mongoDB

Hybrid Datastore

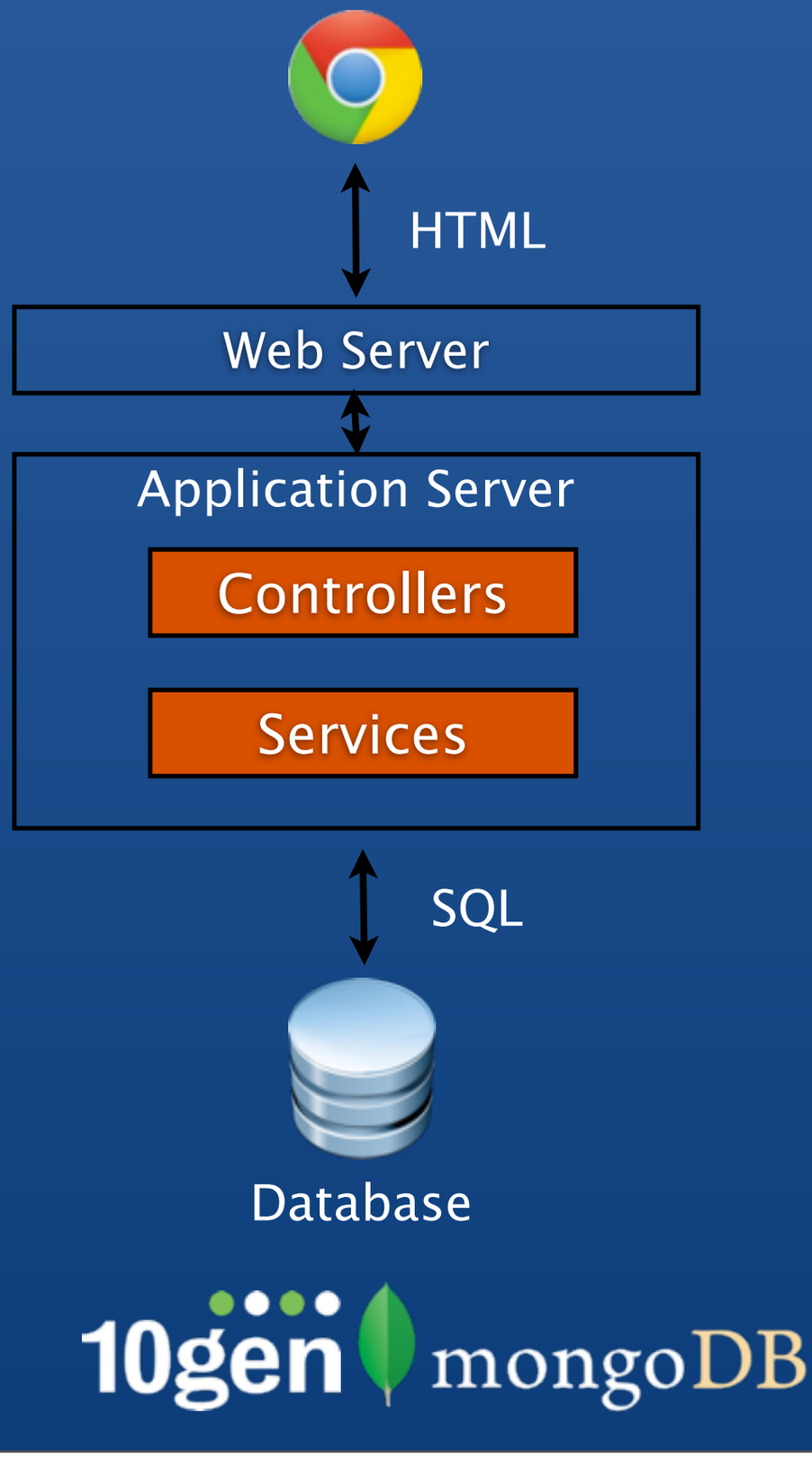
Chris Harris
Email : charris@10gen.com
Twitter : cj_harris5

10gen

mongoDB

Traditional Architecture

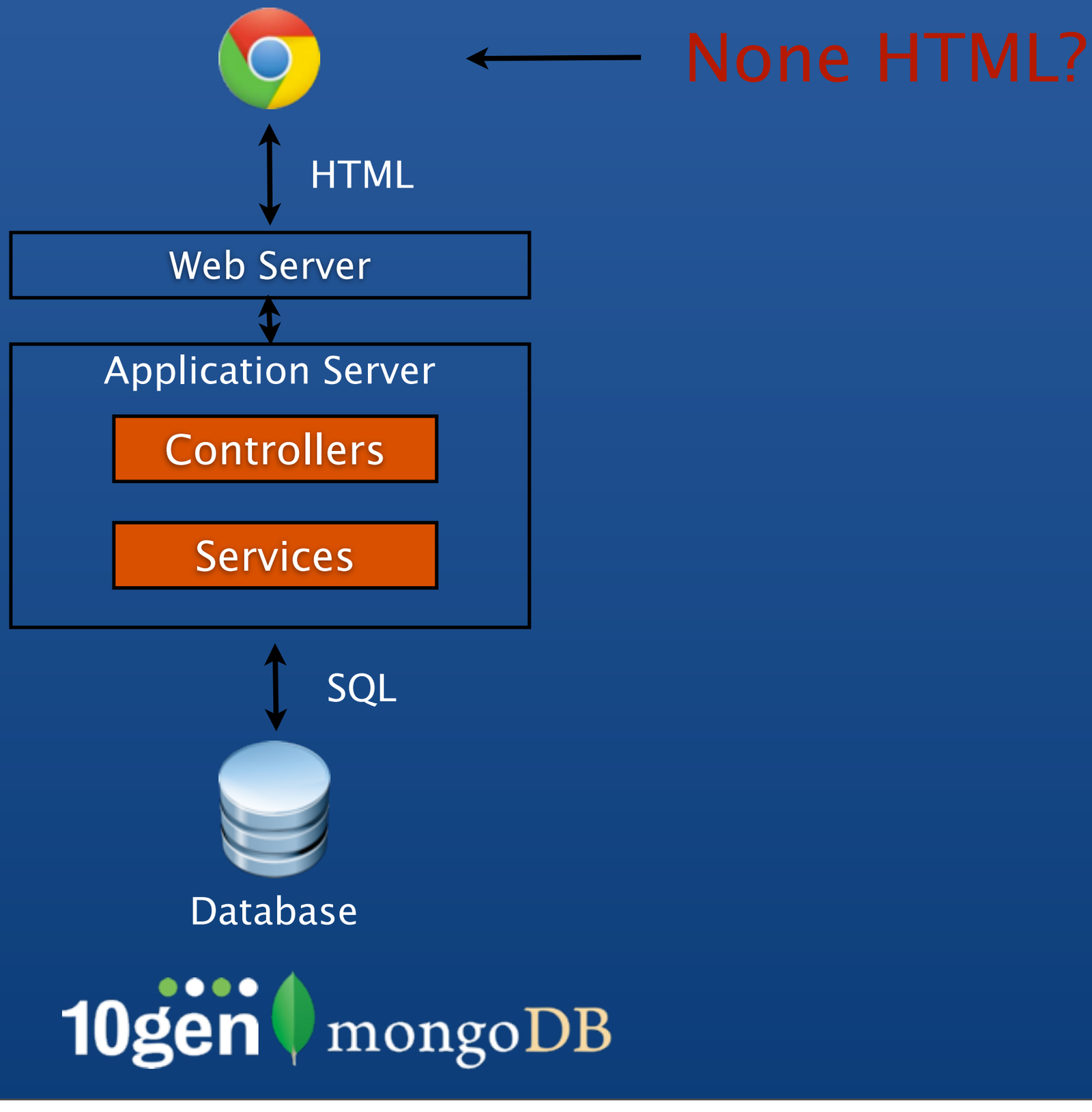
Traditional Architecture



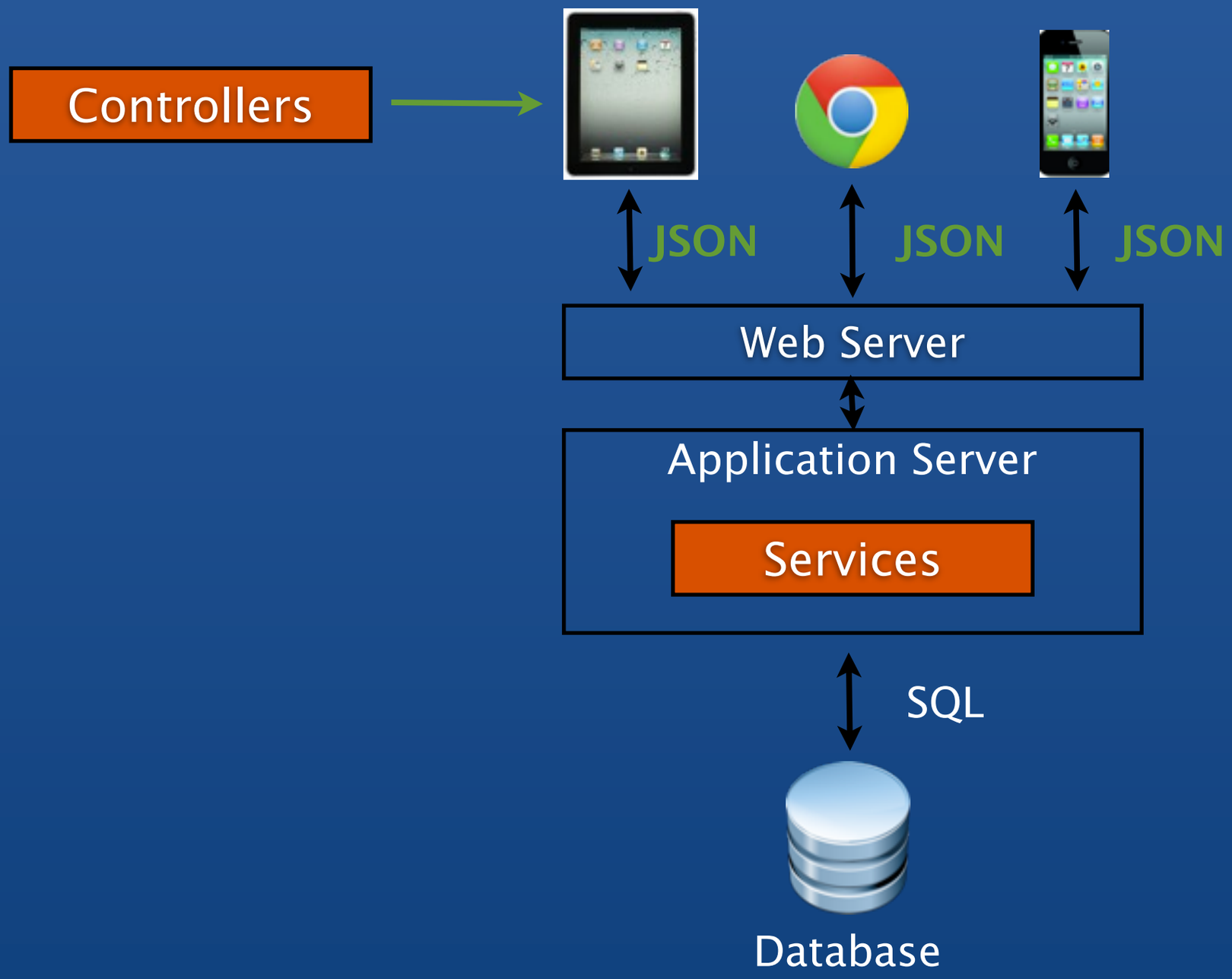
10gen mongoDB

Challenge #1 - HTML 5

Multiple Client Types



HTML 5


Controllers
JSON
JSON
JSON
Web Server
Application Server
Services
SQL
Database
10gen
mongoDB

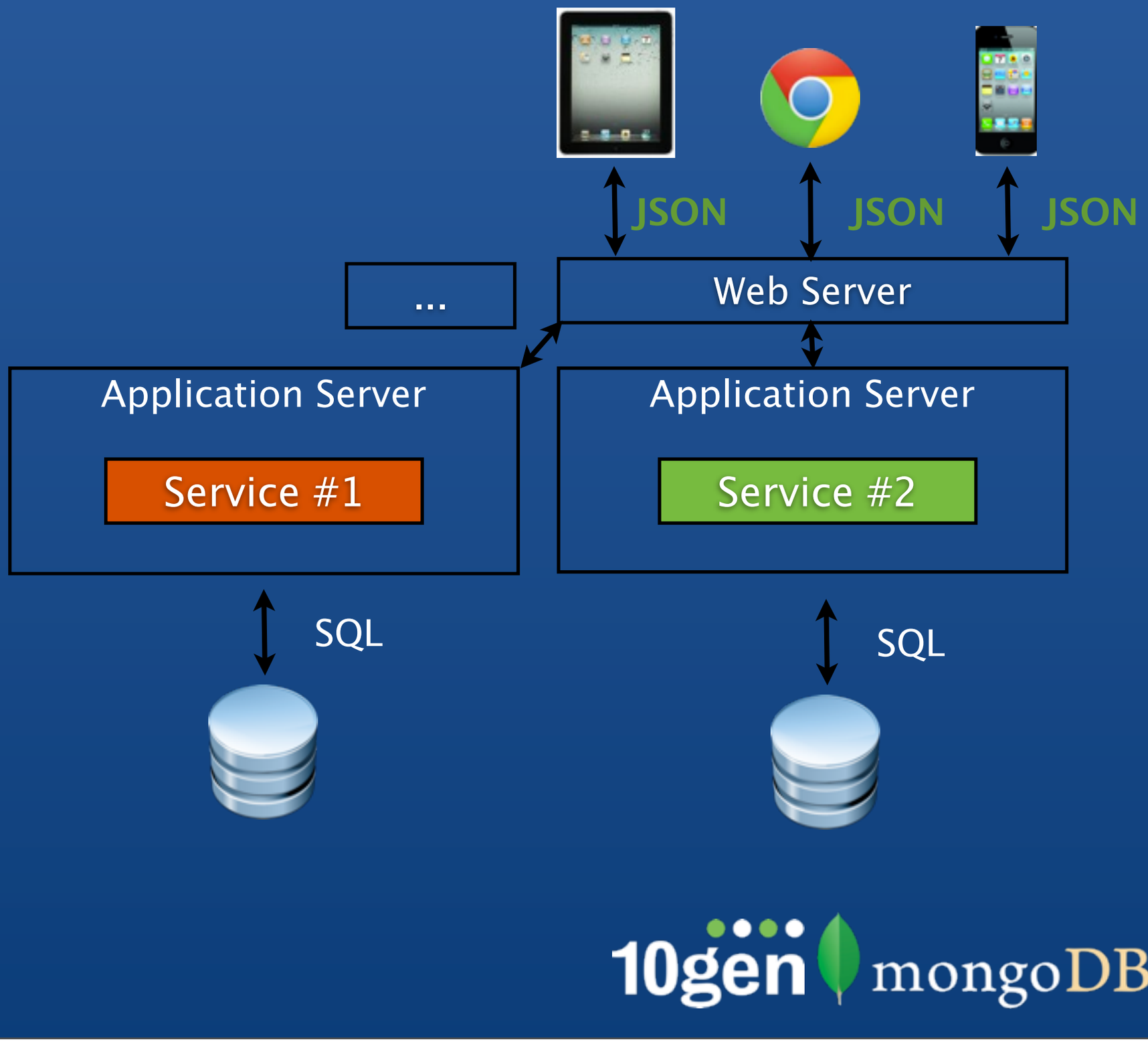
The Move to Services

The controllers have moved to the client

Expose small JSON Services

+ Death to the monolithic deployment

Service Architecture



The Service becomes the API

JSON Service becomes a API to client and other applications.

The clients are not bound to the underlaying data store

- Complexity due to mismatch between the JSON and SQL

10gen mongoDB

Migrating to MongoDB

10gen mongoDB

Here is a “simple” SQL Model

```
mysql> select * from book;
+----+-----------------------------------------------------+
| id | title                                               |
+----+-----------------------------------------------------+
|  1 | The Demon-Haunted World: Science as a Candle in the Dark |
|  2 | Cosmos                                              |
|  3 | Programming in Scala                                |
+----+-----------------------------------------------------+
3 rows in set (0.00 sec)

mysql> select * from bookauthor;
+---------+-----------+
| book_id | author_id |
+---------+-----------+
|       1 |         1 |
|       2 |         1 |
|       3 |         2 |
|       3 |         3 |
|       3 |         4 |
+---------+-----------+
5 rows in set (0.00 sec)

mysql> select * from author;
+----+-----------+------------+-------------+-------------+---------------+
| id | last_name | first_name | middle_name | nationality | year_of_birth |
+----+-----------+------------+-------------+-------------+---------------+
|  1 | Sagan     | Carl       | Edward      | NULL        | 1934          |
|  2 | Odersky   | Martin     | NULL        | DE          | 1958          |
|  3 | Spoon     | Lex        | NULL        | NULL        | NULL          |
|  4 | Venners   | Bill       | NULL        | NULL        | NULL          |
+----+-----------+------------+-------------+-------------+---------------+
4 rows in set (0.00 sec)
```

10gen mongoDB

The Same Data in MongoDB

```
{
    "_id" : ObjectId("4dfa6baa9c65dae09a4bbda5"),
    "title" : "Programming in Scala",
    "author" : [
        {
            "first_name" : "Martin",
            "last_name" : "Odersky",
            "nationality" : "DE",
            "year_of_birth" : 1958
        },
        {
            "first_name" : "Lex",
            "last_name" : "Spoon"
        },
        {
            "first_name" : "Bill",
            "last_name" : "Venners"
        }
    ]
}
```

10gen mongoDB

Aggregate Example – twitter

```
db.tweets.aggregate(
  {$match:
    {"user.friends_count":   { $gt: 0 },
     "user.followers_count": { $gt: 0 }
    }
  },
  {$project:
    { location:  "$user.location",
      friends:   "$user.friends_count",
      followers: "$user.followers_count"
    }
  },
  {$group:
    {_id:       "$location",
     friends:   {$sum: "$friends"},
     followers: {$sum: "$followers"}
    }
  }
);
```

Predicate

Parts of the document you want to project

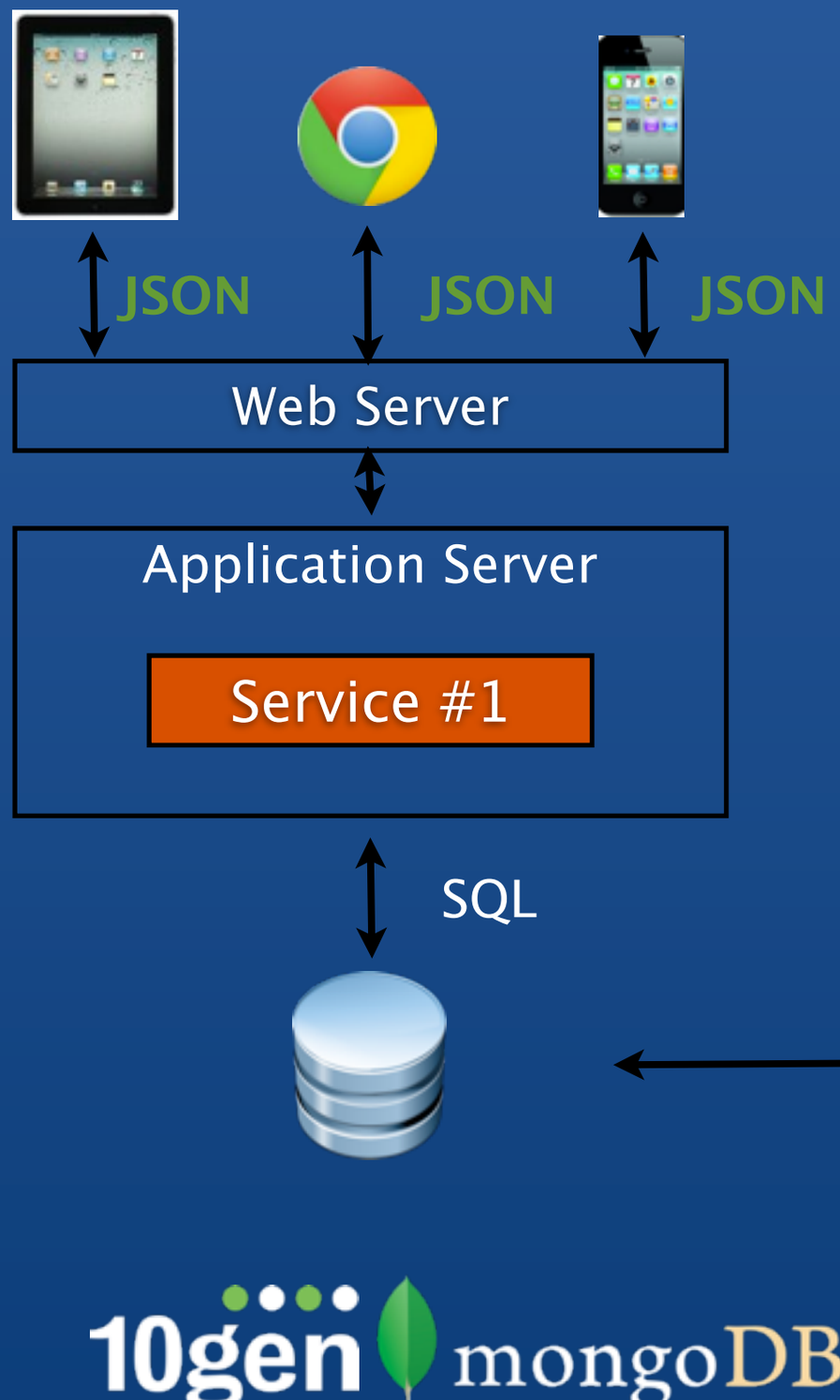
Function to apply to the result set

10gen

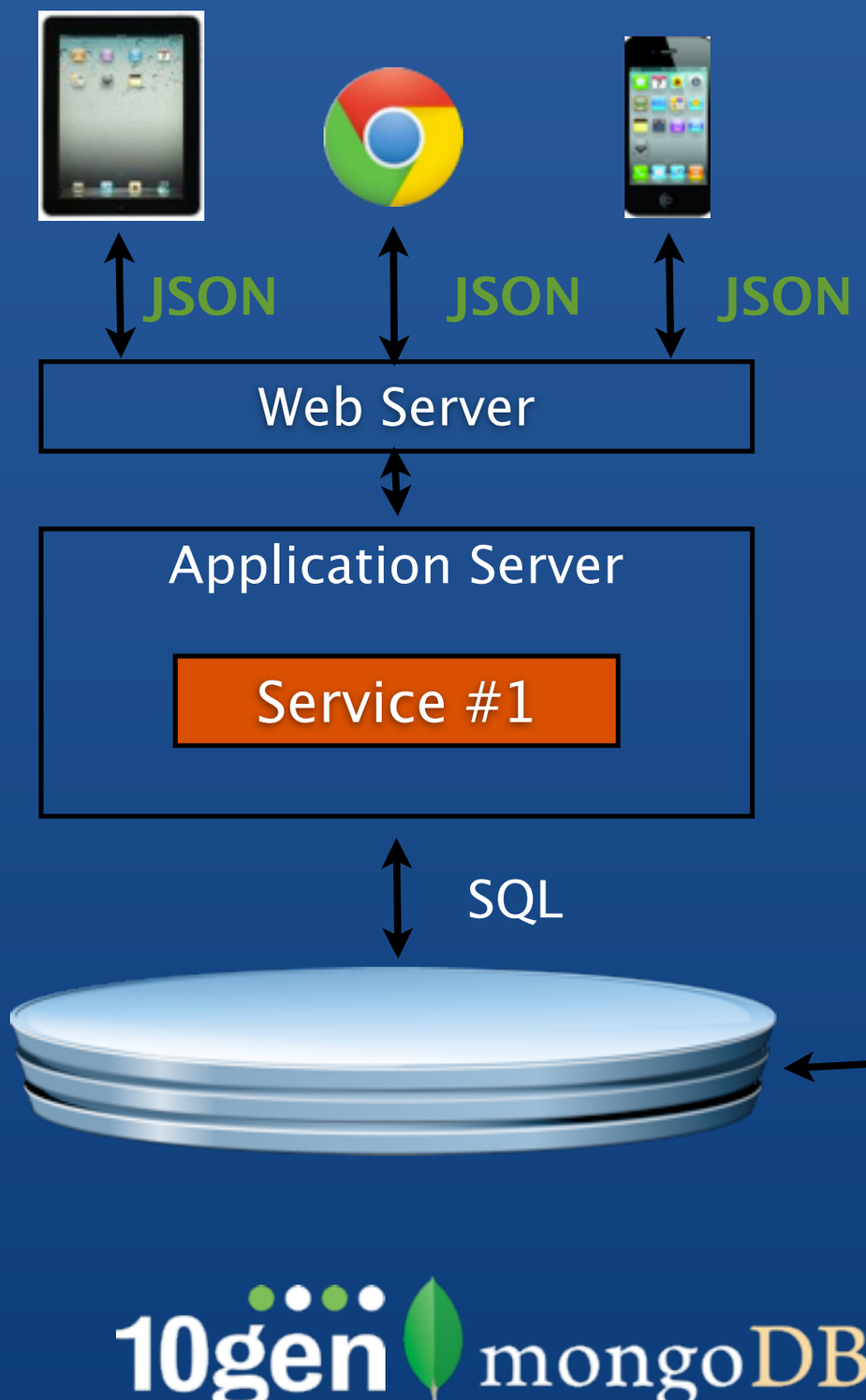
mongoDB

Challenge #2 - Write Volumes

Need to Scale Datasource

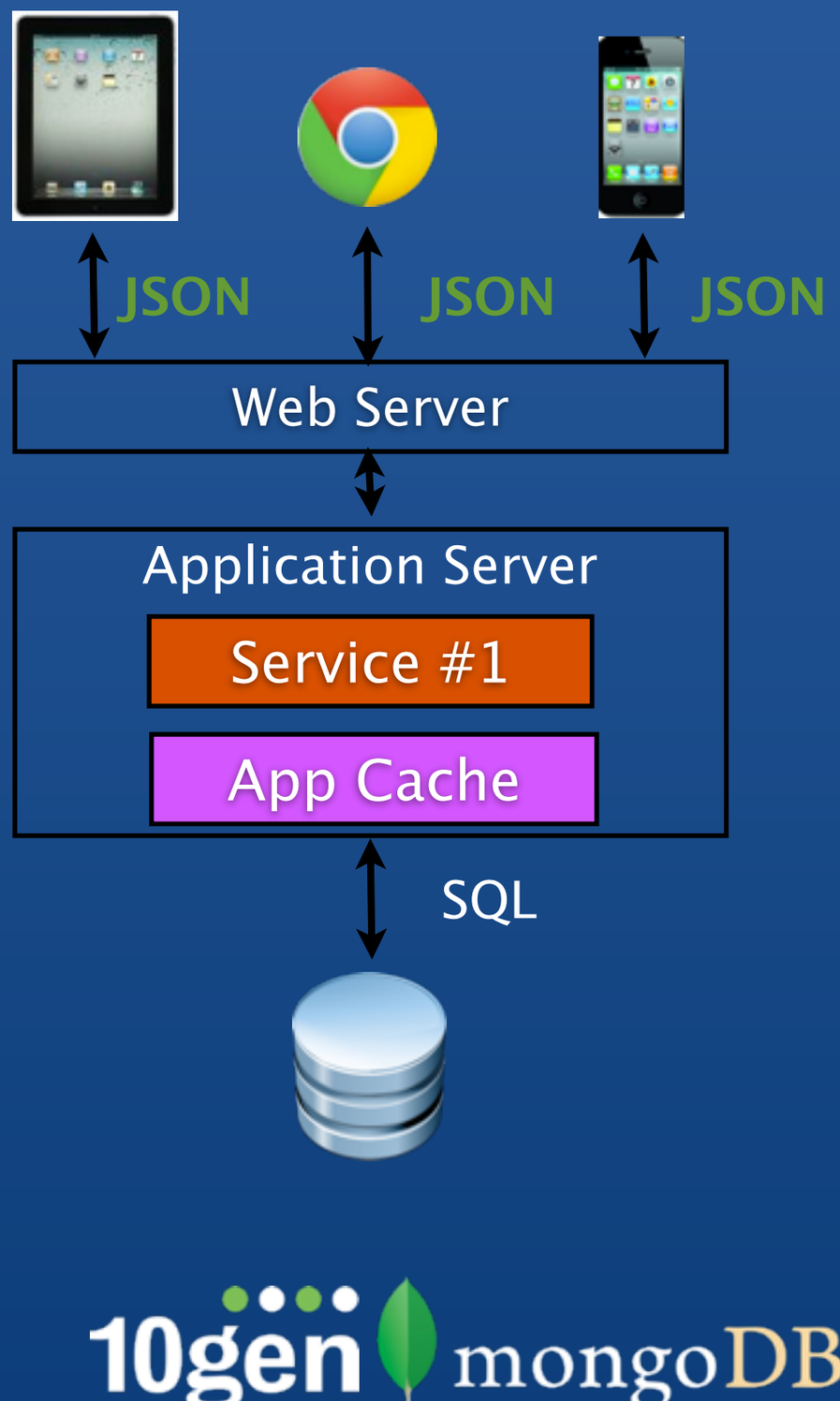


Need to Scale Datasource



10gen mongoDB

Application Cache?



10gen mongoDB

Issues

+ Read Only data comes from a Cache

- Writes slow down as need to update the Cache and the Database

- Need to keep cache data in sync between Application Servers

10gen mongoDB

http://community.qlikview.com/cfs-filesystemfile.ashx/__key/CommunityServer.Blogs.Components.WeblogFiles/theqlikviewblog/Cutting-Grass-with-Scissors-_2D00_-2.jpg

http://www.bitquill.net/blog/wp-content/uploads/2008/07/pack_of_harvesters.jpg

Big Data at a Glance

Large Dataset

Primary Key as “username”

10gen | the MongoDB company

mongoDB

Big Data at a Glance

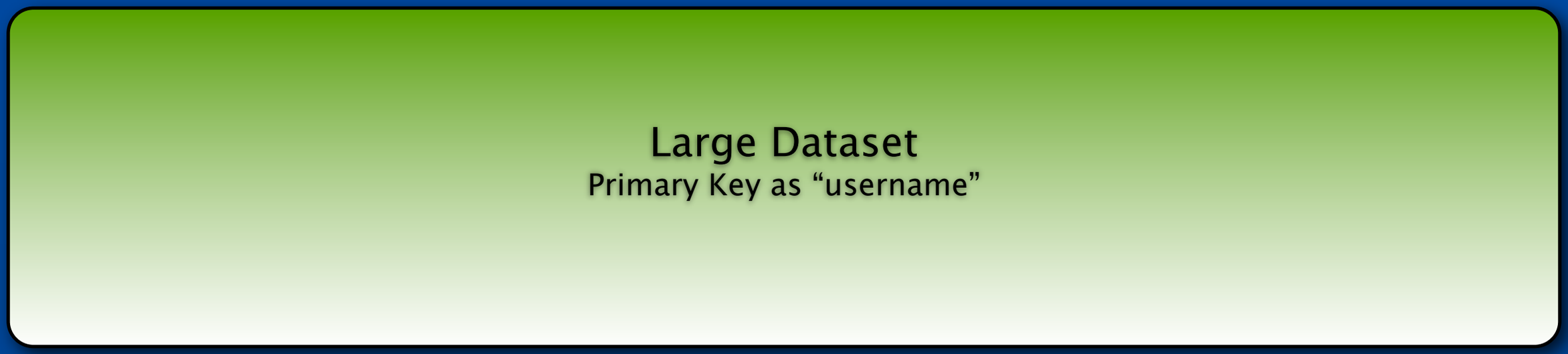
Large Dataset
Primary Key as “username”

- Systems like Google File System (which inspired Hadoop’s HDFS) and MongoDB’s Sharding handle the scale problem by chunking

10gen | the MongoDB company

mongoDB

Big Data at a Glance



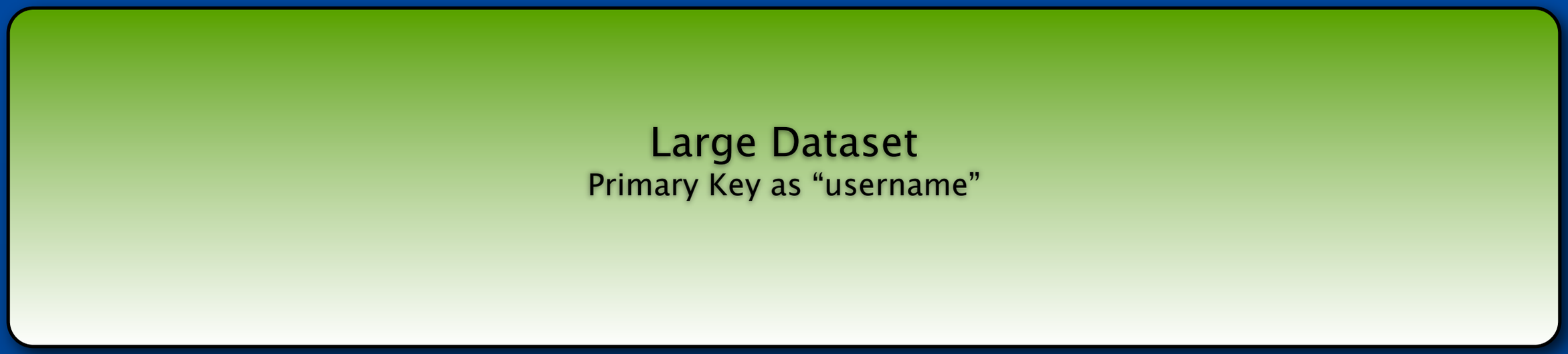
- Systems like Google File System (which inspired Hadoop’s HDFS) and MongoDB’s Sharding handle the scale problem by chunking

- Break up pieces of data into smaller chunks, spread across many data nodes

10gen | the MongoDB company

mongoDB

Big Data at a Glance



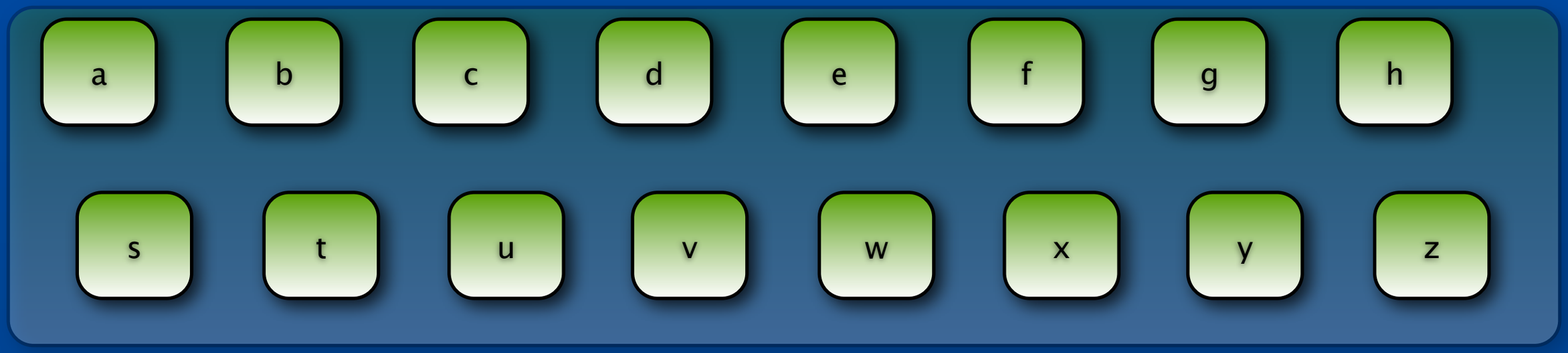
- Systems like Google File System (which inspired Hadoop’s HDFS) and MongoDB’s Sharding handle the scale problem by chunking

- Break up pieces of data into smaller chunks, spread across many data nodes
    - Each data node contains many chunks

10gen | the MongoDB company

mongoDB

Big Data at a Glance



- Systems like Google File System (which inspired Hadoop's HDFS) and MongoDB's Sharding handle the scale problem by chunking

- Break up pieces of data into smaller chunks, spread across many data nodes
  - Each data node contains many chunks
  - If a chunk gets too large or a node overloaded, data can be rebalanced

10gen | the MongoDB company

mongoDB

Big Data at a Glance

10gen | the MongoDB company

mongoDB

Big Data at a Glance

Large Dataset

Primary Key as “username”

x b v t d f z s

h e u c w a y g

MongoDB Sharding (as well as HDFS) breaks data into chunks (~64 mb)

10gen | the MongoDB company

mongoDB

Scaling

Large Dataset

Primary Key as “username”

x b v t d f z s

h e u c w a y g

Representing data as chunks allows many levels of scale across n data nodes

10gen | the MongoDB company

mongoDB

Scaling

Data Node 1

25% of chunks

Data Node 2

25% of chunks

Data Node 3

25% of chunks

Data Node 4

25% of chunks

x b v t d f z s

h e u c w a y g

Representing data as chunks allows many levels of scale across n data nodes

10gen | the MongoDB company

mongoDB

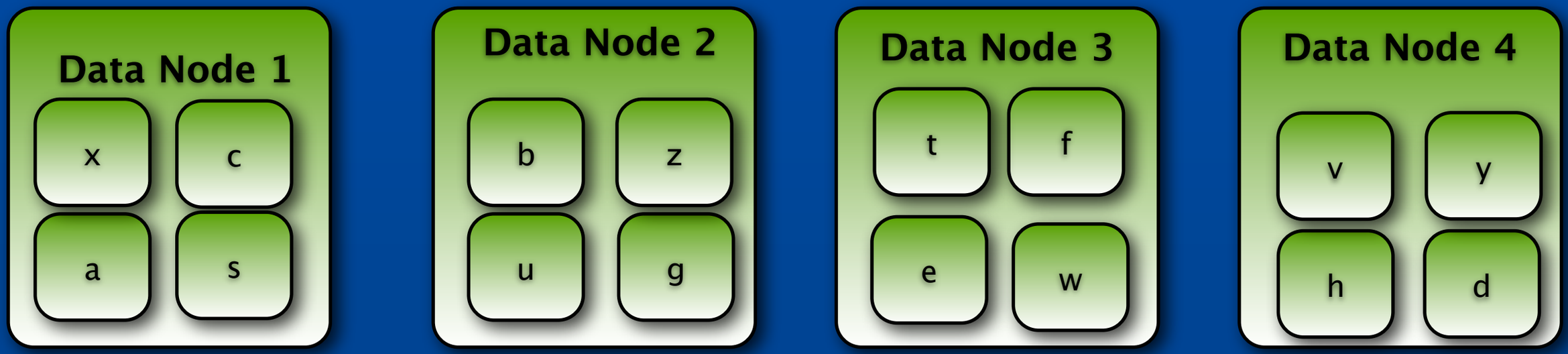
Scaling

The set of chunks can be evenly distributed across n data nodes

10gen | the MongoDB company

mongoDB

Scaling

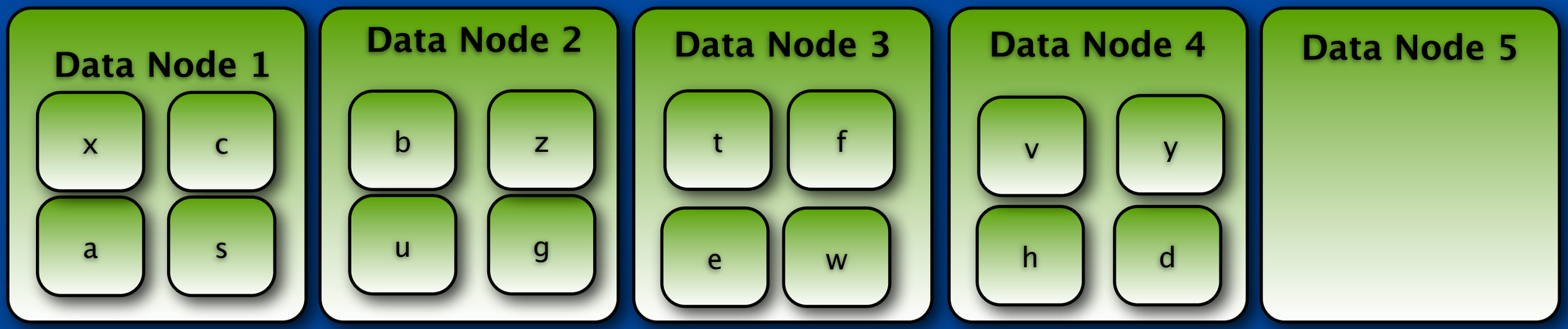


The set of chunks can be evenly distributed across n data nodes

10gen | the MongoDB company

mongoDB

Add Nodes: Chunk Rebalancing

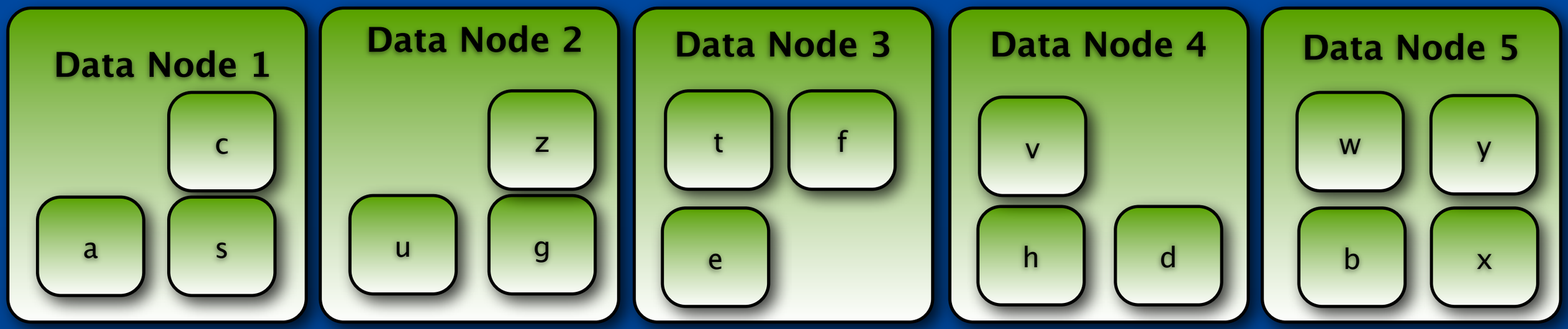


The goal is equilibrium – an equal distribution.
As nodes are added (or even removed)
chunks can be redistributed for balance.

10gen | the MongoDB company

mongoDB

Add Nodes: Chunk Rebalancing

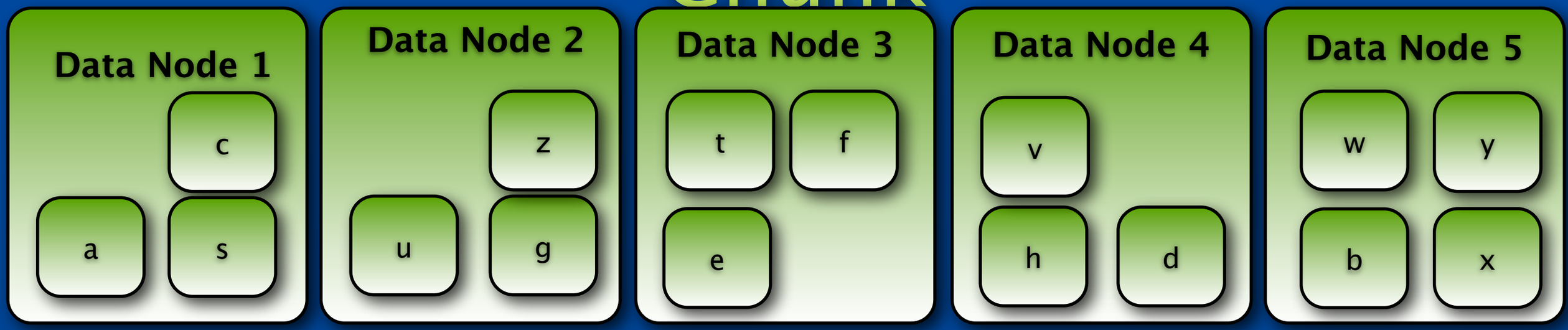


The goal is equilibrium – an equal distribution.
As nodes are added (or even removed)
chunks can be redistributed for balance.

10gen | the MongoDB company

mongoDB

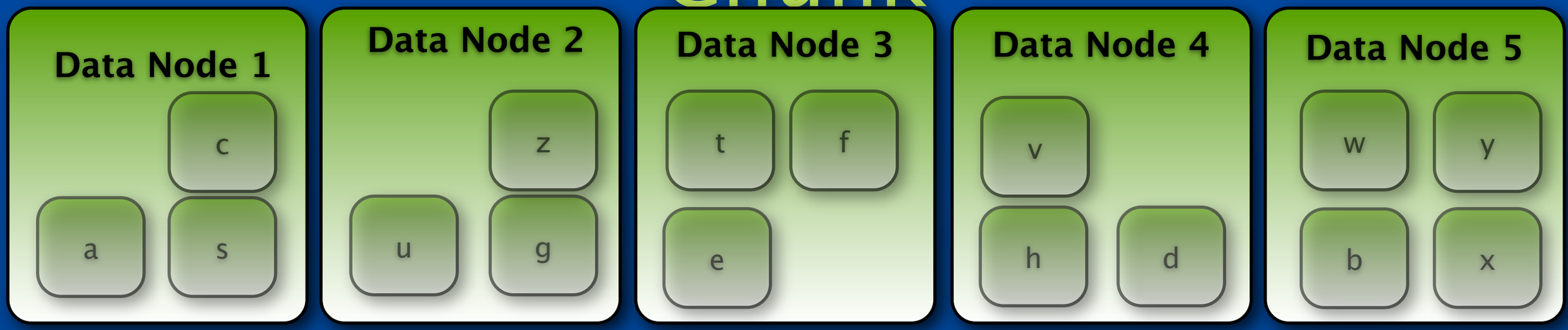
Writes Routed to Appropriate Chunk



10gen | the MongoDB company

mongoDB

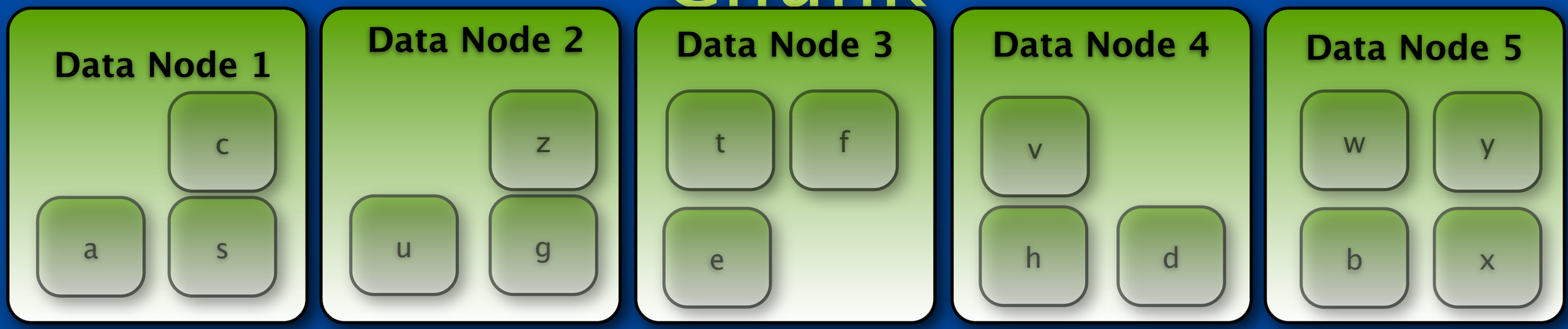
Writes Routed to Appropriate Chunk



10gen | the MongoDB company

mongoDB

Writes Routed to Appropriate Chunk

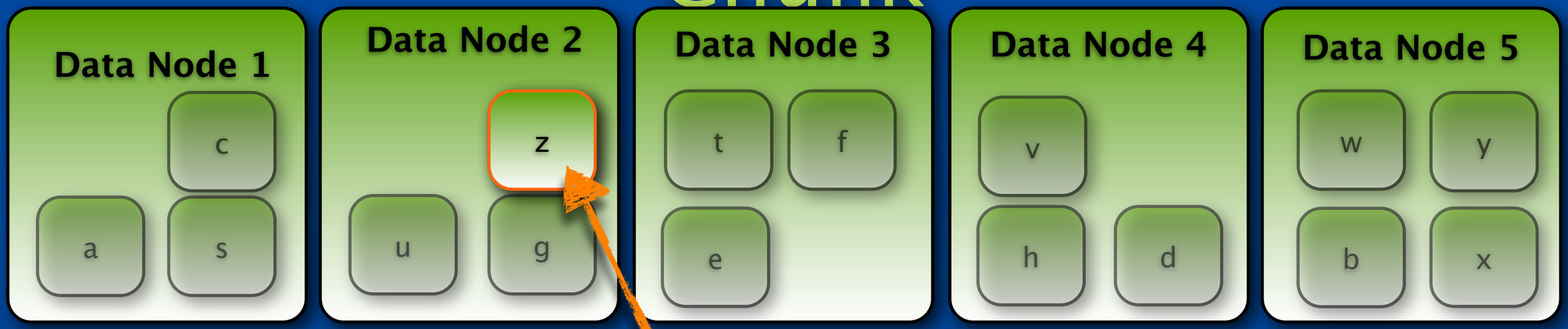


Writes are efficiently routed to the appropriate node & chunk

10gen | the MongoDB company

mongoDB

Writes Routed to Appropriate Chunk



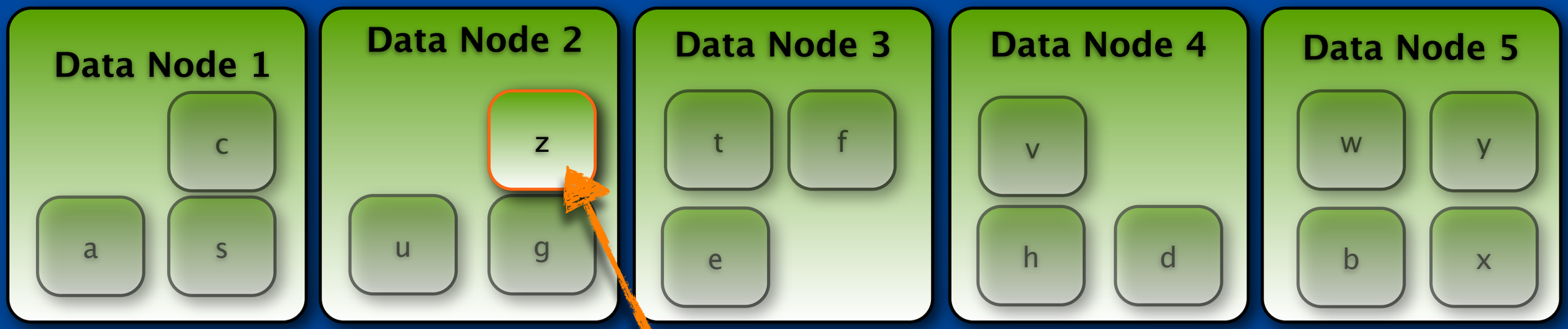
Write to key“ziggy”

Writes are efficiently routed to the appropriate node & chunk

10gen | the MongoDB company

mongoDB

Chunk Splitting & Balancing



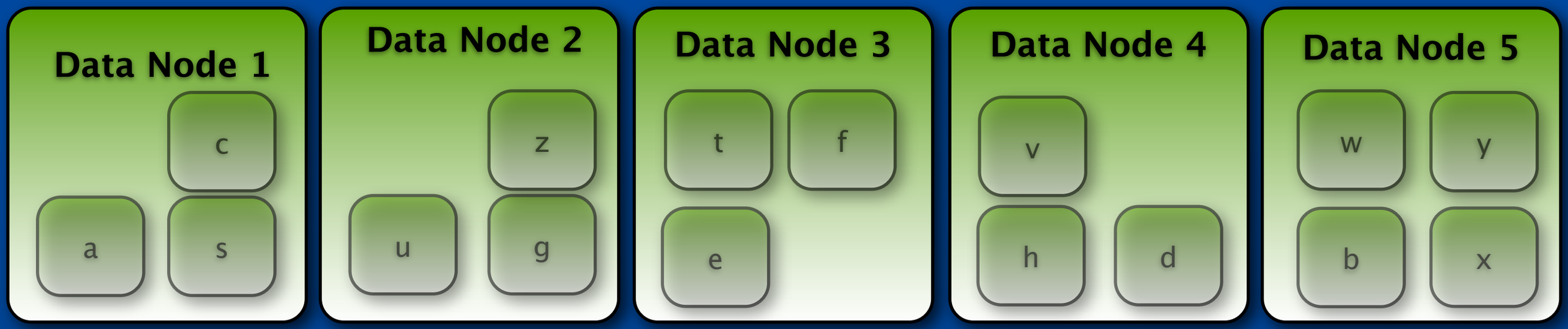
Write to key“ziggy”

If a chunk gets too large (default in MongoDB – 64mb per chunk),
It is split into two new chunks

10gen | the MongoDB company

mongoDB

Chunk Splitting & Balancing

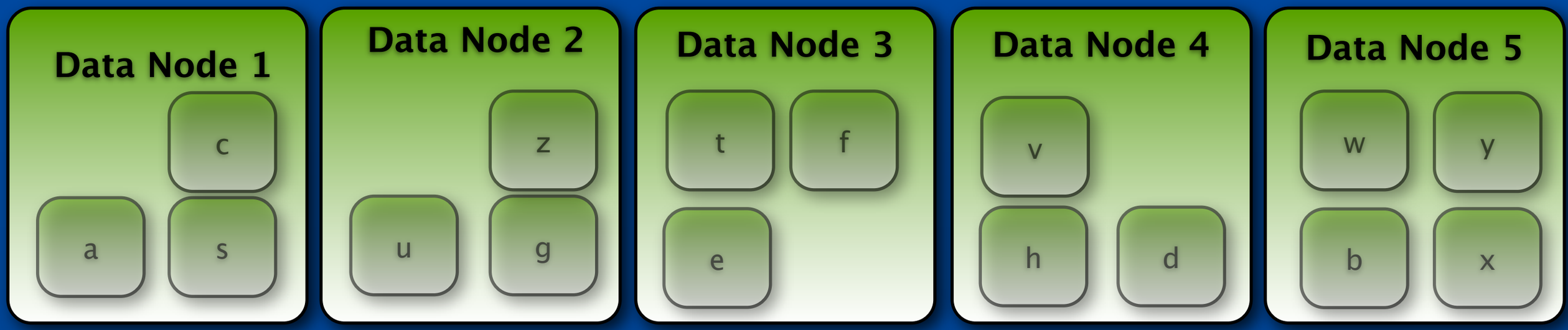


If a chunk gets too large (default in MongoDB – 64mb per chunk),
It is split into two new chunks

10gen | the MongoDB company

mongoDB

Chunk Splitting & Balancing

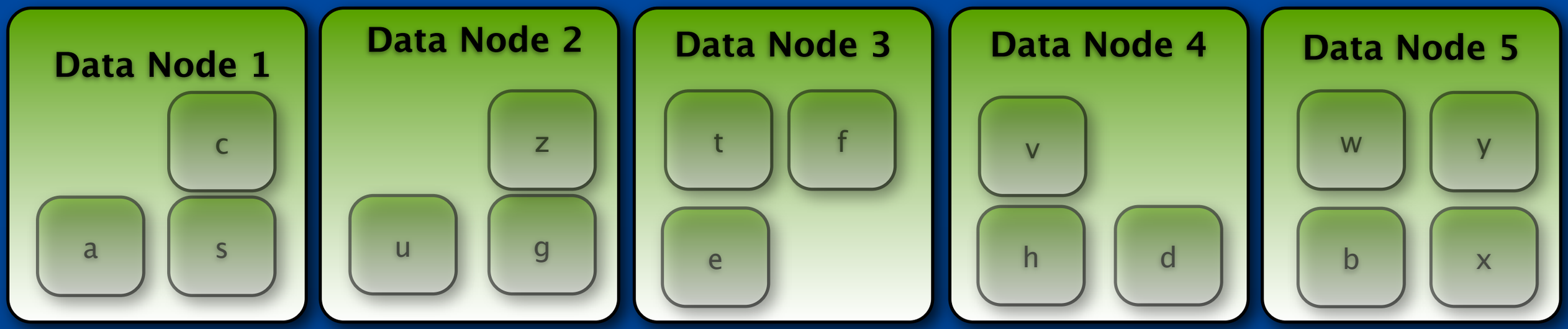


If a chunk gets too large (default in MongoDB – 64mb per chunk),
It is split into two new chunks

10gen | the MongoDB company

mongoDB

Chunk Splitting & Balancing

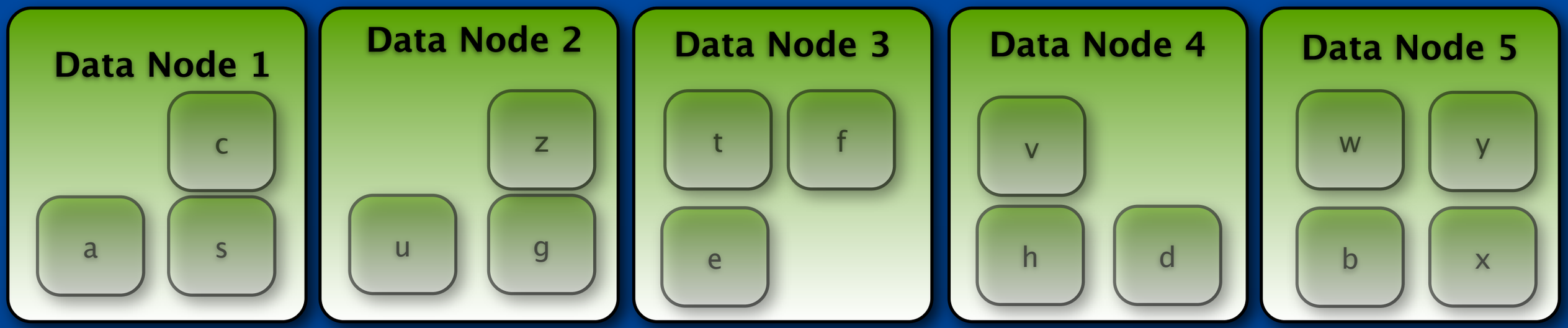


If a chunk gets too large (default in MongoDB – 64mb per chunk),
It is split into two new chunks

10gen | the MongoDB company

mongoDB

Chunk Splitting & Balancing

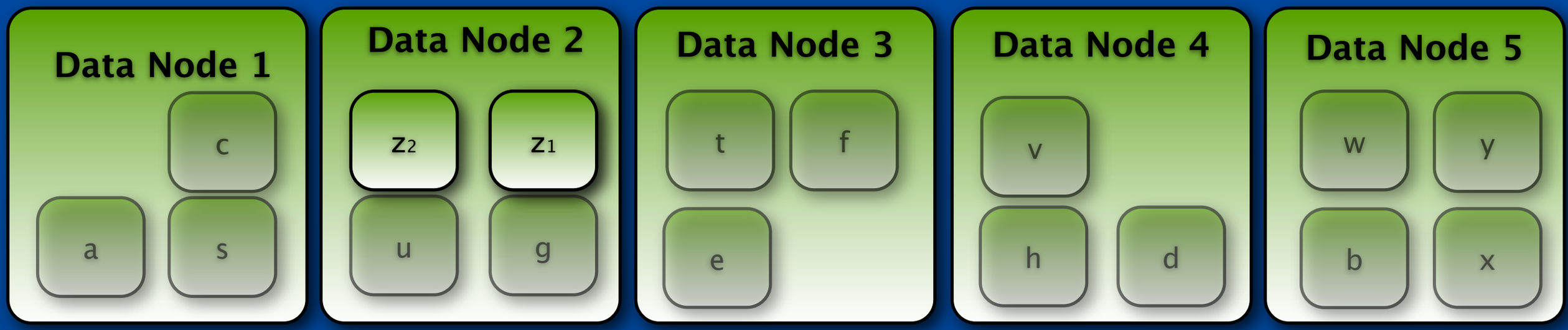


If a chunk gets too large (default in MongoDB – 64mb per chunk),
It is split into two new chunks

10gen | the MongoDB company

mongoDB

Chunk Splitting & Balancing

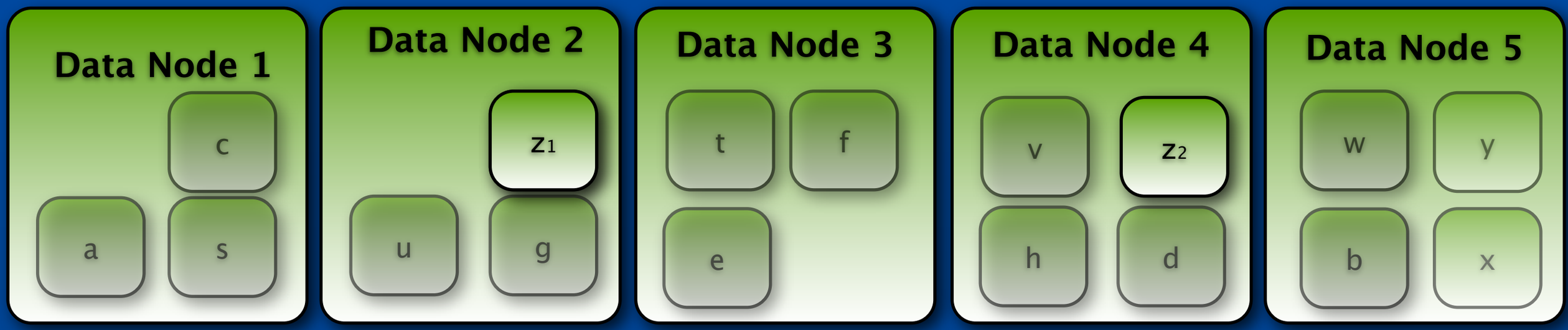


Each new part of the **Z** chunk (left & right) now contains half of the keys

10gen | the MongoDB company

mongoDB

Chunk Splitting & Balancing

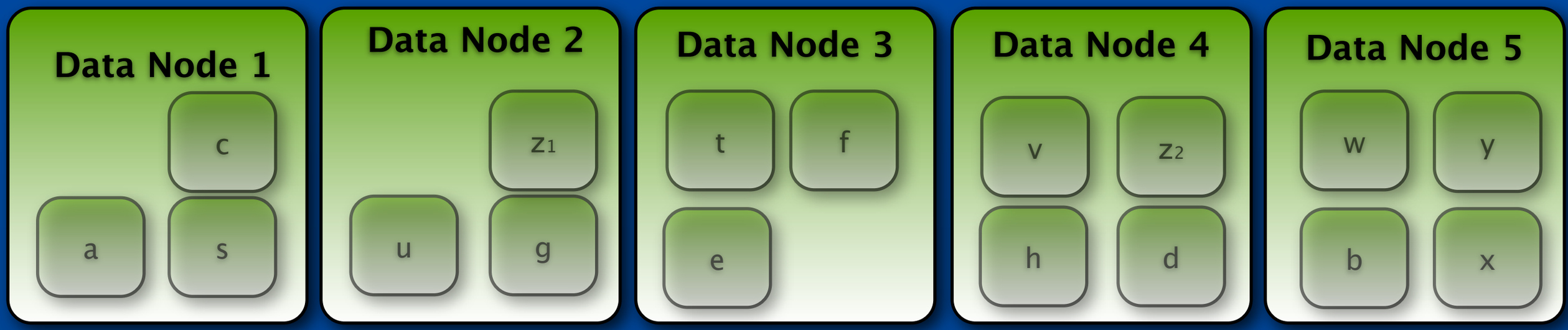


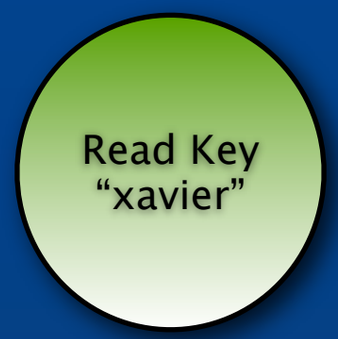
As chunks continue to grow and split, they can be rebalanced to keep an equal share of data on each server.

10gen | the MongoDB company

mongoDB

Reads with Key Routed Efficiently





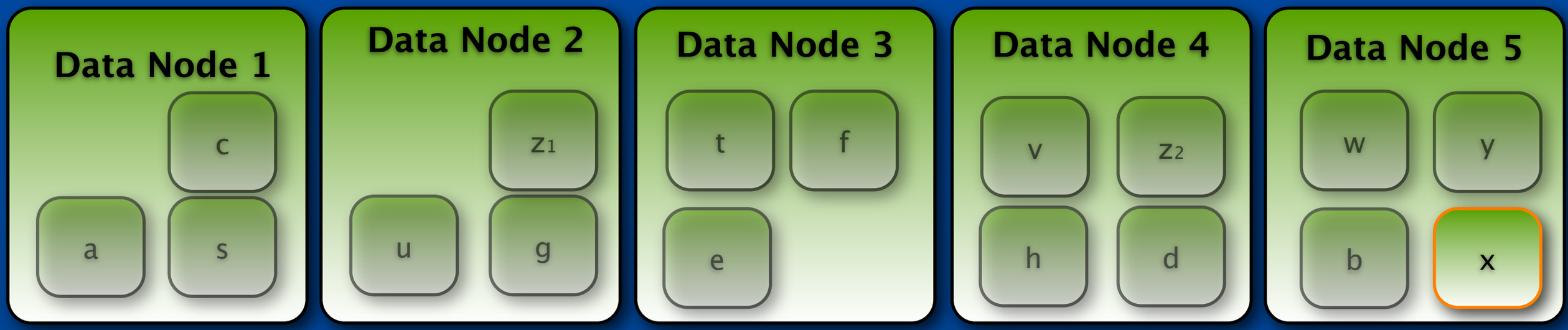
Reading a single value by Primary Key

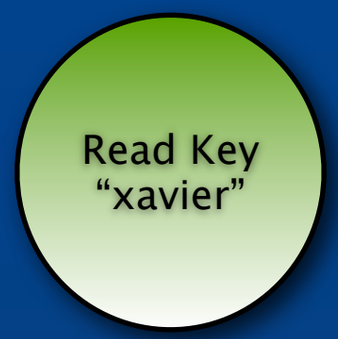
Read routed efficiently to specific chunk containing key

10gen | the MongoDB company

mongoDB

Reads with Key Routed Efficiently





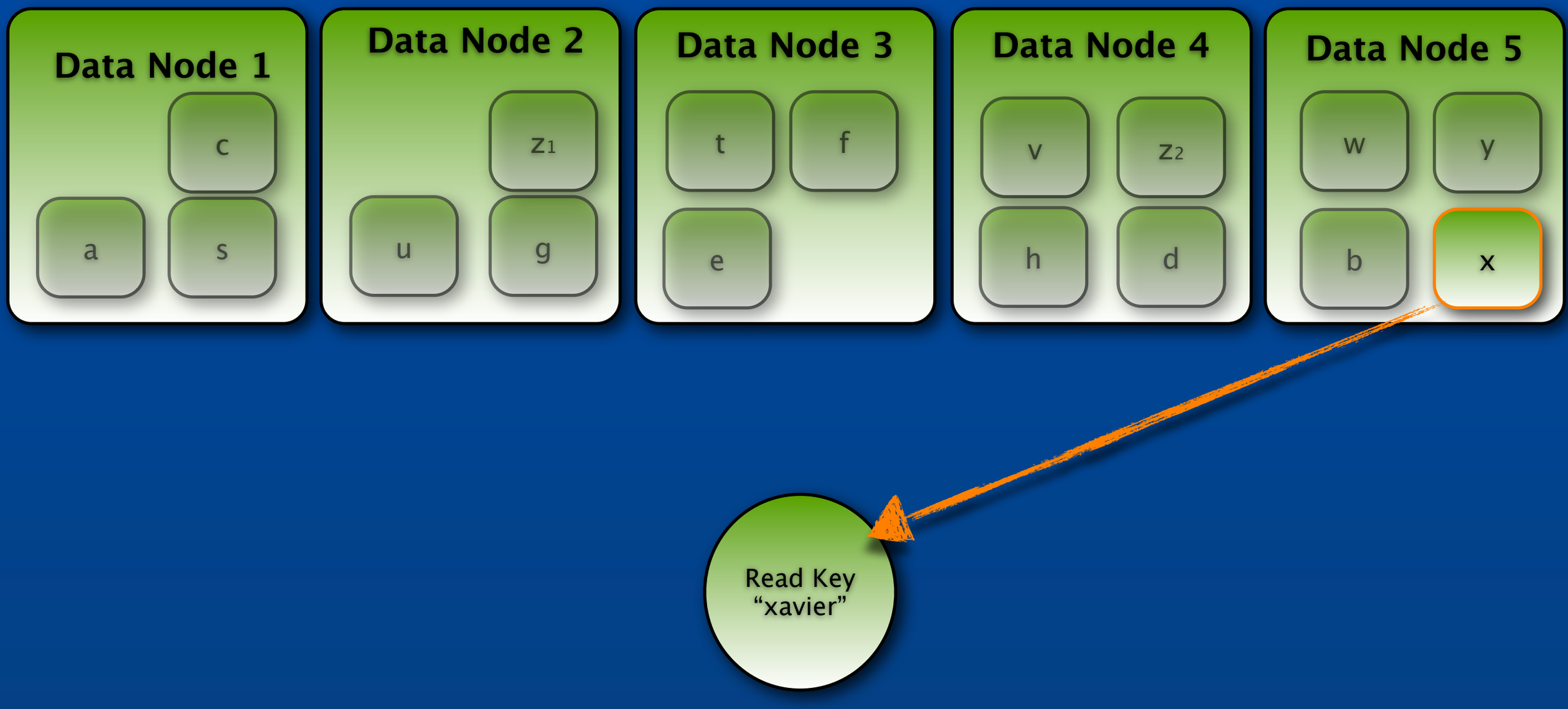
Reading a single value by Primary Key

Read routed efficiently to specific chunk containing key

10gen | the MongoDB company

mongoDB

Reads with Key Routed Efficiently



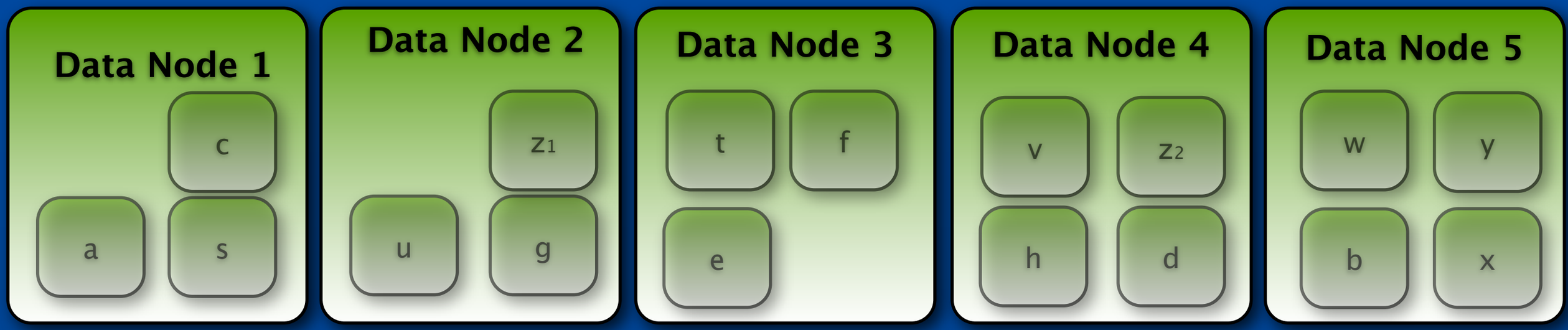
Reading a single value by Primary Key

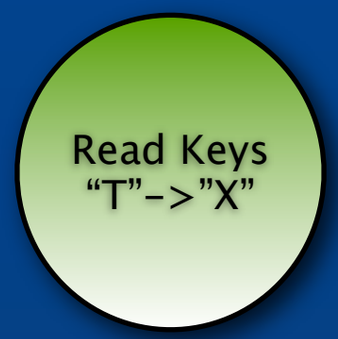
Read routed efficiently to specific chunk containing key

10gen | the MongoDB company

mongoDB

Reads with Key Routed Efficiently





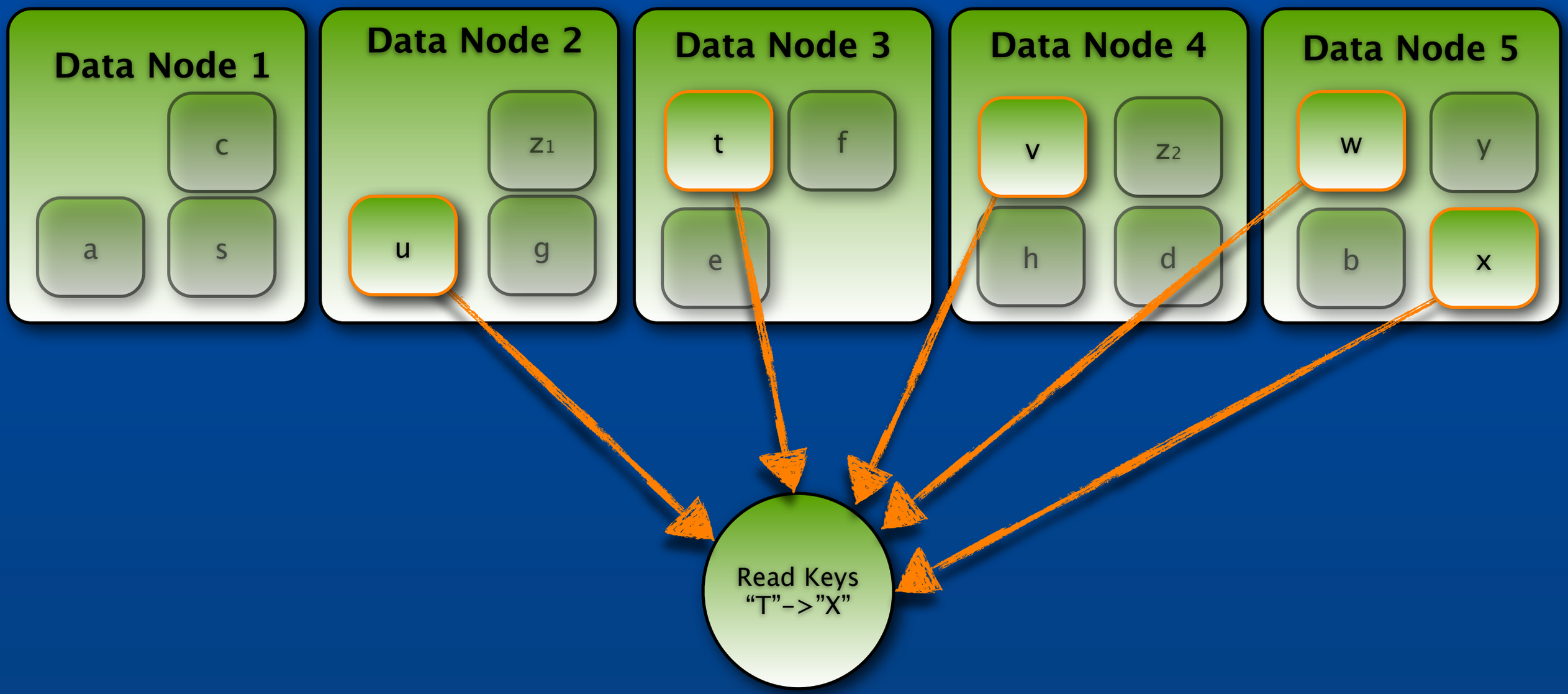
Reading multiple values by Primary Key

Reads routed efficiently to specific chunks in range

10gen | the MongoDB company

mongoDB

Reads with Key Routed Efficiently



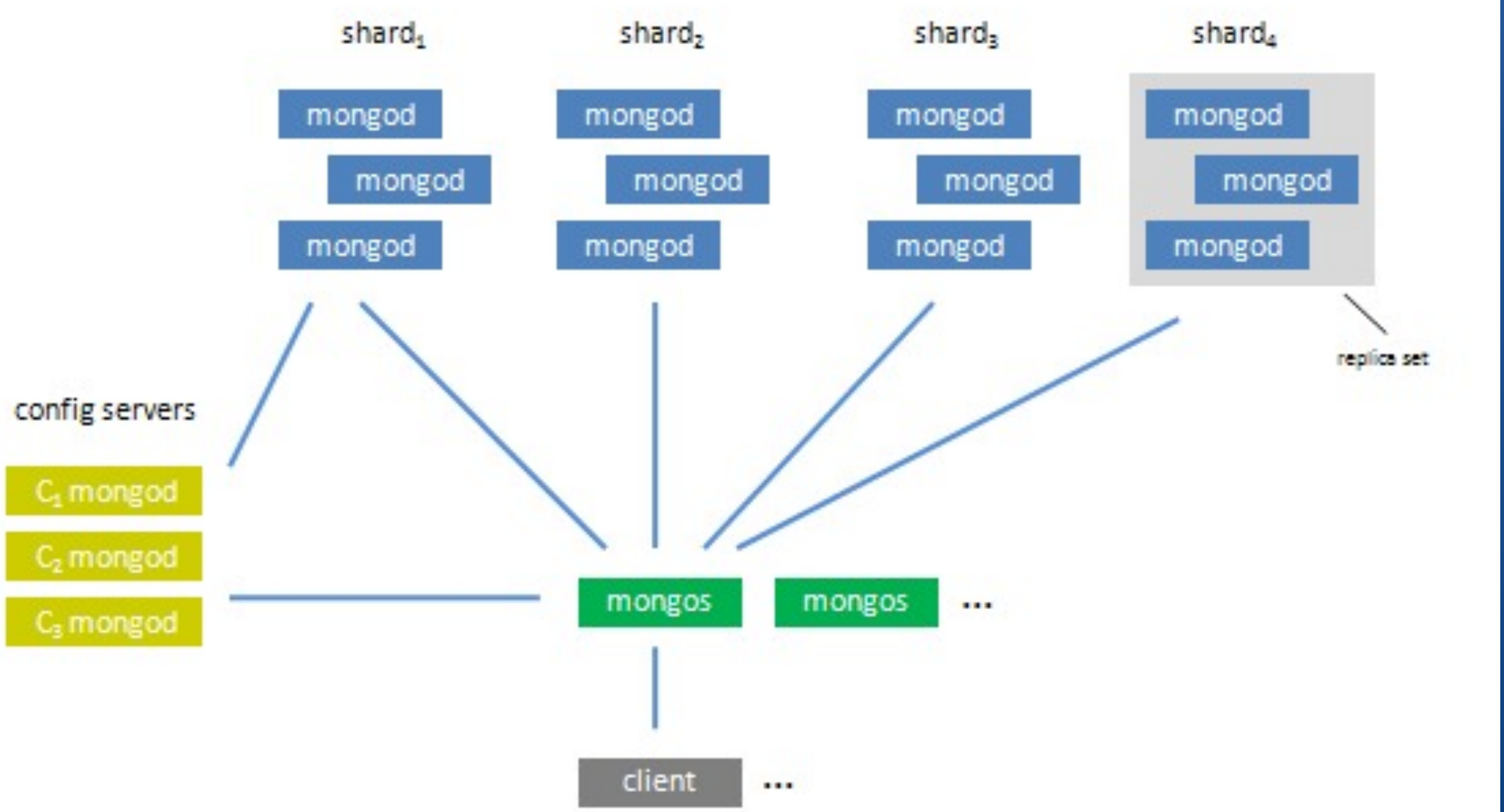
Reading multiple values by Primary Key

Reads routed efficiently to specific chunks in range

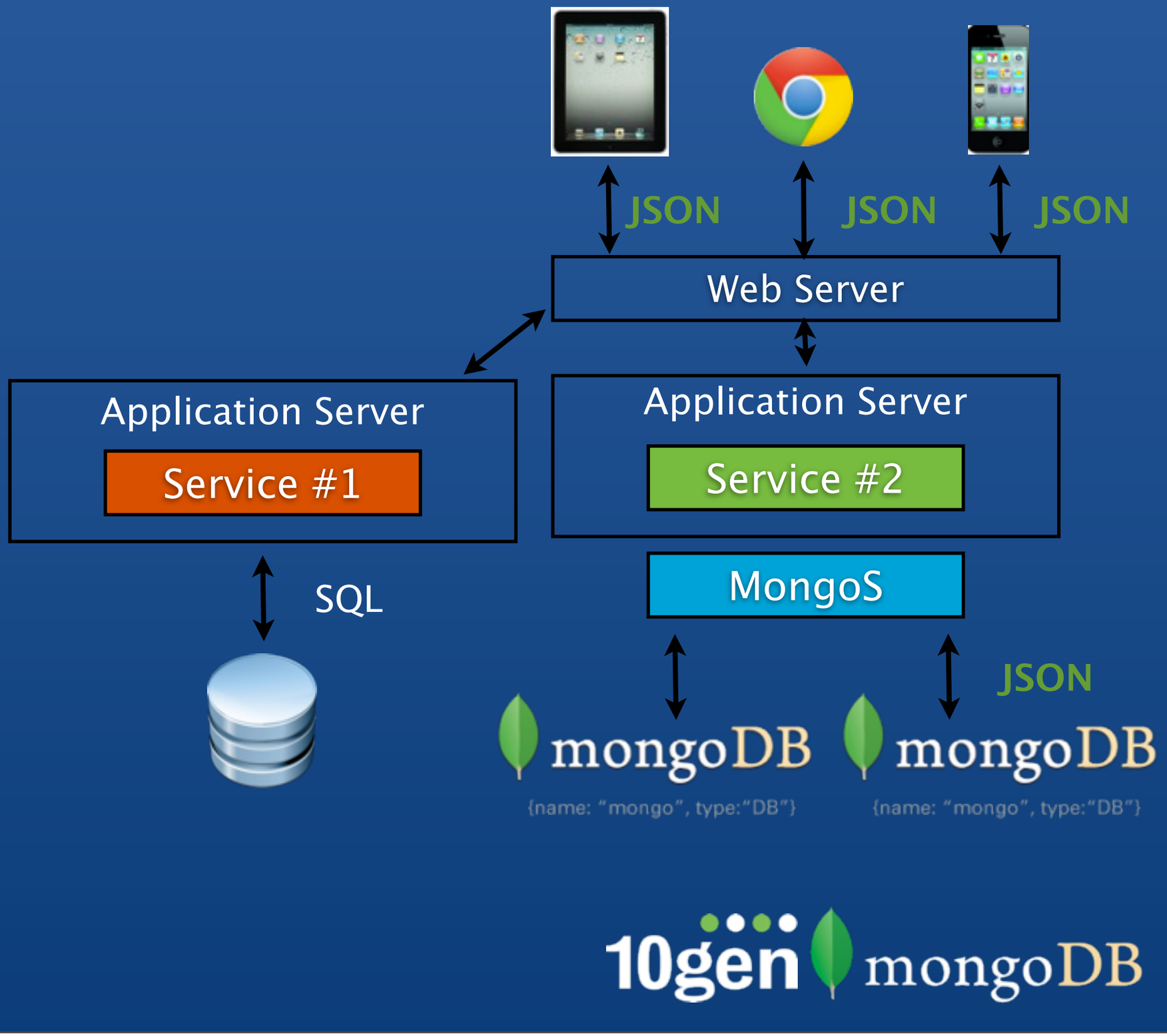
10gen | the MongoDB company

mongoDB

Architecture

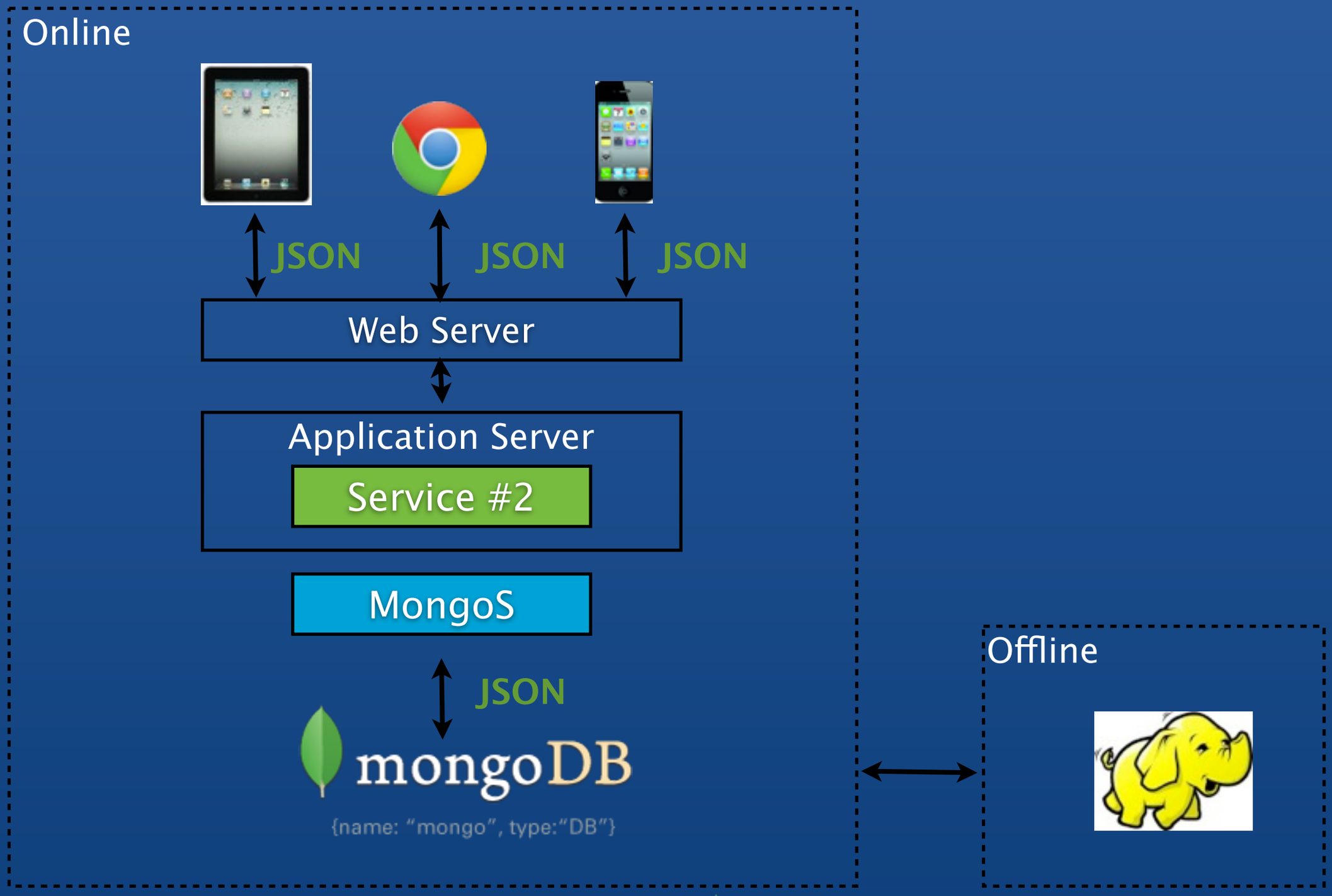


Adding MongoS



Challenge #3 -Offline Processing

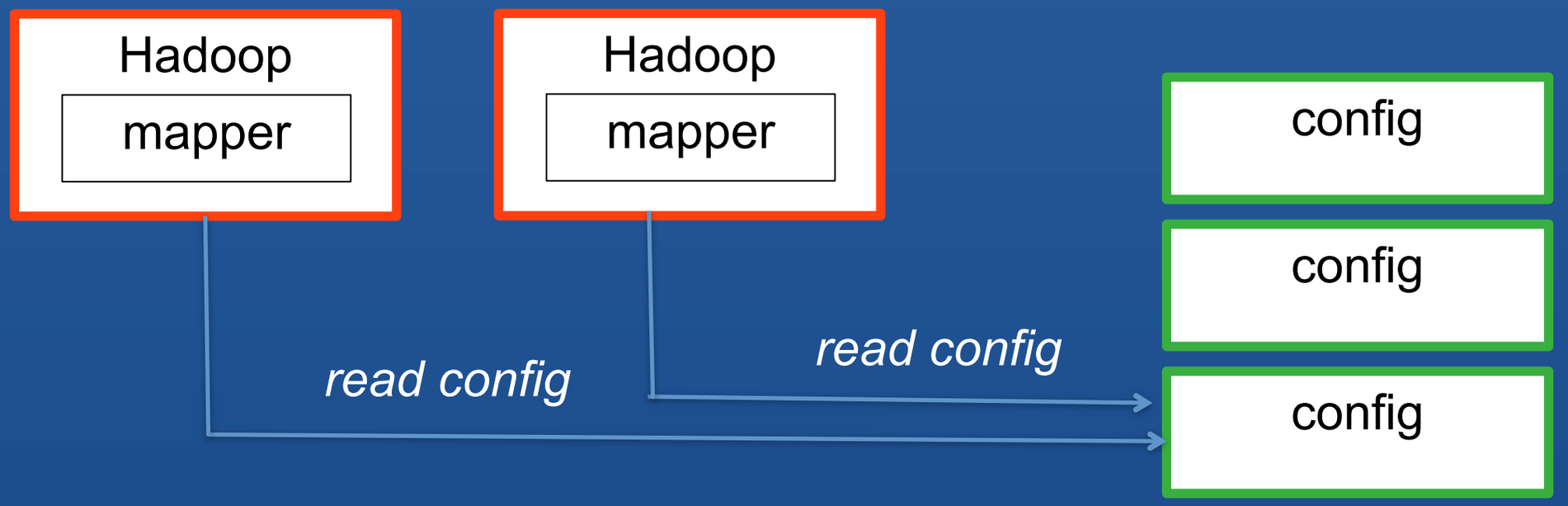
Online / Offline

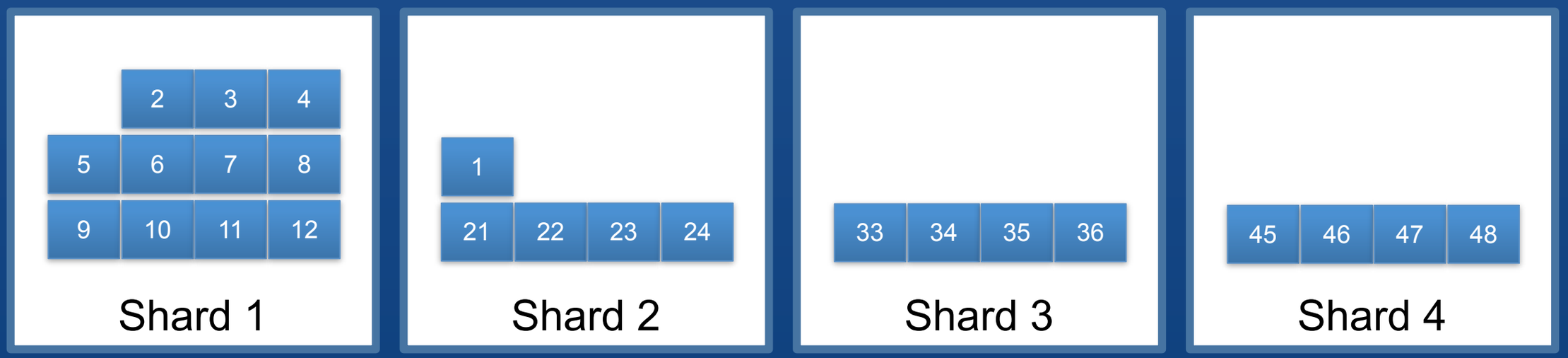


10gen mongoDB

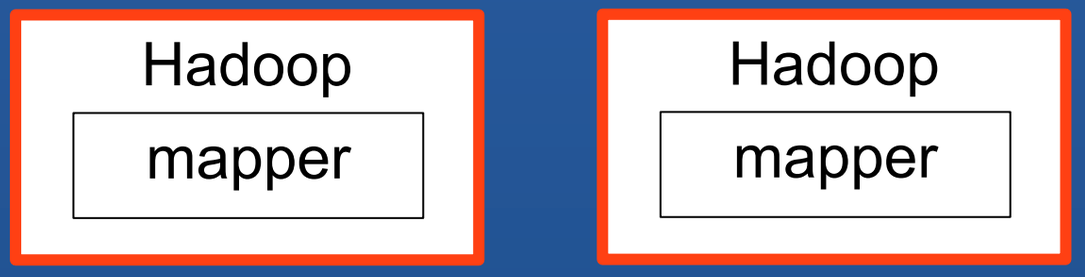
MongoDB and Hadoop

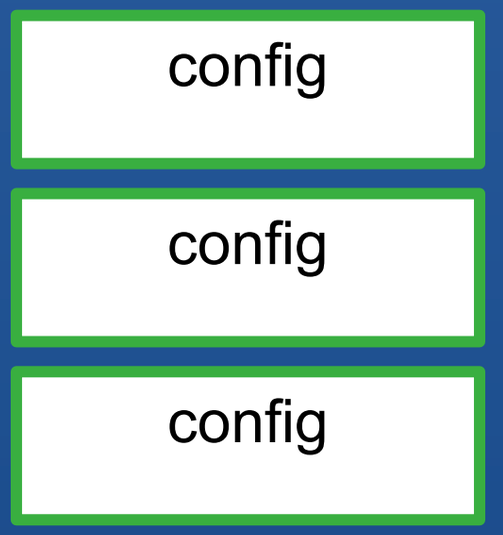
MongoDB & Hadoop

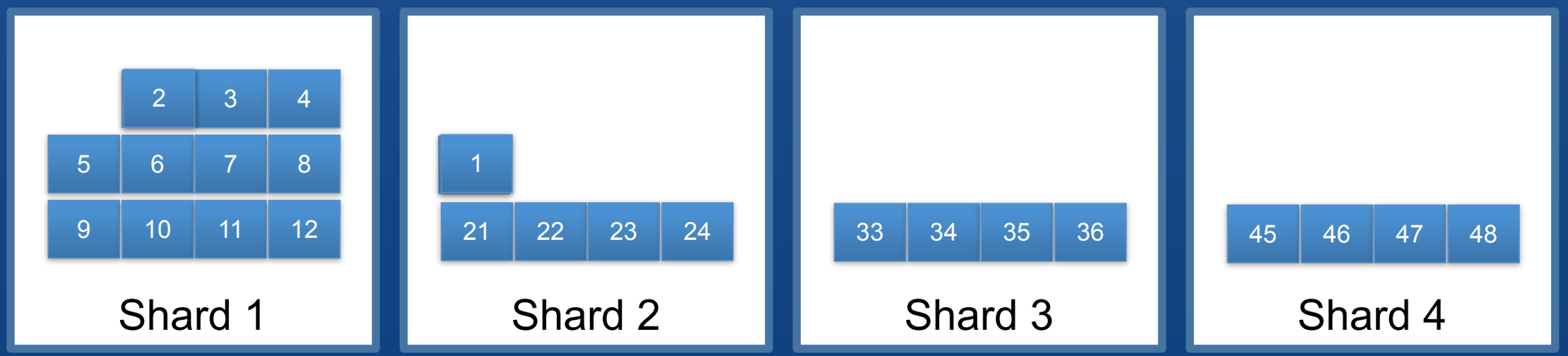




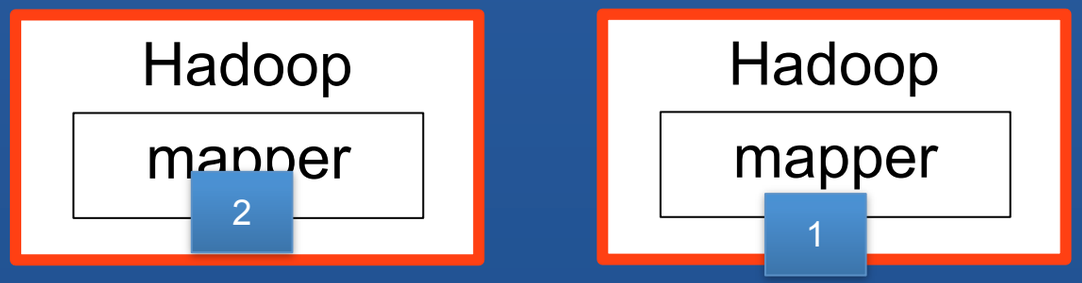
MongoDB & Hadoop

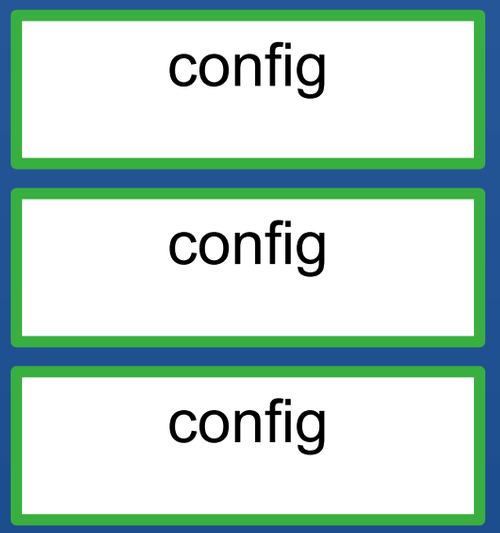


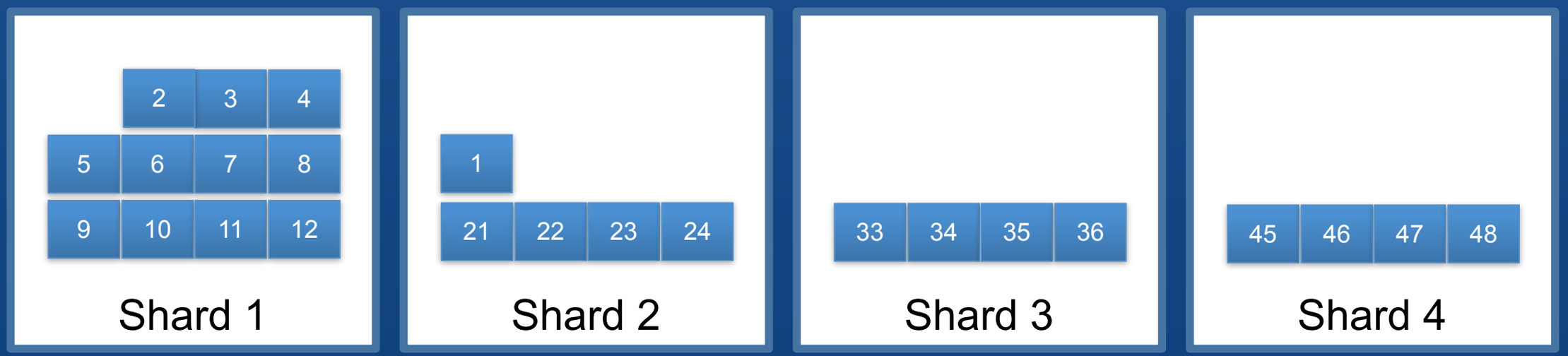




MongoDB & Hadoop







Word Count - Map

Classic Hadoop

```
public void map(LongWritable key, Text value, Context context) throws ..{
        String line = value.toString();
        StringTokenizer tokenizer = new StringTokenizer(line);
        while (tokenizer.hasMoreTokens()) {
            word.set(tokenizer.nextToken());
            context.write(word, one);
        }
}
```

MongoDB Hadoop Adapter

```
public void map(Object key, BSONObject value, Context context ) throws ....{
    StringTokenizer itr = new StringTokenizer( value.get( "line" ).toString() );
    while (tokenizer.hasMoreTokens()) {
        word.set(tokenizer.nextToken());
        context.write(word, one);
    }
}
```

10gen

mongoDB

Word Count - Reduce

Classic Hadoop

```
 public void reduce( Text key, Iterable<IntWritable> values, Context context )
            throws IOException, InterruptedException{
        int sum = 0;
        for ( final IntWritable val : values ){
            sum += val.get();
        }
        context.write( key, new IntWritable(sum));
}
```

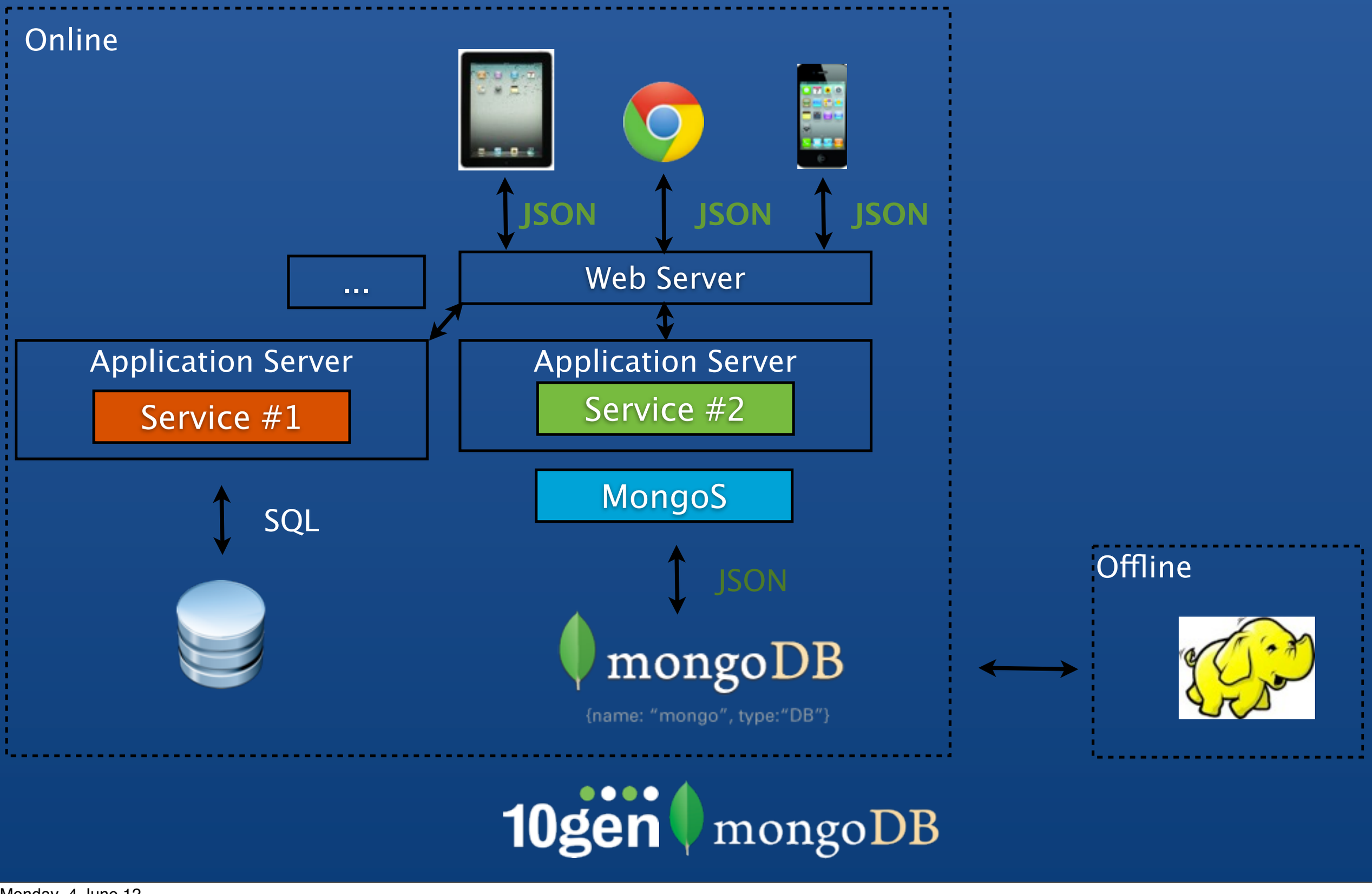
MongoDB Hadoop Adapter is the same code!

10gen

mongoDB

New Hybrid World

New Hybrid World



download at mongodb.org

We're Hiring !

Chris Harris
Email : charris@10gen.com
Twitter : cj_harris5

conferences, appearances
http://www.10gen.com/events