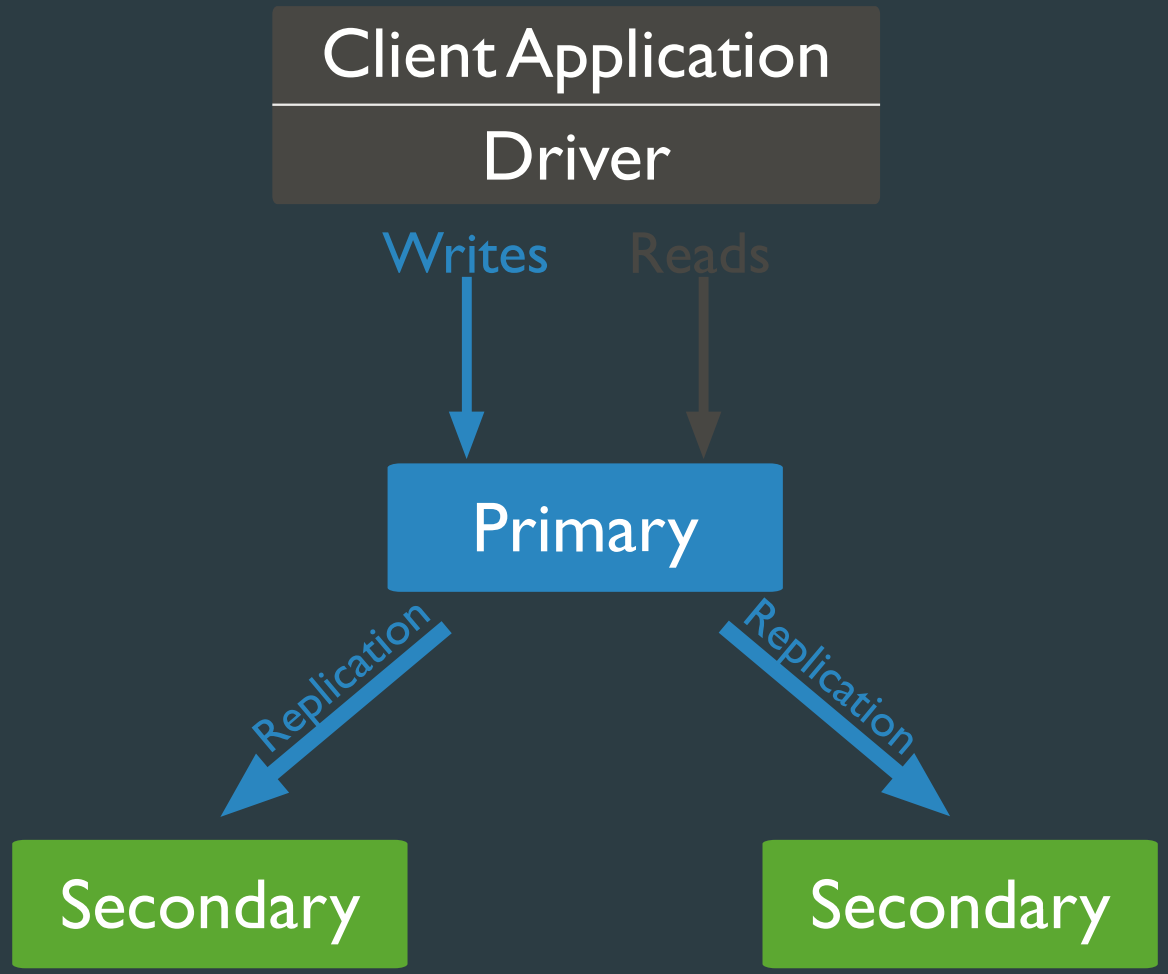
Agregacioni upiti

- populatePhones(855, 5550000, 5650000)
- db.phones.group({
  initial: { count:0 },
  reduce: function(phone, output) { output.count++; },
  cond: { 'components.number': { $gt : 5599999 } },
  key: { 'components.area' : **true** }
  })
- db.phones.group({
  initial: { count:0 },
  reduce: function(phone, output) { output.count++; },
  cond: { 'components.number': { $gt : 5599999 } }
  })

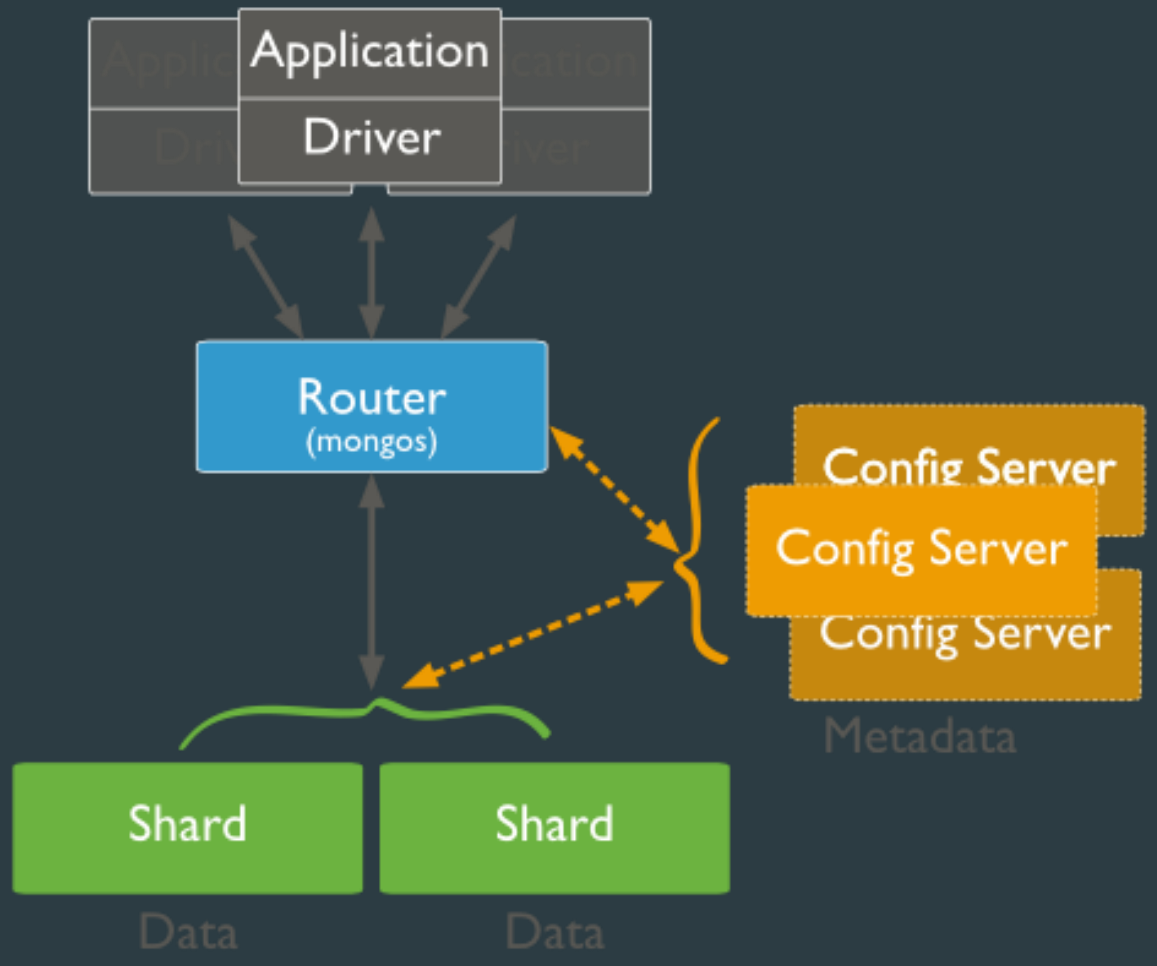
Mapreduce


db.runCommand({'mapReduce'...})
MONGOS
REDUCE
MONGOD1
REDUCE
MAP
MAP
MAP
MONGOD2
REDUCE
MAP
MAP
MAP

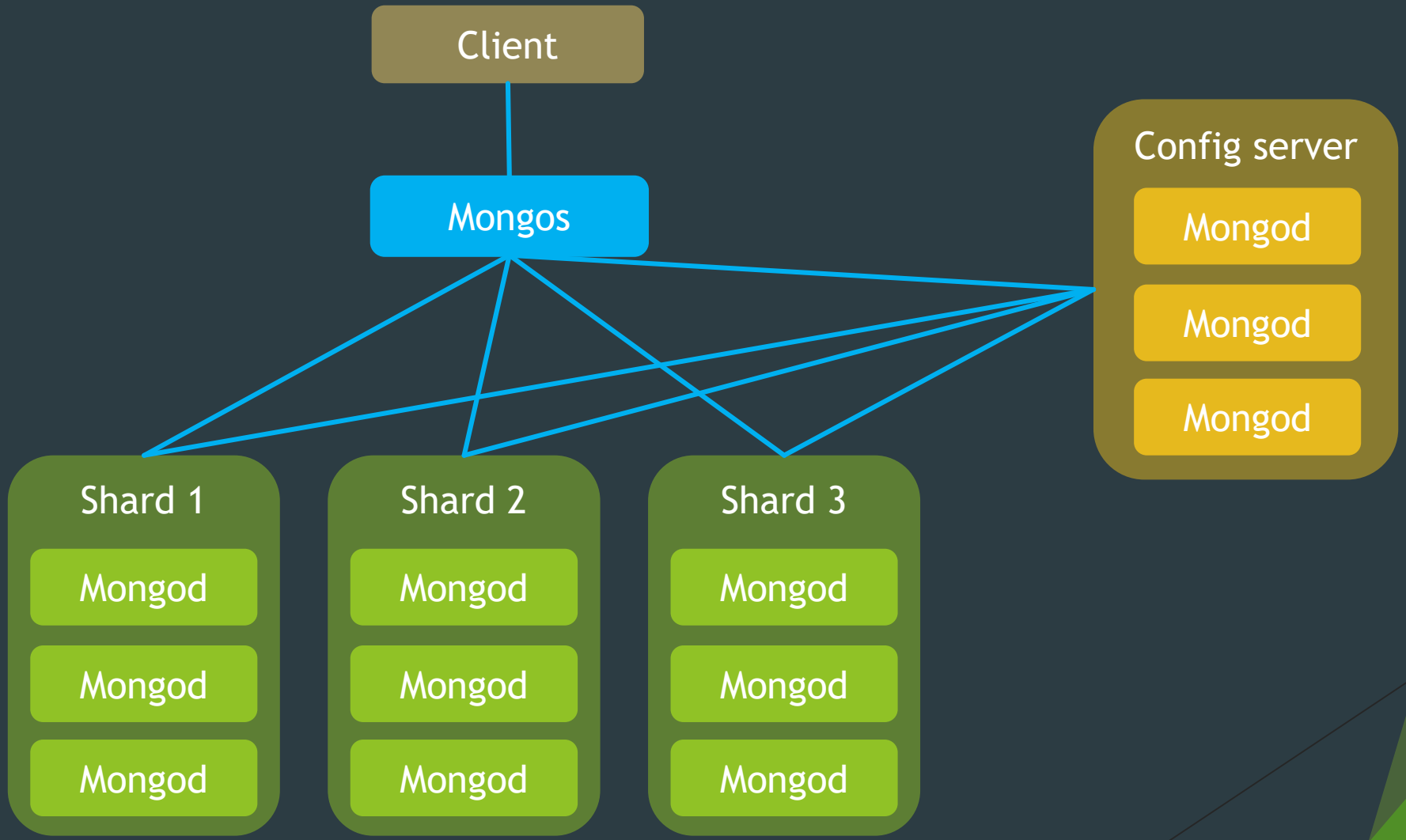
Replicirajući setovi



Particionisanje



Replicirajući setovi + Particionisanje



Hvala na pažnji.