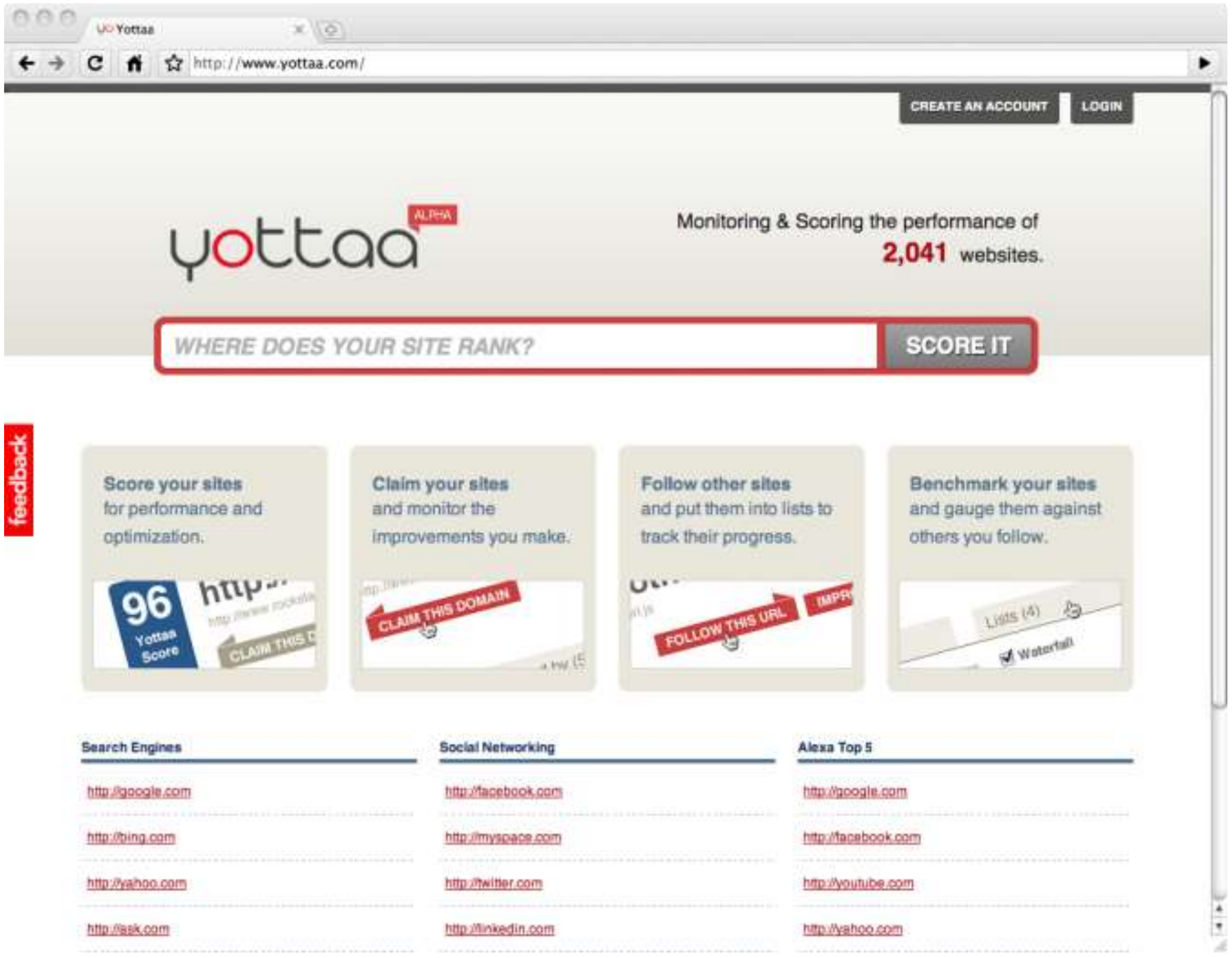
Scalable Event Analytics with Ruby on Rails &MongoDB

Ruby Conf China 2010

Jared Rosoff (@forjared)
jrosoff@yottaa.com

Yottaa!!!! (www.yottaa.com)



Monitoring & Scoring the performance of **2,041** websites.

Score your sites for performance and optimization.

Claim your sites and monitor the improvements you make.

Follow other sites and put them into lists to track their progress.

Benchmark your sites and gauge them against others you follow.

Search Engines	Social Networking	Alexa Top 5
http://google.com	http://facebook.com	http://google.com
http://bing.com	http://myspace.com	http://facebook.com
http://yahoo.com	http://twitter.com	http://youtube.com
http://ask.com	http://linkedin.com	http://yahoo.com

Overview

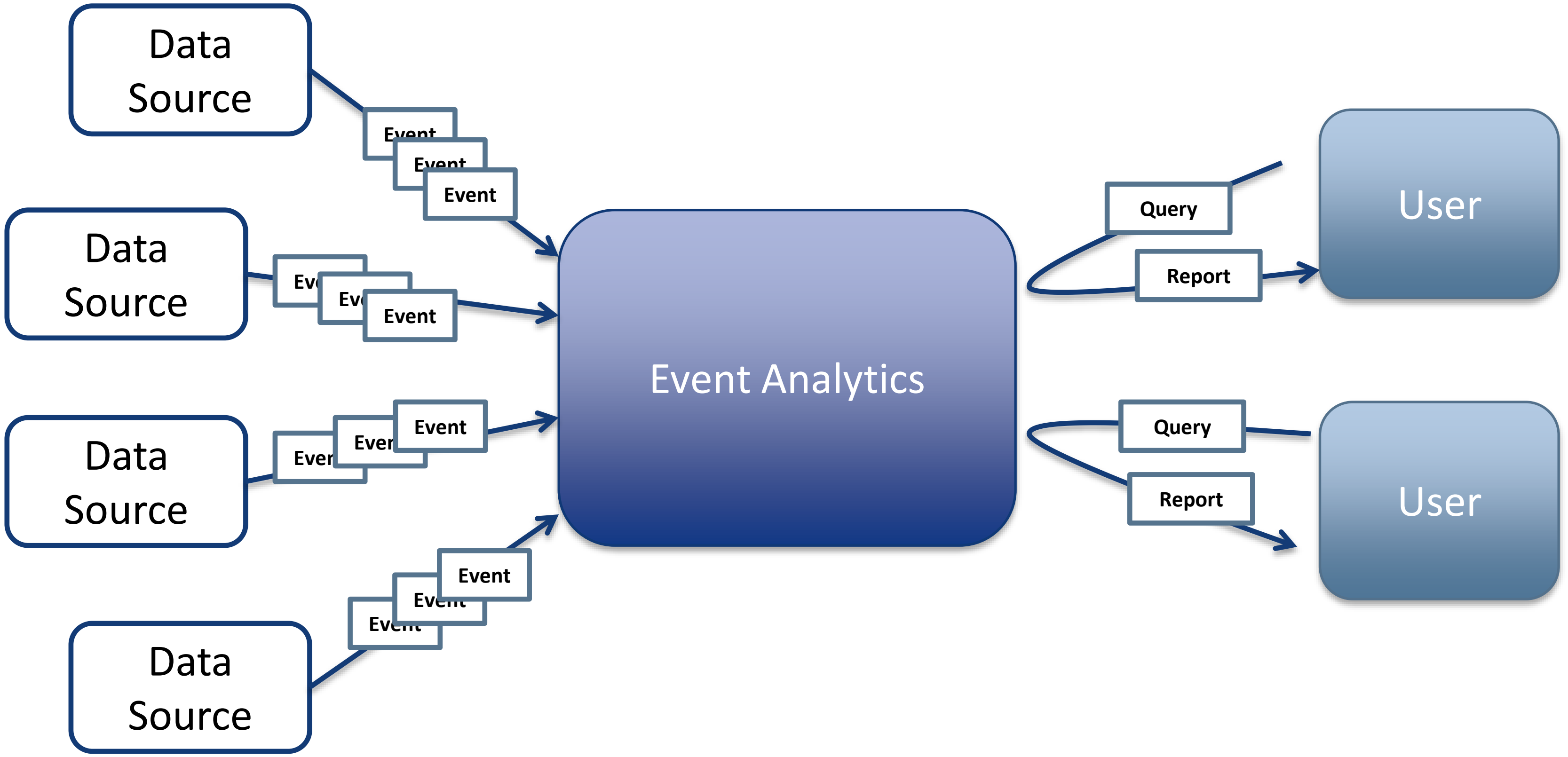
- **Ruby at Scale**
- **What is Event Analytics?**
- **What are the different ways you could do it?**
- **How we did it**

yottaa

Ruby At Scale?

http://www.flickr.com/photos/laughingsquid

Event Analytics



yottaa

High Write Volume

- Each new data source adds X requests per second
- Data never stops arriving

Continuous Data Growth

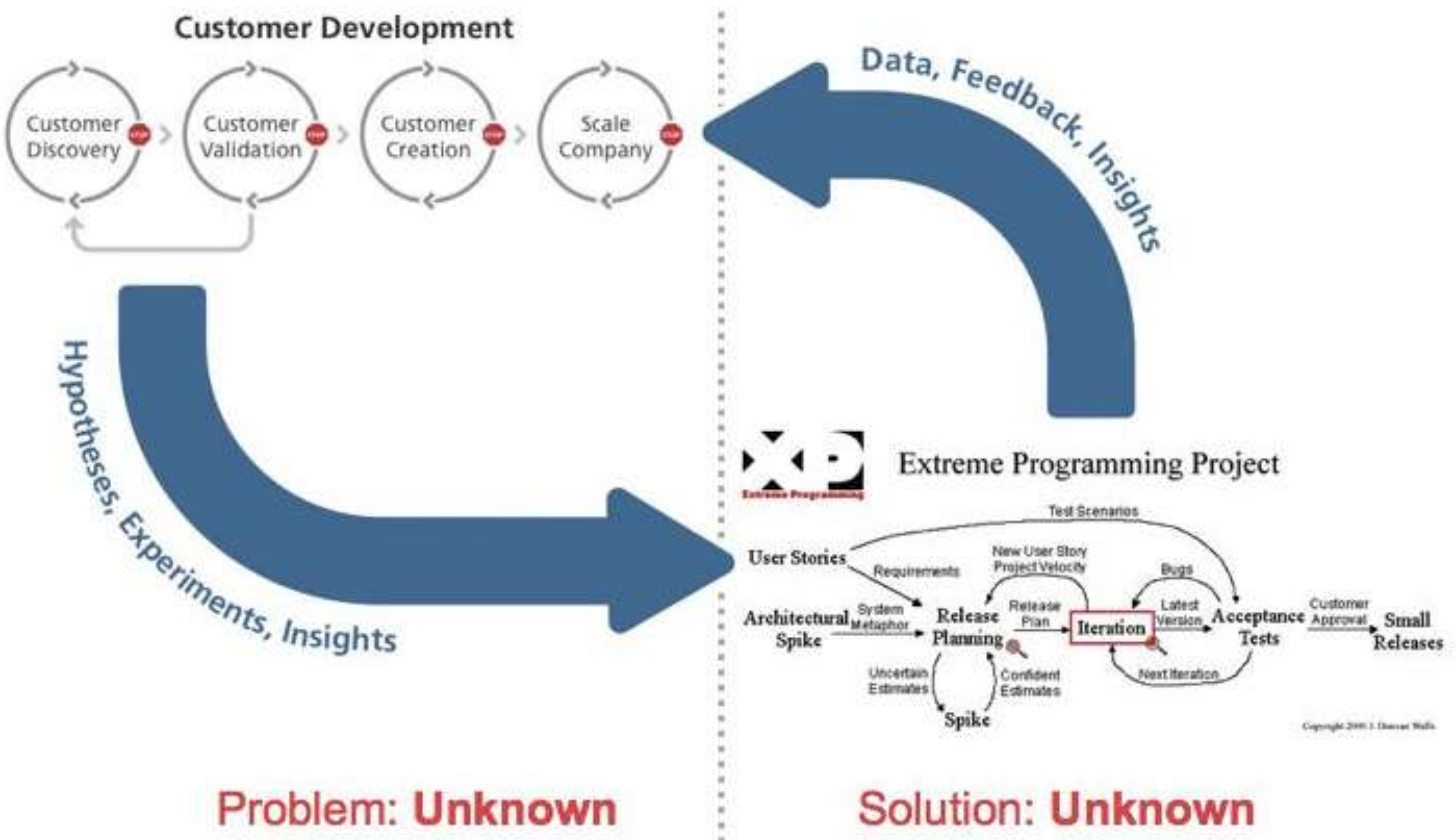
- We only add more data
- Historical data is valuable

Flexible Data Exploration

- Ad hoc queries
- Complex aggregations

yottaa

Oh and we are a startup



Source: Eric Ries
http://startuplessonslearned.blogspot.com

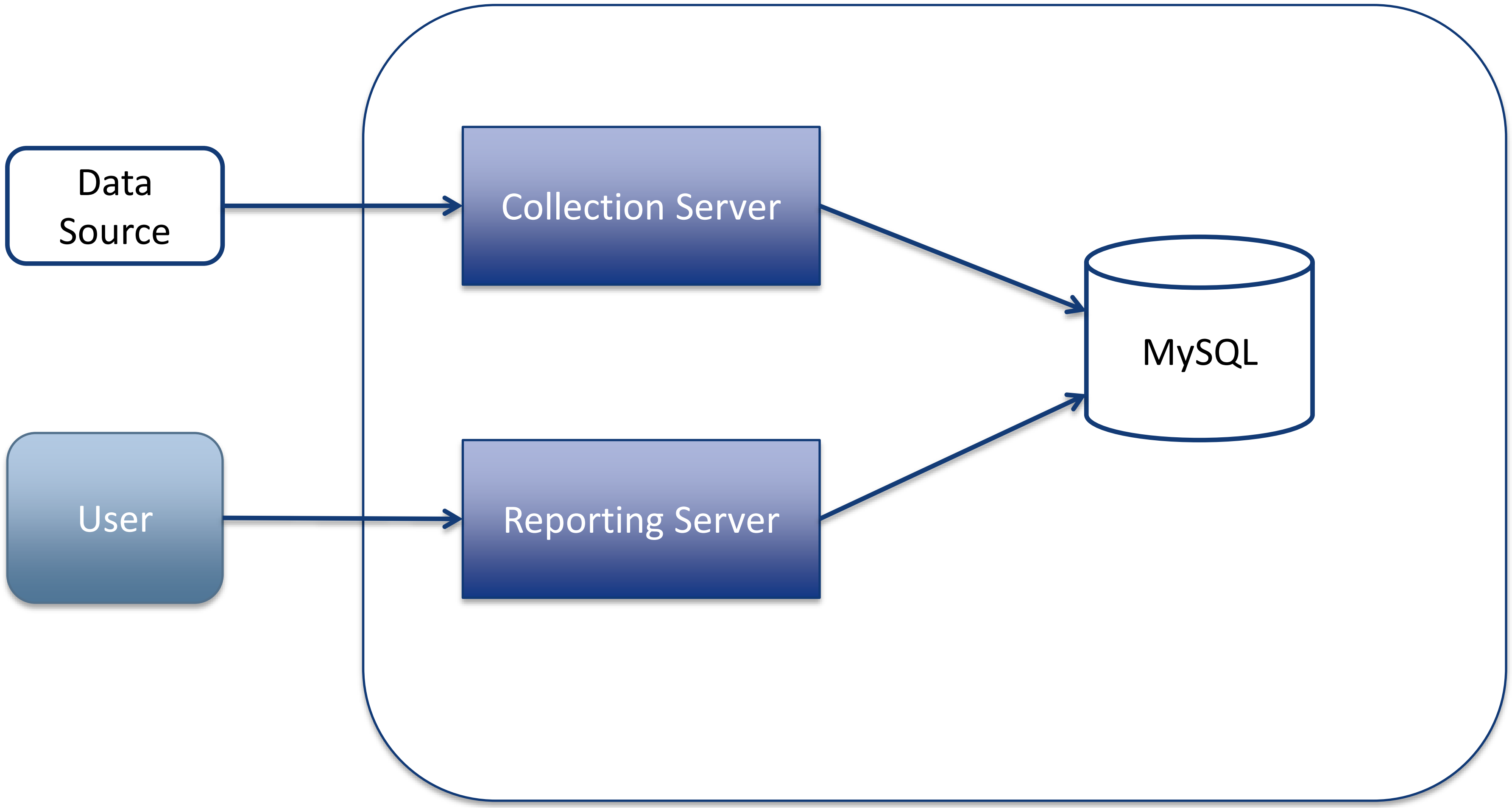
Our requirements:

On Launch Day	
# of data sources	15
# of events per minute	80
# GBs data stored	20

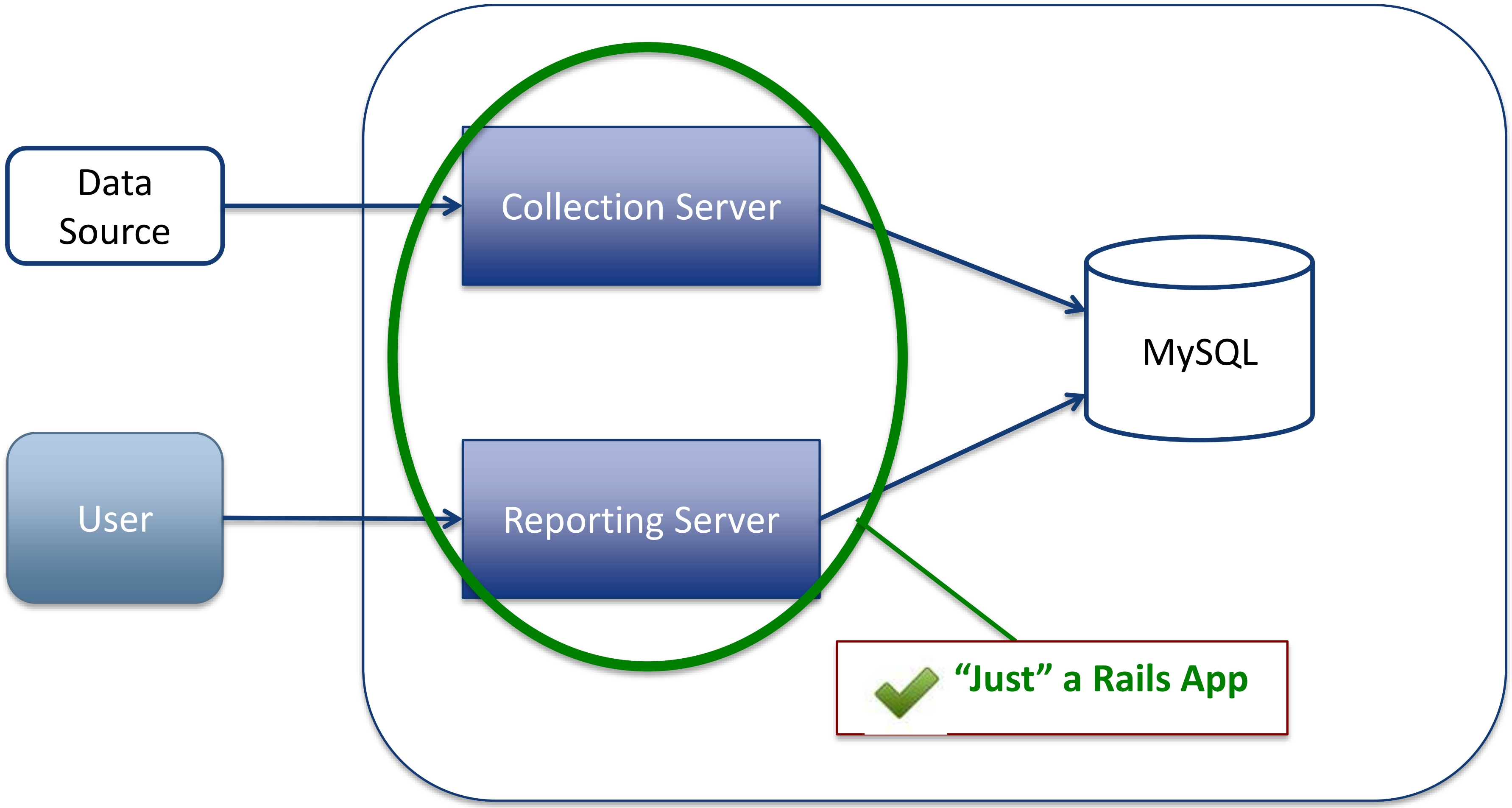
3 months later (projected)	
# of data sources	45
# of events per minute	5600
# GBs data stored	100

yottaa

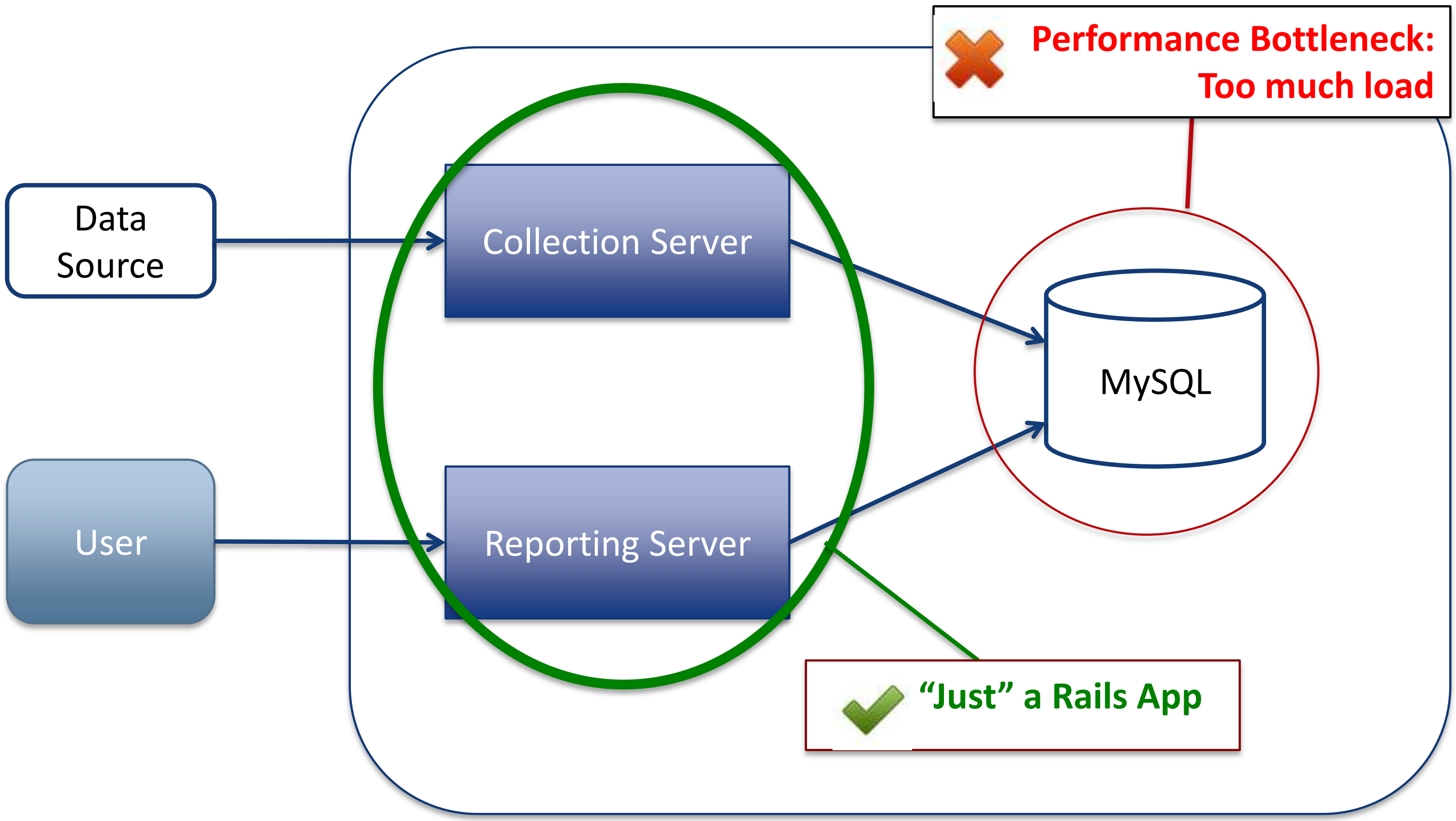
Rails default architecture



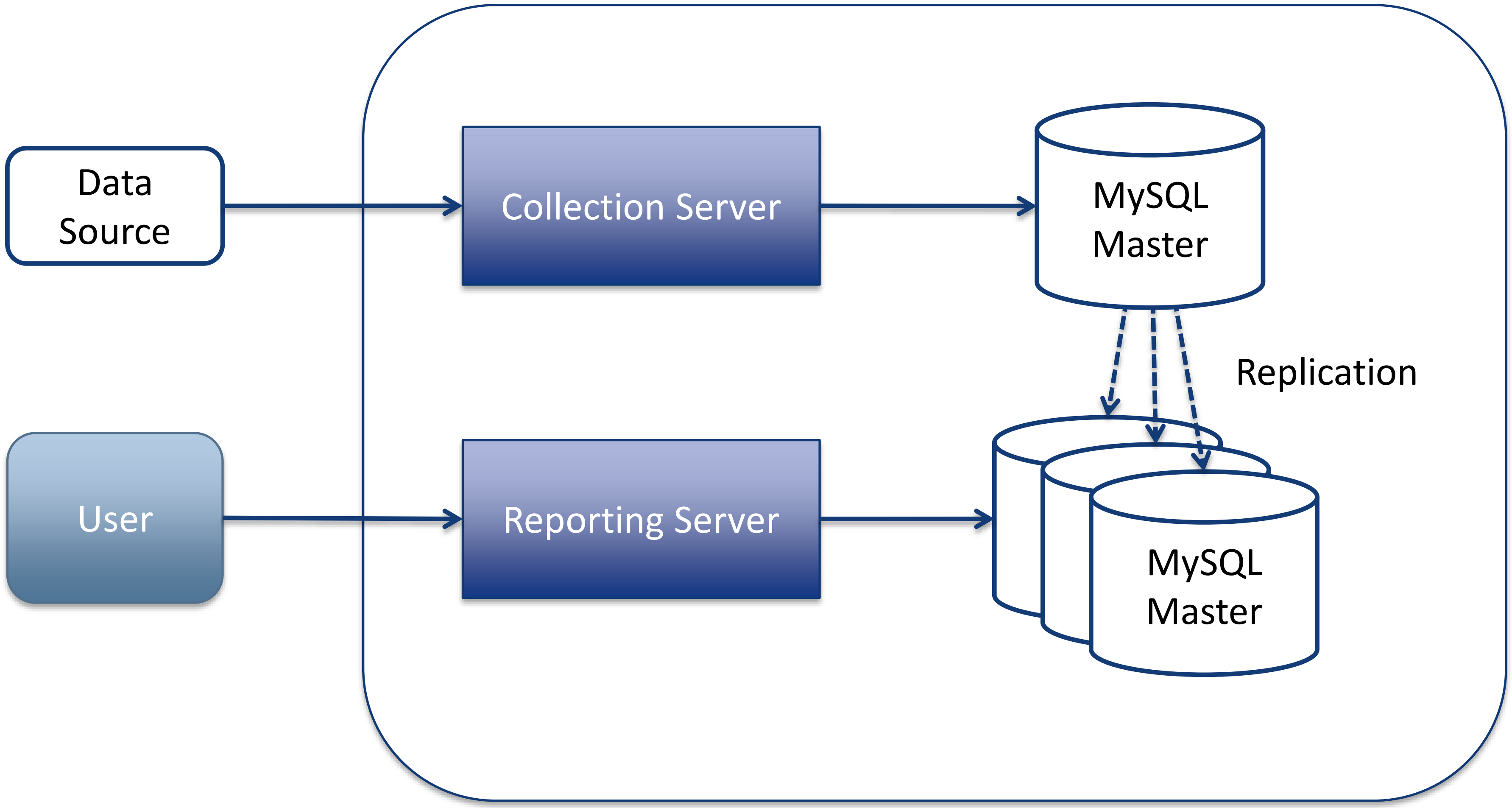
Rails default architecture



Rails default architecture

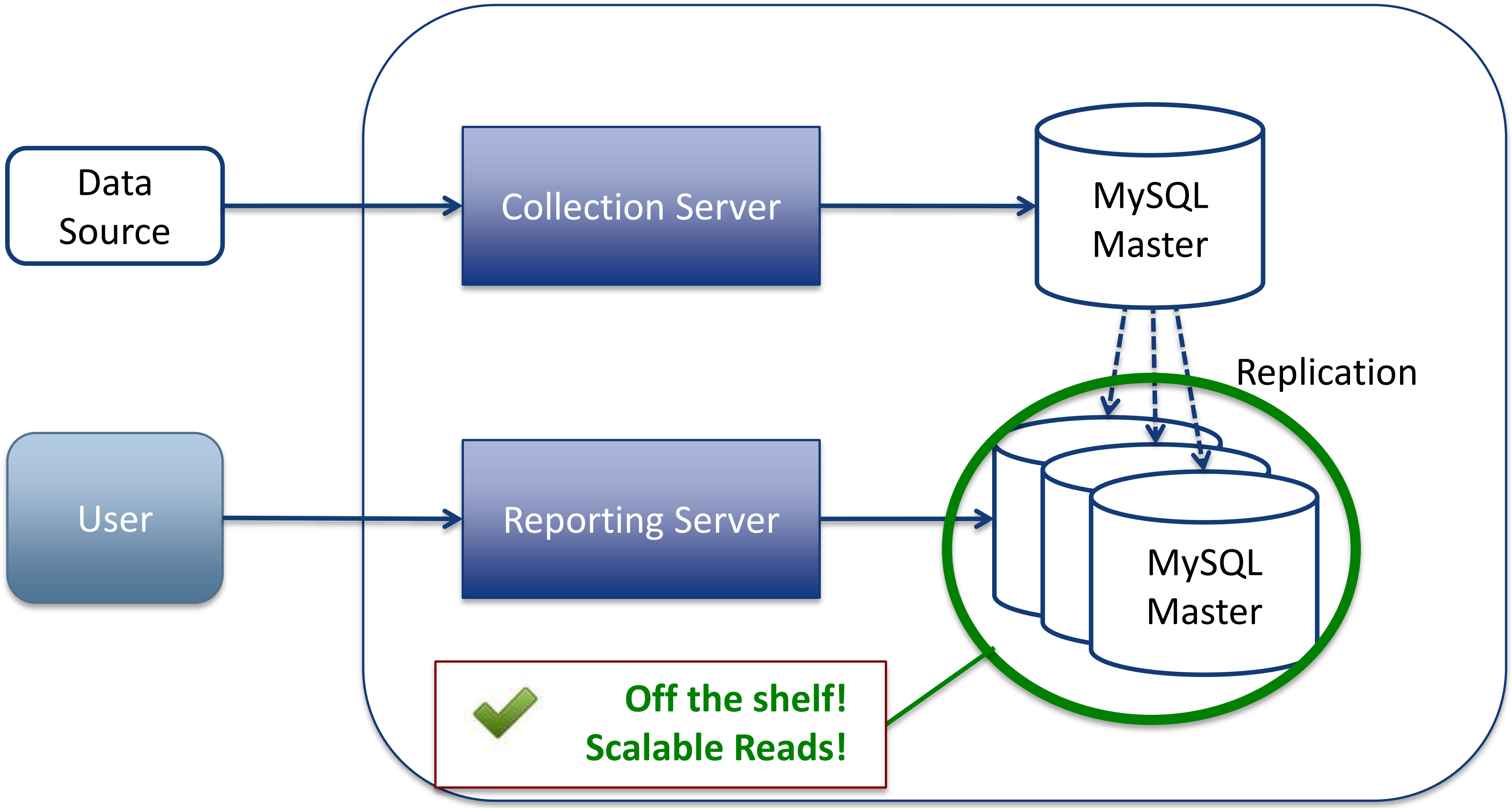


Let's add replication!

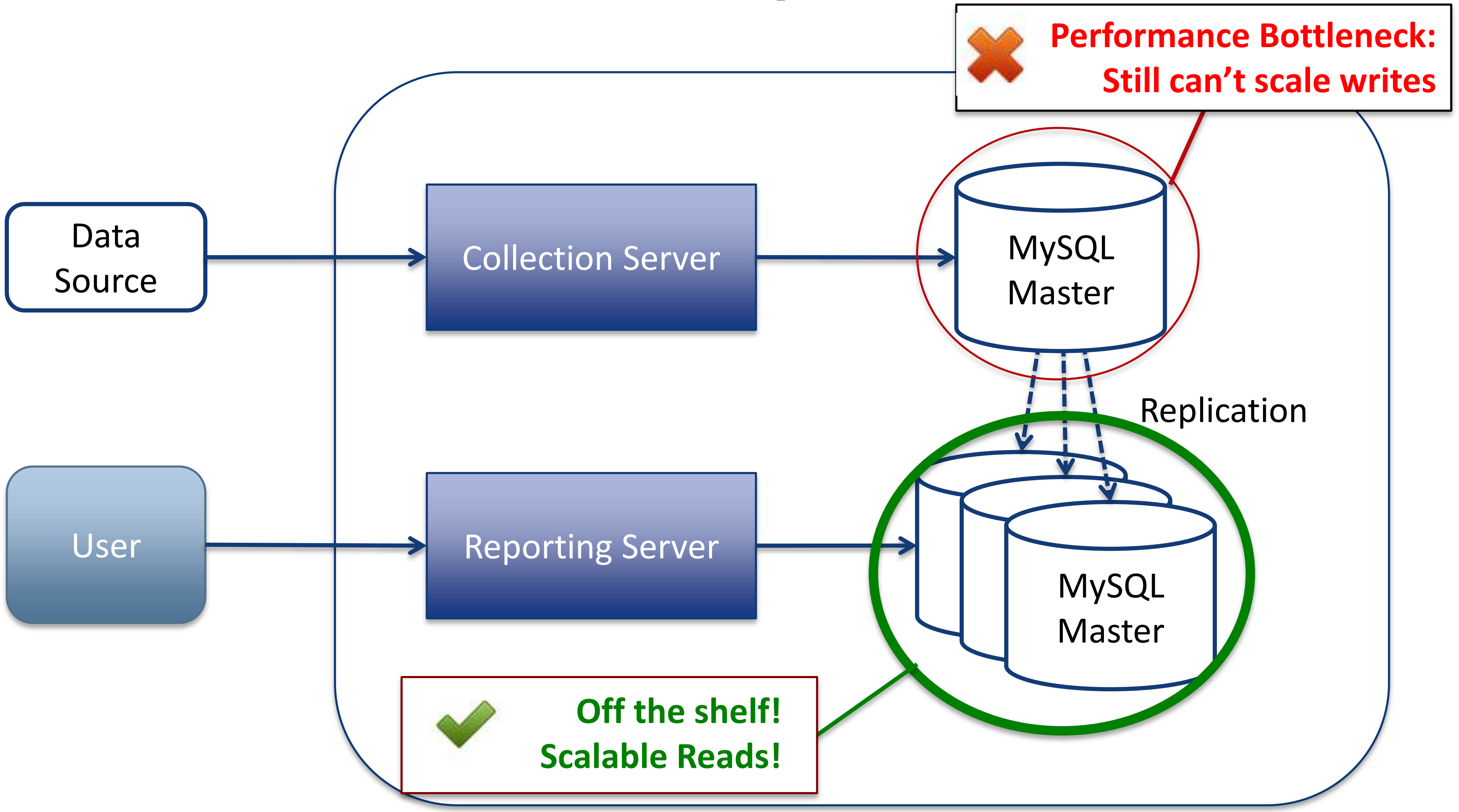


yottaa

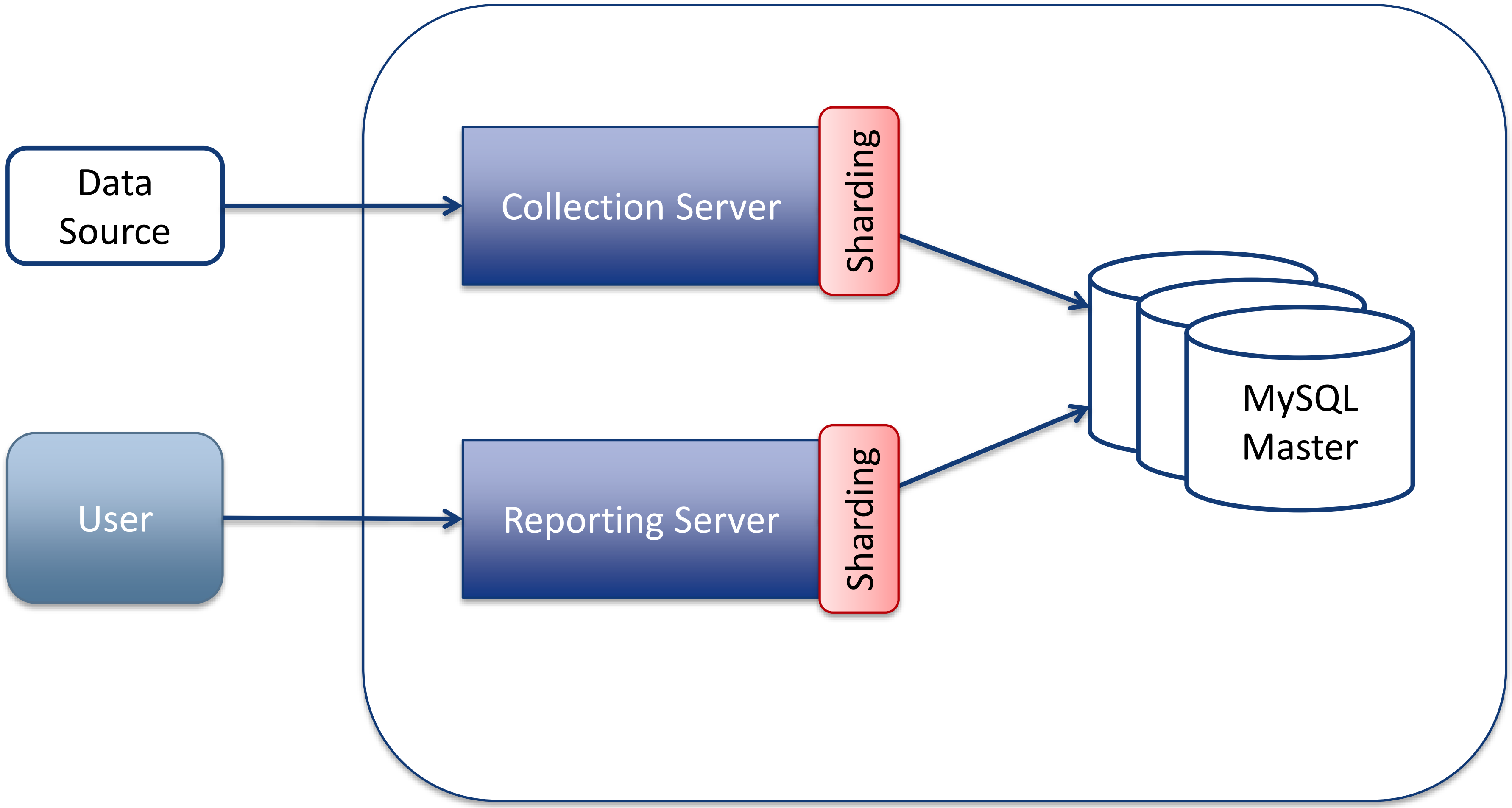
Let's add replication!



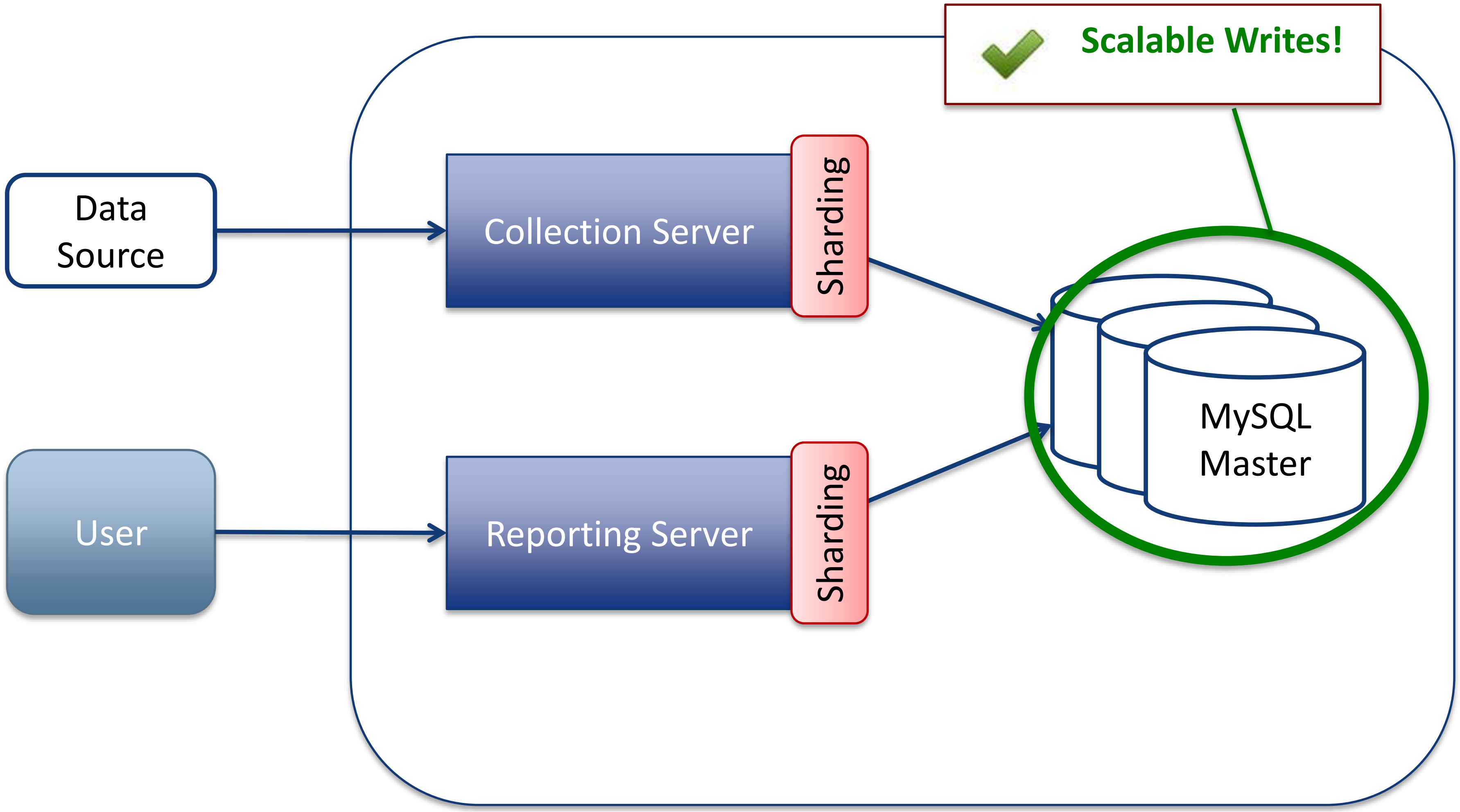
Let's add replication!



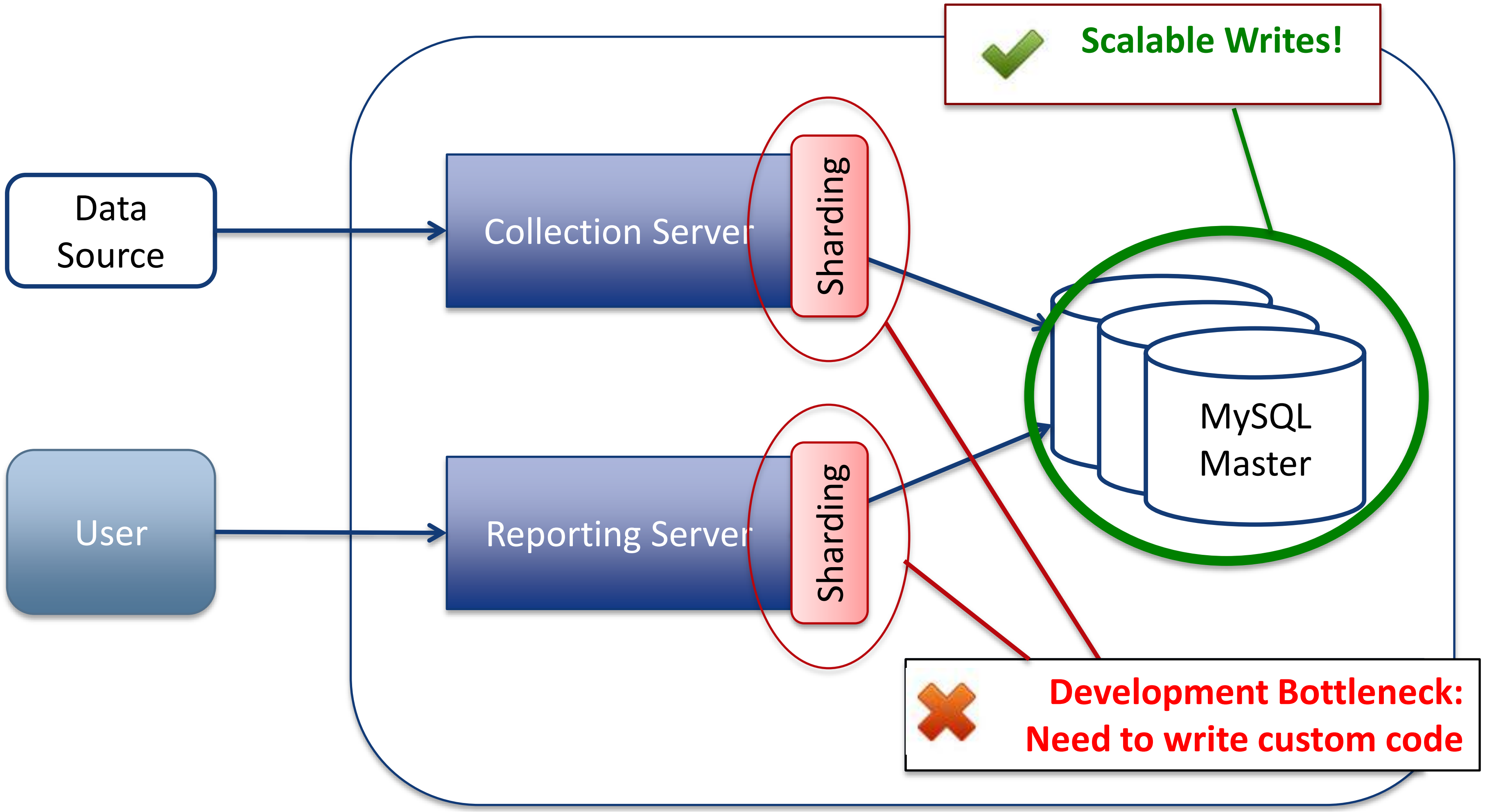
What about sharding?



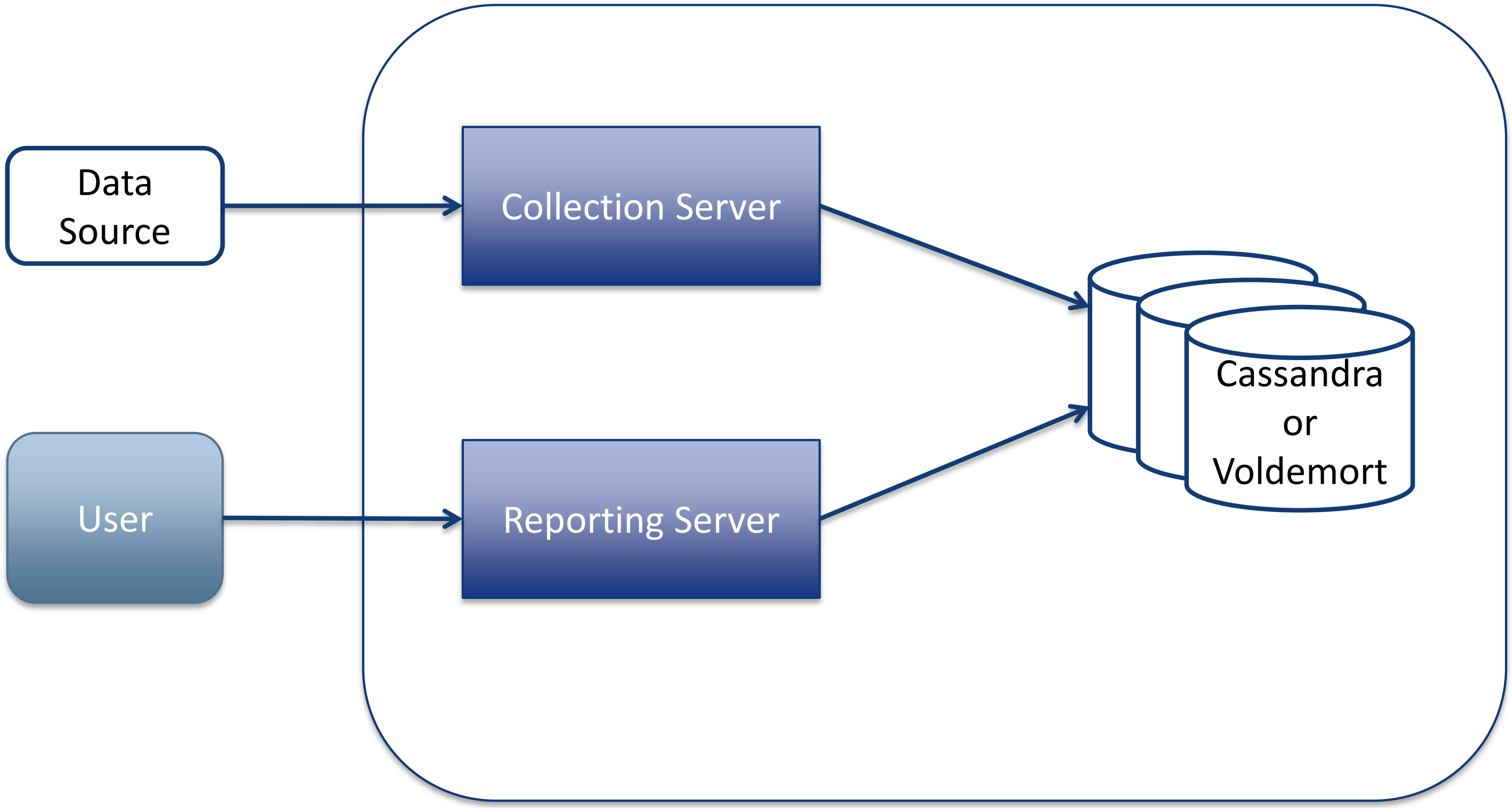
What about sharding?



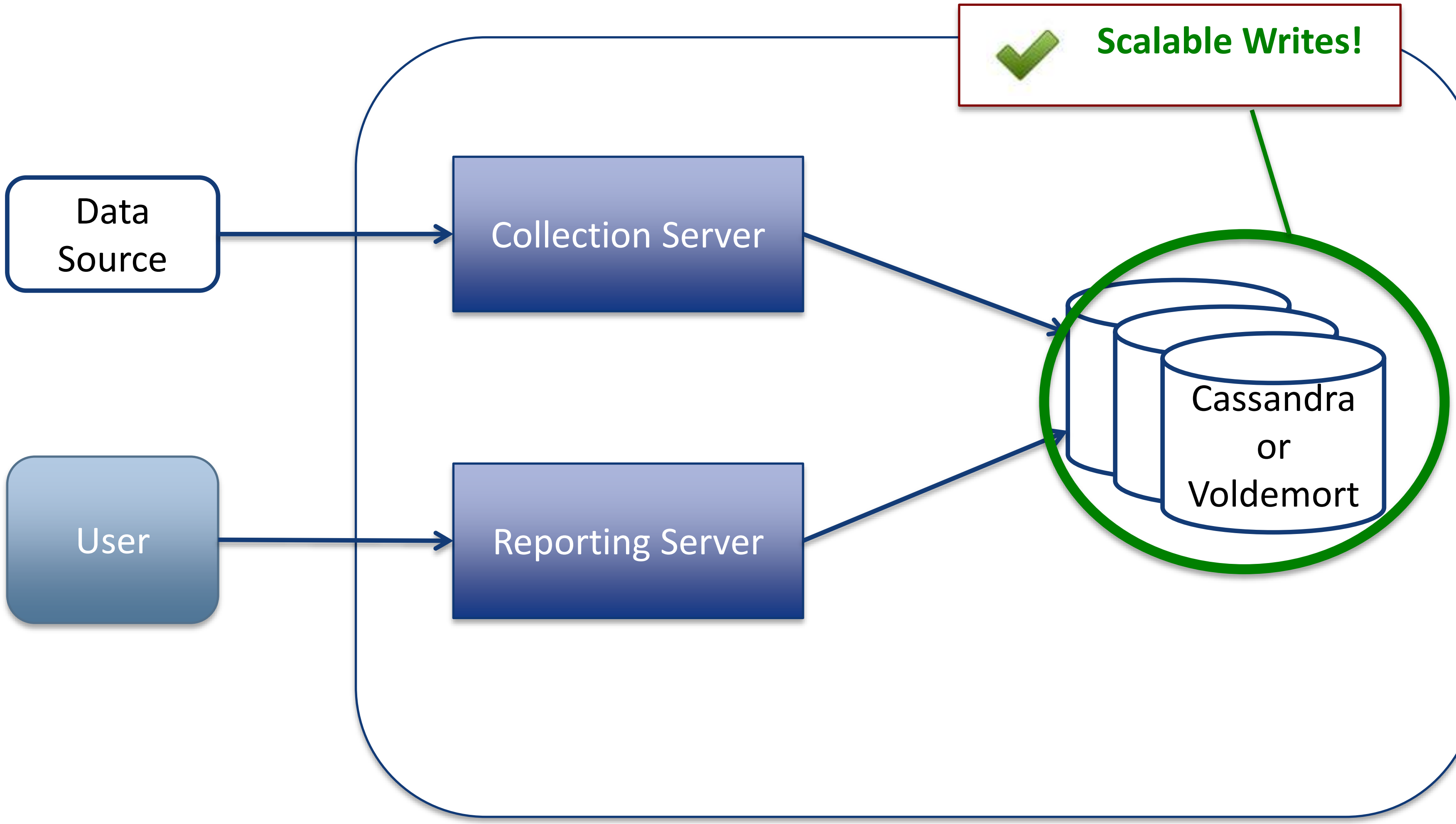
What about sharding?



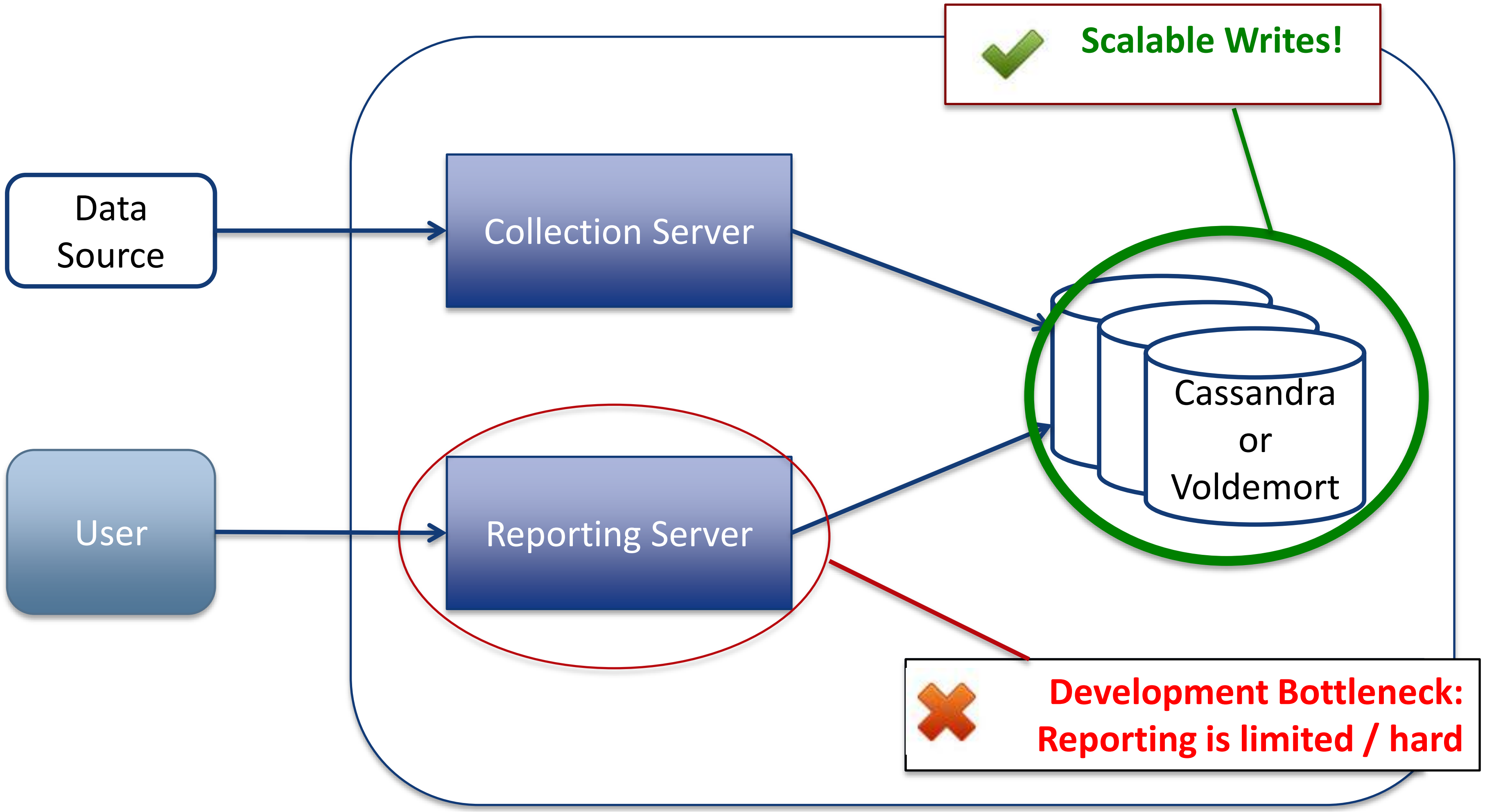
Key Value stores to the rescue?



Key Value stores to the rescue?

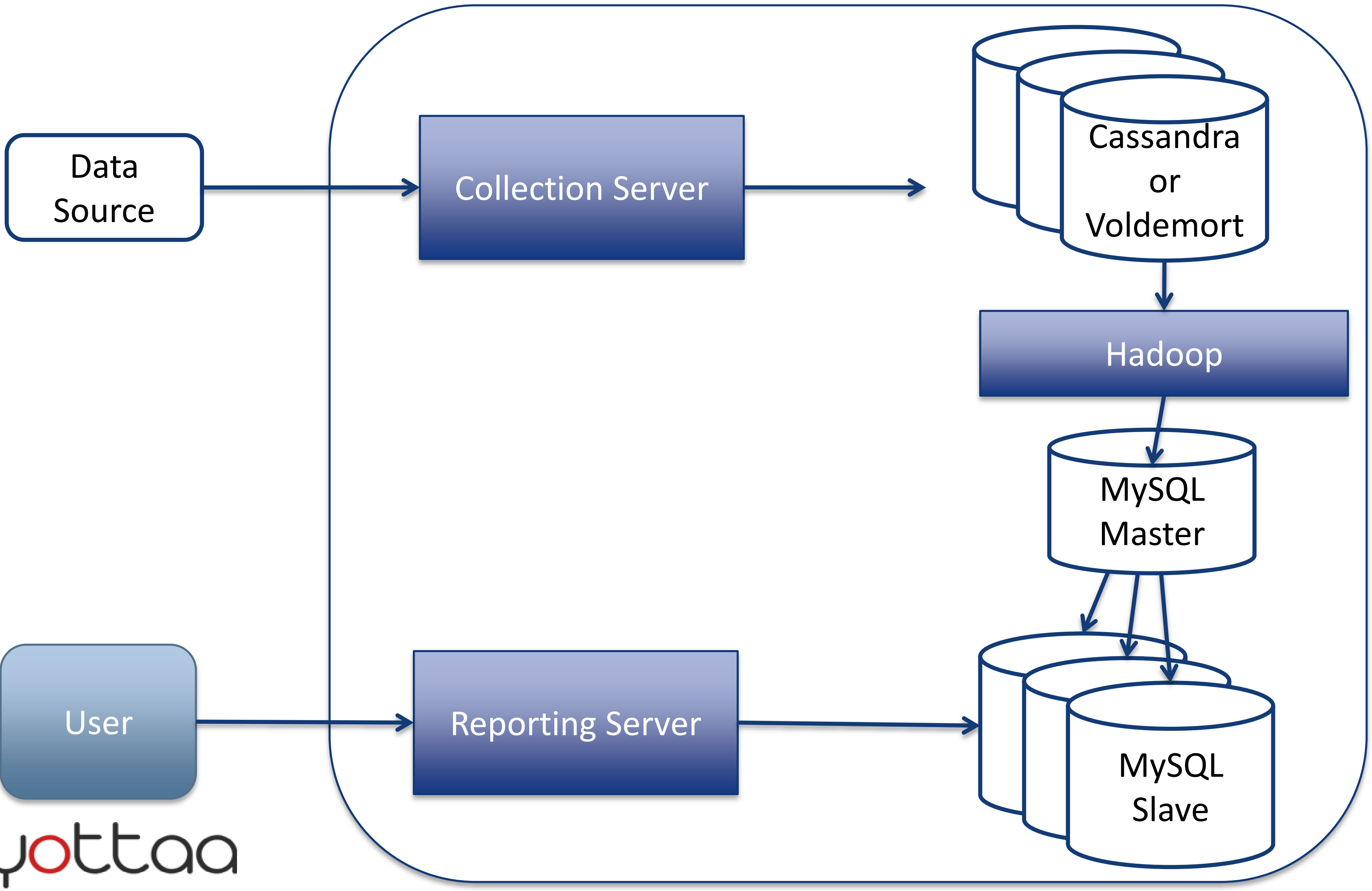


Key Value stores to the rescue?



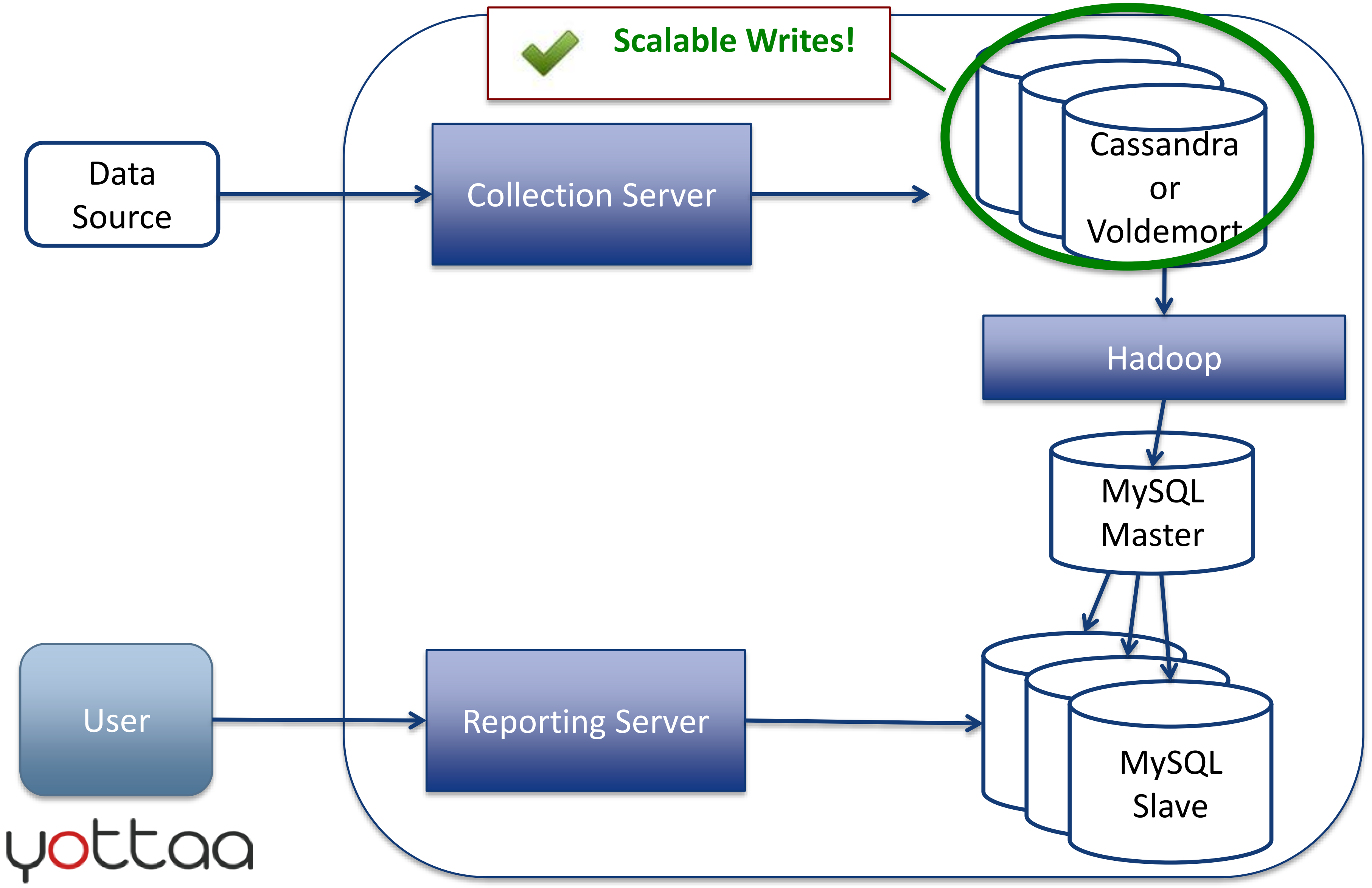
yottaa

Can I Hadoop my way out of this?

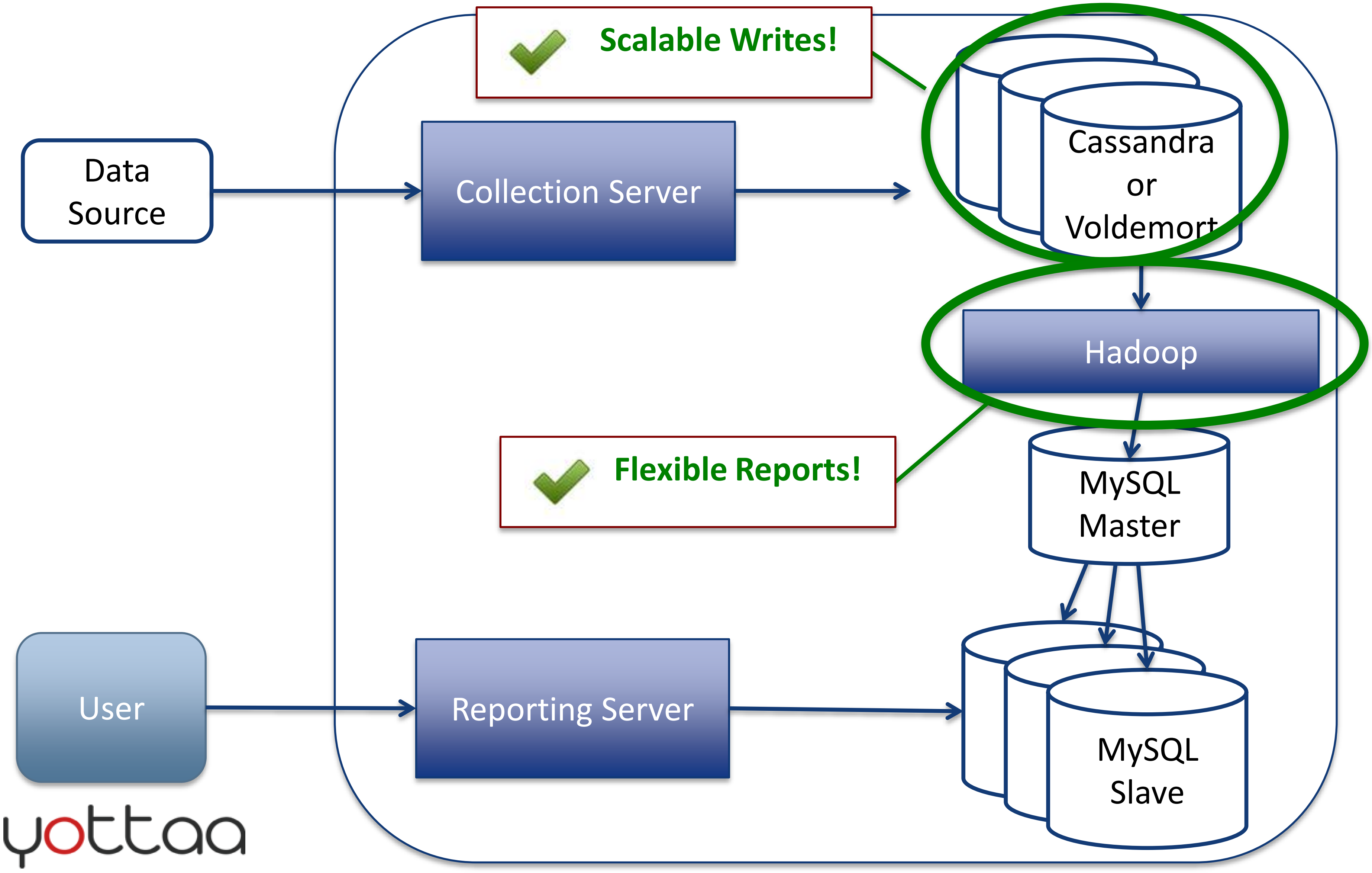


yottaa

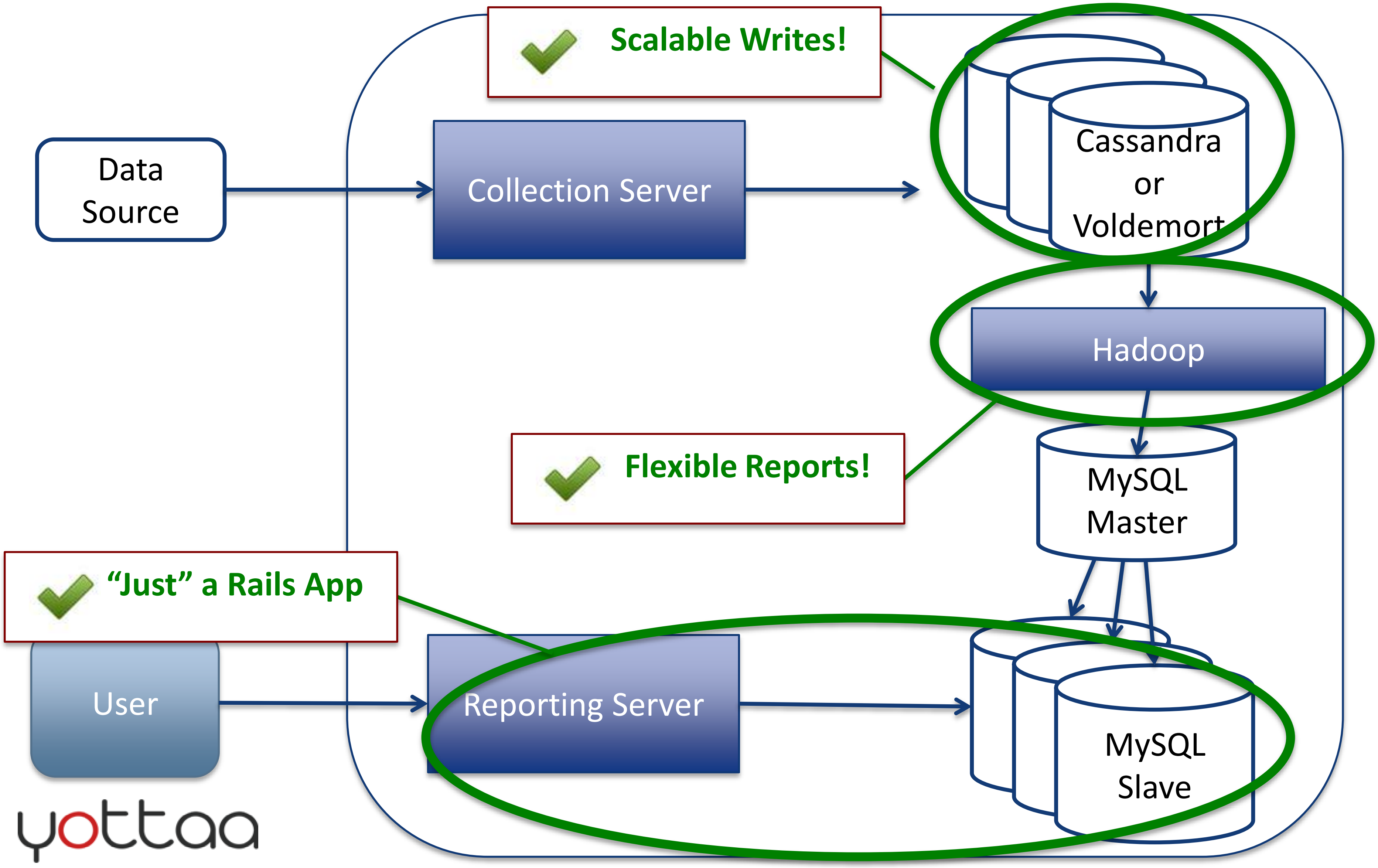
Can I Hadoop my way out of this?



Can I Hadoop my way out of this?

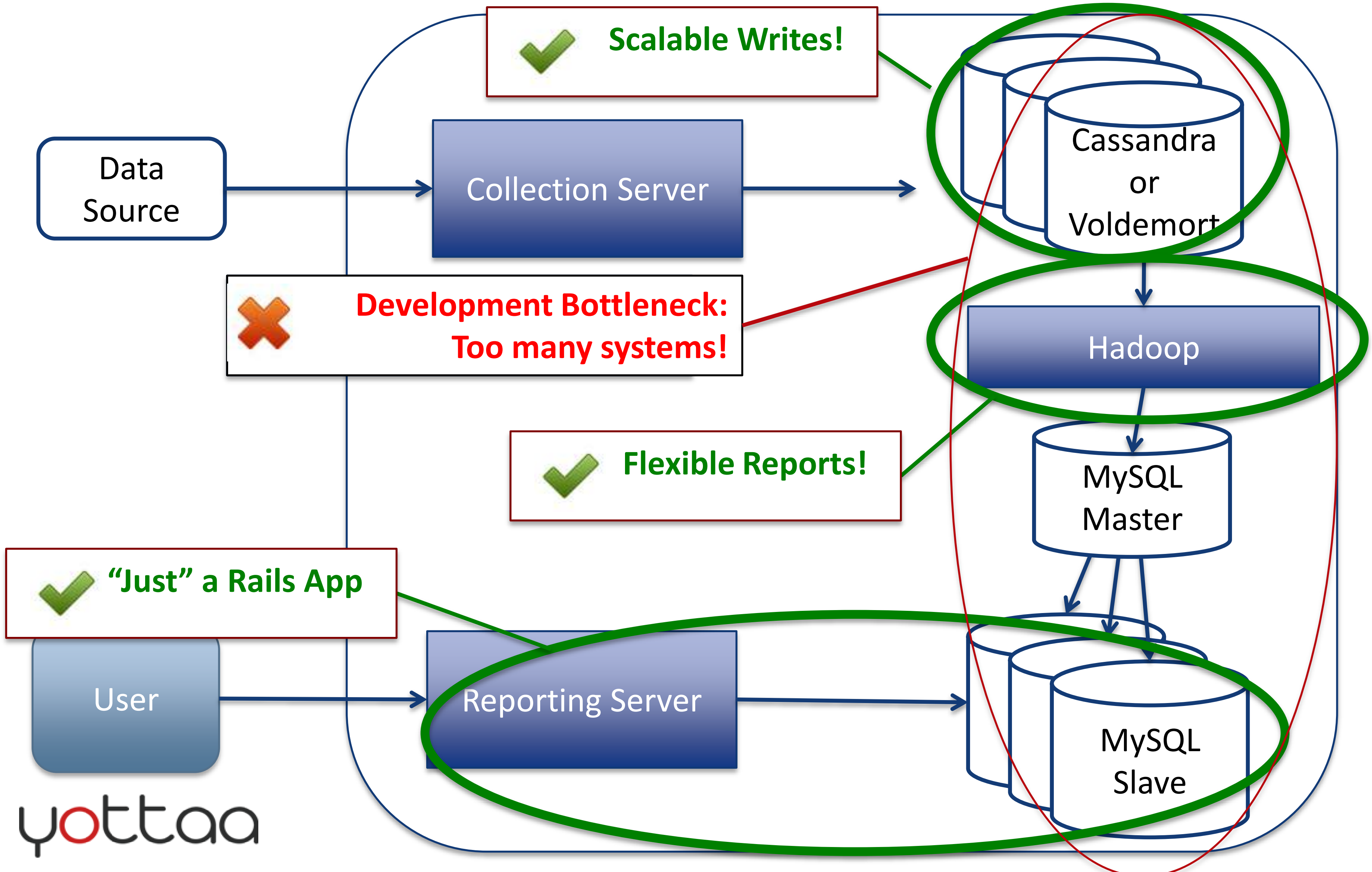


Can I Hadoop my way out of this?



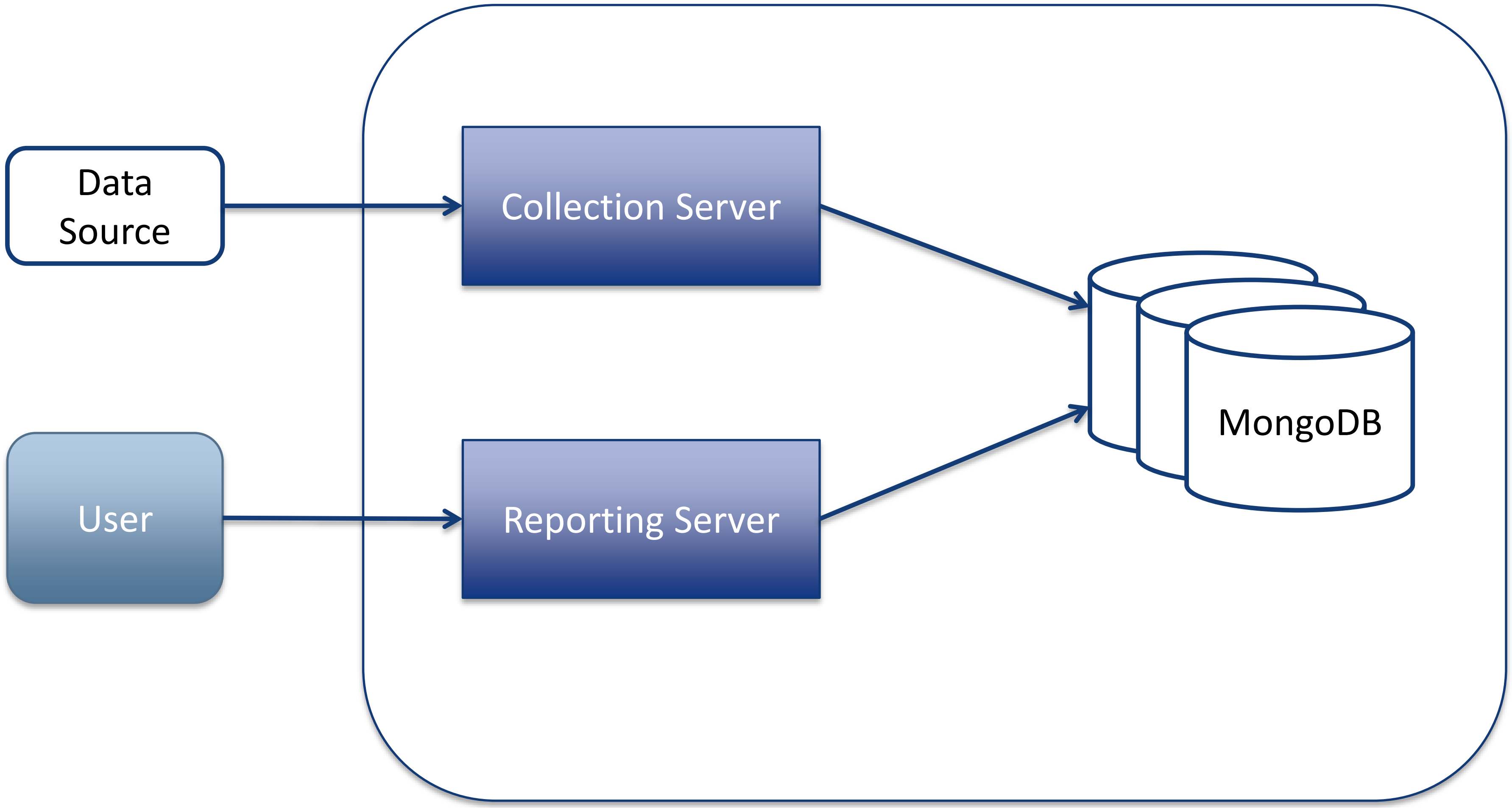
yottaa

Can I Hadoop my way out of this?

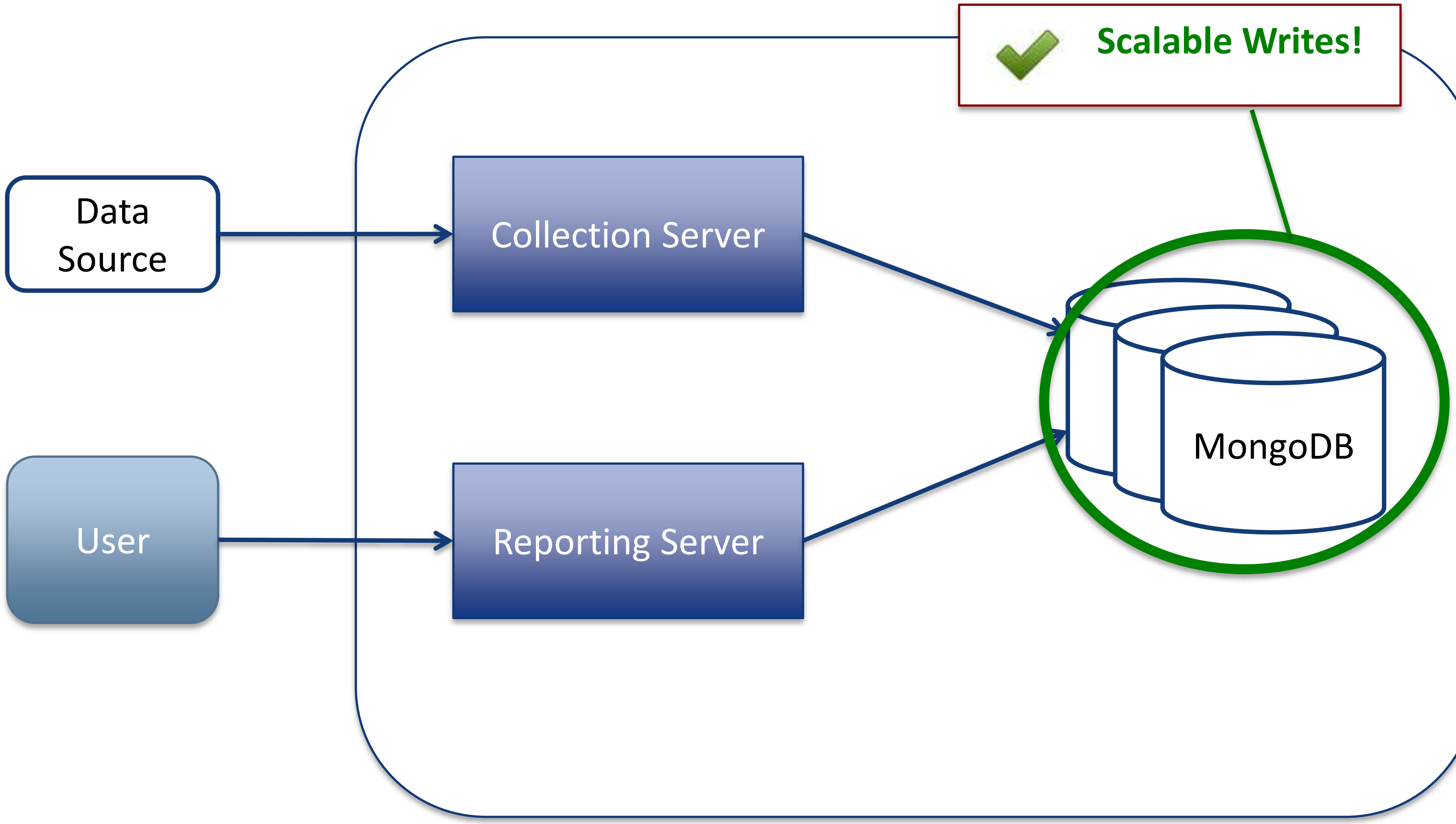


yottaa

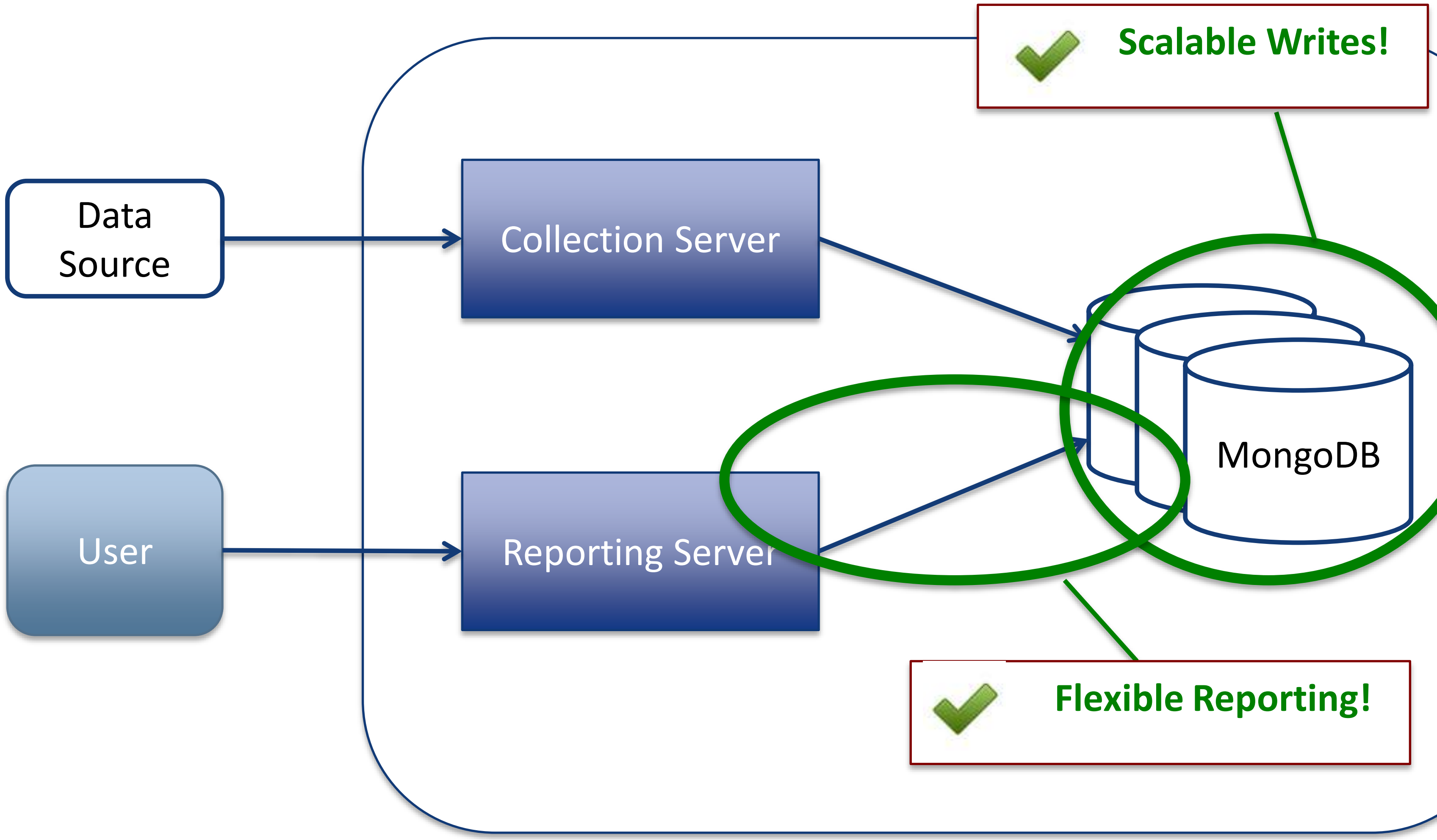
MongoDB!


Data Source
Collection Server
User
Reporting Server
MongoDB

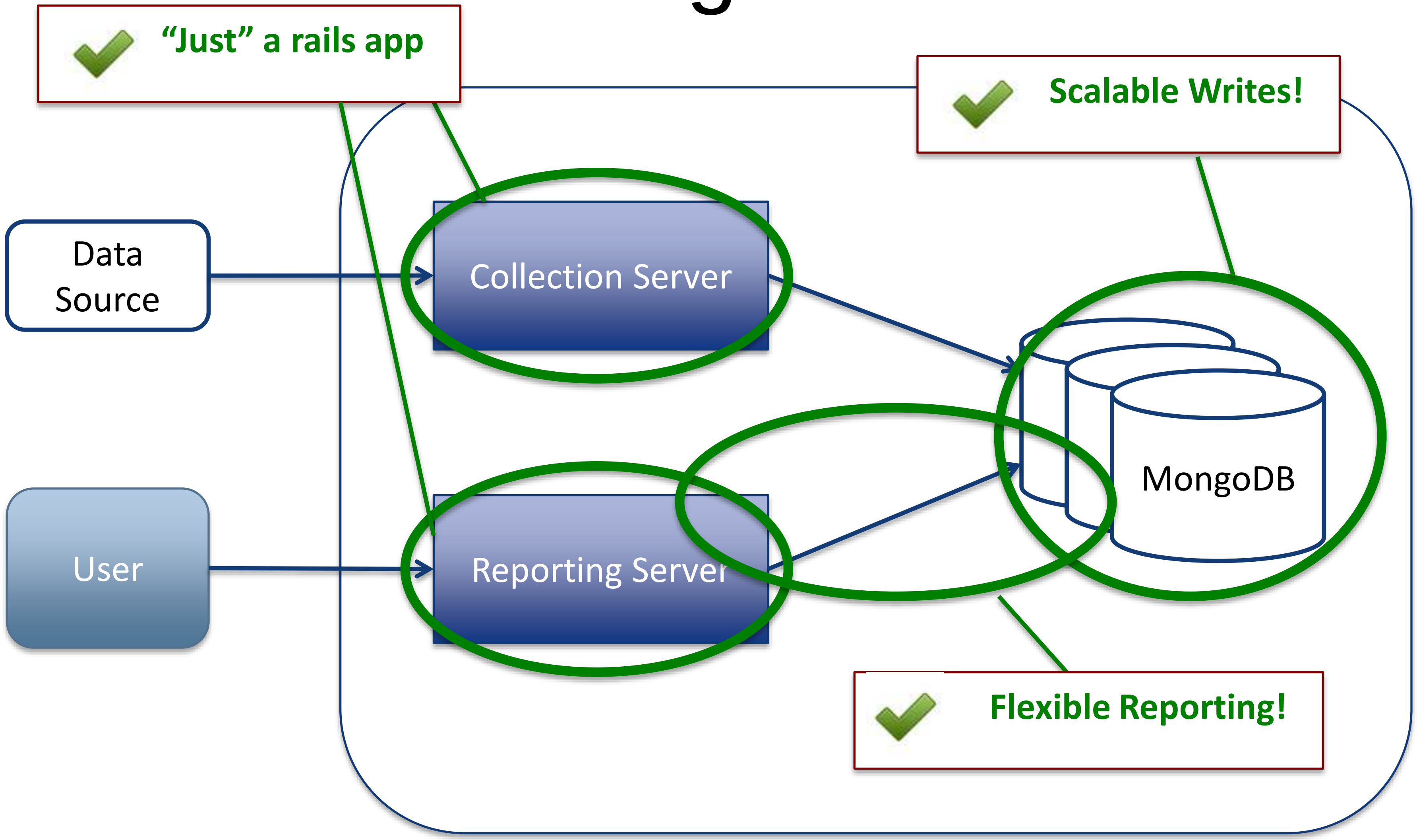
MongoDB!

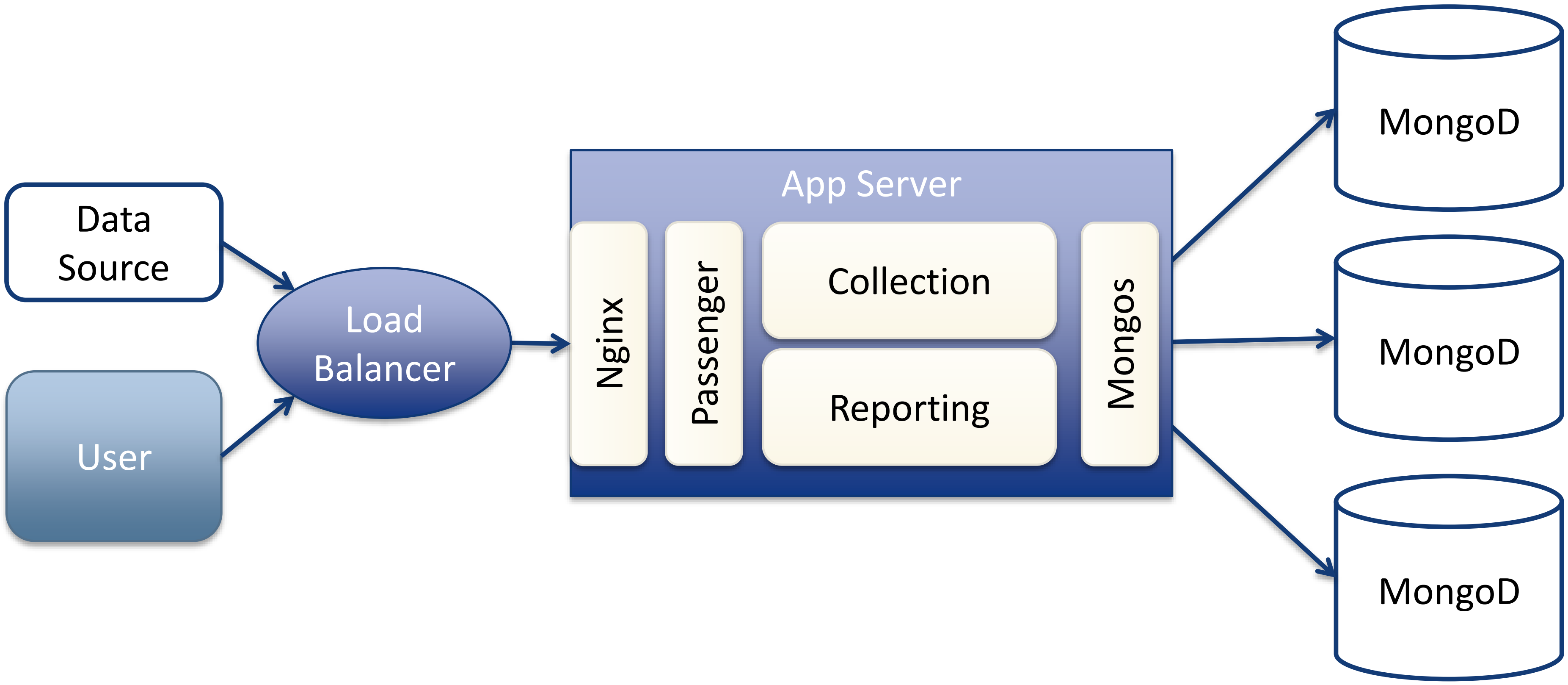

Scalable Writes!
Data Source
Collection Server
User
Reporting Server
MongoDB

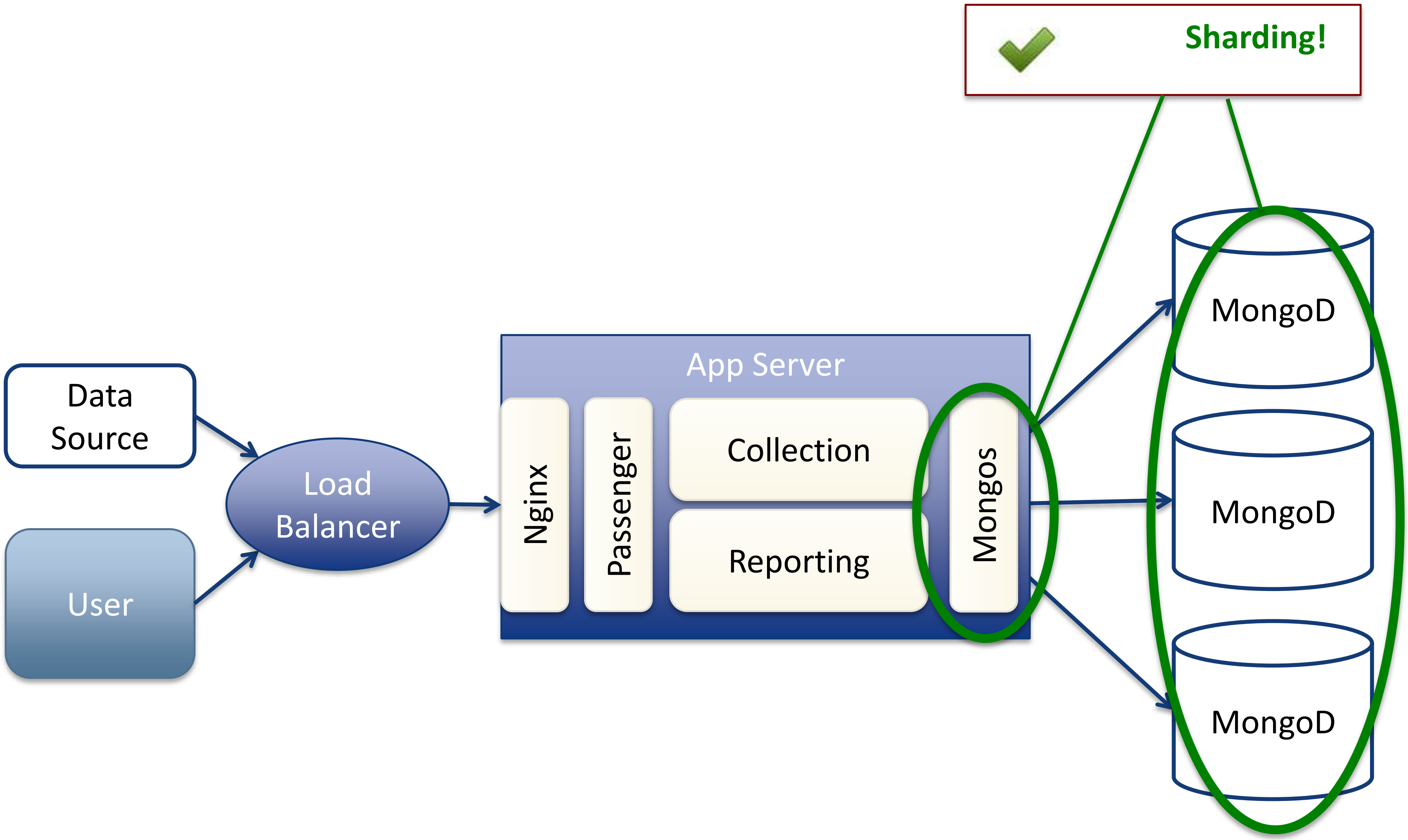
MongoDB!

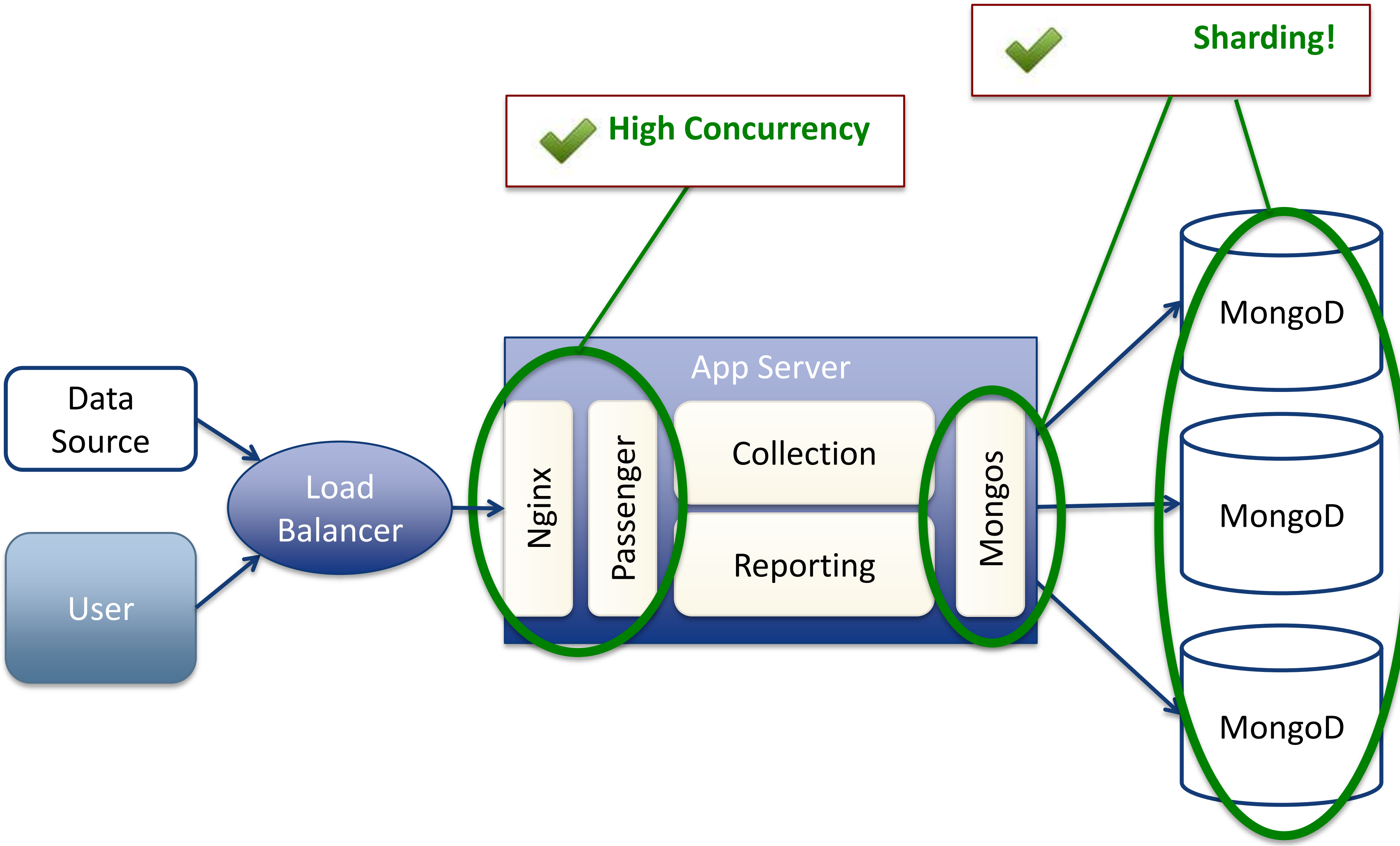

Scalable Writes!
Data Source
Collection Server
MongoDB
User
Reporting Server
Flexible Reporting!

MongoDB!

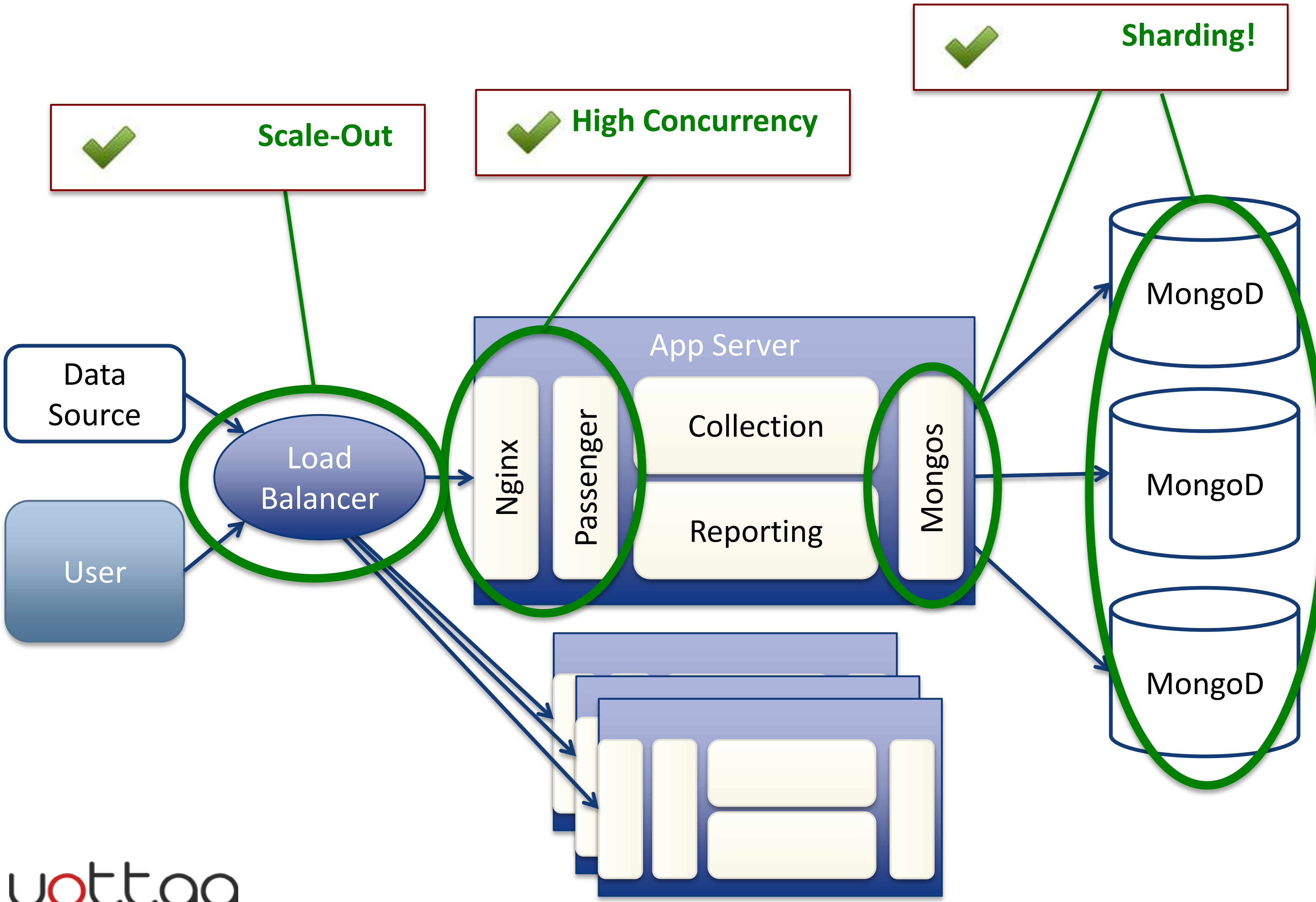

✓ "Just" a rails app
✓ Scalable Writes!
Data Source
Collection Server
User
Reporting Server
MongoDB
✓ Flexible Reporting!


Data Source
User
Load Balancer
App Server
Nginx
Passenger
Collection
Reporting
Mongos
MongoD
MongoD
MongoD


✓ Sharding!
Data Source
User
Load Balancer
App Server
Nginx
Passenger
Collection
Reporting
Mongos
MongoD
MongoD
MongoD

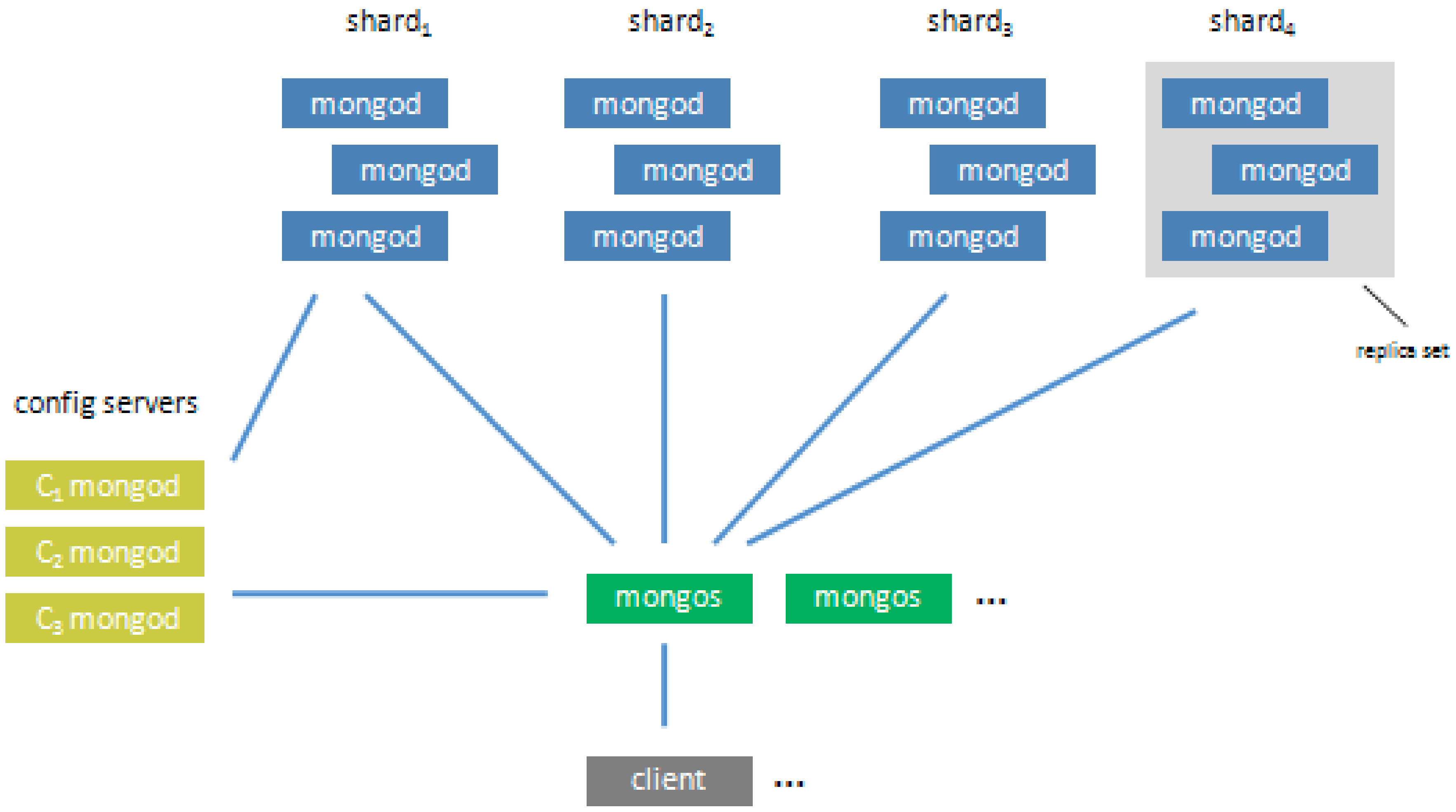


Sharding!
High Concurrency
App Server
Data Source
Load Balancer
User
Nginx
Passenger
Collection
Reporting
Mongos
MongoD
MongoD
MongoD


Scale-Out
High Concurrency
Sharding!
Data Source
User
Load Balancer
App Server
Nginx
Passenger
Collection
Reporting
Mongos
MongoD
MongoD
MongoD

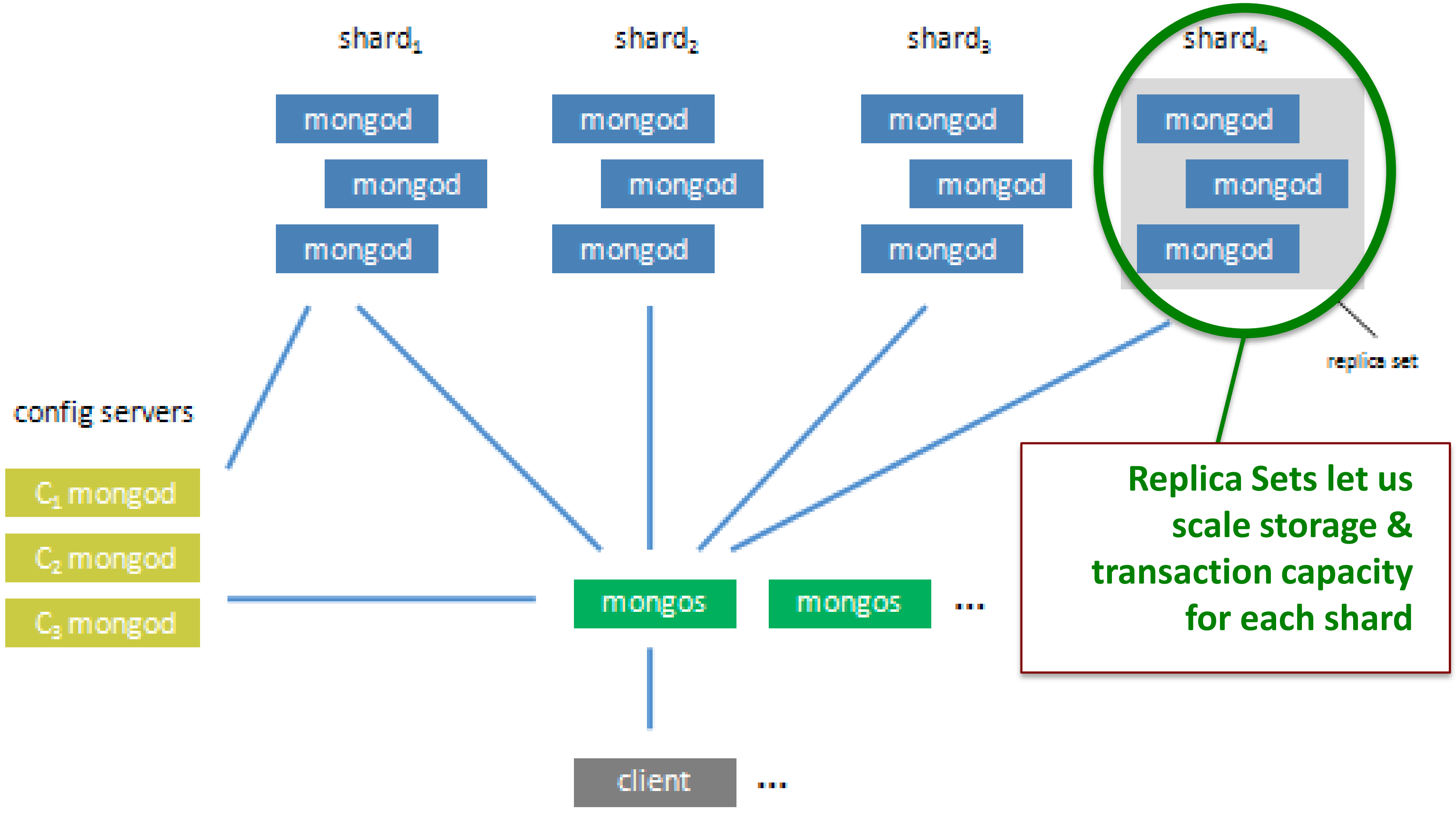
yottaa

MongoDBSharding



yottaa

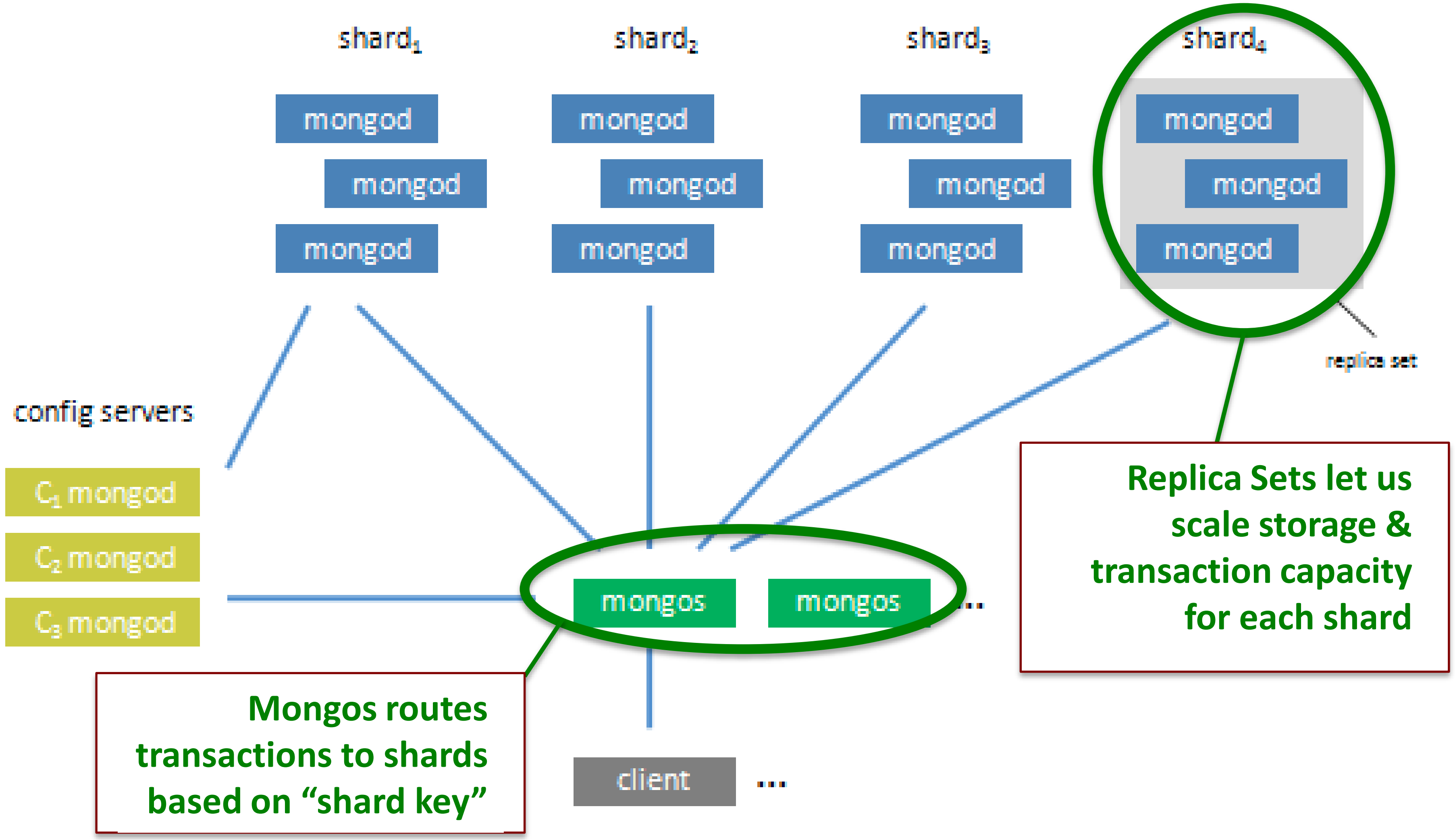
MongoDBSharding



Replica Sets let us scale storage & transaction capacity for each shard

yottaa

MongoDBSharding

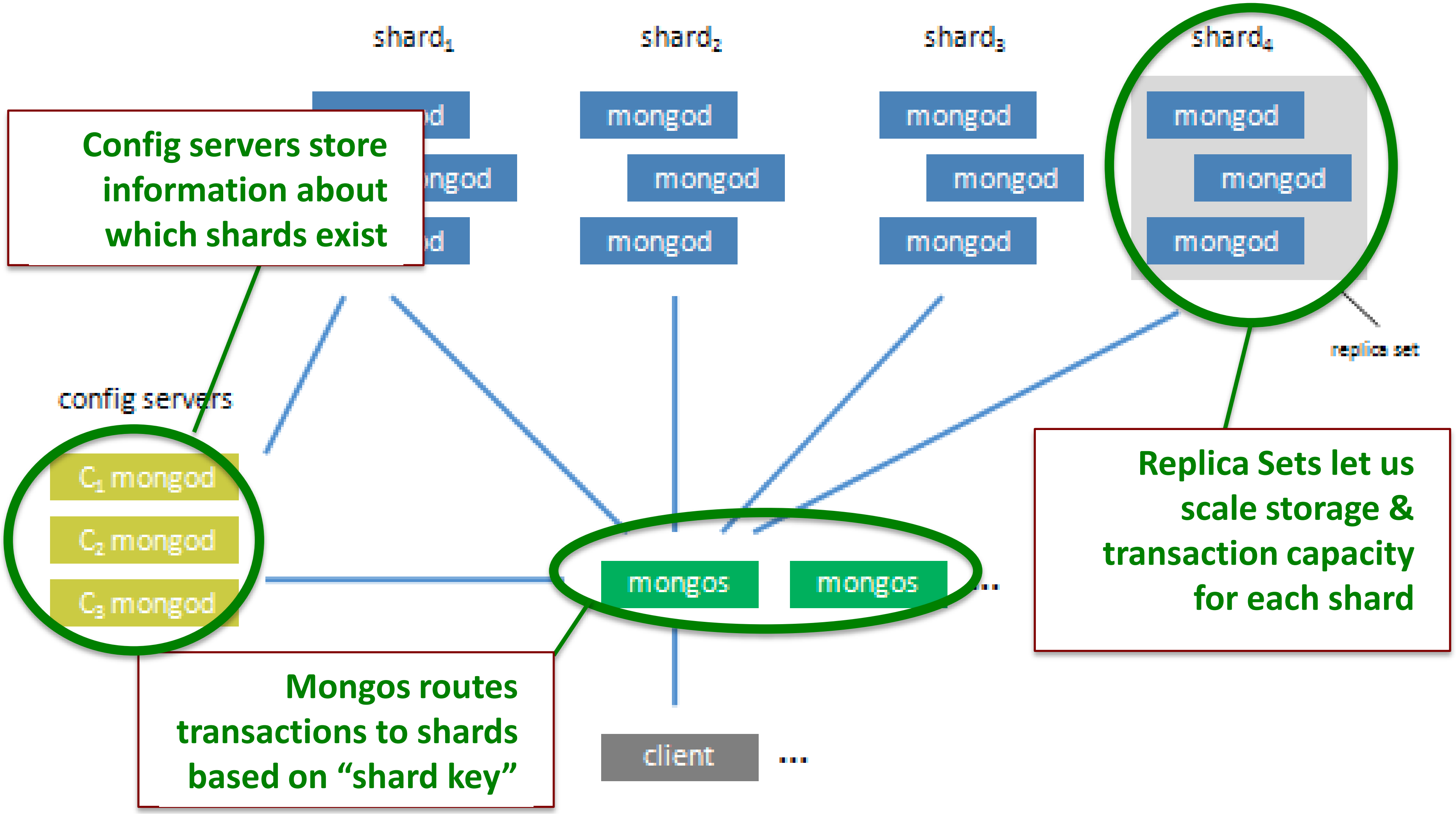


Replica Sets let us scale storage & transaction capacity for each shard

Mongos routes transactions to shards based on "shard key"

yottaa

MongoDBSharding

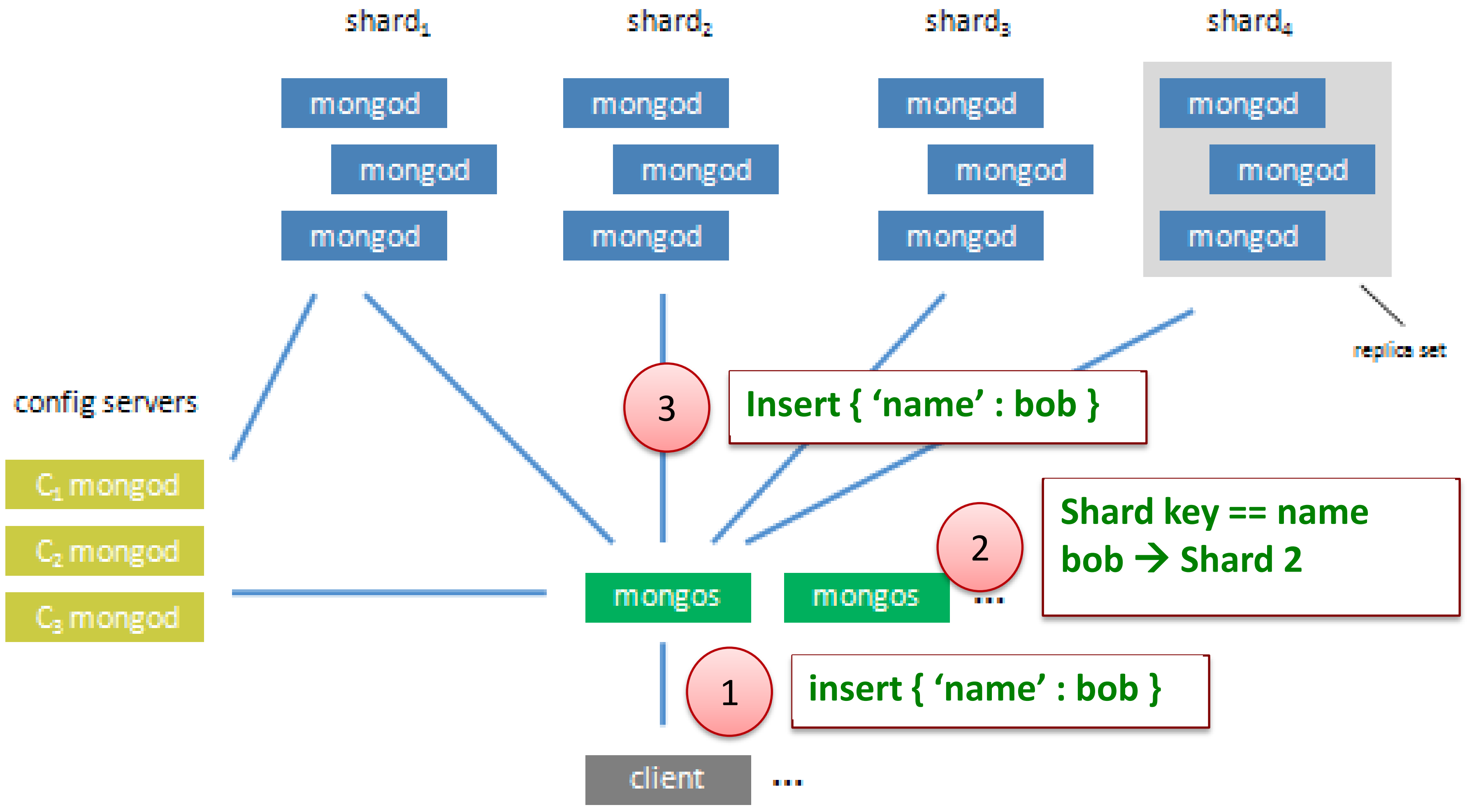


Config servers store information about which shards exist

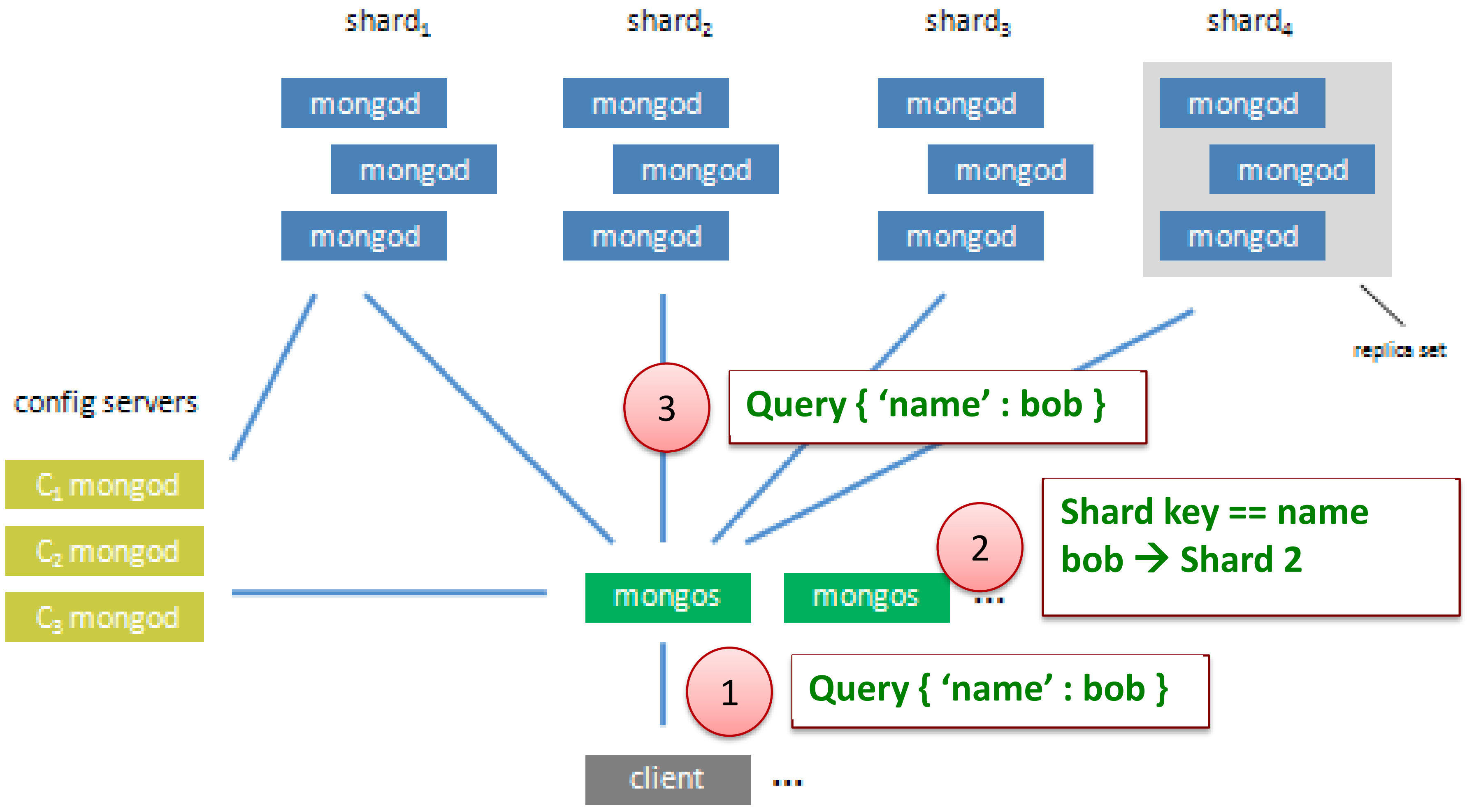
Replica Sets let us scale storage & transaction capacity for each shard

Mongos routes transactions to shards based on “shard key”

Inserting

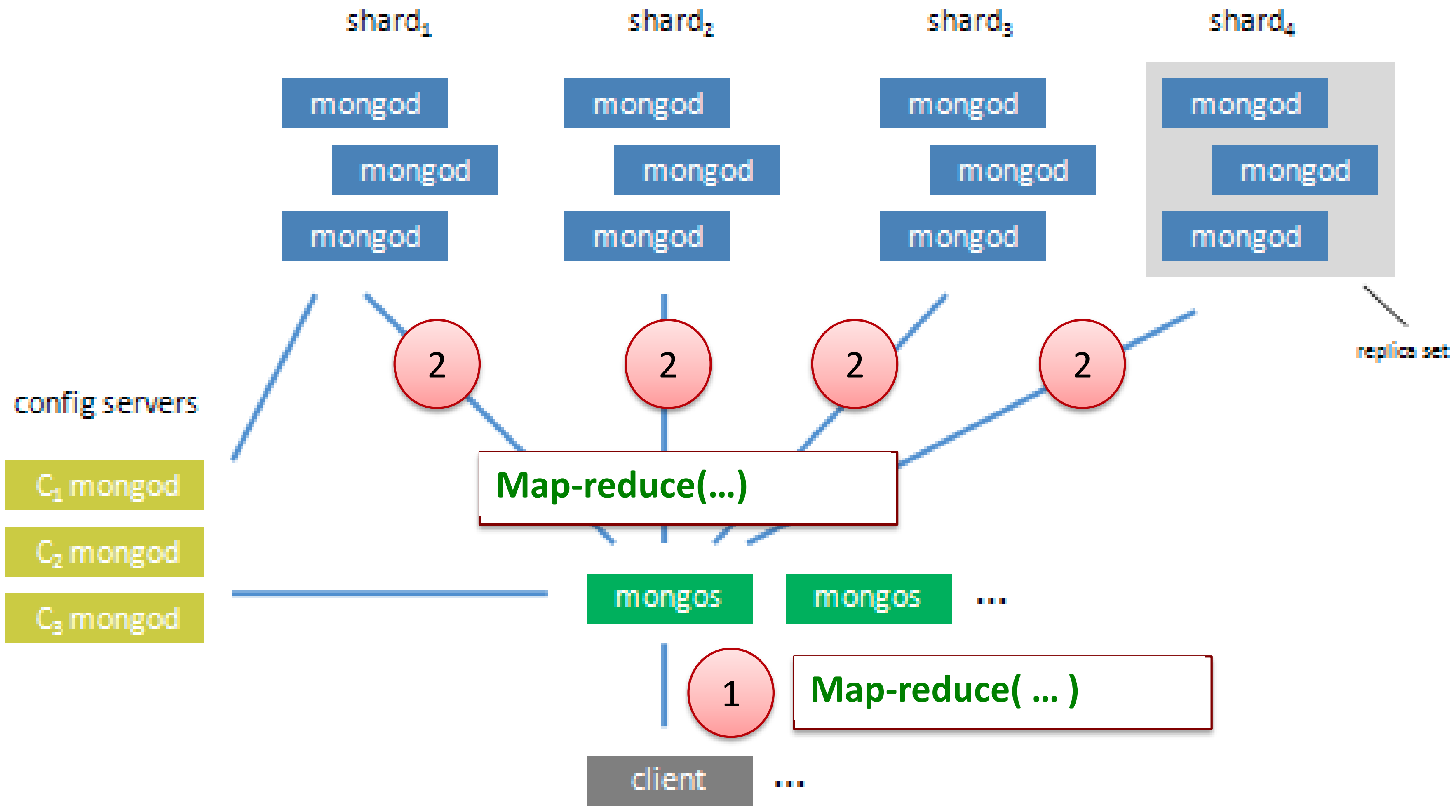

shard1
shard2
shard3
shard4
mongod
mongod
mongod
replica set
config servers
C1 mongod
C2 mongod
C3 mongod
mongos
mongos
...
client
...
3
Insert { 'name' : bob }
2
Shard key == name
bob → Shard 2
1
insert { 'name' : bob }

Querying



shard1
shard2
shard3
shard4
mongod
mongod
mongod
replica set
config servers
C1 mongod
C2 mongod
C3 mongod
mongos
mongos
...
client
...
3
Query { 'name' : bob }
2
Shard key == name
bob → Shard 2
1
Query { 'name' : bob }

Map Reduce



yottaa

Working with Mongo

- **MongoMapper makes it look like ActiveRecord**
- **Documents are more natural than rows in many cases**
- **Map-Reduce rocks (but needs better support in rails)**

http://www.flickr.com/photos/elhamalawy/2526783078/

```ruby
class Page
  include MongoMapper::Document
  key :url, :required => true, :indexed => true
  many :views, :class => View
end

class View
  include MongoMapper::Document
  key :created_at
  key :user_id
  belongs_to :page

  def before_save
    created_at = Time.now
  end
end
```

```ruby
class Page
  include MongoMapper::Document
  key :url, :required => true, :indexed => true
  many :views, :class => View
  many :links, :class => Link
end

class Link
  include MongoMapper::EmbeddedDocument
  belongs_to :page
  key :href
end
```

Ruby

Mongo

```
{
    "_id" : ObjectId("4bd0fd4814b55319f0000004"),
    "url" : "http://www.myawsomesite.com"
    "links" : [
        { "_id" :  ObjectId("4be3183f6a10fda8de0000f5"),
          "href" : "http://someothersite.com/page" },
        { "_id" : ObjectId("4be3183f6a10fda8de0000f6"),
          "href" : "/about_us.html" }
    ]
}
```

yottaa

```
class PageViewsByMonth

  def map
    <<MAP
      function() {
        emit( { 'page_id': this.page_id,
                'day' : new Date( this.time.getYear(),
                                  this.time.getMonth() ) }, 1 )
      }
    MAP
  end

  def reduce
    <<REDUCE
      function(key,values) {
        sum = 0;
        values.forEach(function(value) {
          sum += value;
        })
        return sum;
      }
    REDUCE
  end

  def build
    Views.collection.map_reduce( map, reduce )
  end

end
```

Runs over all the objects in the views table, counting how many times a page was viewed

Adds up all the counts for a unique url / date combination

Run the map reduce job and return a collection containing the results

Results

- **Version 1 of our analytics system took 2 weeks with 1 engineer**
  - We have since added a lot more complexity, but we did it incrementally
- **We replaced MySQL entirely with MongoDB**
  - No need for joins, transactions
  - Every table is now a document collection
- **It’s fast!**
  - 63ms – Average response time for sending data to server
  - 93ms – Average response time for displaying reports

yottaa