

MongoDB

Introduction

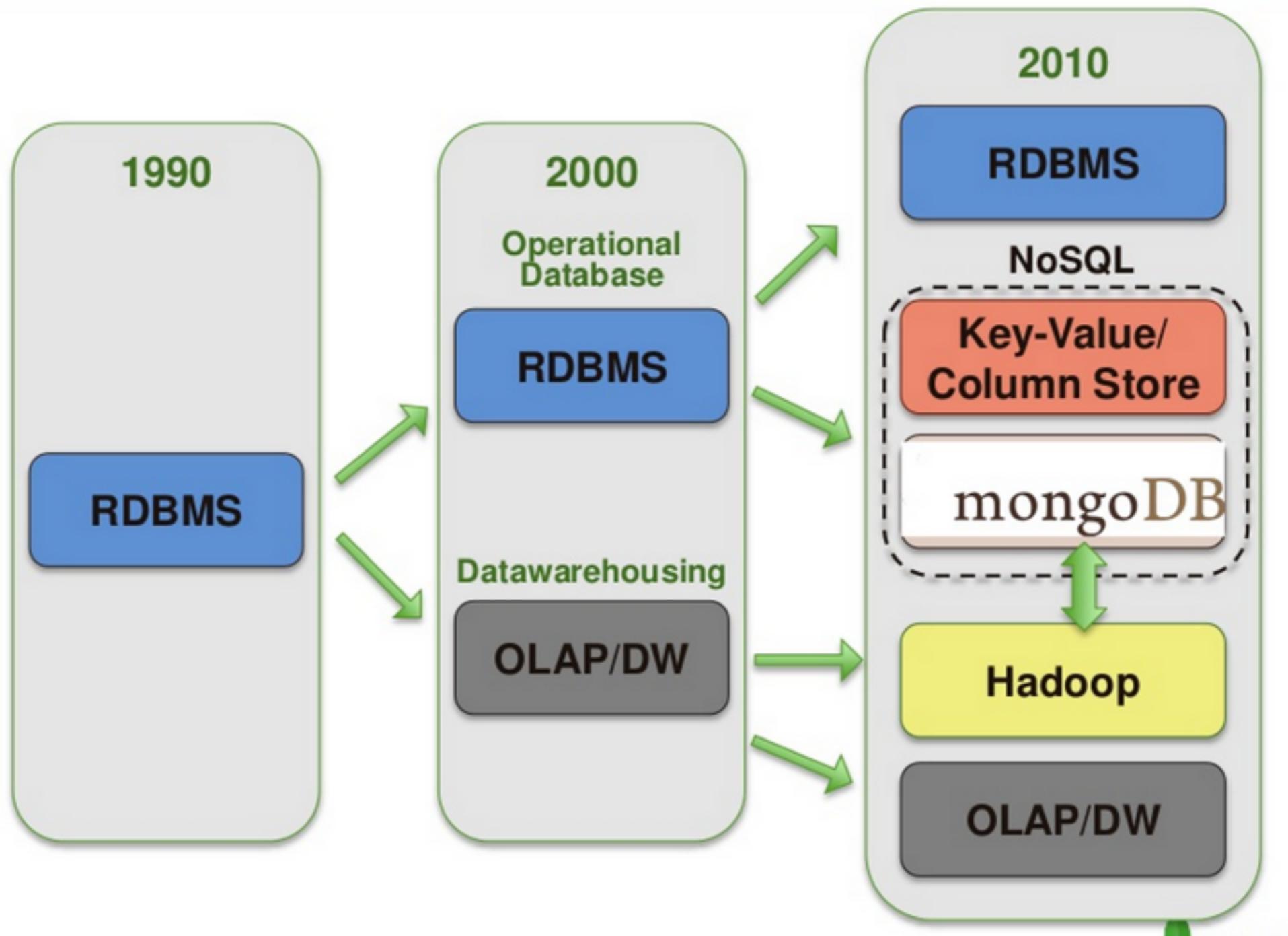
Current World

RDBMS

NOSQL

NEXT GEN
OLAP

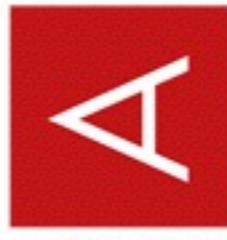
Evolution of database

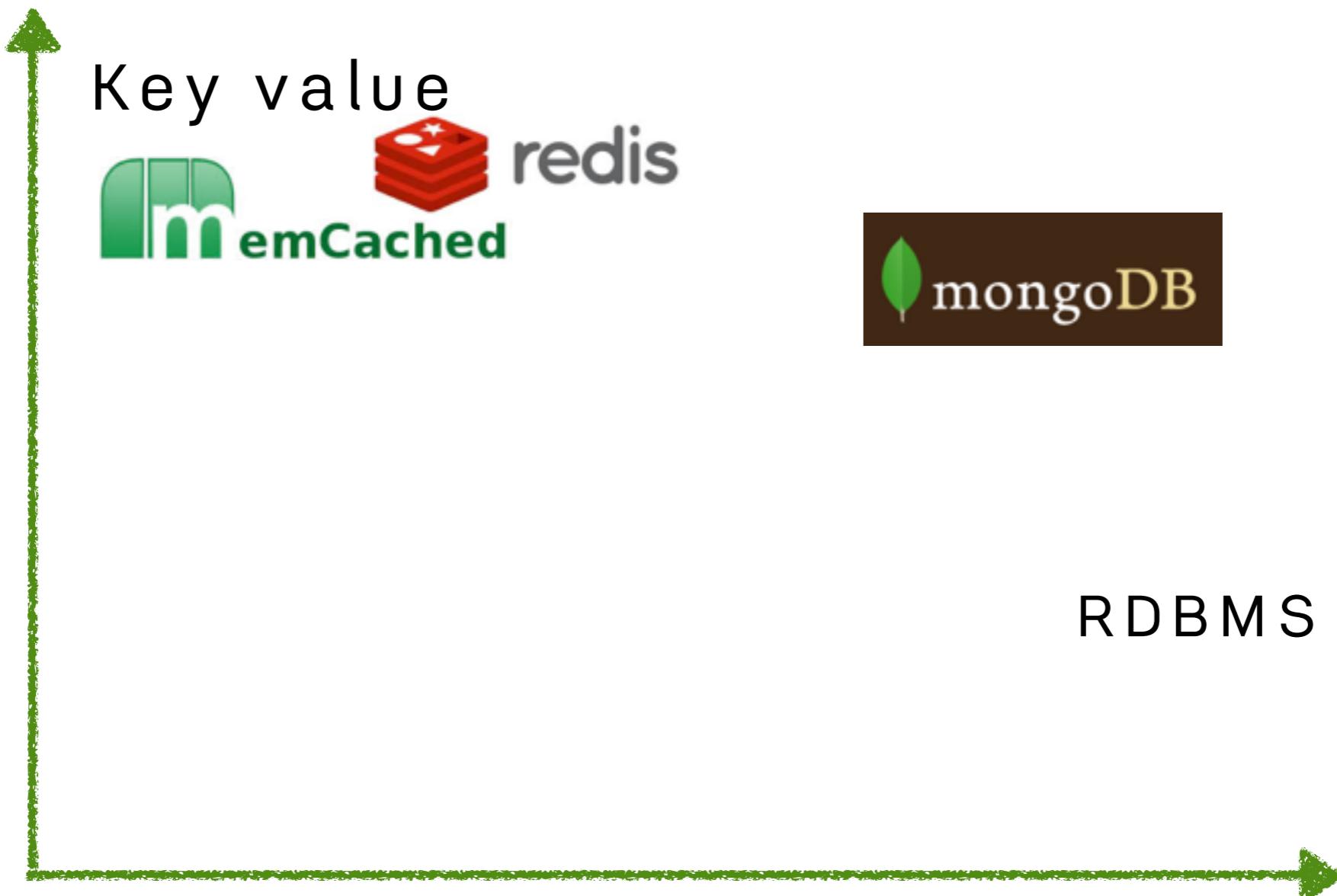


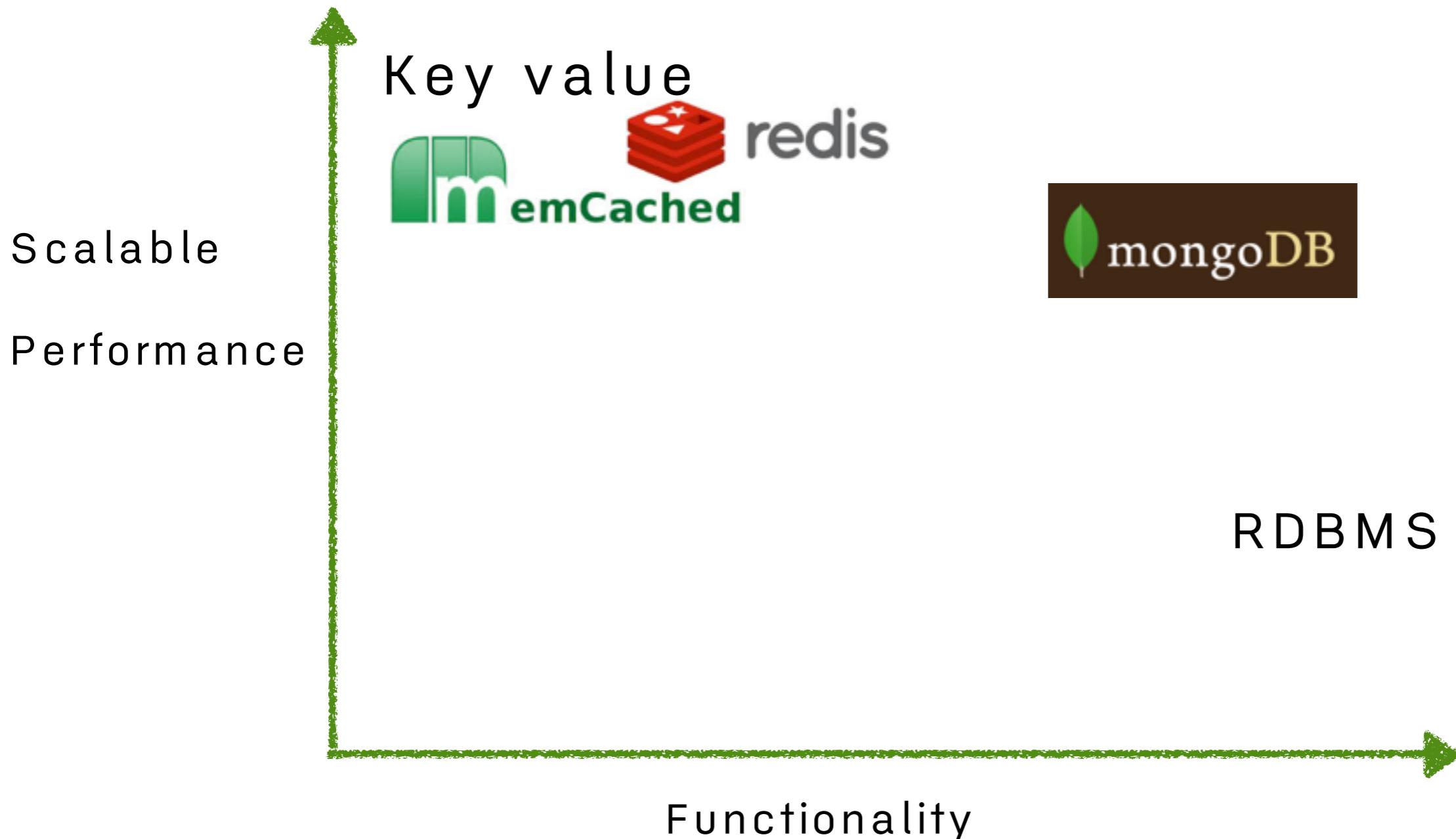
NoSQL ?

- Not Only SQL
- Non-relational database
- No joins
- No complex transaction
- Focus on Horizontal scalable
- Need new data model

NoSQL World

Document Database	Graph Databases
  	  
Wide Column Stores	Key-Value Databases
  	    





What is MongoDB ?

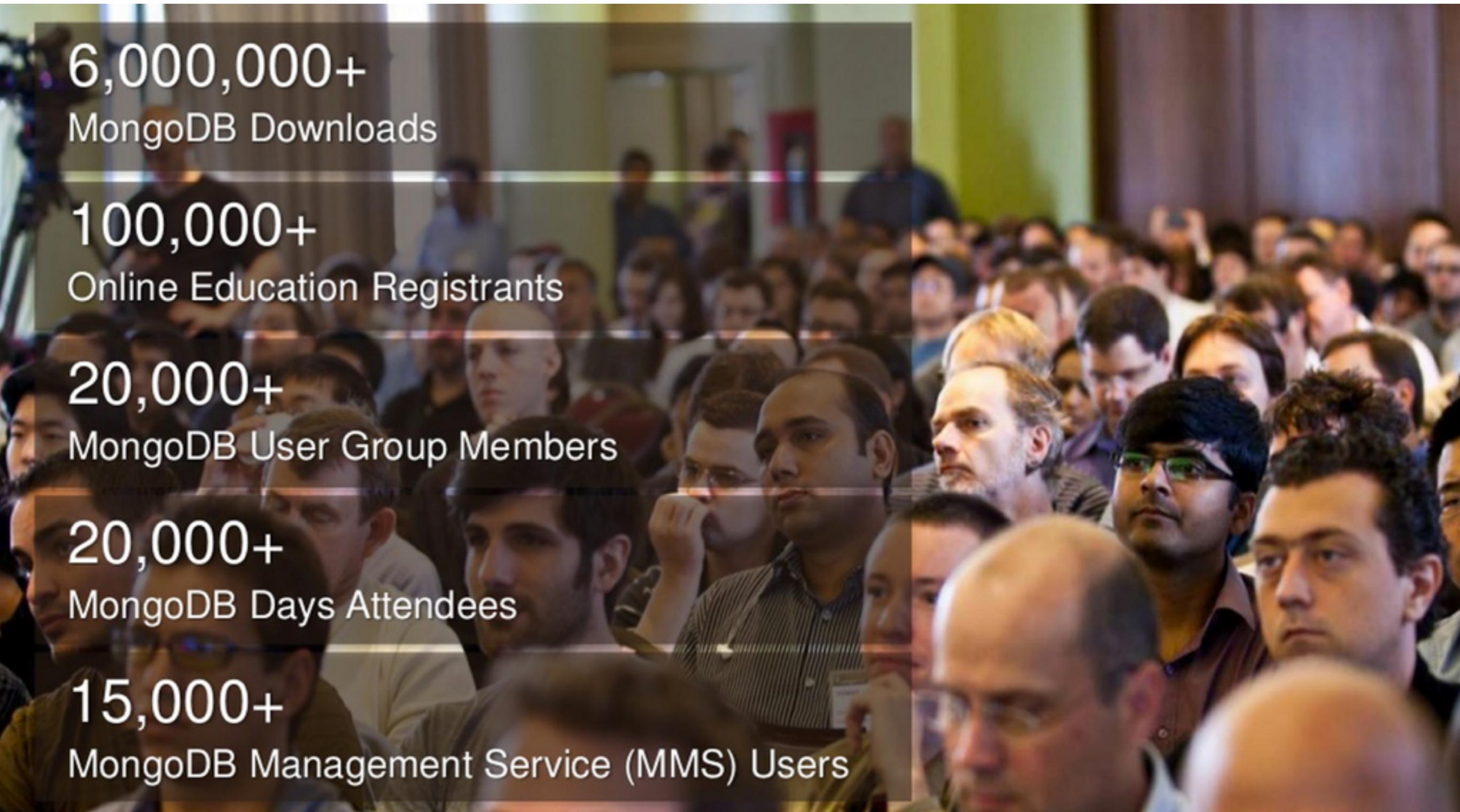
- **Humongous**
- Scalable
- High-performance
- Open source
- Schema free, flexible
- Document-oriented database

hu·mon·gous
/(h)yoo'ಮäNGgəs,-'məNG-/
adjective NORTH AMERICAN *informal*
huge; enormous.
"a humongous steak"

What is MongoDB ?

- No-joins
- No distributed transaction
- Atomic transaction per document

Community

A large audience of people, mostly men, sitting in rows, facing towards the left side of the frame, suggesting they are attending a presentation or conference. The background is slightly blurred.

6,000,000+
MongoDB Downloads

100,000+
Online Education Registrants

20,000+
MongoDB User Group Members

20,000+
MongoDB Days Attendees

15,000+
MongoDB Management Service (MMS) Users

One size fit all



One size fit all ไม่มีอยู่จริง !!



มาดู feature กันดีกว่า

JSON document as BSON

```
{ "hello" : "world" }
```



```
\x16\x00\x00\x00\x02hello  
\x00\x06\x00\x00\x00world  
\x00\x00
```

<http://bsonspec.org/>

Flexible schema

Book schema

```
{  
  "name" : "MongoDB",  
  "detail" : "... ..."  
}
```

```
{  
  "name" : "MongoDB",  
  "detail" : "... ...",  
  "tags" : ["computer"]  
}
```

Dynamic query

Javascript-based query language



```
db.books.find( {tags : "computer"} )
```

ອິນ້າ ອຶກມາກມາຍ

Atomic Update

Auto sharding

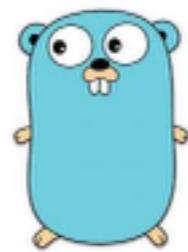
Replication

Many supported platform & language

Programming language



Java



Ruby



JavaScript



Perl



Python



Haskell

<http://docs.mongodb.org/ecosystem/drivers/>

Best use case

Scale out

Web

Caching

High volume

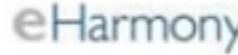
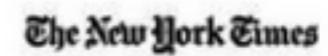
Less good use case

Highly transactional

Business intelligence

Require SQL

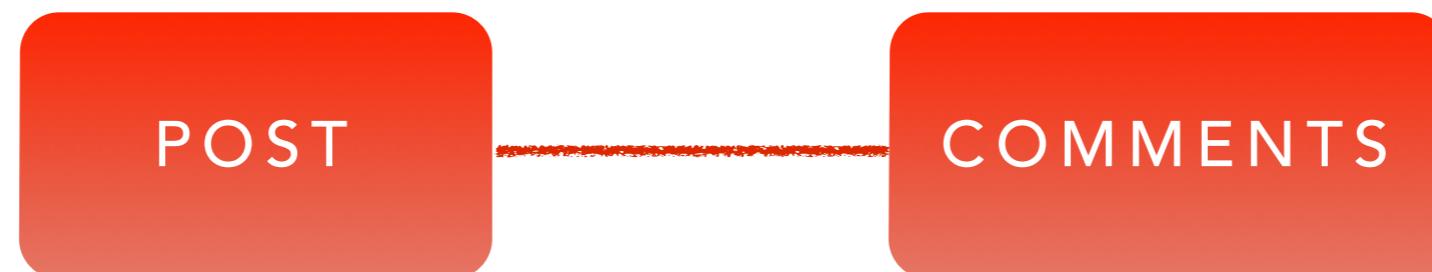
Use case

Big Data	Product & Asset Catalogs	Security & Fraud	Internet of Things	Database-as-a-Service
  	  	  Top Investment and Retail Banks Intelligence Agencies	Top Global Shipping Company Top US Retailer  Top Industrial Equipment Manufacturer	  Top Media Company Top Investment and Retail Banks
Mobile Apps	Customer Data Management	Data Hub	Social & Collaboration	Content Management
  	  	   	   	     

Download และ ติดตั้ง



Hello MongoDB



01_blog.txt

Post

```
{   _id: "1",
    topic: "Hello my blog",
    text: "this is my first blog",
    tag: [ "intro", "mongodb" ],
    author: "somkiat",
    date: new Date()
}
```

Comment

```
{   text: "Cool post!",  
    author: "somkiat",  
    date: new Date()  
}
```

New post

```
post =  
  
{   _id: "1",  
    topic: "Hello my blog",  
    text: "this is my first blog",  
    tag: [ "intro", "mongodb" ],  
    author: "somkiat",  
    date: new Date()  
}
```

```
db.posts.insert(post)
```

Add comment with embedding

```
db.posts.update(  
  {  
    _id: post._id  
  },  
  {  
    $push: { comments : comment }  
  }  
)
```

Let's go with query

Post by author

```
db.posts.find(  
  { author: "somkiat" }  
)
```

Last 10 posts

```
db.posts.find()  
      .sort( {date:-1} )  
      .limit(10)
```

1 = Ascending
-1 = Decending

<http://docs.mongodb.org/manual/reference/operator/aggregation/sort/>

Post from 15 Jan 2015

```
jan_1_2015 = new Date(2015, 1, 15)
```

```
db.posts.find(  
  {  
    date: { $gte: jan_1_2015}  
  }  
)
```

<http://docs.mongodb.org/manual/reference/operator/query-comparison/>

Post with a tag

```
db.posts.find(  
  { tag: "mongodb" }  
)
```

Improve performance

```
db.posts.ensureIndex(  
    { tag: 1 }  
)
```

For mongodb 3.0

```
db.posts.createIndex(  
    { tag: 1 }  
)
```

<http://docs.mongodb.org/manual/reference/method/db.collection.ensureIndex/>

Count posts

```
db.posts.count()
```

```
db.posts.find(  
  { tag: "mongodb" }  
).count()
```

Basic paging

```
page = 0
```

```
page_size = 10
```

```
db.posts.find().limit(page_size)
```

```
.skip( page * page_size )
```

<http://docs.mongodb.org/manual/reference/method/cursor.skip/>

Query on embedded document

```
db.posts.find( {"comments.author": "somkiat"} )
```

```
db.posts.ensureIndex( {"comments.author": 1} )
```

```
db.posts.createIndex( {"comments.author": 1} )
```

Migrate :: add category

```
post = {  
    tag: [ "intro", "mongodb" ],  
    author: "somkiat",  
    date: new Date(),  
    category: ["book"]  
}
```

post_id = db.posts.save(post)

<http://docs.mongodb.org/manual/reference/method/js-collection/> SPRINT3R

Advance query

\$gt, \$lt, \$gte, \$lte, \$ne, \$all, \$in, \$nin

```
db.posts.find({  
    $where: "this.author == 'somkiat' ||  
              this.topic == 'foo'"  
})
```

ຄູນເຫັນປໍລູກາໄໝ ?

```
db.posts.find( {"comments.author": "somkiat"} )
```

```
> db.posts.find( {"comments.author": "somkiat"} ).pretty()
{
    "_id" : "1",
    "topic" : "Hello my blog",
    "text" : "this is my first blog",
    "tag" : [
        "intro",
        "mongodb"
    ],
    "author" : "somkiat",
    "date" : ISODate("2015-03-09T17:59:46.089Z"),
    "category" : [
        "book"
    ],
    "comments" : [
        {
            "text" : "Cool post!",
            "author" : "somkiat",
            "date" : ISODate("2015-03-09T18:19:54.180Z")
        },
        {
            "text" : "Cool post!",
            "author" : "joe",
            "date" : ISODate("2015-03-09T18:22:49.245Z")
        }
    ]
}
```