# Everyday


Open-source, high-performance, schema-free, document-oriented database.

# Basic Terms

Document
BSON
Collection
GridFS

# Document

{ “**definition**”: “Basic unit of storage in MongoDB.”,
“**relational analog**” : “record”,
“**sample types**” : [ “utf-8 string”,
“integer”,
“object”],
“**special types**”: [“utc date”,
“binary”,
/regular expression/],
“**storage format**”: “BSON”}

# BSON

## Binary JSON

- Efficient, fast, rich in types
- Client serializes to BSON
- Data stored at BSON in MongoDB
- MongoDB fluent in BSON

## General-purpose

- RPC / Custom protocols

# Collection

**Group of Documents**

- Analogous to Relational Tables.
- Supports up to 40 indexes.
- Informal namespacing:
  - blog.posts, blog.authors
- Schema-free.

# GridFS

**Storing large files in MongoDB**

- GridFS: specification for chunking files.
- Chunked for performing range operations.
- File metadata stored in a files

# Technical Bits

Administration / Drivers
Javascript Shell
Replication
Sharding

# Administration

## Socket / Wire Protocol

- Drivers for
  - Java
  - Perl
  - PHP
  - Python
  - Ruby

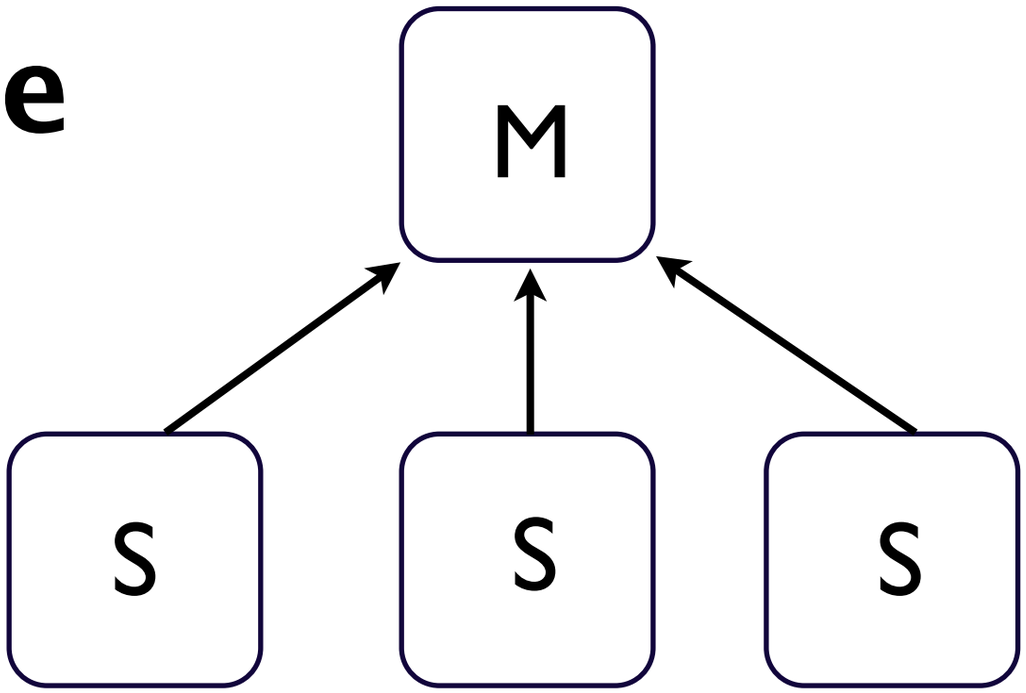
Technical

# Javascript Shell

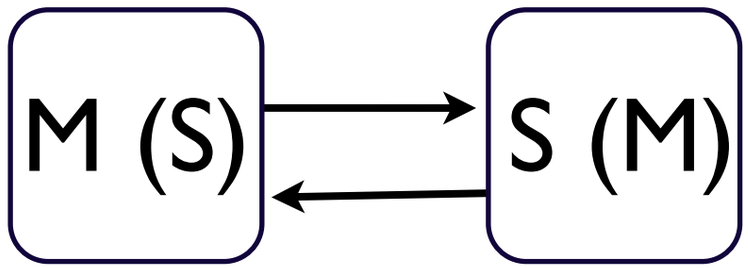
```
MongoDB shell version: 1.1.2
url: test
connecting to: test
type "help" for help
> db.addresses.find({}, {"street": 1, "location": 1})
{ "_id" : ObjectId("4ae75756090ef60873460185"), "street" : "17 W. 18th St." }
{ "_id" : ObjectId("4ae7576f090ef60873460186"), "street" : "149 Fifth Ave" }
{ "_id" : ObjectId("4ae75943090ef60873460187"), "street" : "25 W. 34 St.", "location" : 0 }
{ "_id" : ObjectId("4ae75943090ef60873460188"), "street" : "25 W. 34 St.", "location" : 1 }
{ "_id" : ObjectId("4ae75943090ef60873460189"), "street" : "25 W. 34 St.", "location" : 2 }
{ "_id" : ObjectId("4ae75943090ef6087346018a"), "street" : "25 W. 34 St.", "location" : 3 }
{ "_id" : ObjectId("4ae75943090ef6087346018b"), "street" : "25 W. 34 St.", "location" : 4 }
{ "_id" : ObjectId("4ae75943090ef6087346018c"), "street" : "25 W. 34 St.", "location" : 5 }
{ "_id" : ObjectId("4ae75943090ef6087346018d"), "street" : "25 W. 34 St.", "location" : 6 }
{ "_id" : ObjectId("4ae75943090ef6087346018e"), "street" : "25 W. 34 St.", "location" : 7 }
{ "_id" : ObjectId("4ae75943090ef6087346018f"), "street" : "25 W. 34 St.", "location" : 8 }
{ "_id" : ObjectId("4ae75943090ef60873460190"), "street" : "25 W. 34 St.", "location" : 9 }
{ "_id" : ObjectId("4ae75943090ef60873460191"), "street" : "25 W. 34 St.", "location" : 10 }
{ "_id" : ObjectId("4ae75943090ef60873460192"), "street" : "25 W. 34 St.", "location" : 11 }
{ "_id" : ObjectId("4ae75943090ef60873460193"), "street" : "25 W. 34 St.", "location" : 12 }
{ "_id" : ObjectId("4ae75943090ef60873460194"), "street" : "25 W. 34 St.", "location" : 13 }
{ "_id" : ObjectId("4ae75943090ef60873460195"), "street" : "25 W. 34 St.", "location" : 14 }
{ "_id" : ObjectId("4ae75943090ef60873460196"), "street" : "25 W. 34 St.", "location" : 15 }
{ "_id" : ObjectId("4ae75943090ef60873460197"), "street" : "25 W. 34 St.", "location" : 16 }
{ "_id" : ObjectId("4ae75943090ef60873460198"), "street" : "25 W. 34 St.", "location" : 17 }
has more
> db.addresses.count()
202
>
```

# Replication

## Master - Slave

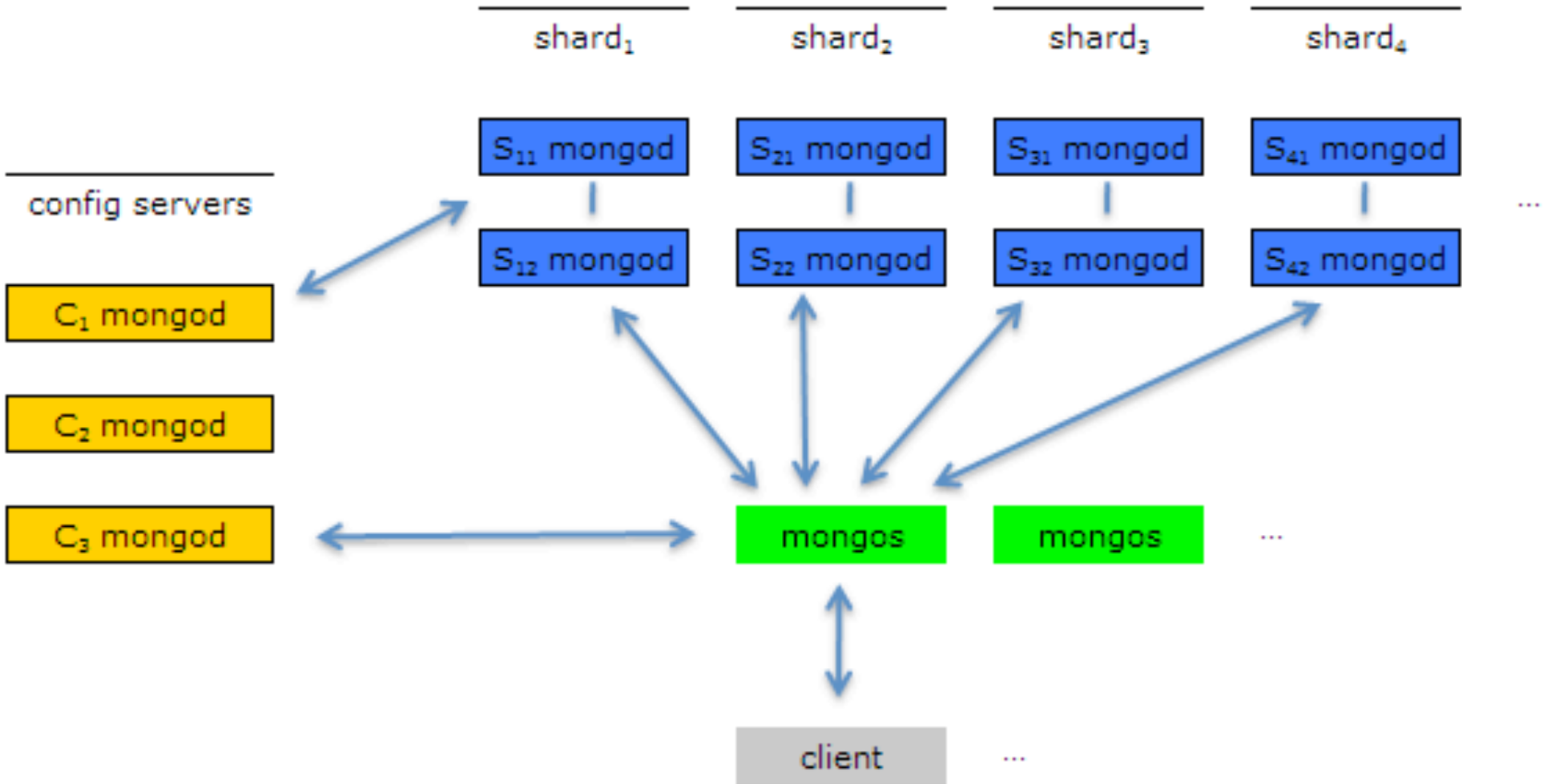



## Replica Pairs



Technical

# Auto Sharding



# Features

Sophisticated, Dynamic Queries
Map/Reduce
Upserts
Capped Collections

# Sample Document

{“**title**”: “NoSQL Database Internals”,

“**date**”: “Tue Oct 27 2009 17:24:18 GMT-0400 (EDT)”

“**tags**”: [“nosql”, “databases”, “c++”],

“**comments**”: [{“author”: “Fred”,

“text”: “Quite

# Dynamic Queries

```
db.posts.find({“author”:

“Kyle”})

db.posts.find(“tags”: “nosql”)

db.posts.find(

  {“date”: {“$lte”: Date.now})
```

# Queries / Indexes on Nested Documents

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

db.posts.find("comments.author":
"fred")

db.posts
.find({}, {"comments":1})
```

# Query Operators

- $gt, $lt, $gte, $lte, $ne, $all, $in, $nin
- where()
- db.posts.find({“$where”:  this.author == “kyle” || this.author === nil})

# And...

- Map/Reduce
- Count
- Group
- Distinct
- Regex queries
- Upserts
- Capped Collections

# MongoDB in Production

BusinessInsider.com
TweetCongress.org / Squeejee
SourceForge.net

# BusinessInsider.co

October 24, 2009

THE BUSINESS INSIDER

Log In Using Facebook | Log In | Register

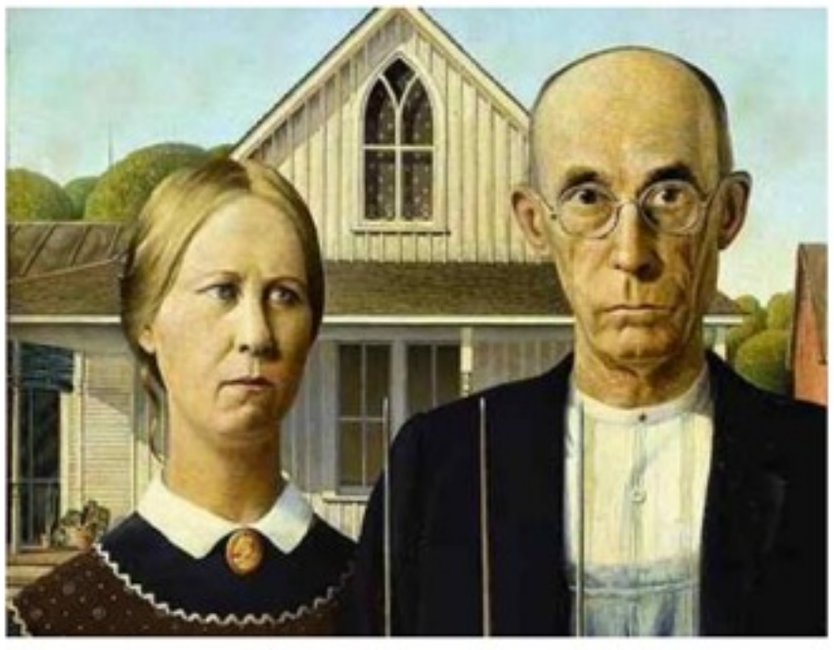


HOME | SAI | CLUSTERSTOCK | MONEY GAME | GREEN SHEET | WAR ROOM | LAW REVIEW | LATEST | THE TAPE | Resources

Hot » iPhone | Android | Research in Motion | Microsoft | China | Stock Market ... more topics

SEARCH

## Debt Stress in Middle Class America



Naked Capitalism | 2,393 | 47

"I tried to use my card about 3 weeks ago at the grocery store and it was denied. Needless to say, I walked out without the food. We don't waste anything, not money,

### The National Association Of REALTORS Is Getting It All Wrong

Henry Blodget

1,548 | 17

### More

Roubini: I hate gold

The media revolts against Obama

Attorney in deep trouble over Craigslist sex ad

Traders livid at CNBC

NEWSLETTER Your Email Address SIGN-UP

Got a tip? Let us know!

Markets

**Dow** -109.13 at 9972.18

**Nasdaq** -10.82 at 2154.47

# Overview

- MongoDB since Jan, 2008
- 12M page views / month
- 2.2M uniques / month

# Data Modeling

- 10 Collections
- Posts with embedded comments
- Settings
- Real-time analytics (heat maps)
- Users with Facebook connect.

# Analytics

- Page views, real-time statistics panel
- Dynamic queries ease data analysis
- Fast updates: non-locking inserts
- Sailthru for click tracking

# Images with GridFS

- All data, images included, in db.
- Meta-data alongside images.
- On-the-fly sizing.
- Possible con: database large.

# Migrations Mitigated

- Simplifies schema changes.
  - No “alter” statements.
- Easily create new fields, collections.
- Certain data migration still necessary.

{"**_id**" :
ObjectId( "f4b9b9146c65f647af..."),
"**name**" : "2008/4/flixster-can-t-sell-
to-i..." ,
"**title**" : "Flixster Can't Sell To IAC, Raises
$5 M..." ,
"**commentsEnabled**" : true ,
"**ts**" : "Fri Apr 04 2008 13:43:00 GMT-0400
(EDT)" , "**cls**" : "entry" ,
"**content**" : "<p>Movie rating site/social
network... "
"**author**" : "Dan Frommer" ,
"**excerpt**" : "Who wants to pay $150
million..." ,
"**channel**" : "allvoxinsider"

"**comments**" :
[{"**author**" : "Alphanaliste" ,
"**email**" :
"harderwisdom@aol.com" , "**url**" :
"" ,
"**ts**" : "Fri Apr 04 2008 15:18:31 GMT-0400
(EDT)" ,
"**text**" : "This junk won't
survive..." ,
"**cid**" :
ObjectId( "f4b9b91407...") },
{"**author**" : "jenkins" ,

# Deployment

- Single database server, 16GB, 15-30%.
- 3 Apache/PHP servers.
- MongoDB for caching.
- Exception: memcached for homepage.

# Tweetcongress.org
# Floxee.com (Squeejee)

# Overview

- MongoDB since May, 2009
- Mash-up millions of tweets
- Ruby Driver, MongoMapper

# Data Modeling

- Version-controlled schema
- Code defines the schema
- Migrations simplified

# Advantages of MongoDB

- Good for API Integration
- Upserts, fast writes
- API data to documents

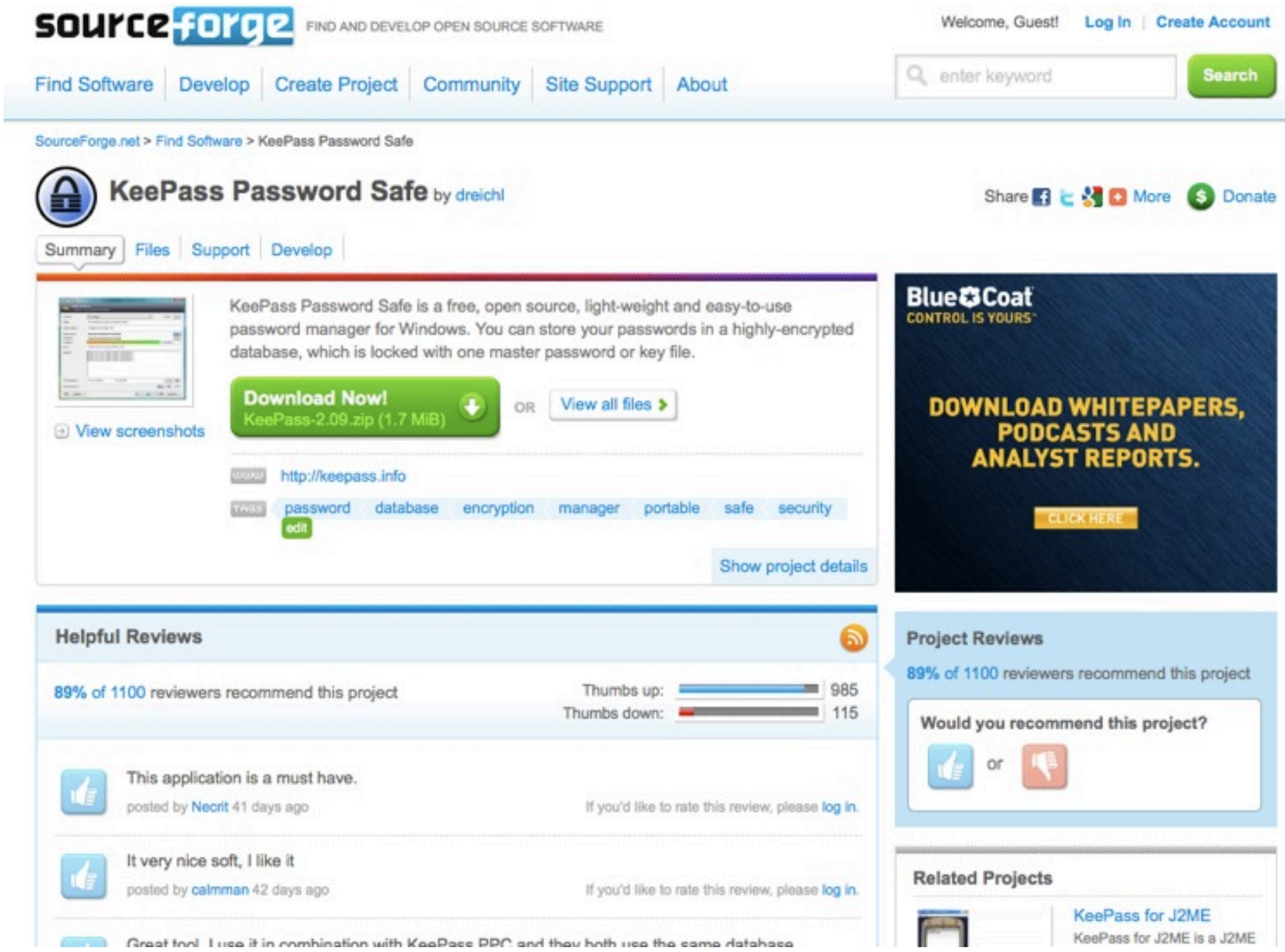
# Advantages of MongoDB

- Modeling associations
  - Embedded document vs Separate Collection
  - Embedded arrays for tagging, simple associations.

# Deployment

- 64-bit EC2, Rackspace Cloud
- Single master, snapshots
- Heroku with MongoHQ for smaller, internal

# SourceForge.net



# Overview

- Goal: Front pages, project pages, download pages stored in single MongoDB document.

- Master / 5-6 Read-only slaves
  - Scaling for reads, reliability.
  - 10x current traffic, with 100x linearly.
- Python Driver with Turbogears

# Why Use

Document Model
Powerful, Dynamic Queries
Binary Storage
Scalability

Pre-compiled binaries
Great documentation
Multi-language support


- http://www.mongodb.org
- irc.freenode.net#mongodb
- mongodb-user on google groups
- kyle@10gen.com