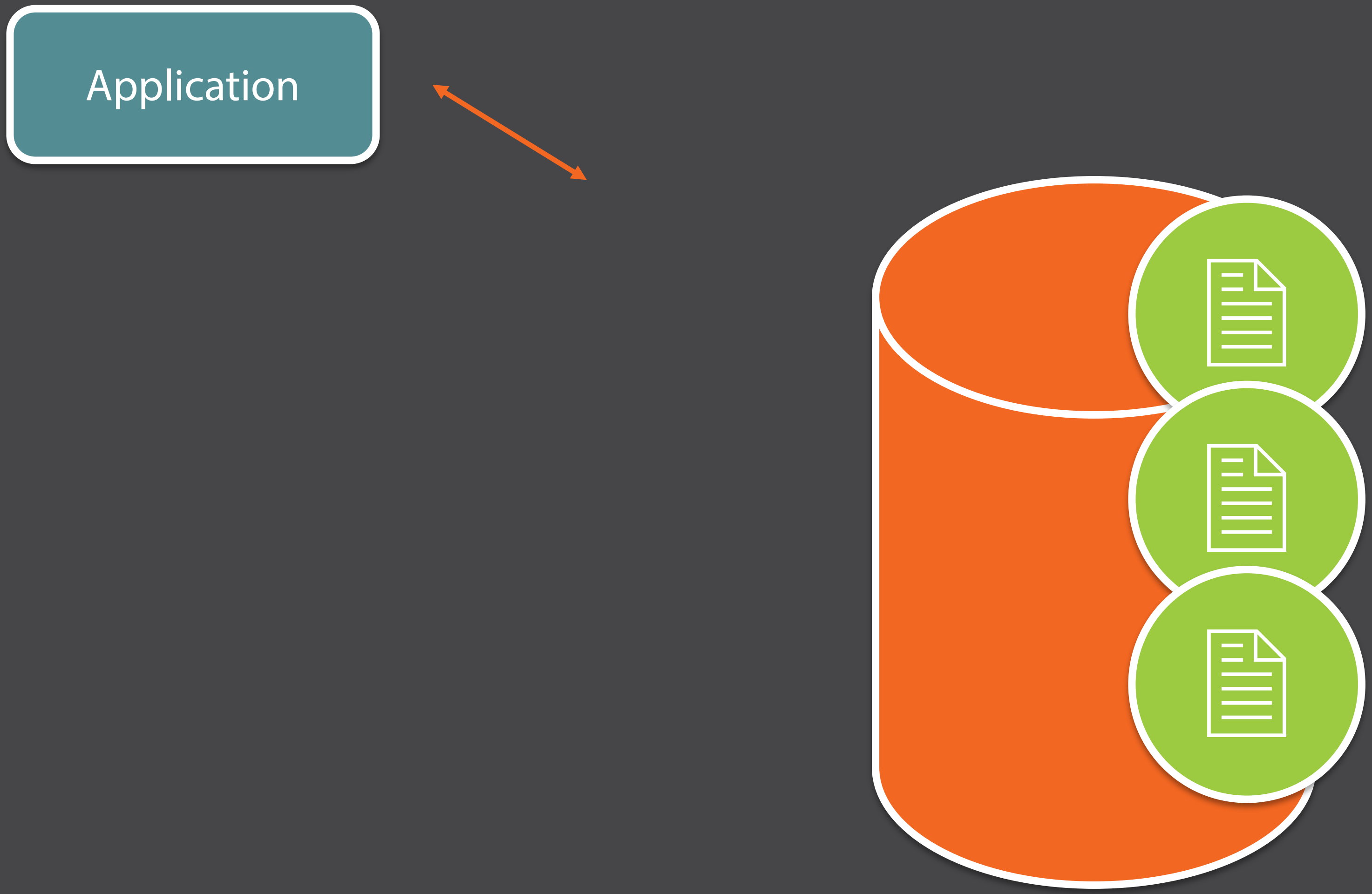
Sharding

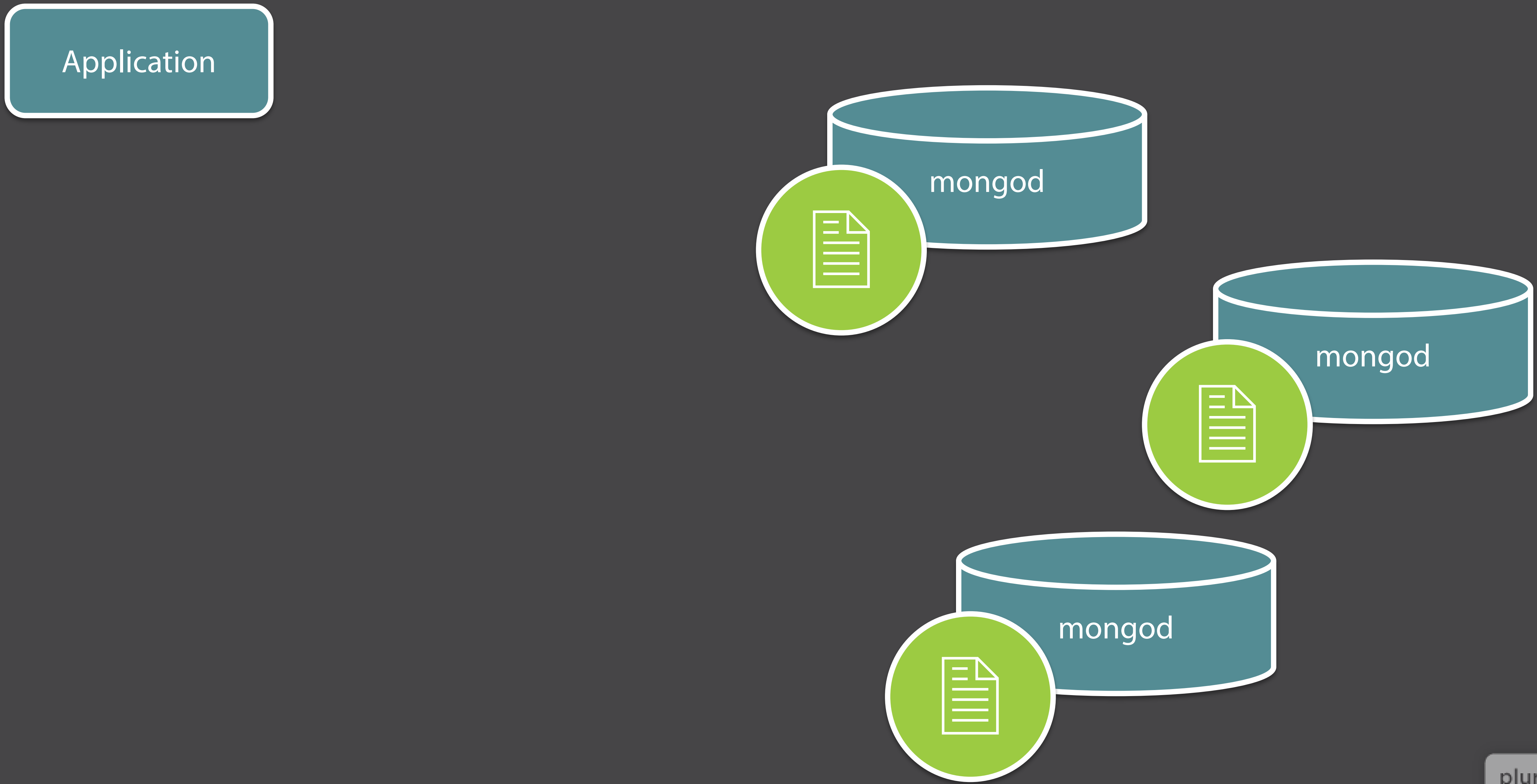
Nuri Halperin

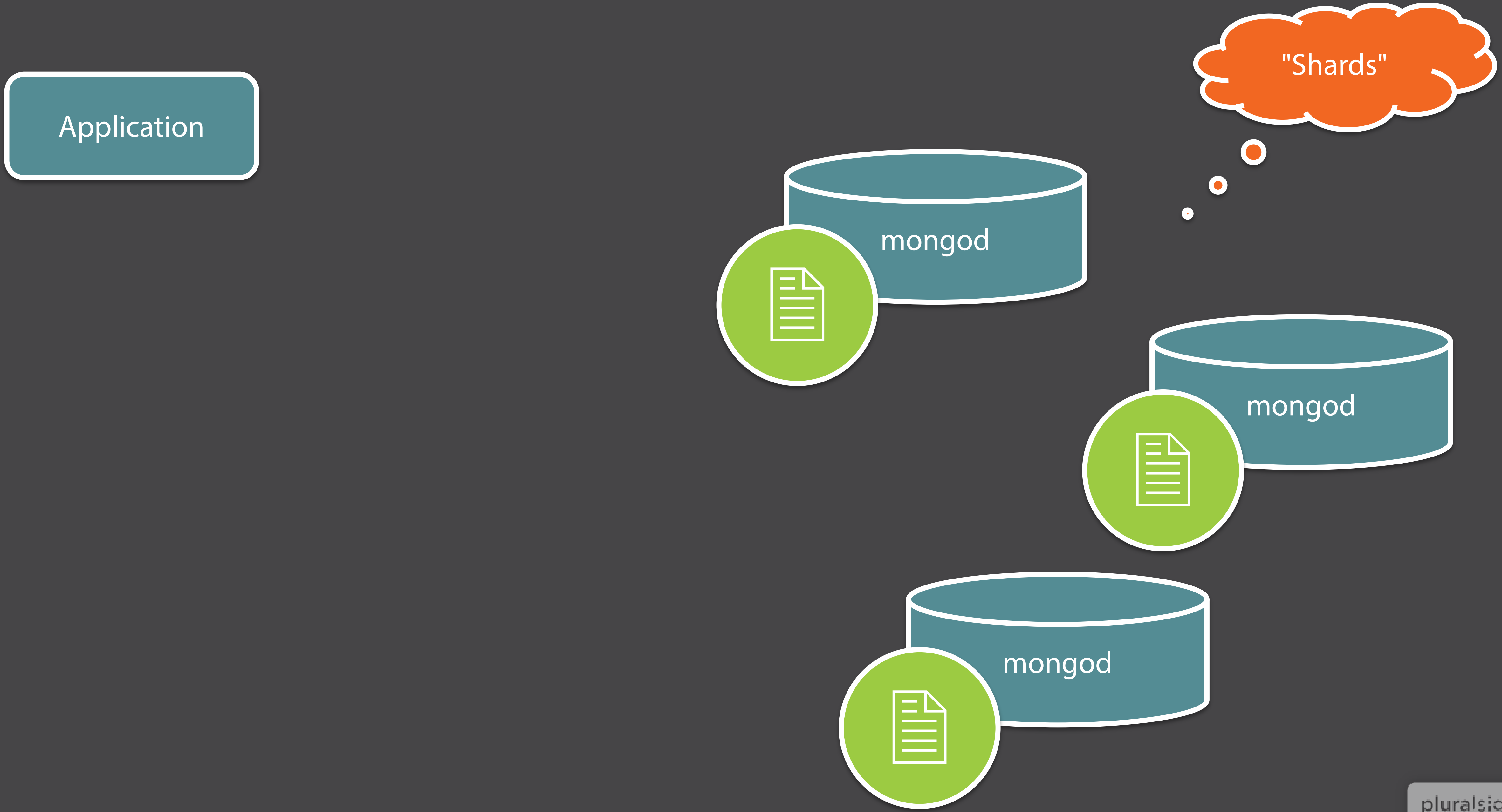
@nurih | www.plusnconsulting.com

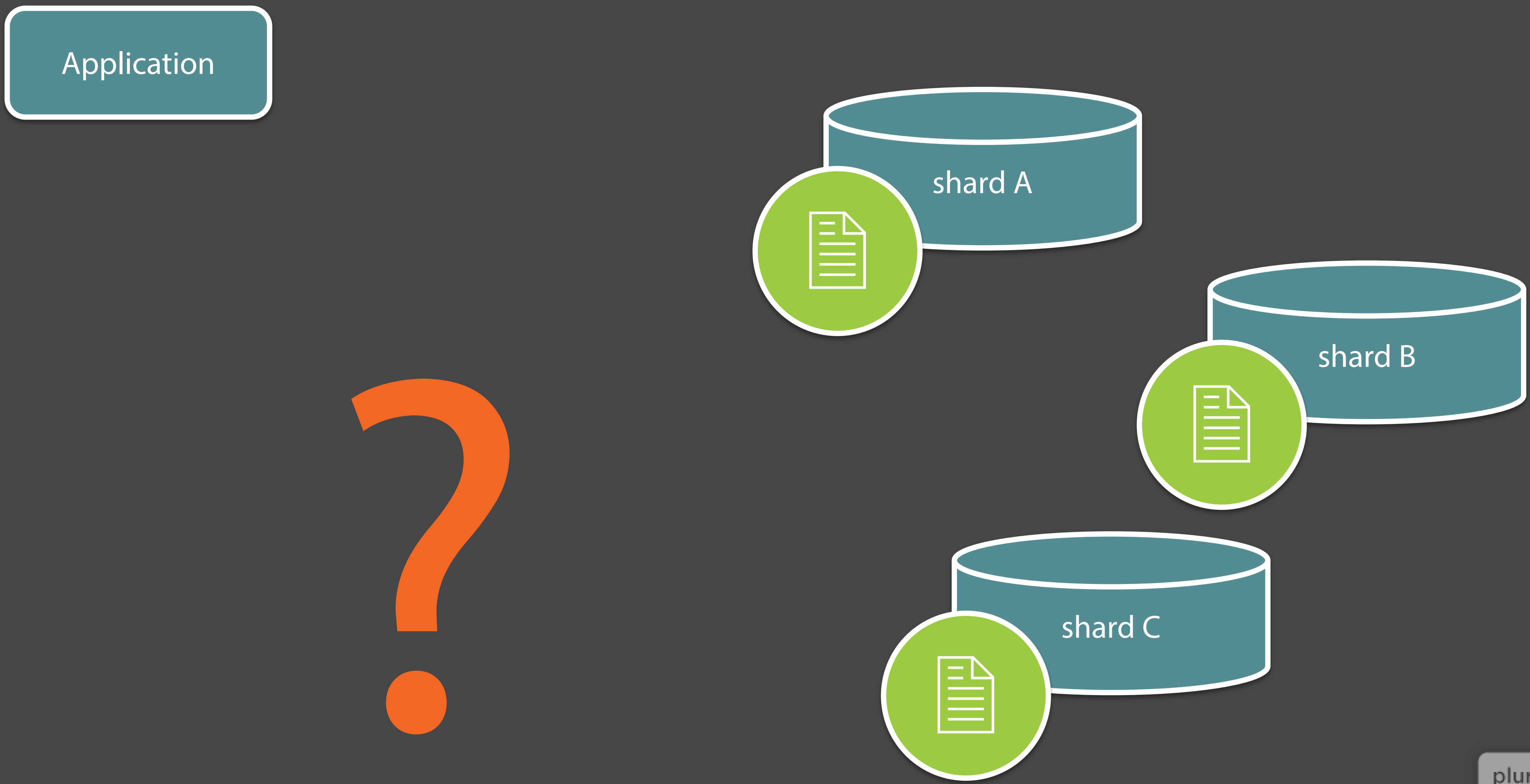
Not Sharded



Sharded




Application
"Shards"
mongod
mongod
mongod

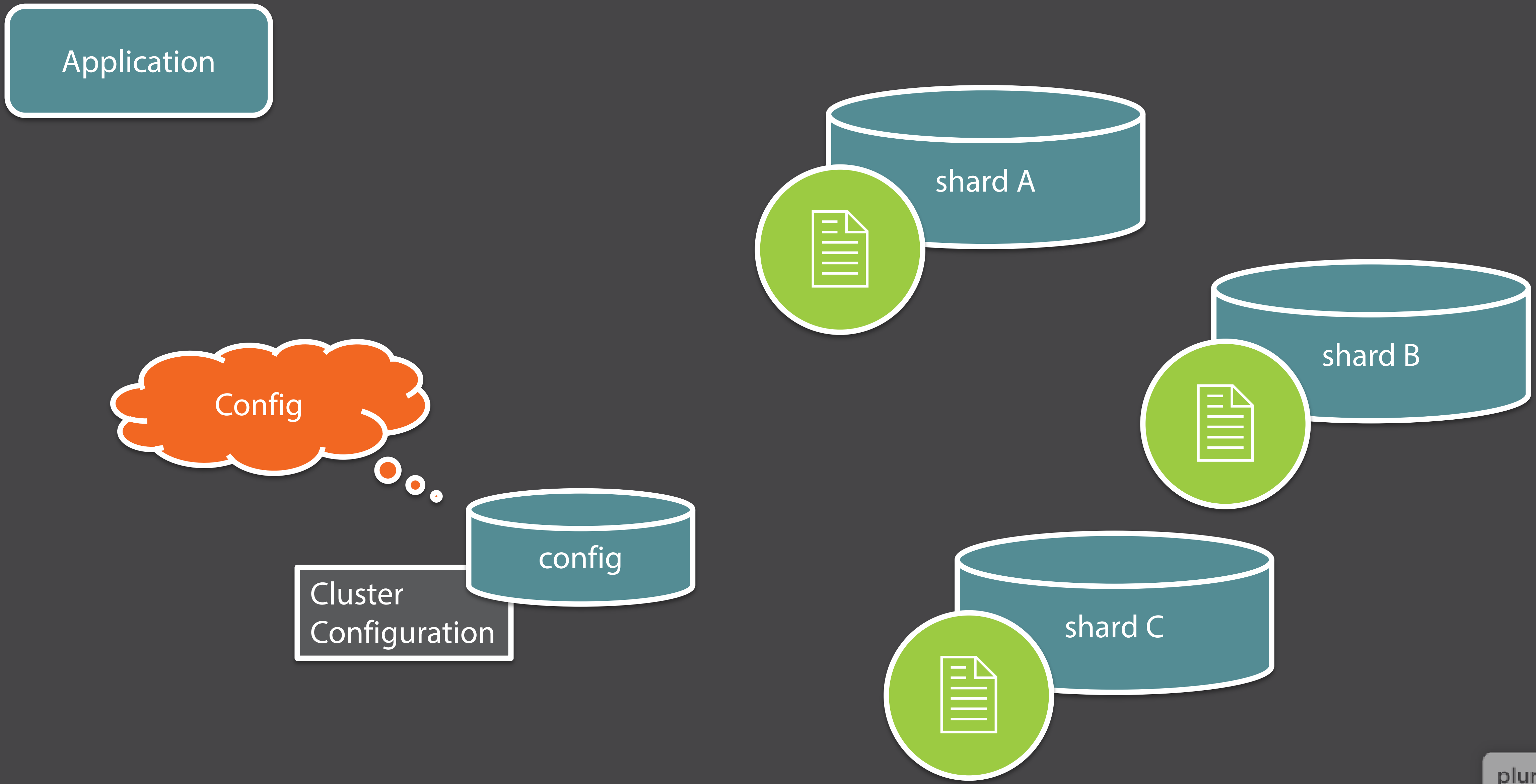


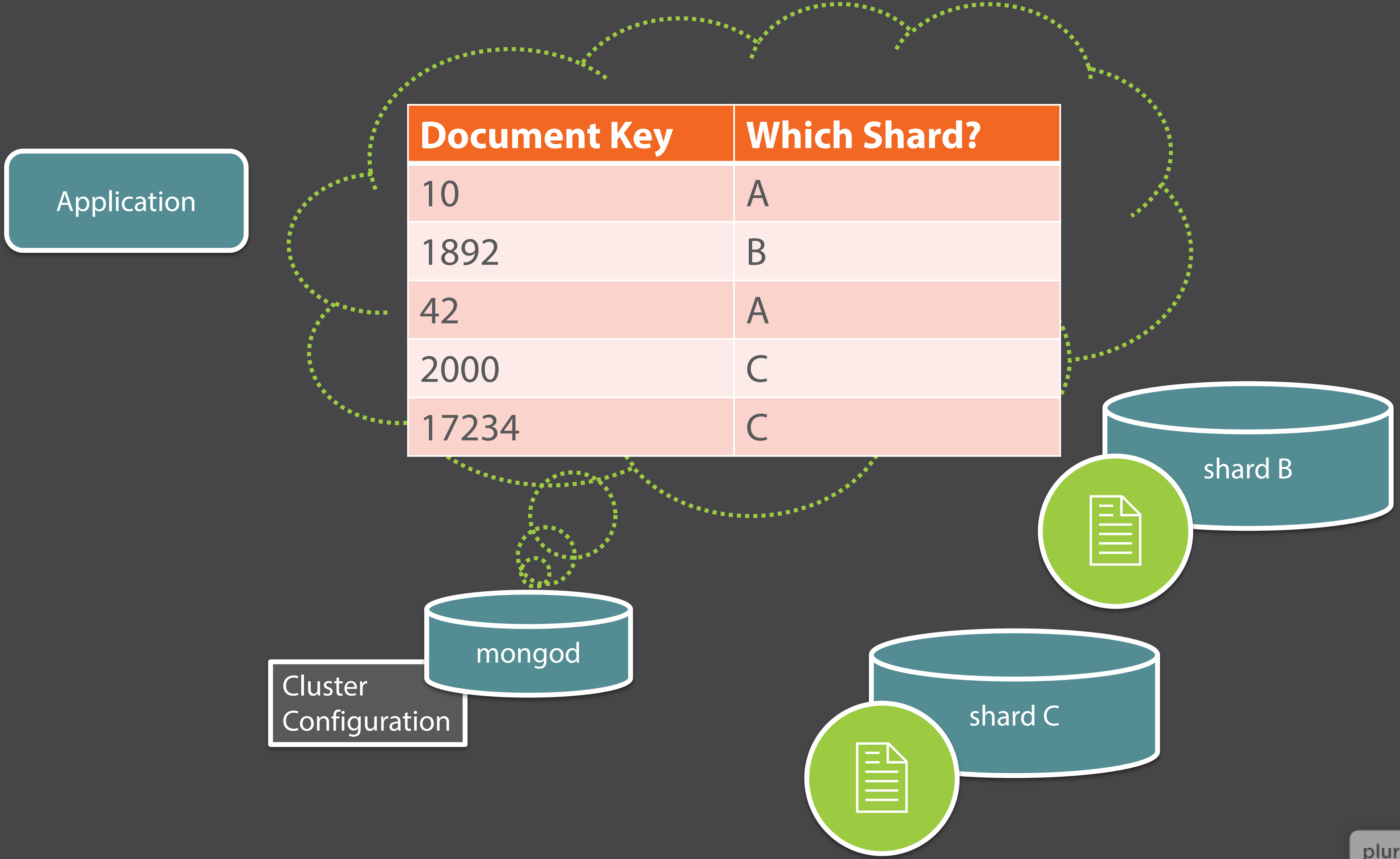
Application

shard A

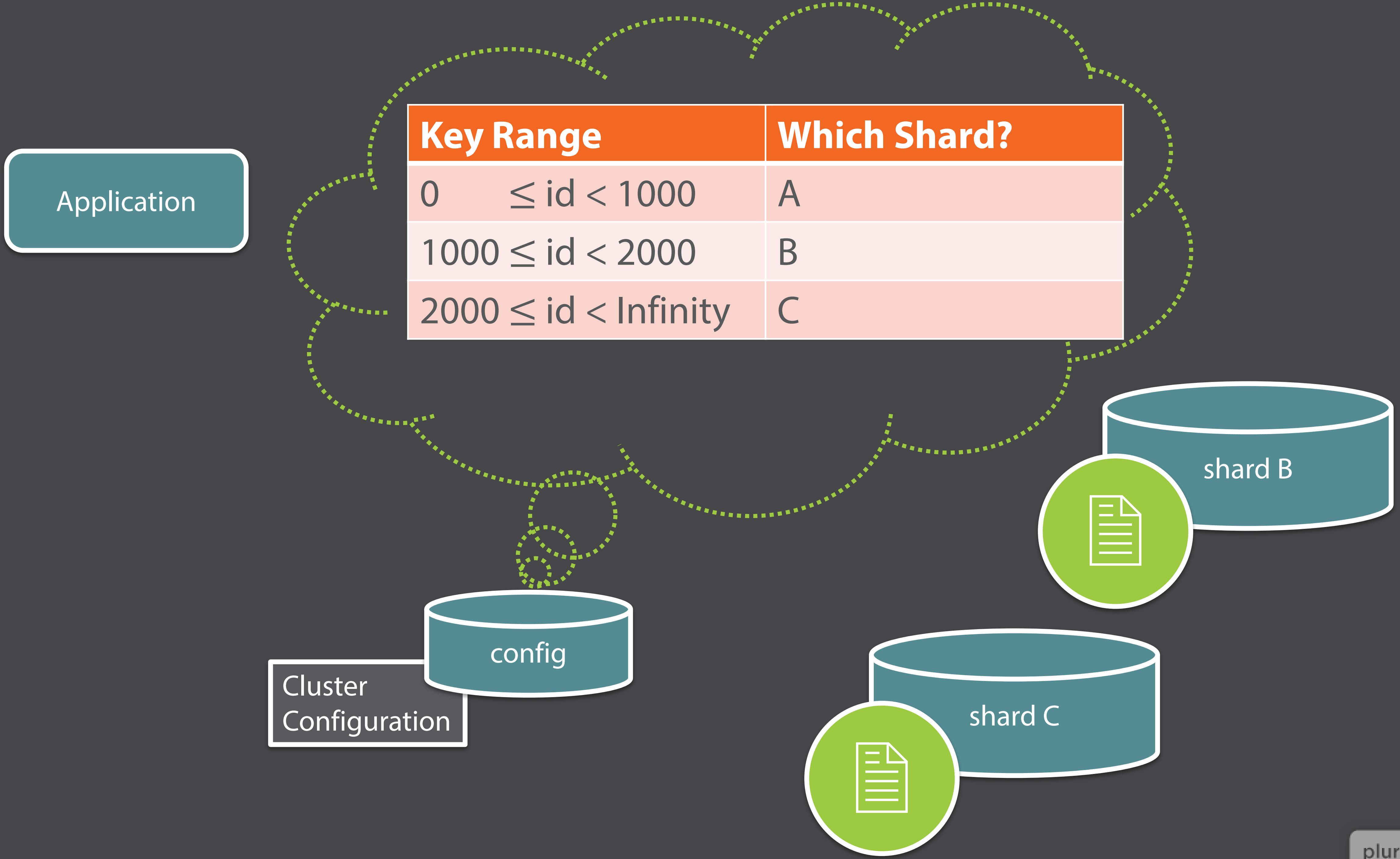
shard B

shard C

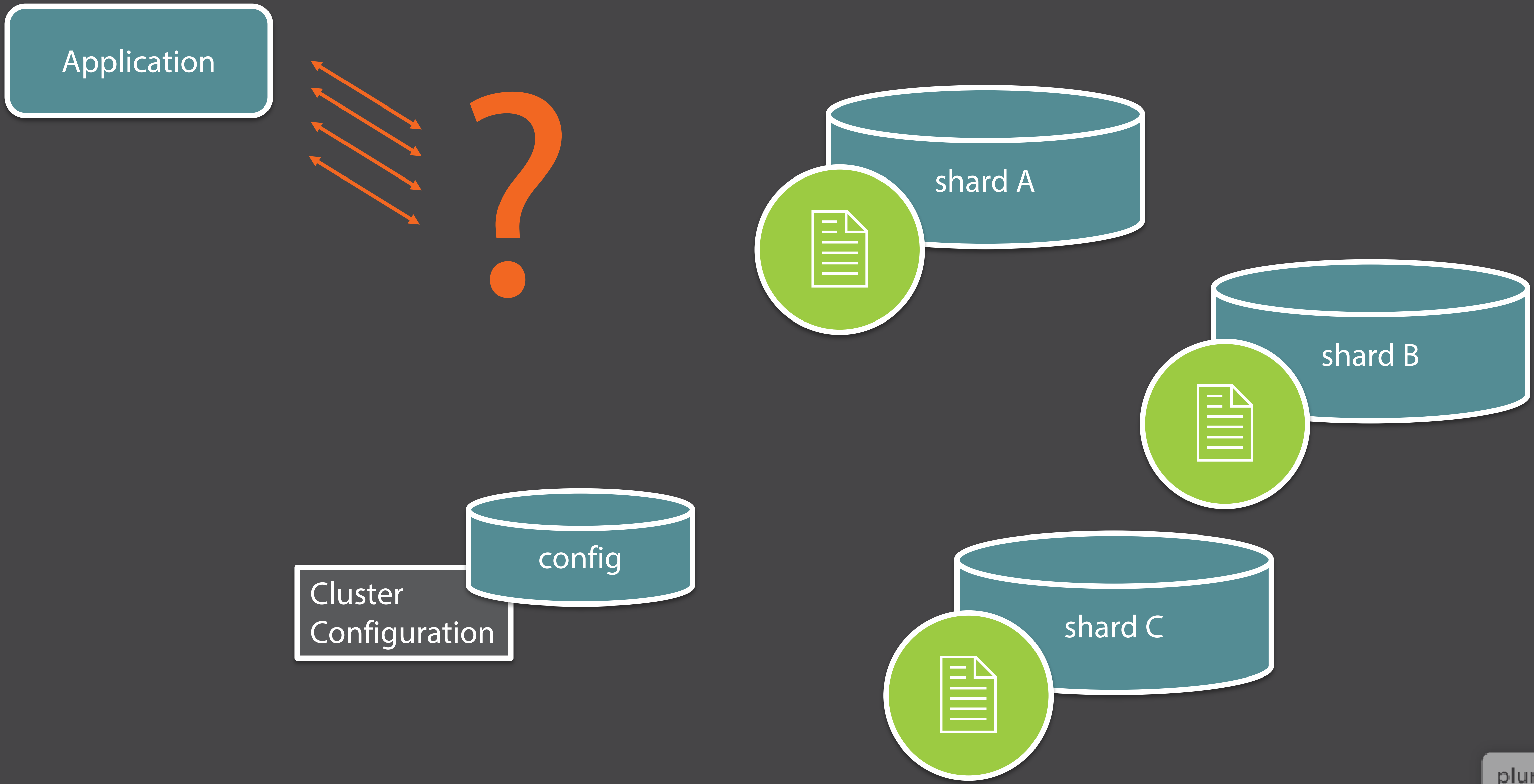

Application
shard A
shard B
shard C
Config
config
Cluster Configuration

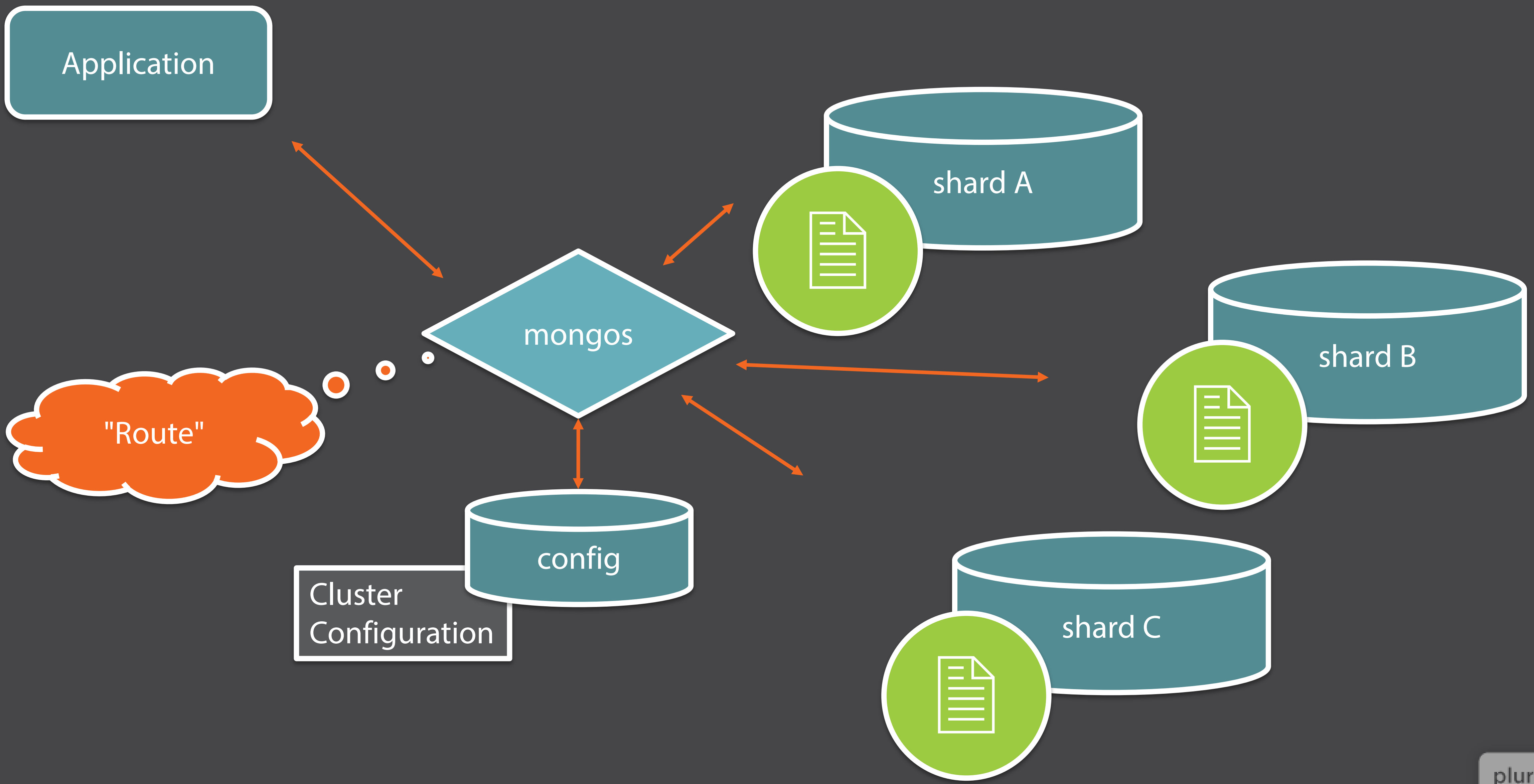


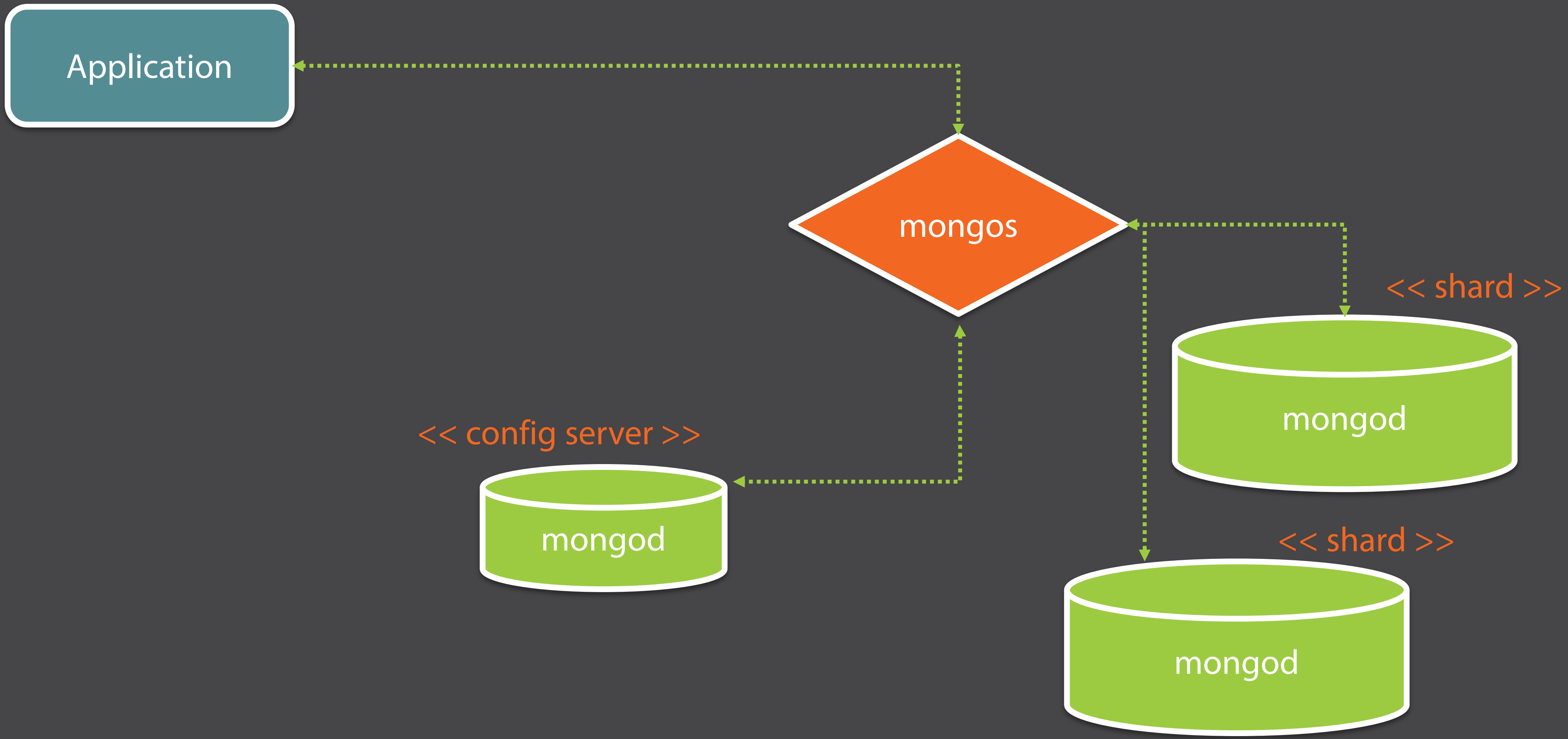
Document Key	Which Shard?
10	A
1892	B
42	A
2000	C
17234	C



Key Range	Which Shard?
$0 \leq \text{id} < 1000$	A
$1000 \leq \text{id} < 2000$	B
$2000 \leq \text{id} < \text{Infinity}$	C


Application
?
shard A
shard B
config
Cluster Configuration
shard C

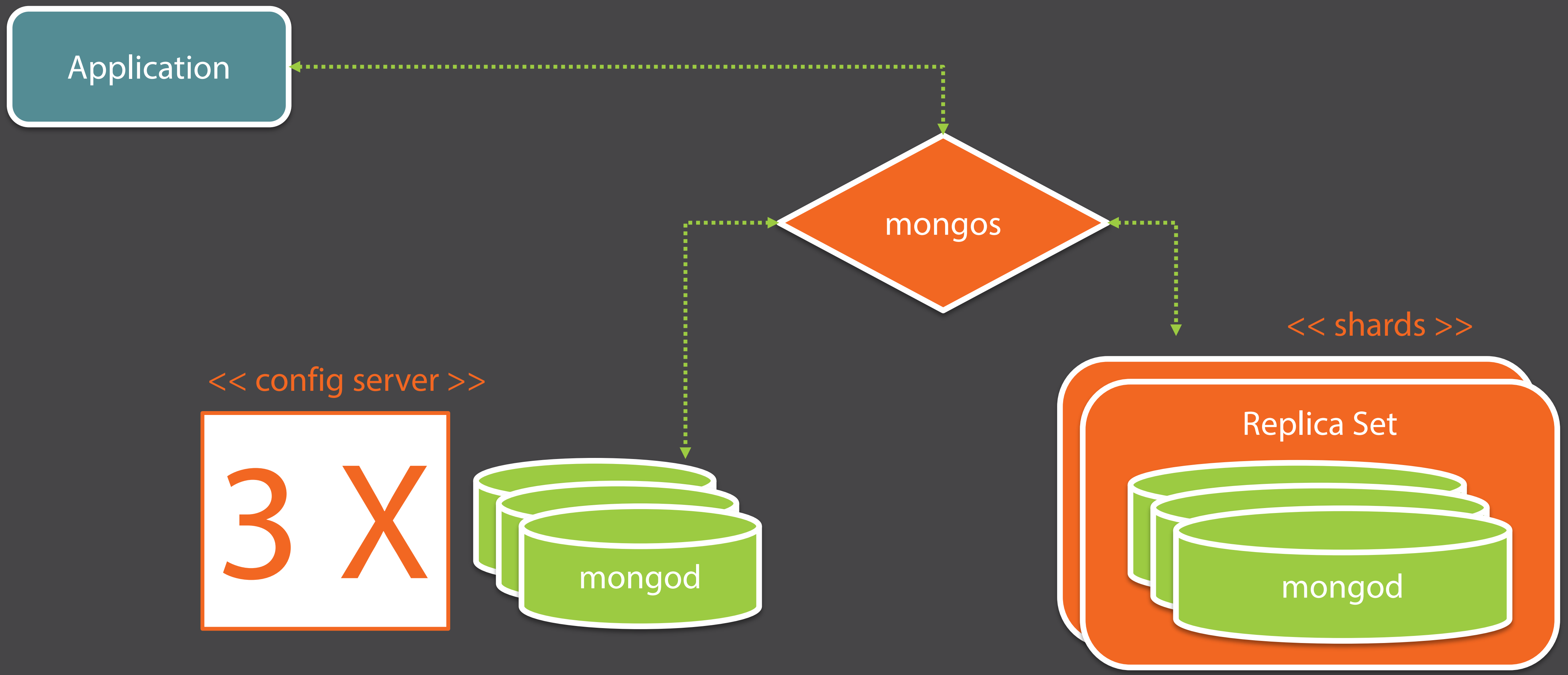

Application
mongos
"Route"
config
Cluster Configuration
shard A
shard B
shard C

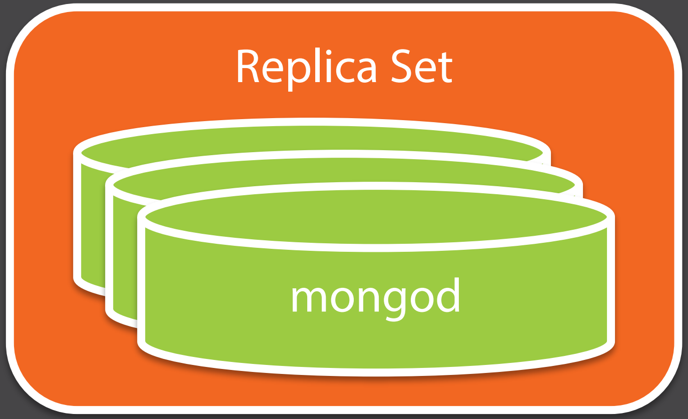

Application
mongos
<< shard >>
mongod
<< config server >>
mongod
<< shard >>
mongod

At a Glance

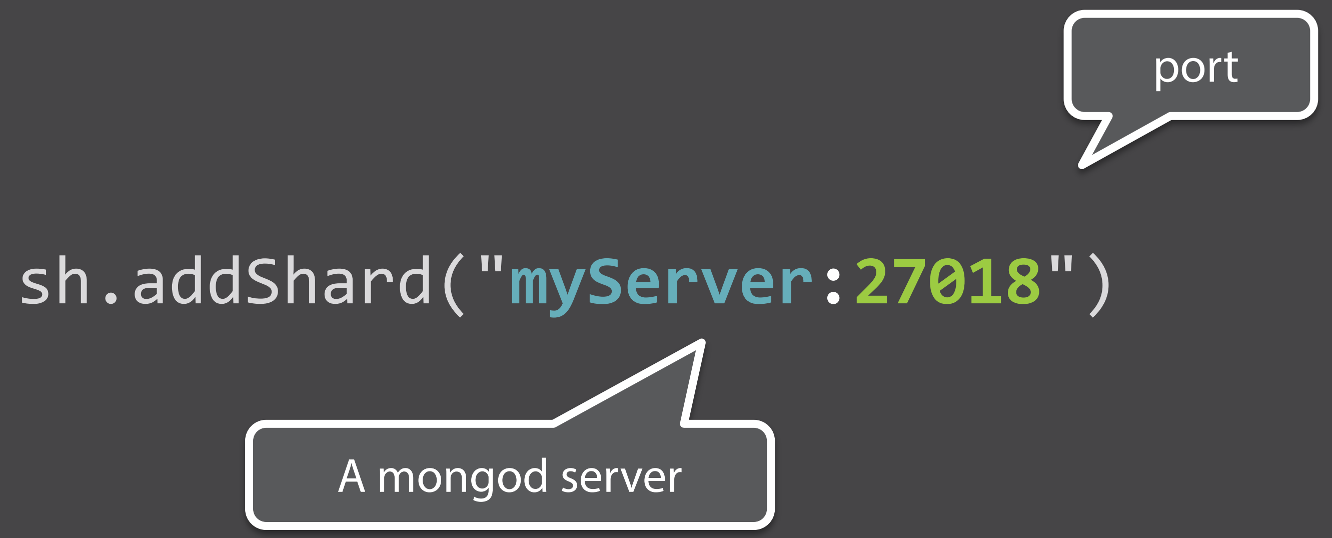
```
mongos --configdb configServer


// inside the shell:
mongos> sh.addShard('mongodServer:port')


mongos> sh.enableSharding('dbName')


mongos> sh.shardCollection('dbName.coll', {'field':1})
```

> Production Gig?
>
> Three Config!
>
> \- Me


Application
mongos
<< config server >>
3 X
mongod
<< shards >>
Replica Set
mongod



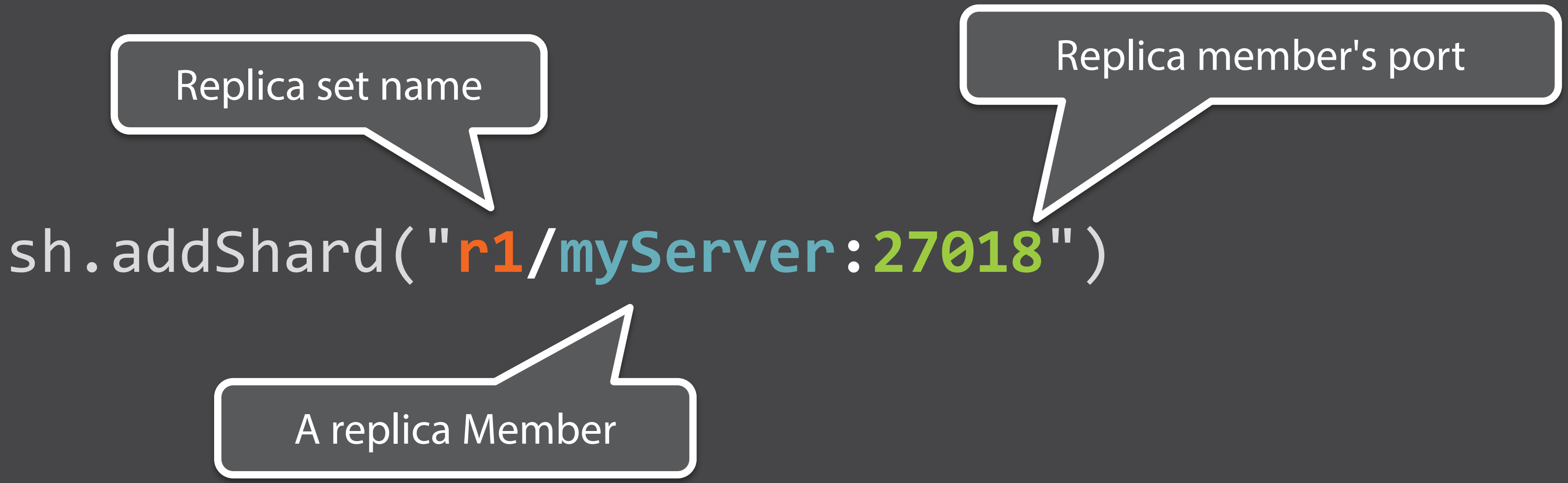
- Durability
- Availability



```
sh.addShard("myServer:27018")
```

Adding a Shard

- Specify server + port



```
sh.addShard("r1/myServer:27018")
```

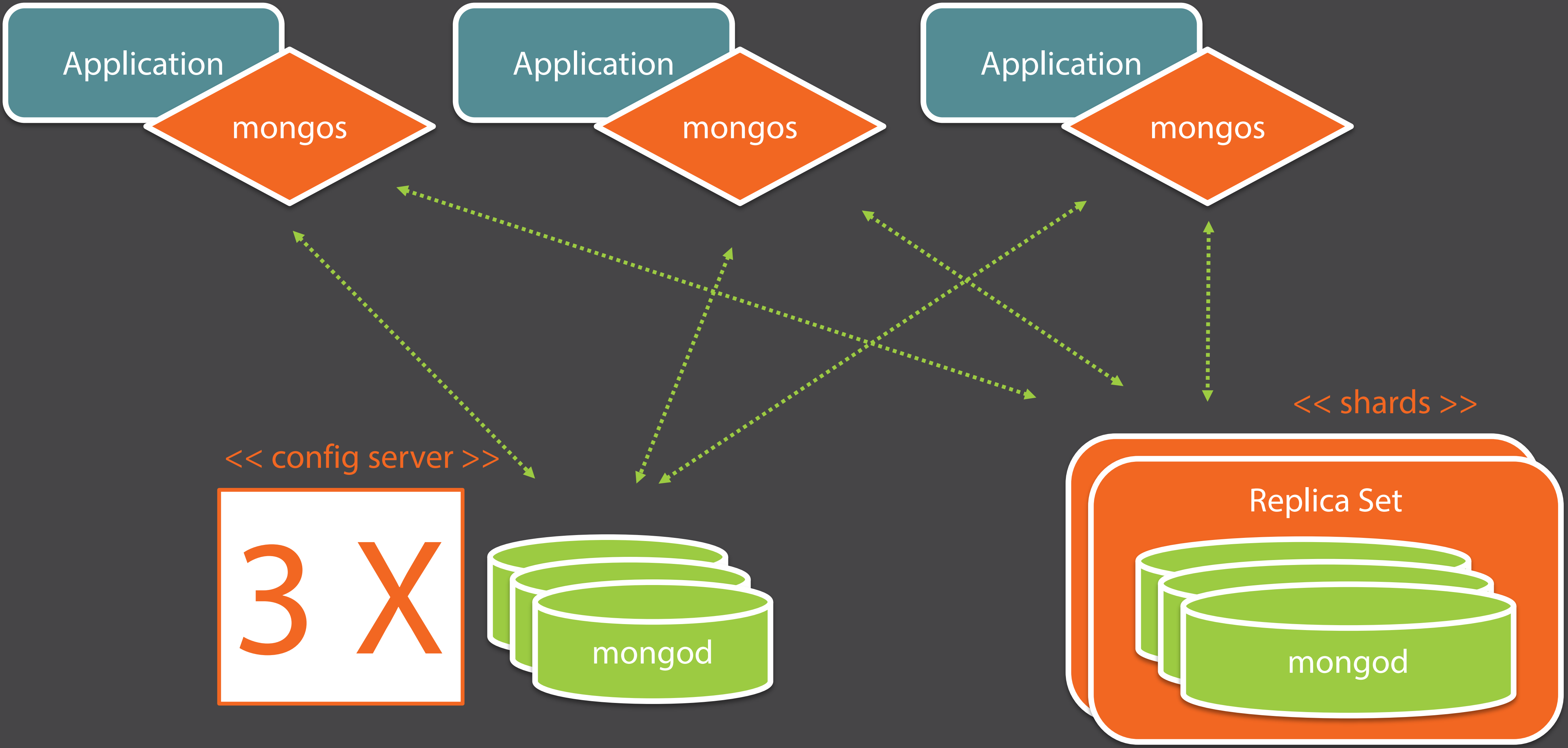
Replica set name

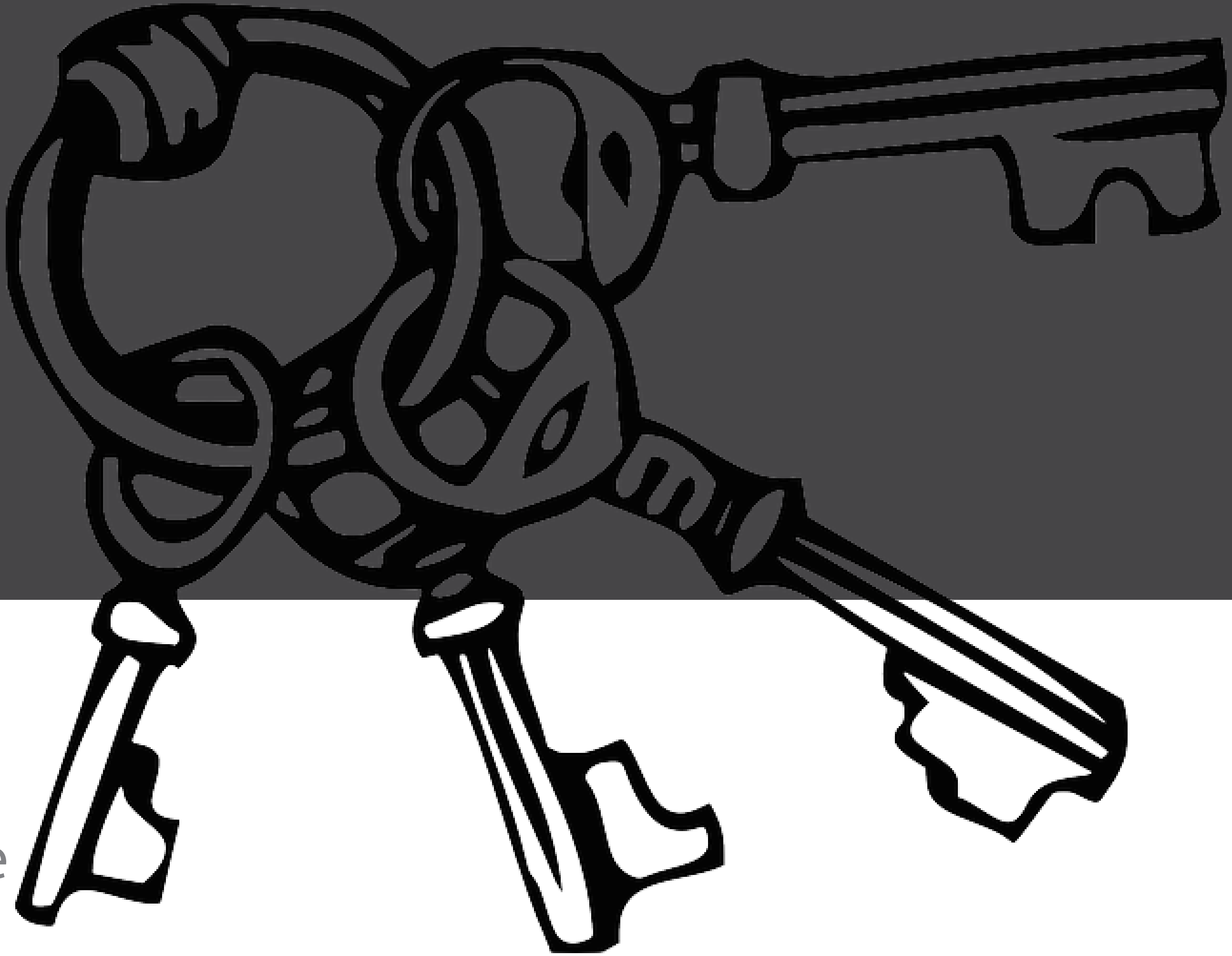
Replica member's port

A replica Member

Adding a Replica Set as a Shard

- Prefix with replica set name
- Specify one of the replica set member + port
- The rest of the members in the replica set are discovered & registered as well

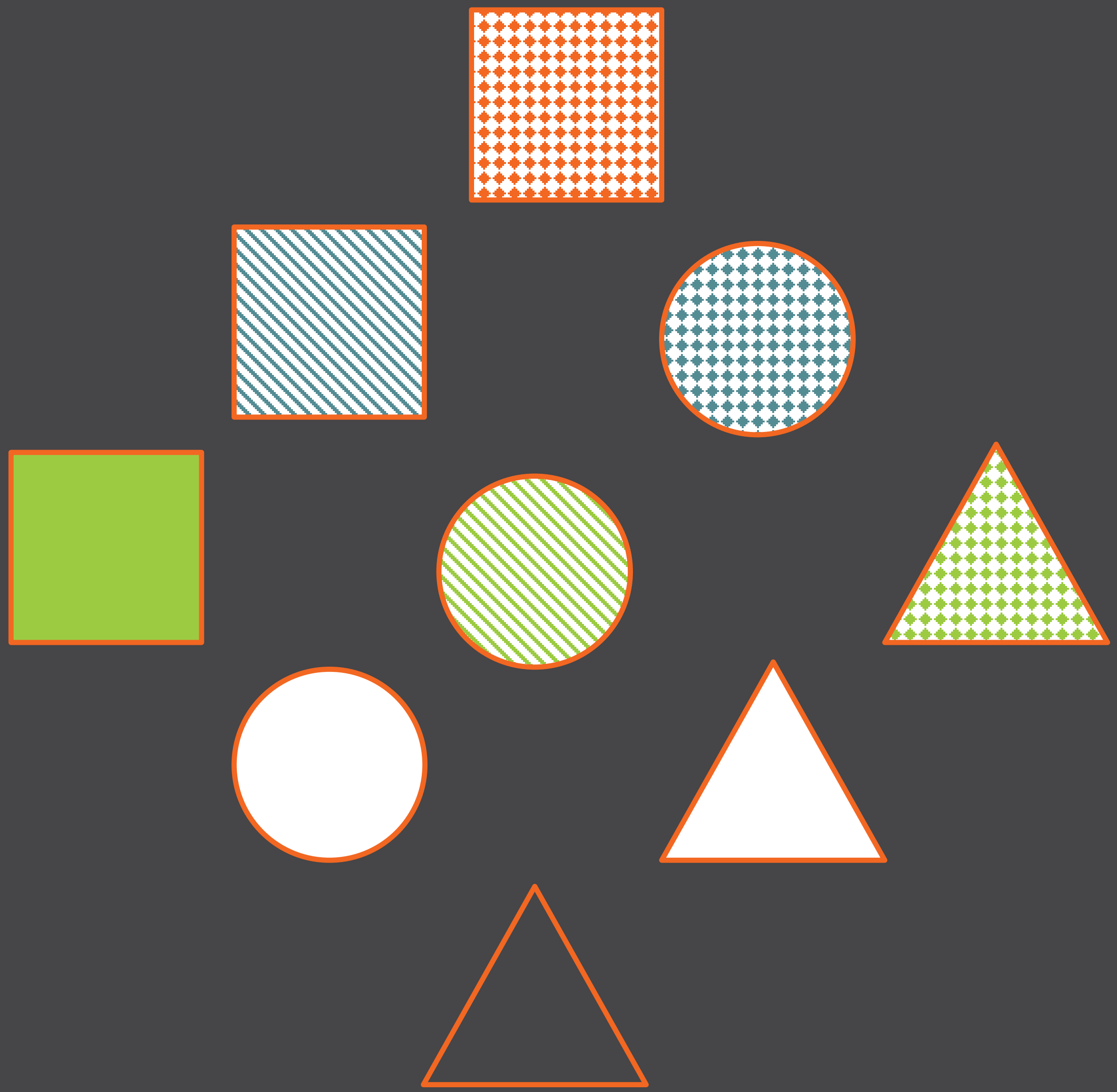

Application
mongos
Application
mongos
Application
mongos
<< config server >>
3 X
mongod
<< shards >>
Replica Set
mongod



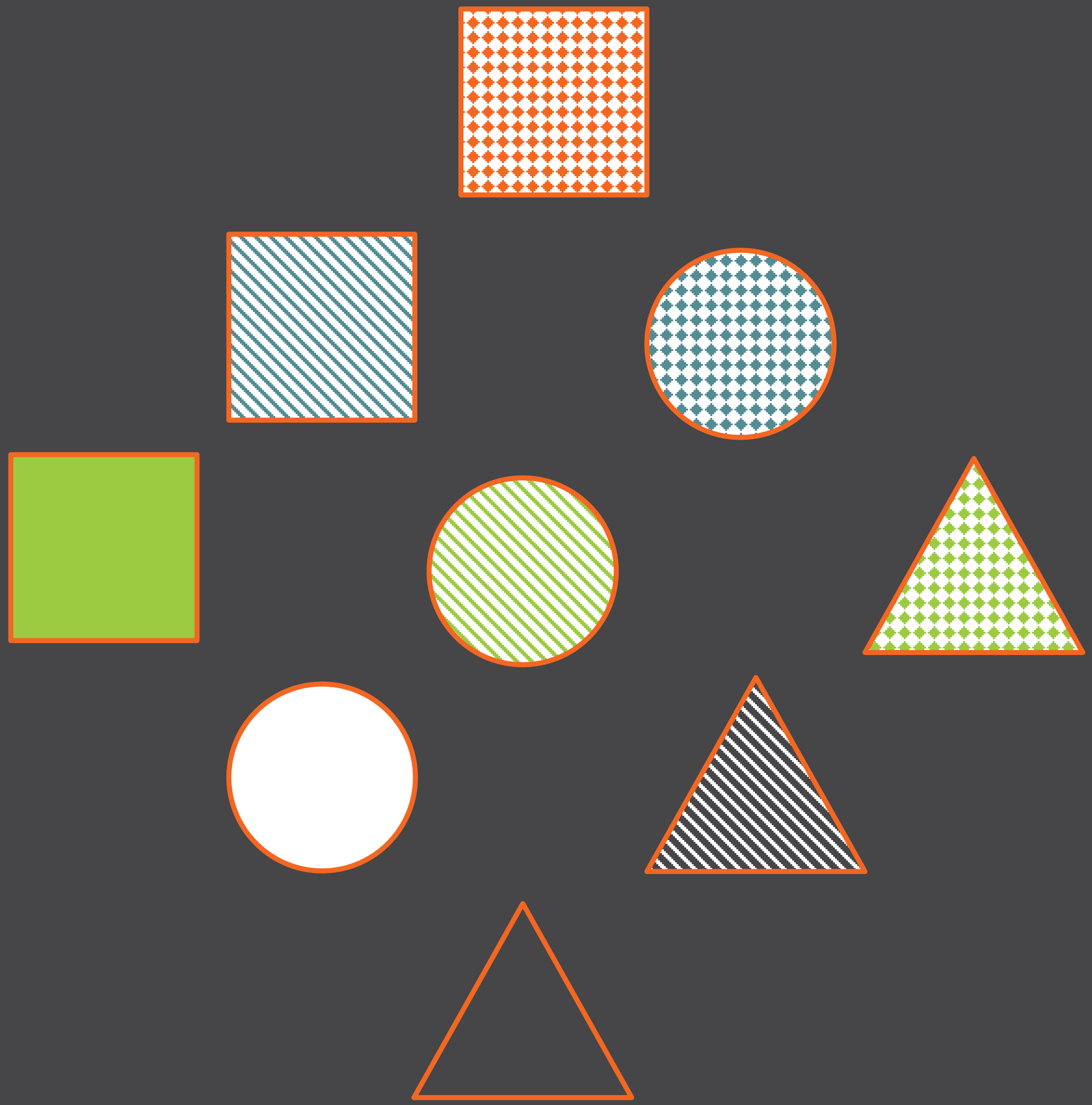
Shard Key Choices

- Important to get "right" the first time
- A balance act to get "right"

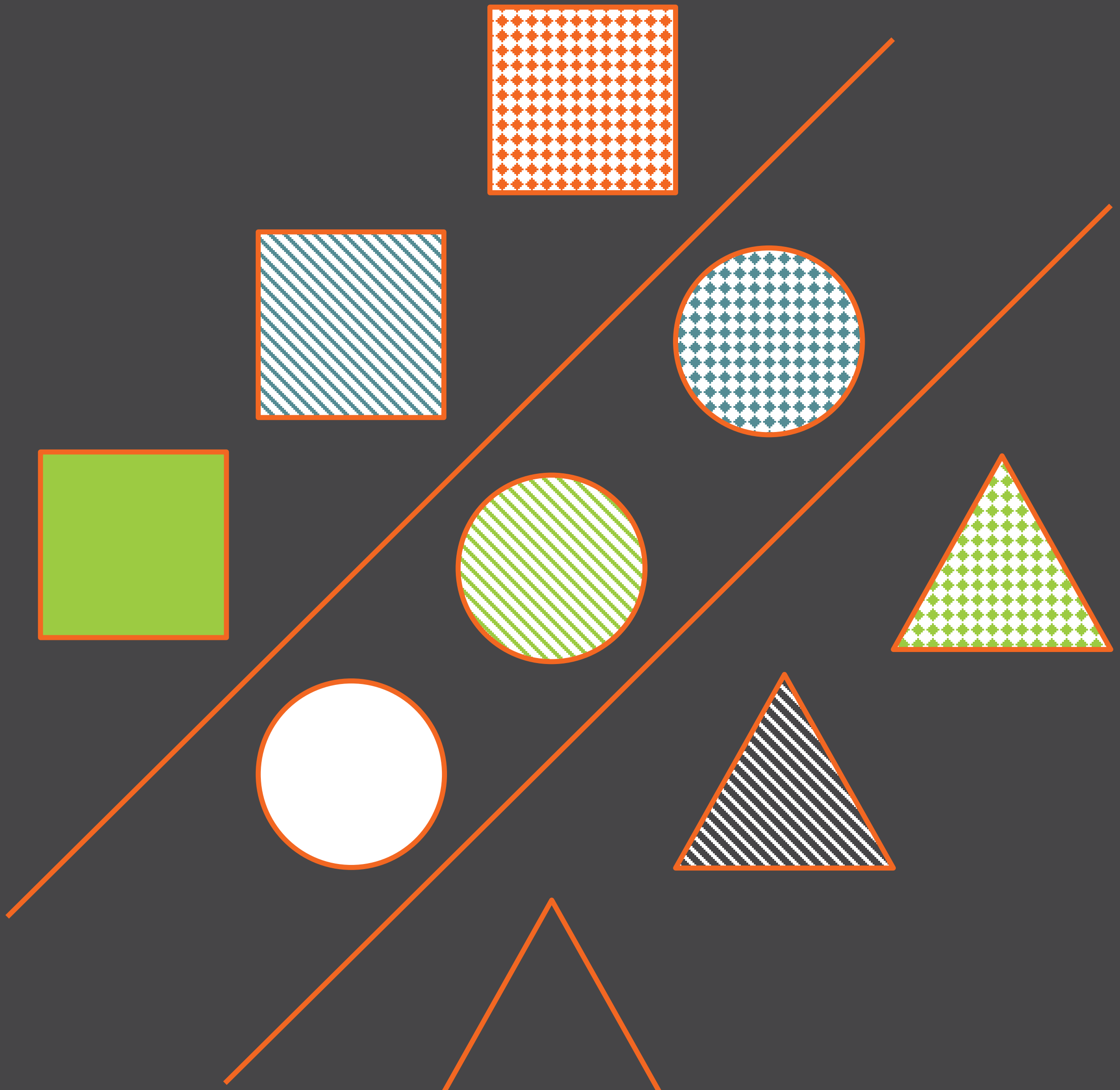
Picking a Shard Key



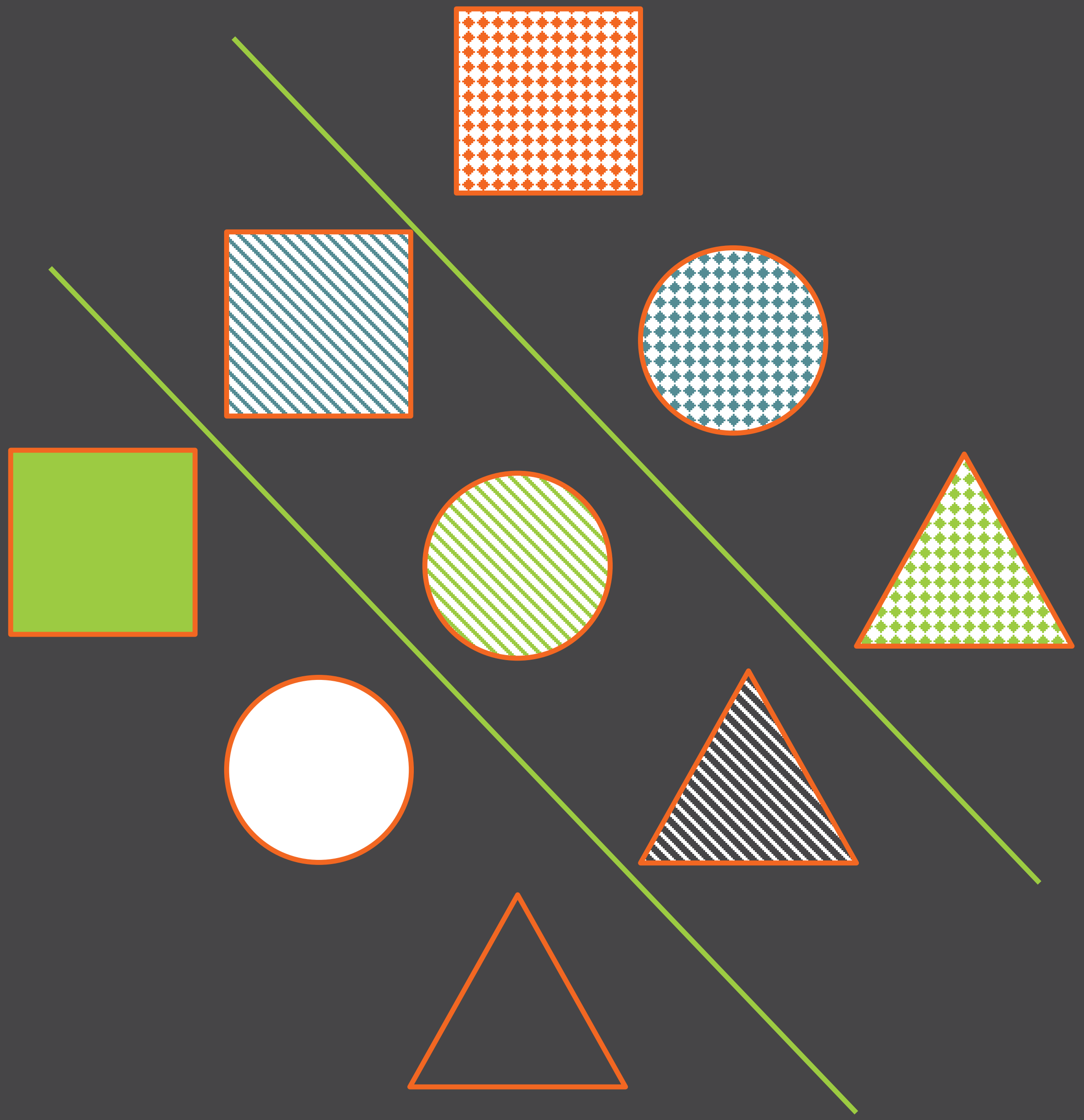
Picking a Shard Key



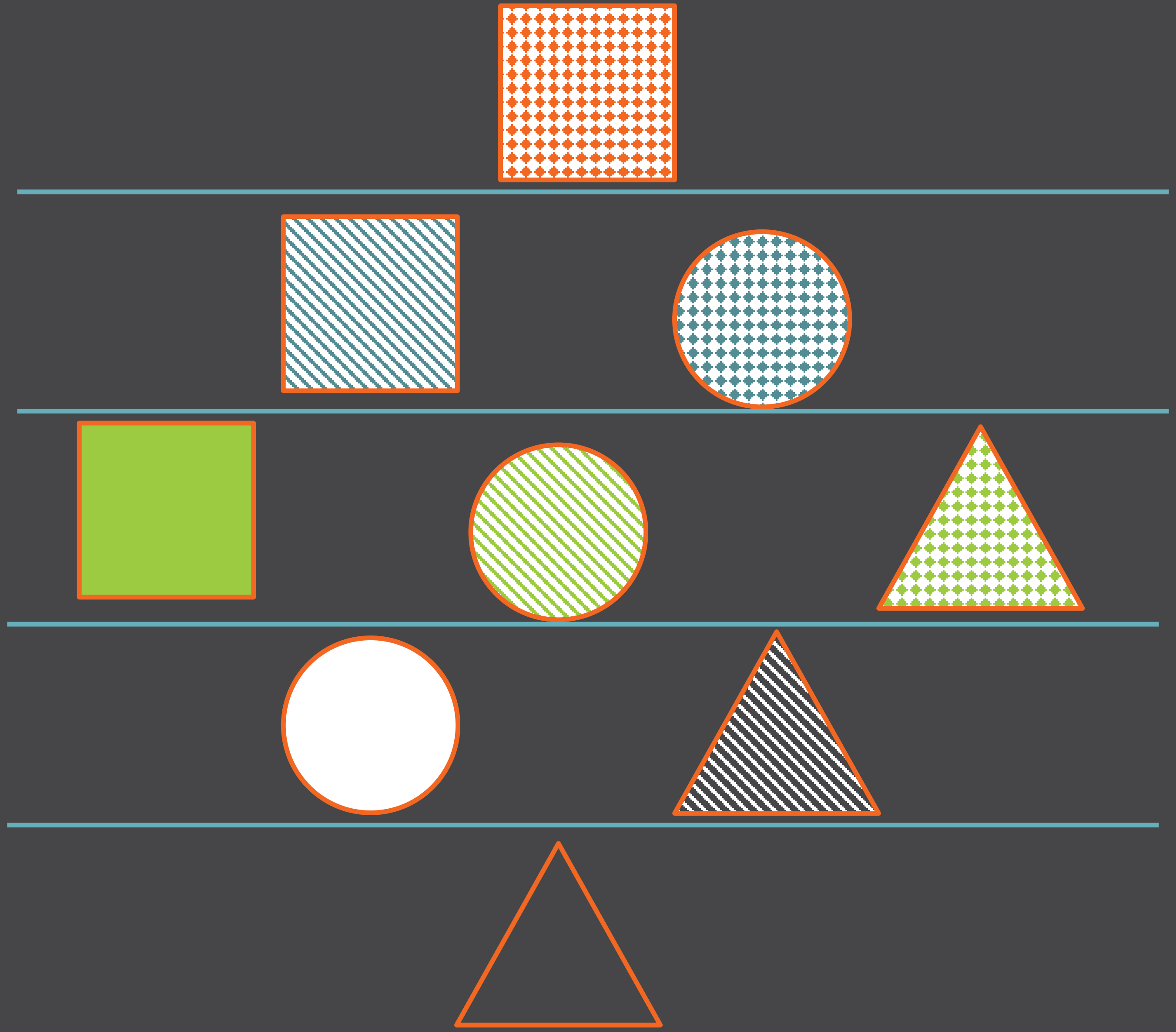
Picking a Shard Key



Picking a Shard Key



Picking a Shard Key

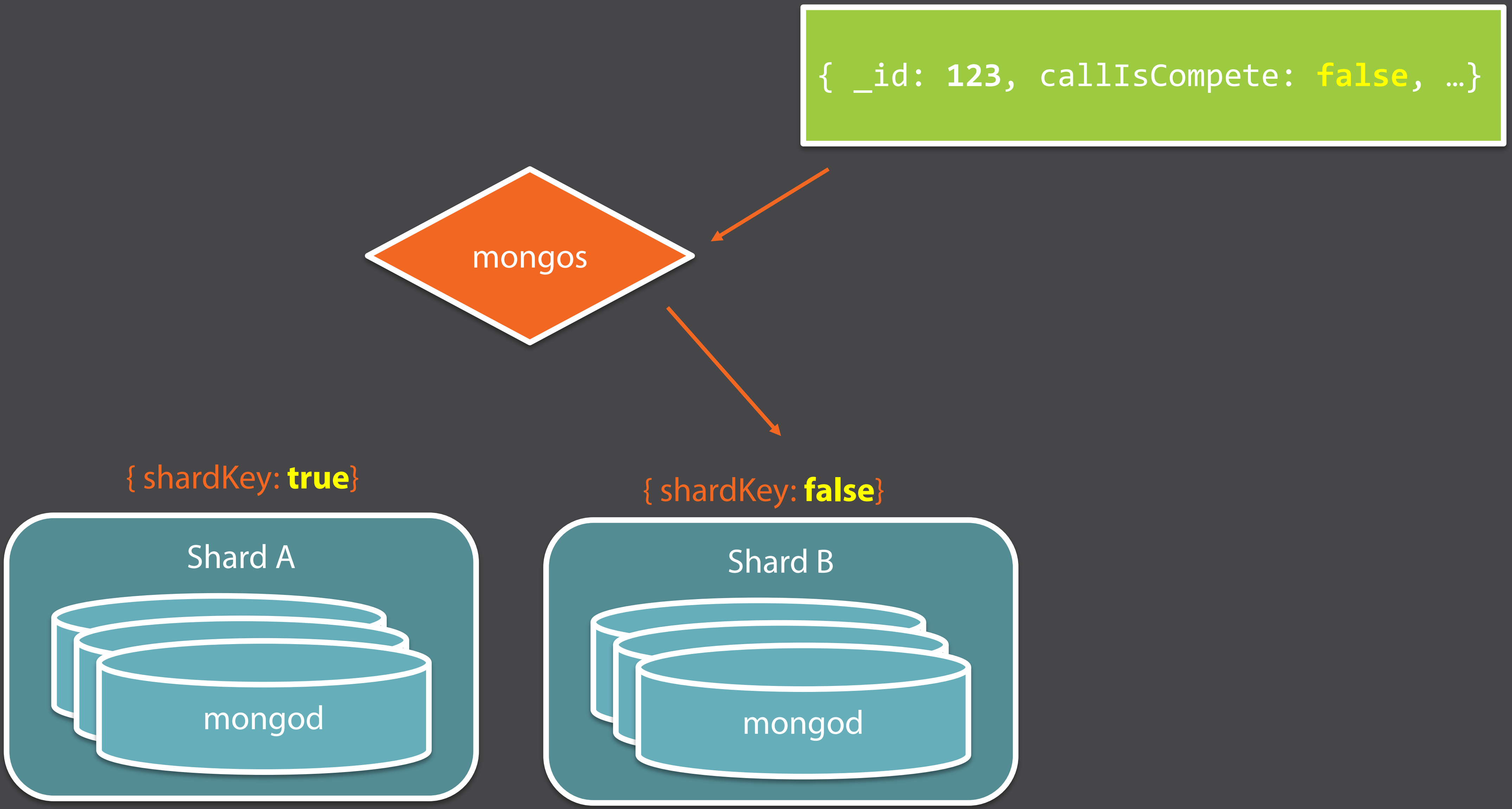


Sharding Rules

Can't change shard key definition

Can't "un-shard" a collection

Shard key field value can't changed



```
{ _id: 123, callIsCompete: false, …}
```

mongos

{ shardKey: **true**}

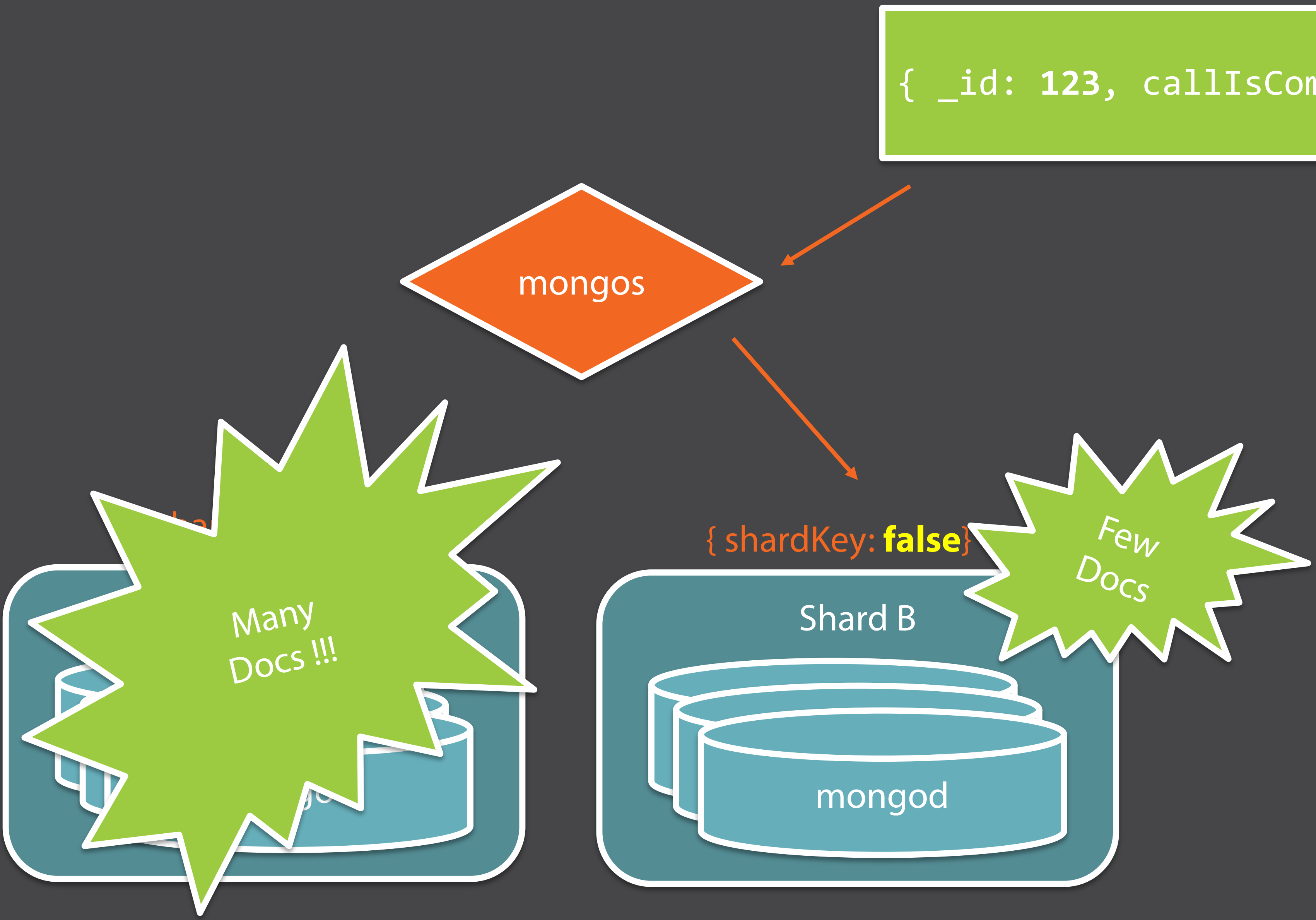
Shard A

mongod

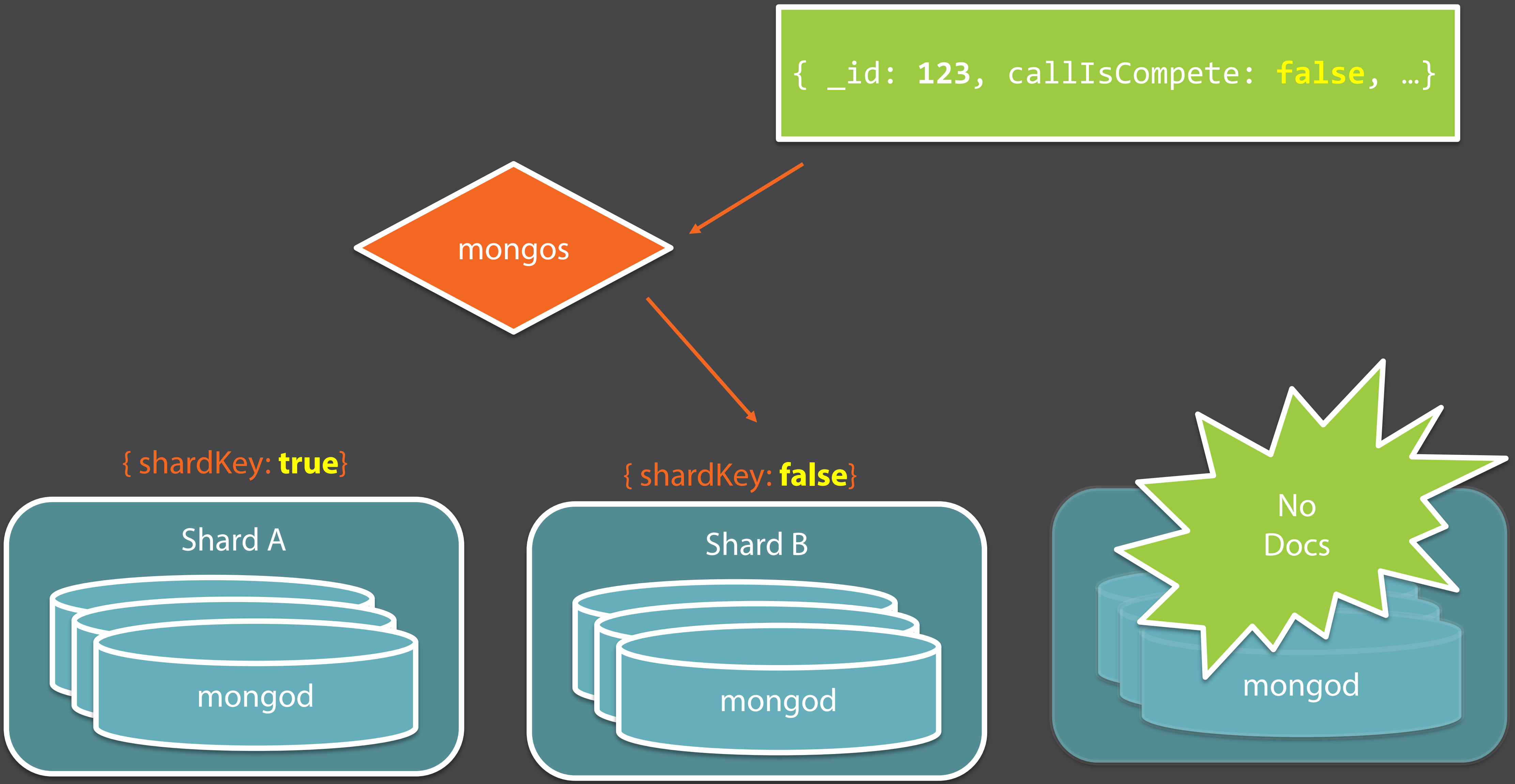
{ shardKey: **false**}

Shard B

mongod



```
{ _id: 123, callIsCompete: false, …}
```



```
{ _id: 123, callIsCompete: false, …}
```

mongos

{ shardKey: **true**}

Shard A

mongod

{ shardKey: **false**}

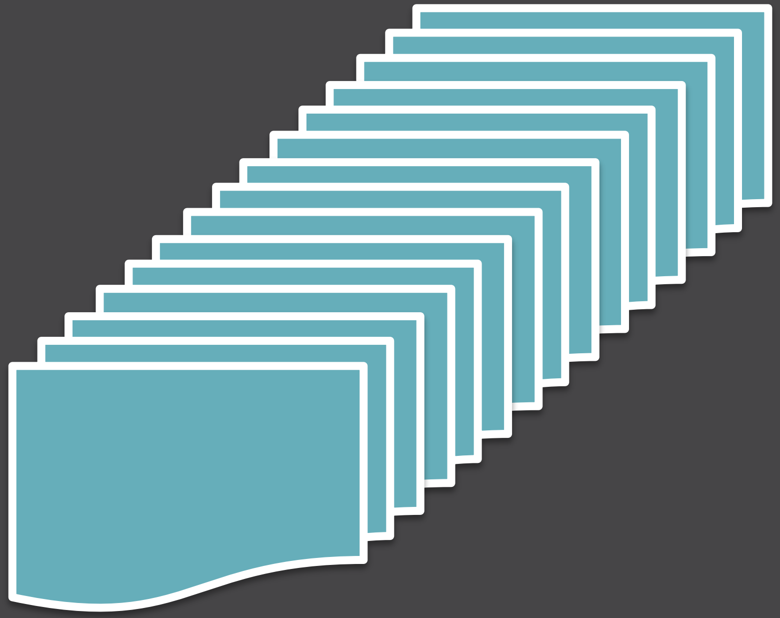
Shard B

mongod

No Docs

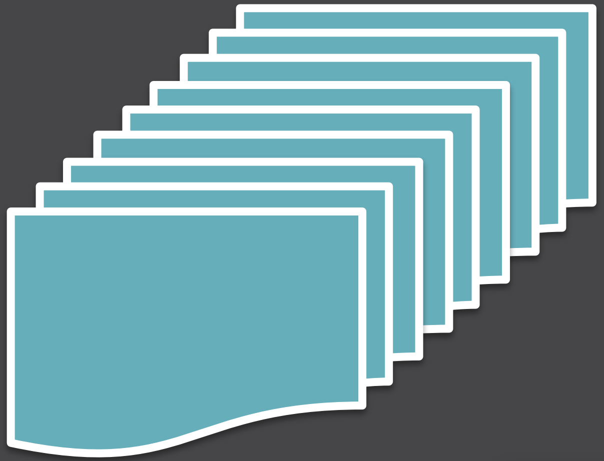
mongod

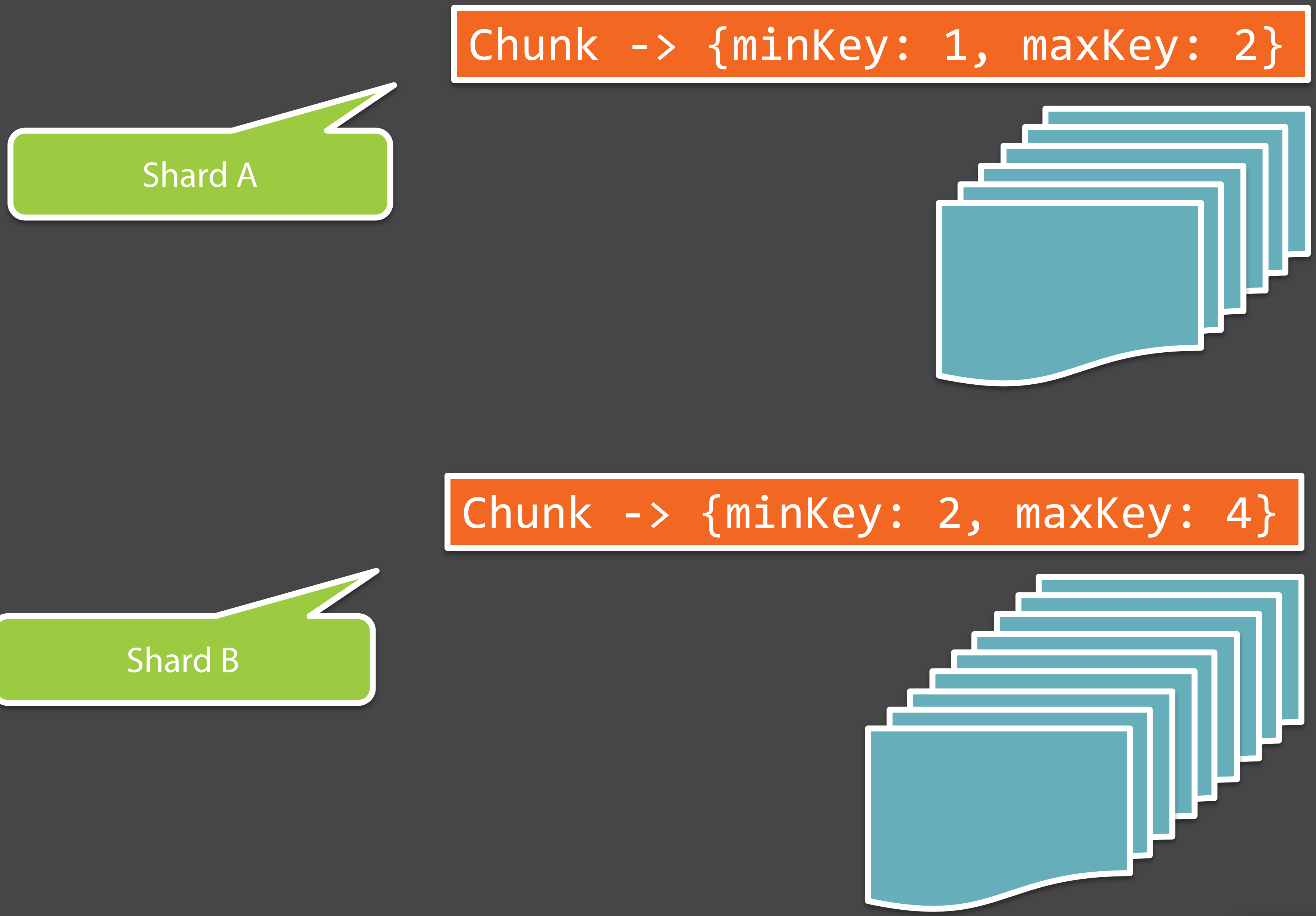
```
Chunk -> {minKey: 1, maxKey: 4}
```



```
Chunk -> {minKey: 1, maxKey: 2}
```

```
Chunk -> {minKey: 2, maxKey: 4}
```



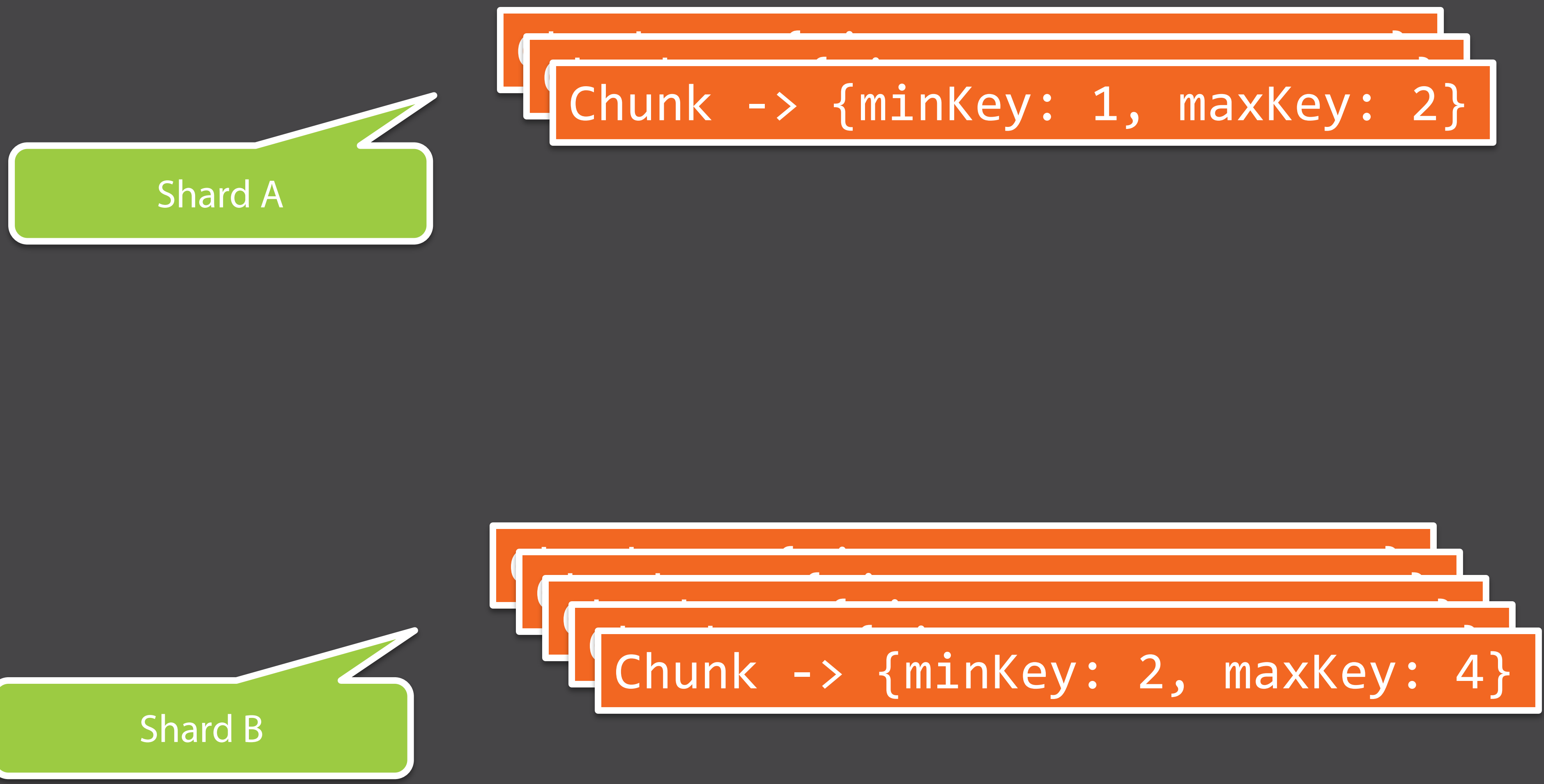


Chunk -> {minKey: 1, maxKey: 2}

Shard A

Chunk -> {minKey: 2, maxKey: 4}

Shard B

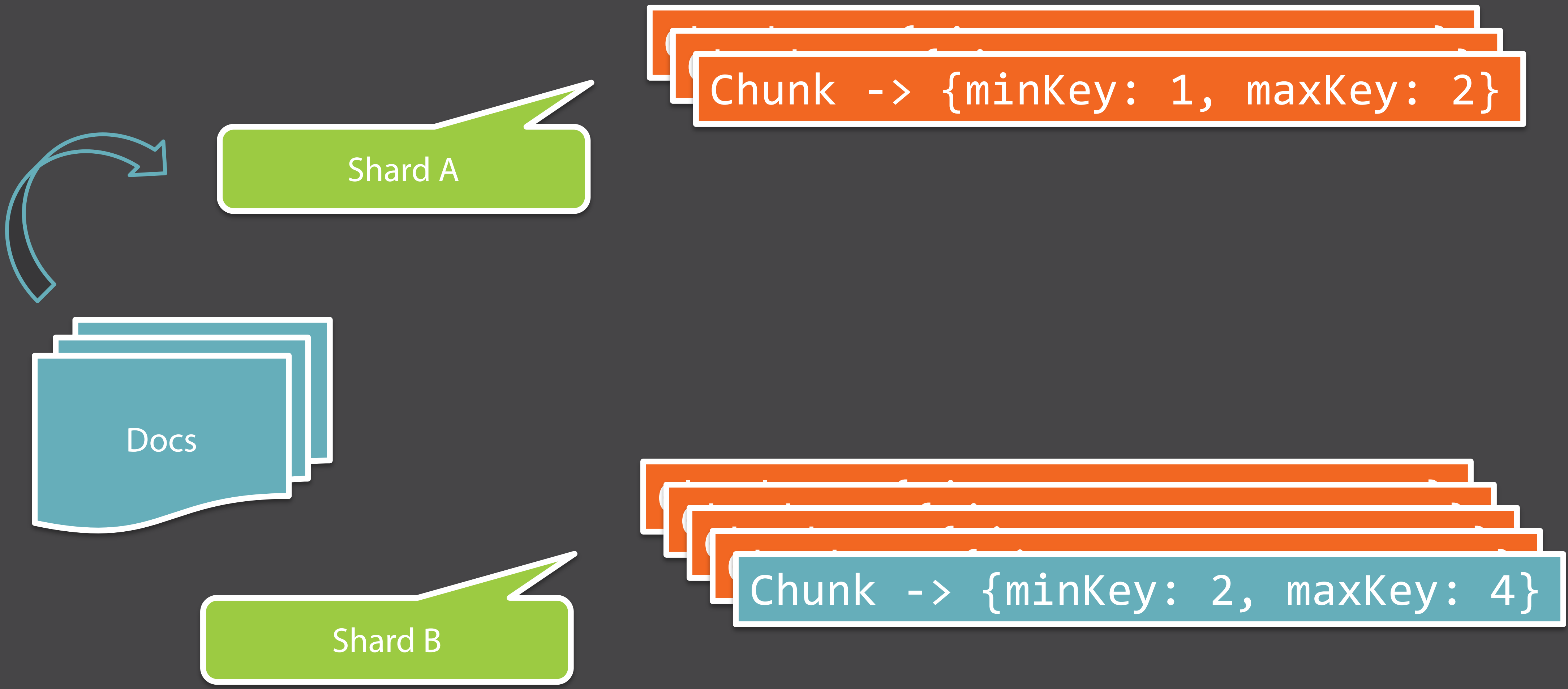


Shard A

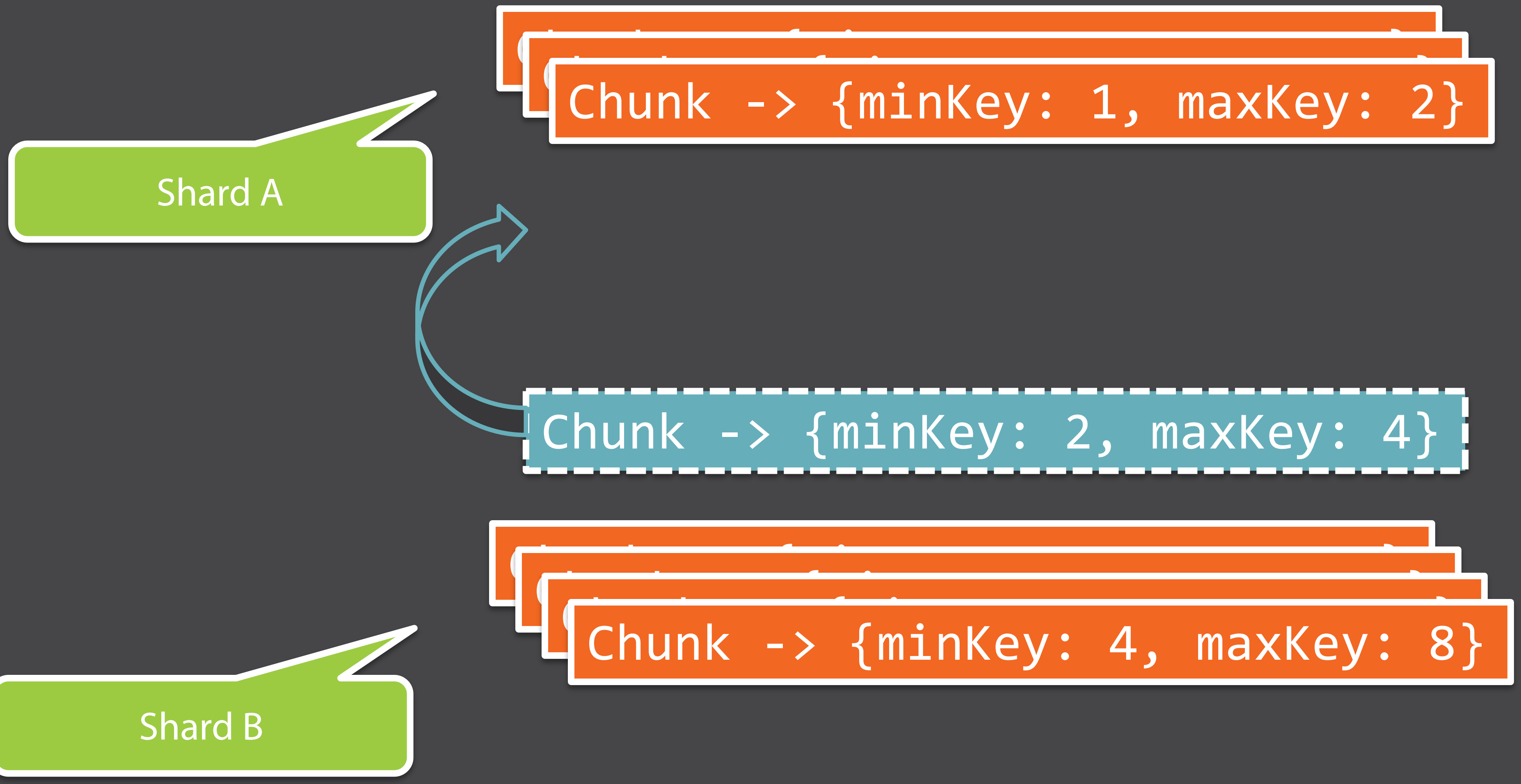
Chunk -> {minKey: 1, maxKey: 2}

Shard B

Chunk -> {minKey: 2, maxKey: 4}



Shard A
Chunk -> {minKey: 1, maxKey: 2}
Docs
Shard B
Chunk -> {minKey: 2, maxKey: 4}


Shard A
Chunk -> {minKey: 1, maxKey: 2}
Chunk -> {minKey: 2, maxKey: 4}
Shard B
Chunk -> {minKey: 4, maxKey: 8}

Sample Document

```
{
    _id: ObjectId(),
    from: { country: '01' , number: '333-555-1212' },
    to: {  country: '01' , number: '444-555-1212' },
    text:'Hi',
    time: ISODate()};
}
```

Choosing a "Good" Key

Your Access Pattern

Theoretical Granularity

Actual Granularity

Choosing a "Good" Key

Your Access Pattern

Theoretical Granularity

Actual Granularity

Country alone
~200 Keys

Choosing a "Good" Key

Country + Number
> Billion Keys

Your
Access Pattern

Theoretical
Granularity

Actual
Granularity

?

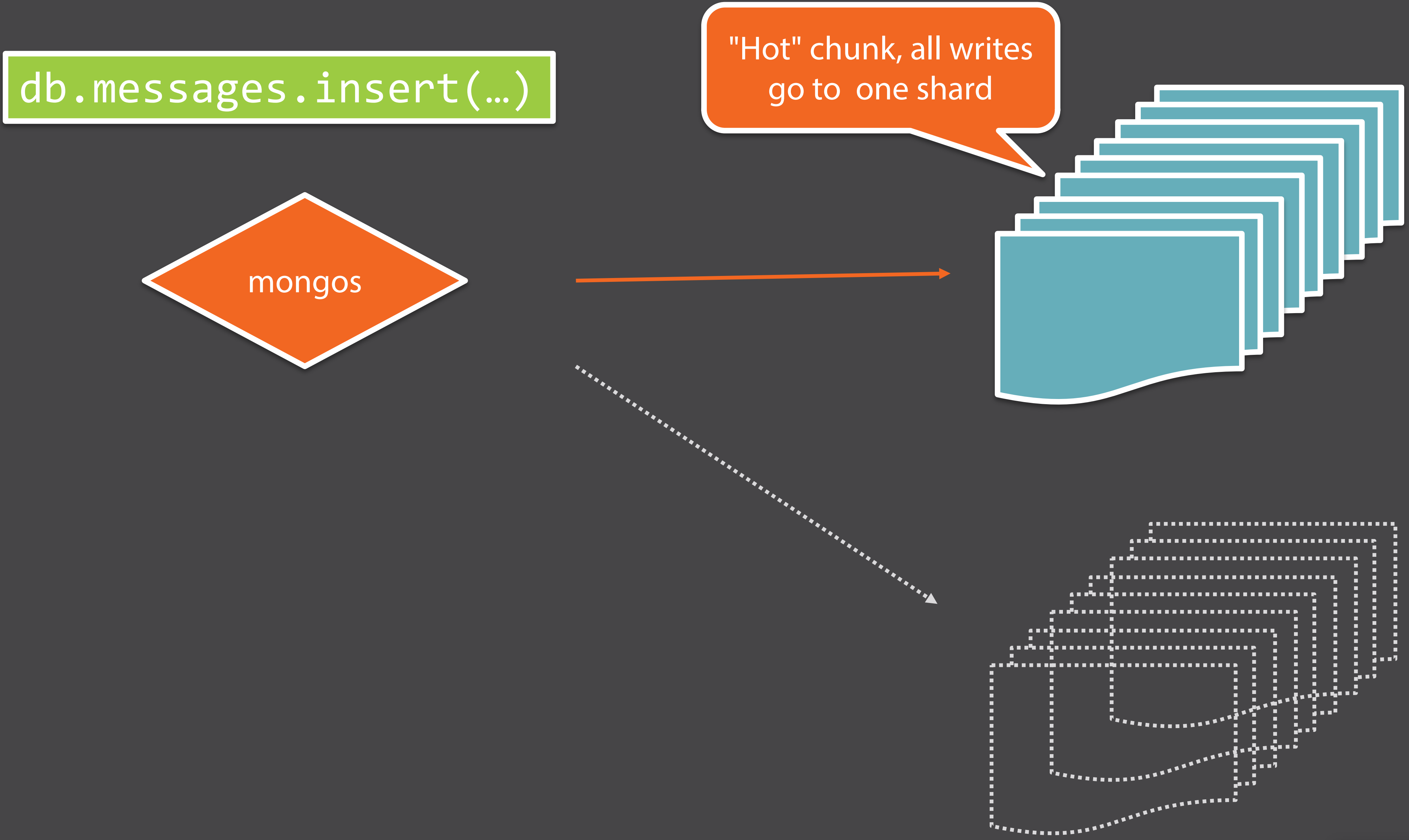
Choosing a "Good" Key

Your Access Pattern

Theoretical Granularity

Actual Granularity

Time stamp, ever increasing



```
db.messages.insert(…)
```

"Hot" chunk, all writes go to one shard

Choosing a "Good" Key

Your Access Pattern

Theoretical Granularity

Actual Granularity

Hashed Key

```
sh.shardCollection(
    "dbName.collectionName",
    {"fieldName": "hashed"})
```

Hashed shard key

- Only on a single field
- Helps even distribution across shards

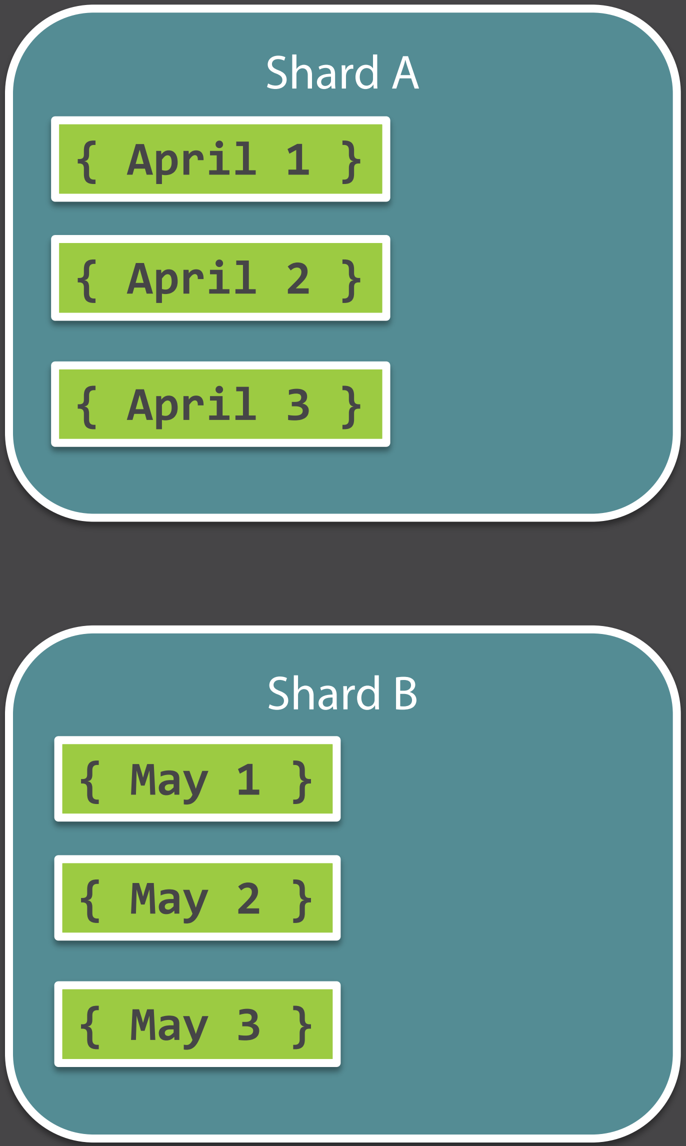
Choosing a "Good" Key

Your Access Pattern

Theoretical Granularity

Actual Granularity

Not Hashed



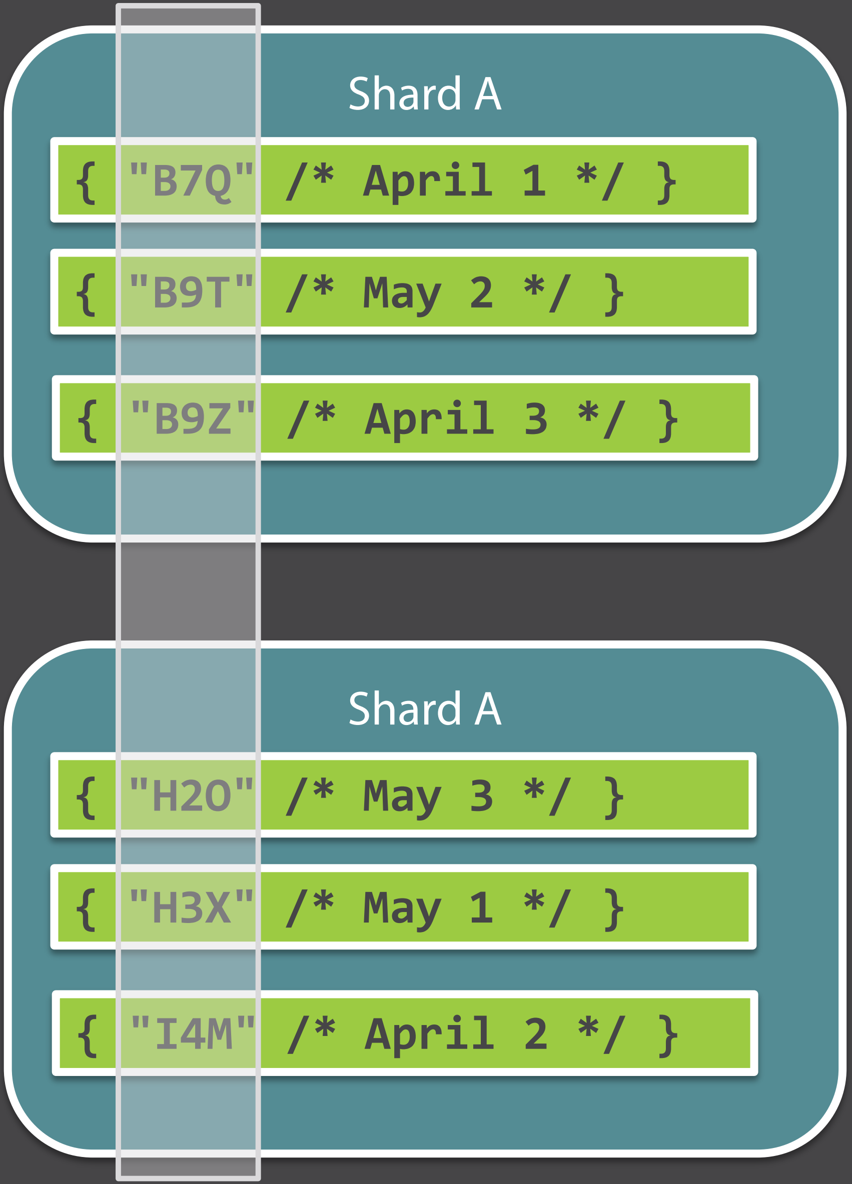
Shard A

```
{ April 1 }
{ April 2 }
{ April 3 }
```

Shard B

```
{ May 1 }
{ May 2 }
{ May 3 }
```

Hashed



Shard A

```
{ "B7Q" /* April 1 */ }
{ "B9T" /* May 2 */ }
{ "B9Z" /* April 3 */ }
```

Shard A

```
{ "H2O" /* May 3 */ }
{ "H3X" /* May 1 */ }
{ "I4M" /* April 2 */ }
```

Choosing a "Good" Key

Establish this …

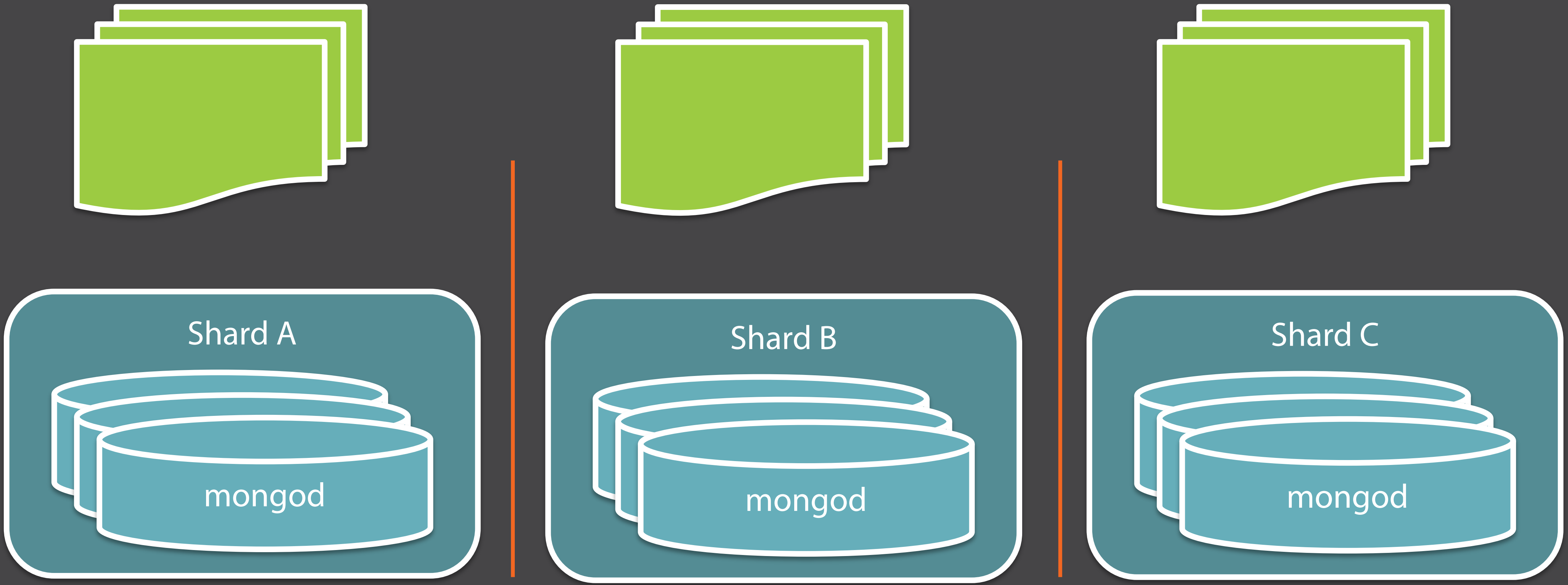
Your Access Pattern

Theoretical Granularity

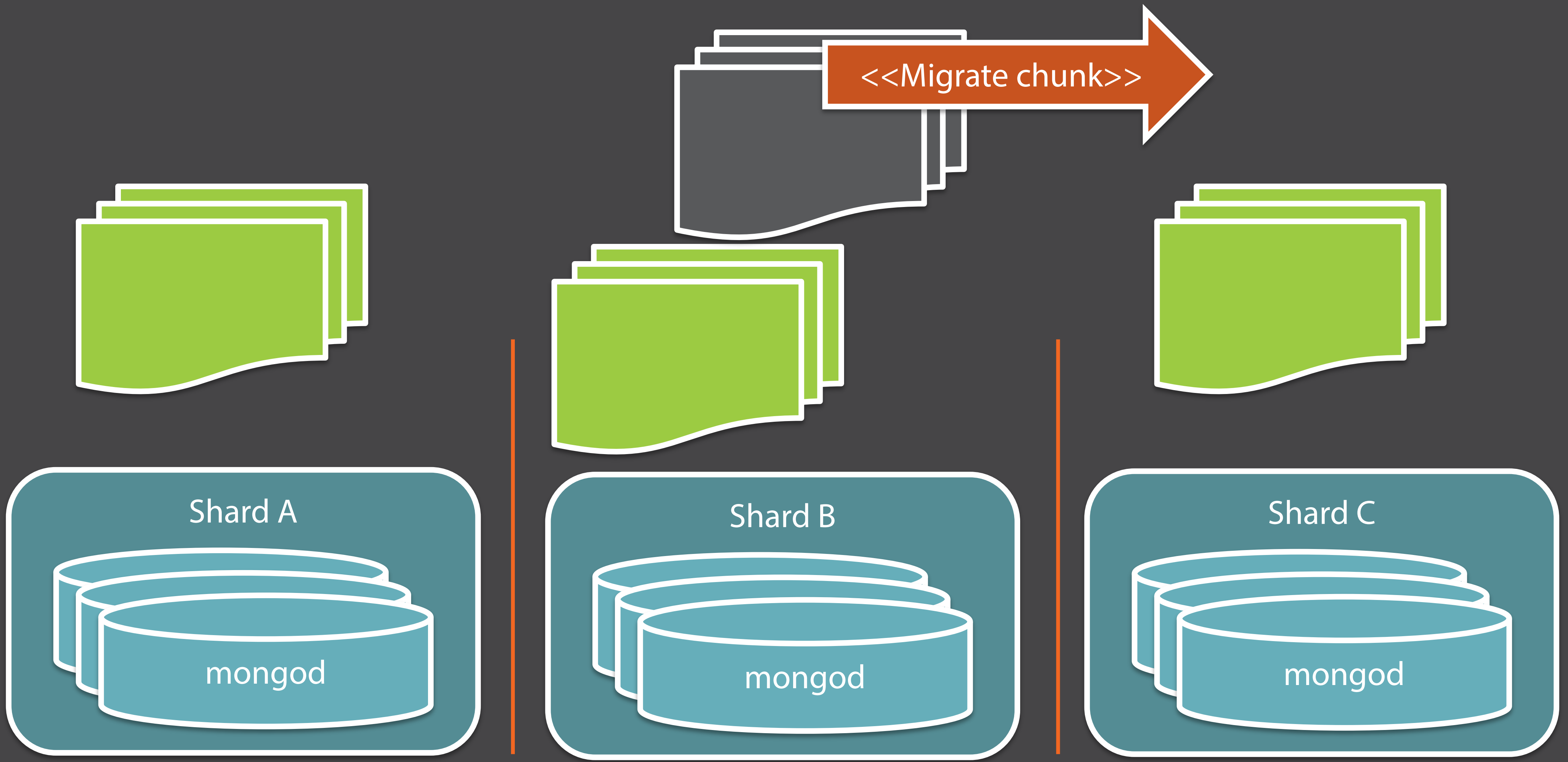
Actual Granularity

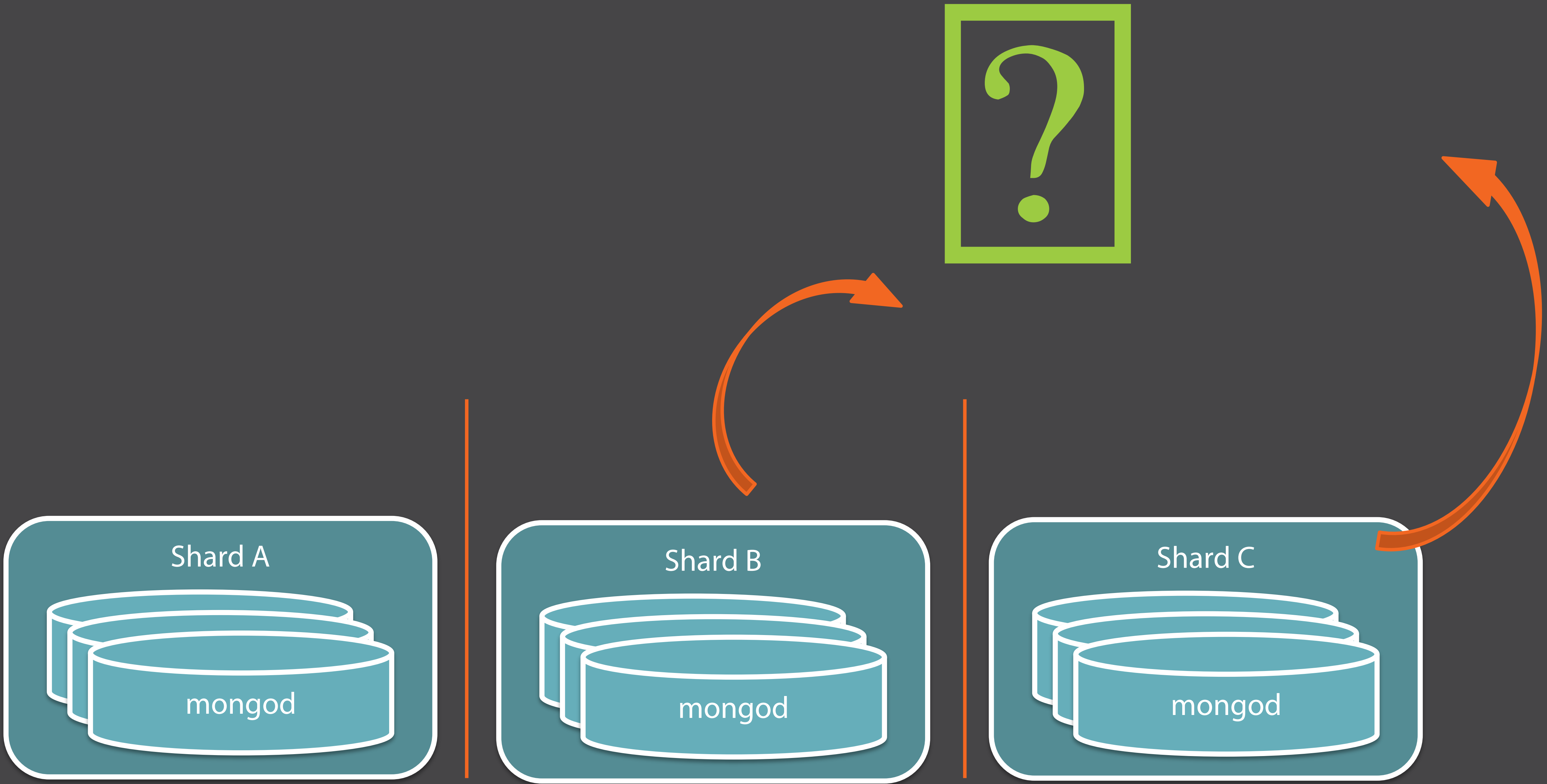
Find key to satisfy these…

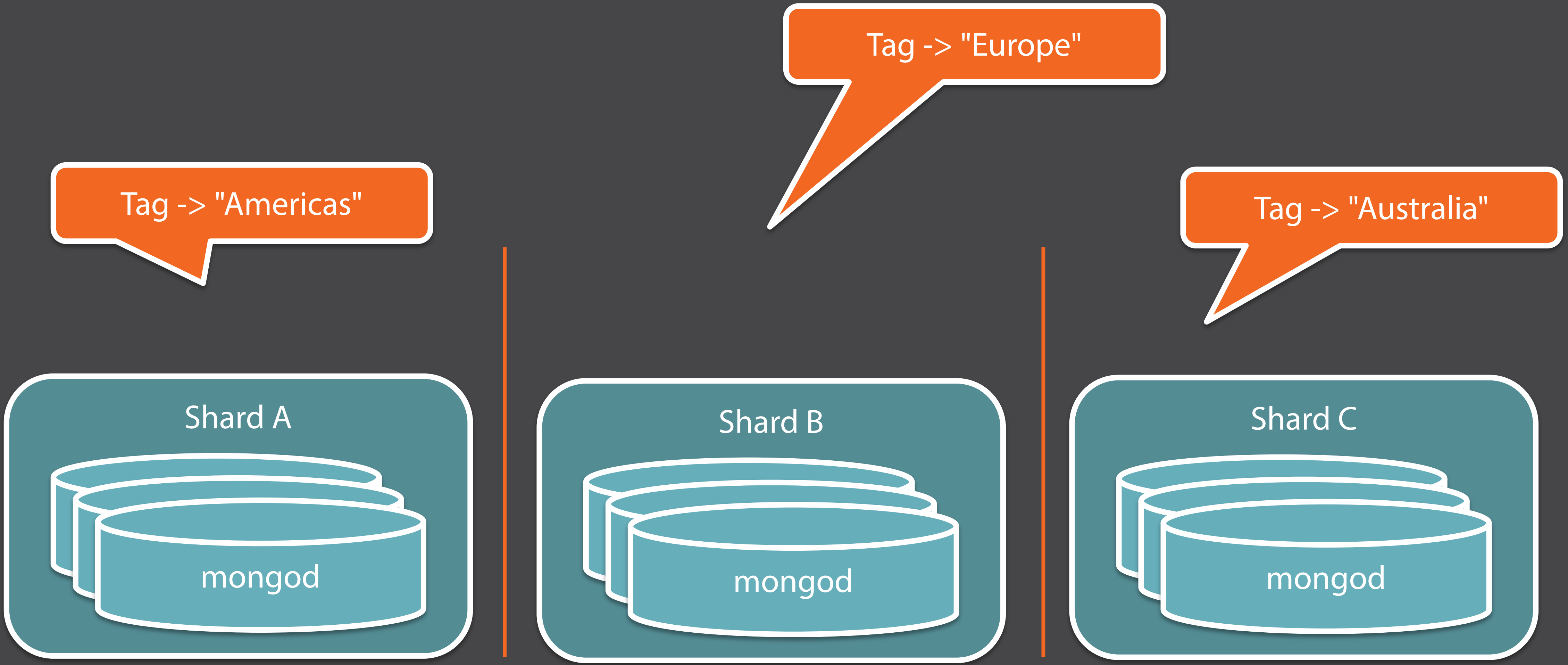
Even Distribution

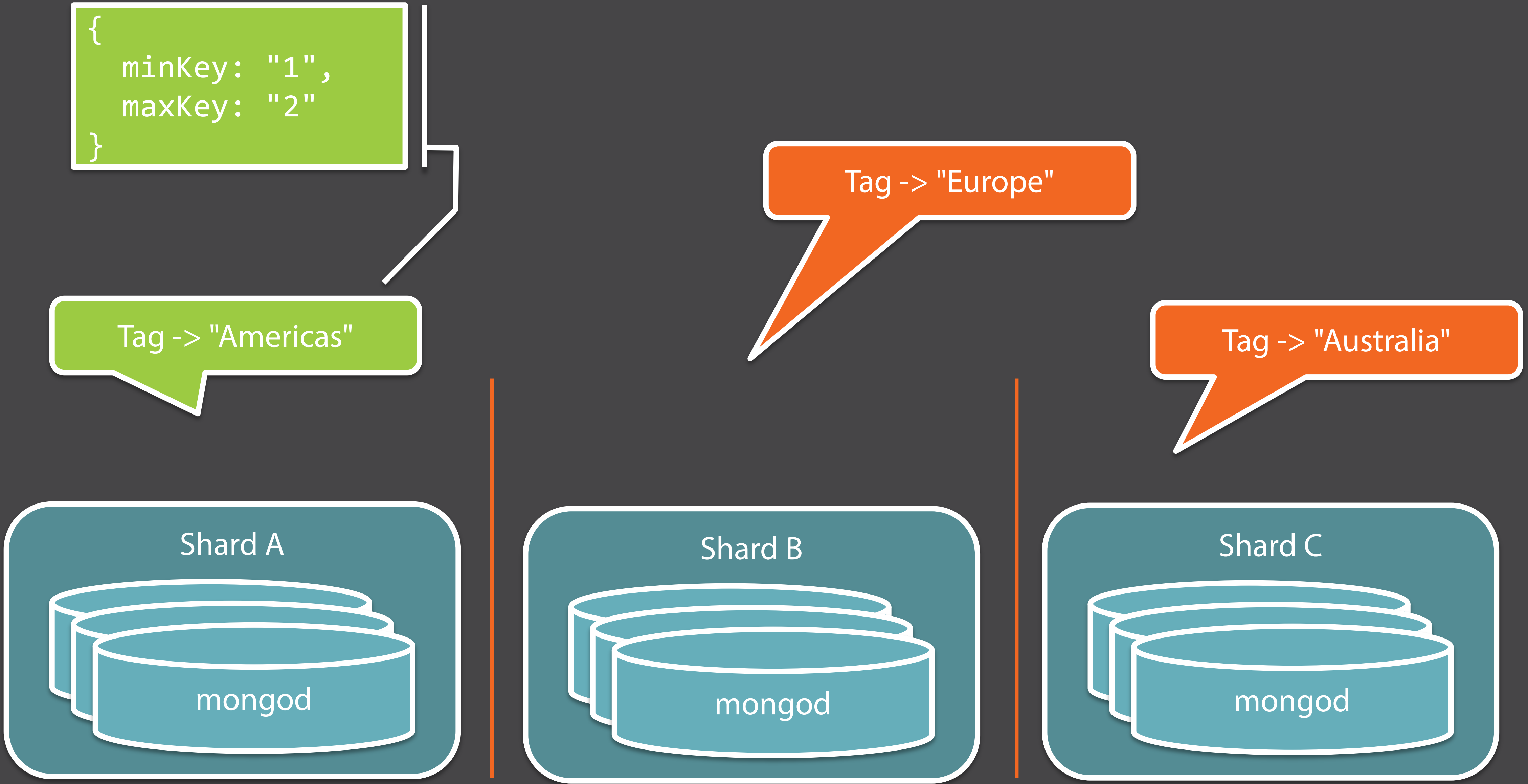


Even Distribution

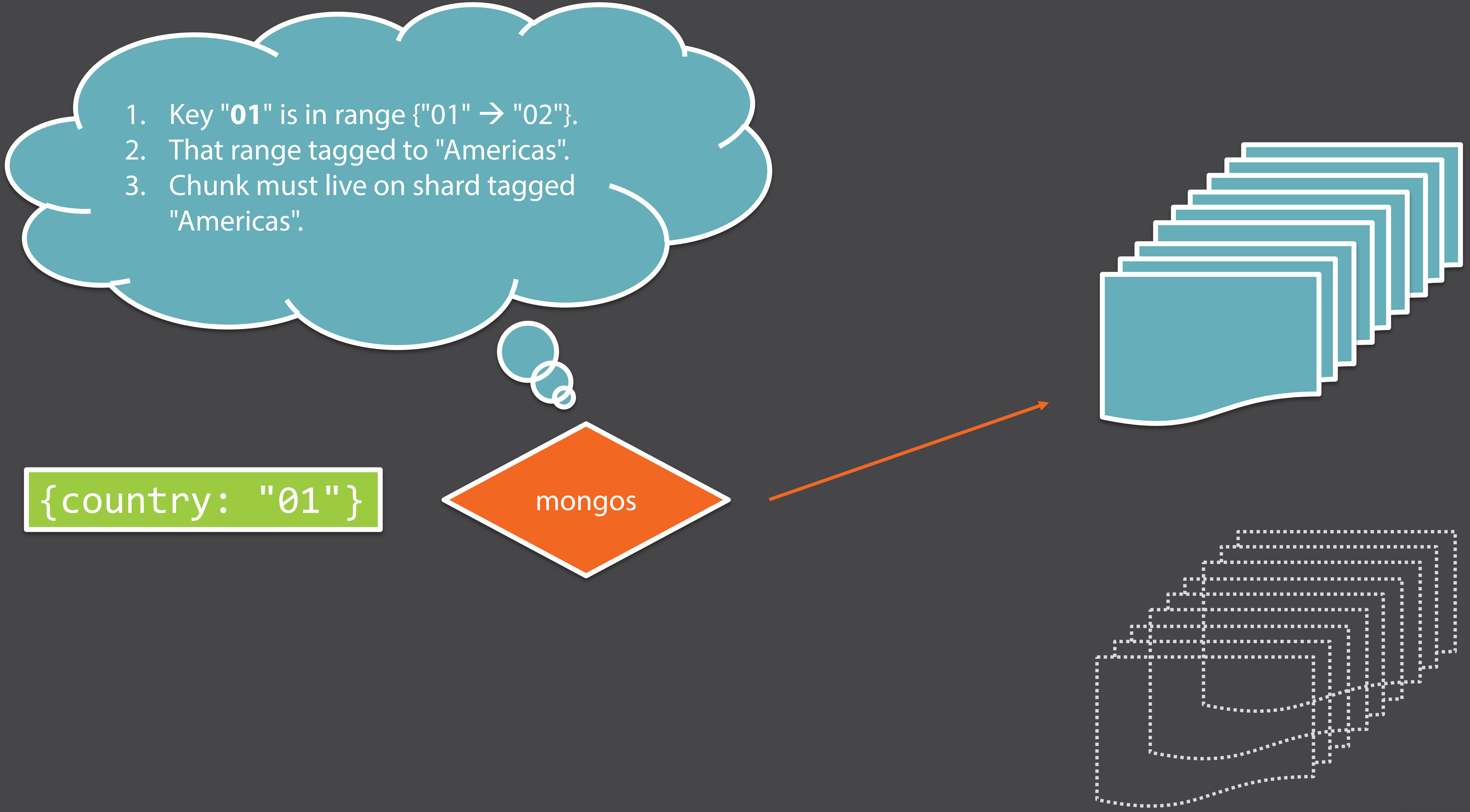



?
Shard A
mongod
Shard B
mongod
Shard C
mongod


Tag -> "Europe"
Tag -> "Americas"
Tag -> "Australia"
Shard A
mongod
Shard B
mongod
Shard C
mongod

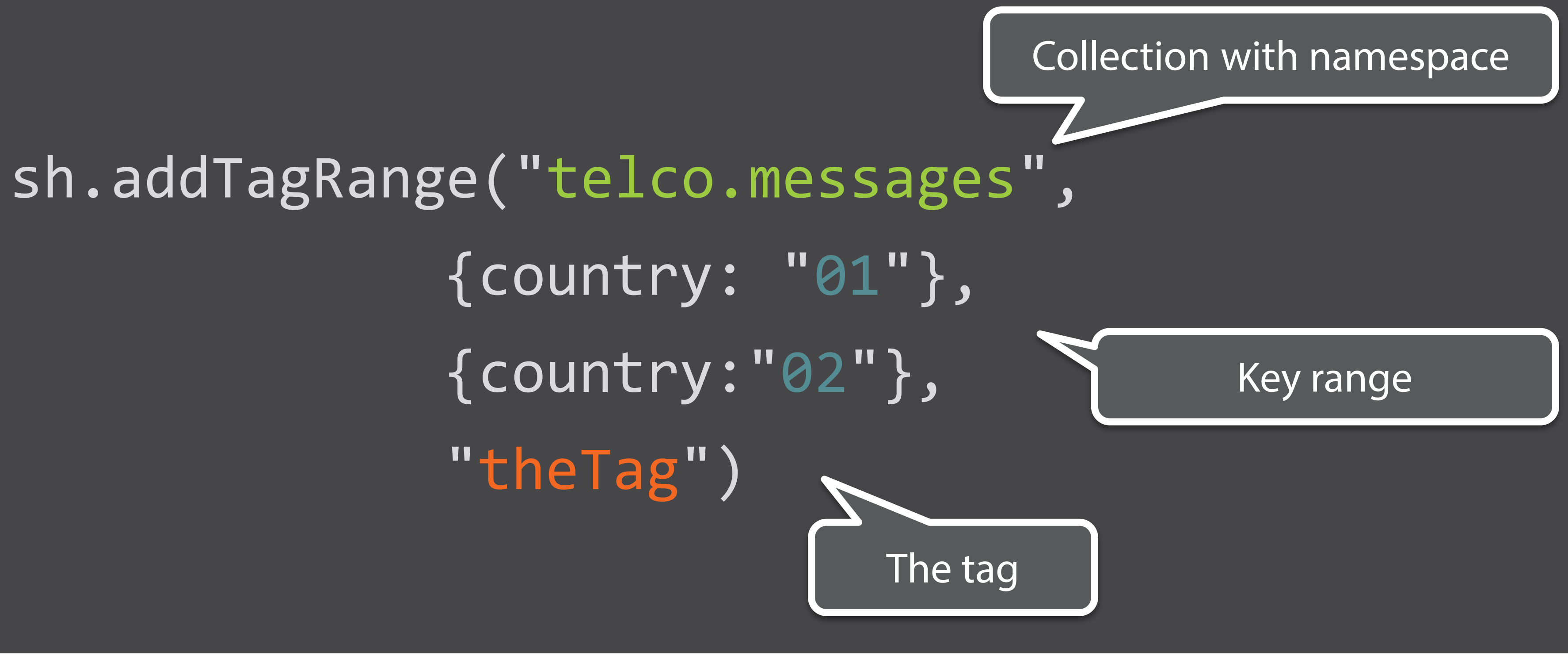


```
{
  minKey: "1",
  maxKey: "2"
}
```



1. Key "**01**" is in range {"01" → "02"}.
2. That range tagged to "Americas".
3. Chunk must live on shard tagged "Americas".

```
{country: "01"}
```



```
sh.addTagRange("telco.messages",
               {country: "01"},
               {country:"02"},
               "theTag")
```

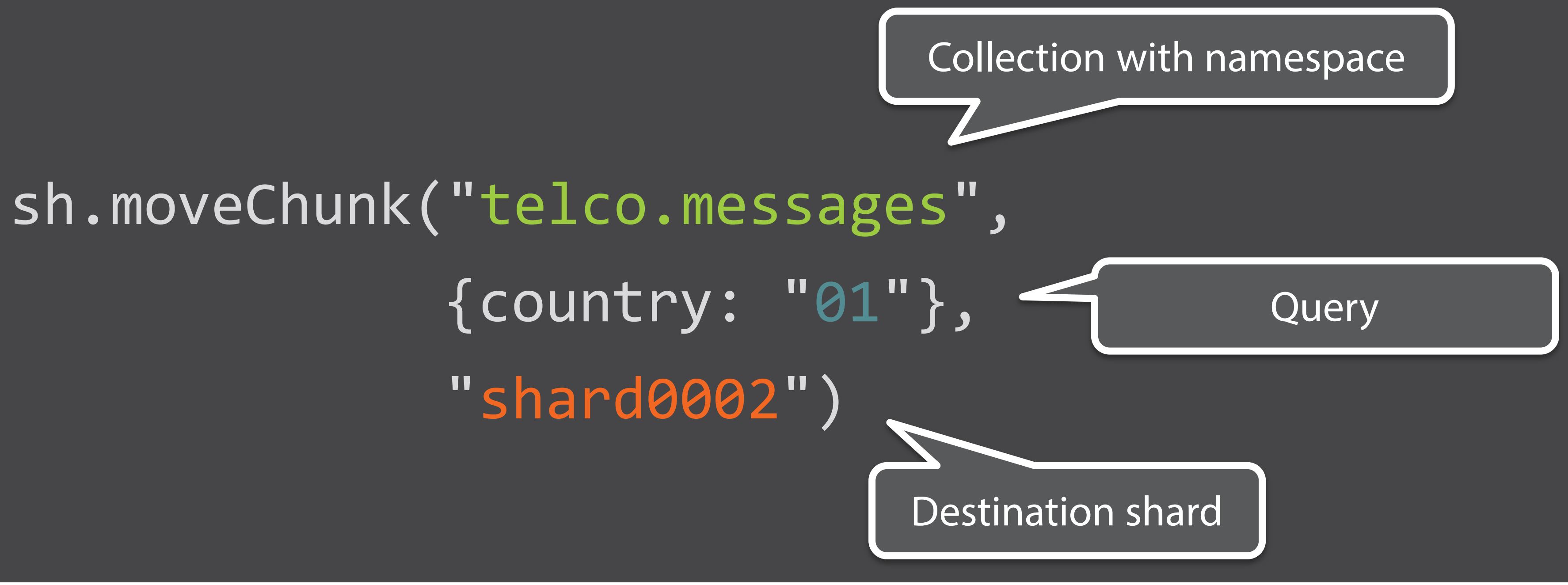
Collection with namespace

Key range

The tag

Creating a tag range definition

- Defines a range that is assigned to a tag
- Use only when connected to mongos
- Helps balancer automatically assign chunks to desired shards



```
sh.moveChunk("telco.messages",
             {country: "01"},
             "shard0002")
```

Moving a chunk

- Query is equality-match on the shard key
- Moves the whole chunk containing the document found by the query

Shard or Not?

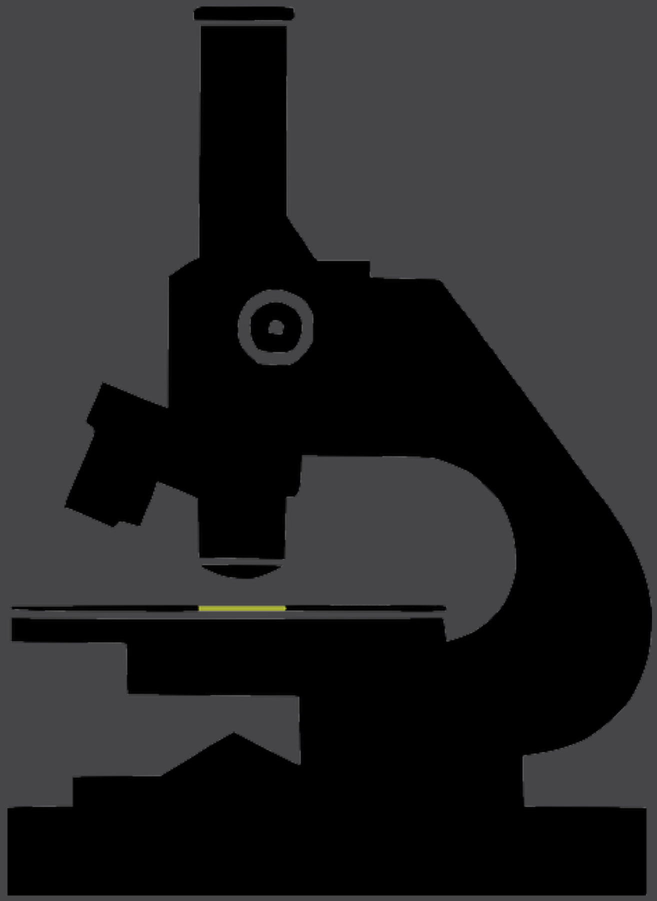
- Will it solve a real issue?
- Current Capacity?
- Application ready for it?
- Complexity?
- Infrastructure Cost?
- When Should I?

Watch for Clues…

Working set, RAM

Watch for Clues…

I/O bottleneck



Watch for Clues…

Free disk space

“PLAN AHEAD”

Summary