software pilots
TRIFORK.

# Frictionless Persistence in .NET with MongoDB

Mogens Heller Grabe
Trifork
mhg@trifork.com

# Agenda

- Document-oriented databases
- Introduction to MongoDB
  - JavaScript, baby!
- How to do it with C#
- Tiny web app sample
- Some (grander) usage scenarios
- Who's using it

# Goals

- Communicate that NoSQL can relieve us of some of the pain we're experiencing
- Spread the word that NoSQL is also very relevant to people engaged in enterprisey platforms in enterprisey environments
- Allow for just enough understanding of MongoDB to allow it to be considered as a possible future technology choice

# Who might be interested?

- Developer who
  - doesn't know too much about MongoDB
  - feels the relational agony
  - is interested in technologies with an agile mindset
  - 's not necessarily a C# dude (but there will be C# ☺)

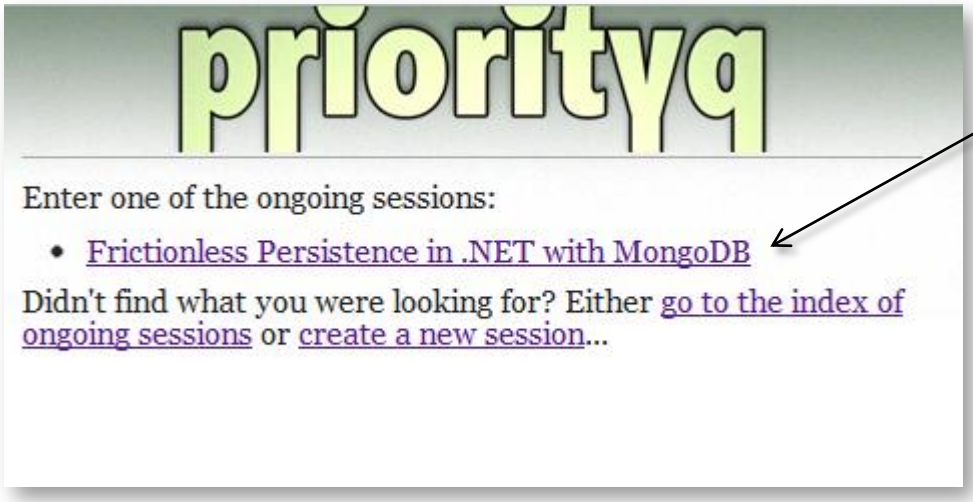
# Who am I?

```
{
    name: ”Mogens Heller Grabe”,
    occupation: ”Software Pilot @ Trifork”,
    preferences: {
        likes: [”CLR”, ”OSS”,
            ”Building great stuff for customers”,
            ”node.js”, ”Ruby”, ”(...)”],
        dislikes: [”Pain”, ”Geeky introductions”]
    },
    contact_info: {
        email: ”mhg@trifork.com”,
        twitter: ”mookid8000”,
        blog: ”http://mookid.dk/oncode”
    }
}
```

# An experiment...

- If you carry some kind of mobile, web-enabled device, and you have a question:

**`http://priorityq.apphb.com`**




go here and ask a question and/or vote on other questions

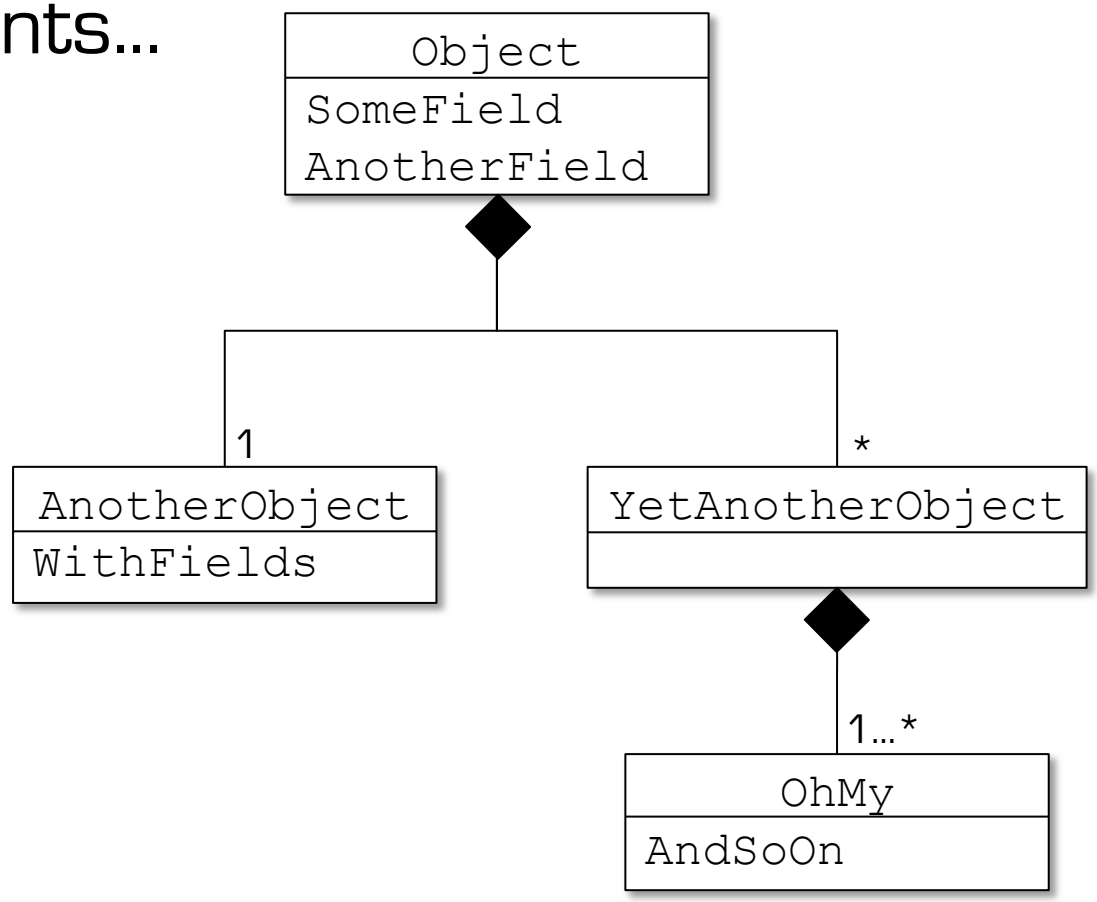
(or if you feel like voting on other people's questions)


First, a few words about

# DOCUMENT-ORIENTED DATABASES

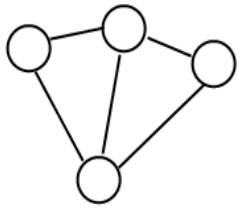
# Document DBs

- Store documents...



# Document DBs

- Common properties
  - Schemaless/dynamic
  - No joins
  - Unit of atomicity: One document
  - Can be distributed
  - Quite performant



# DocumentDBs

- Document-oriented databases
  - Lotus Notes
  - CouchDB
  - MongoDB
  - RavenDB
  - SimpleDB
  - Redis
  - Riak

  (.....etc...etc.....etc.....)


Now it's about time we get an

# INTRODUCTION TO MONGODB

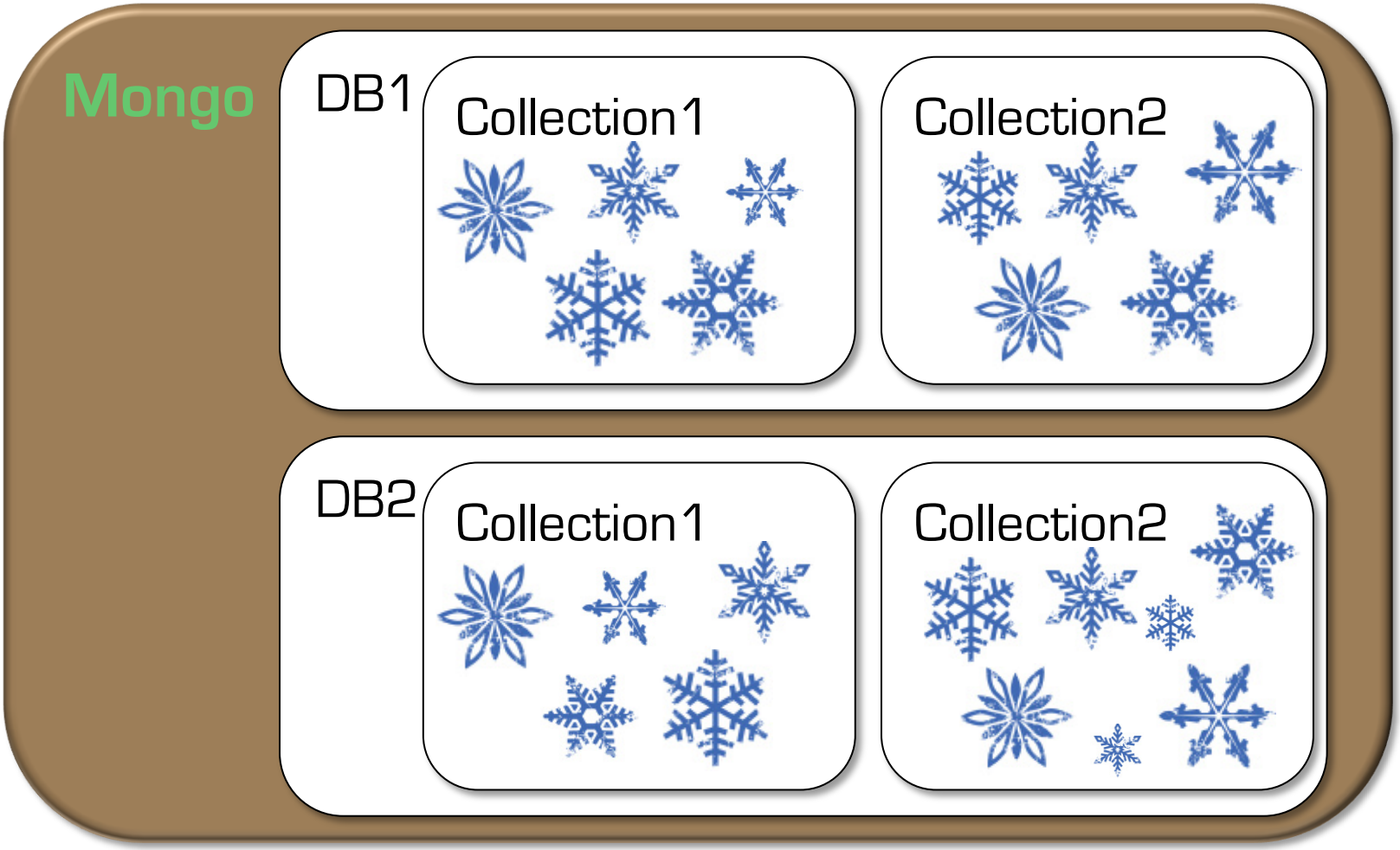
# MongoDB overview

- ”MongoDB as in hu**MONGO**us, not retarded...”
- Developed (and commercially supported) by 10gen
- **Free and open source**
- Currently in a stable **1.8.1** (odd minor version numbers are development versions)

# MongoDB overview

- Document-oriented database
  - like CouchDB, RavenDB, and many others
- Stores JSON (Binary jSON actually, BSON)
  - i.e. there's no fixed schema
- Unlike most document dbs, MongoDB can
  - be ad hoc queried
  - perform in-place updates
    (i.e. it can modify documents on the server)
  - is NOT REST-based
    (*can* expose a REST endpoint though)

# Data organization




Instance ➔ Database ➔ Collection ➔ Document

Snowflakes idea ripped from MongoDB – The Definitive Guide ☺

# Features

- Ad hoc querying
- Indexes
  - Normal, as well as geo-spatial
- Server-side updates

we'll take a peek at these things today...

- Map/reduce
  - Simple map/reduce implementation allows for computations that aggregate across documents and embedded documents
- GridFS
  - Standard for streaming large blobs of data to/from documents
- Durability
  - Either through replication or through journaling (1.8+)
- Scalability and fault tolerance
  - via auto-sharding and replica sets

# Documents, queries, updates

- Documents are JSON
  - `{_id: ObjectId("..."), user_name: "joe"}`
- Queries are documents
  - `{user_name: "joe"}`
  - `{user_name: /^joe.*/}`
  - `{user_name: {$in: ["joe", "moe"]}}`
- Updates are documents
  - `{_id: 21, user_name: "joe", txt: "bla"}`
  - `{$inc: {hits: 1}}`
  - `{$addToSet: {tags: "new_tag_to_add"}}`

# Let's see...

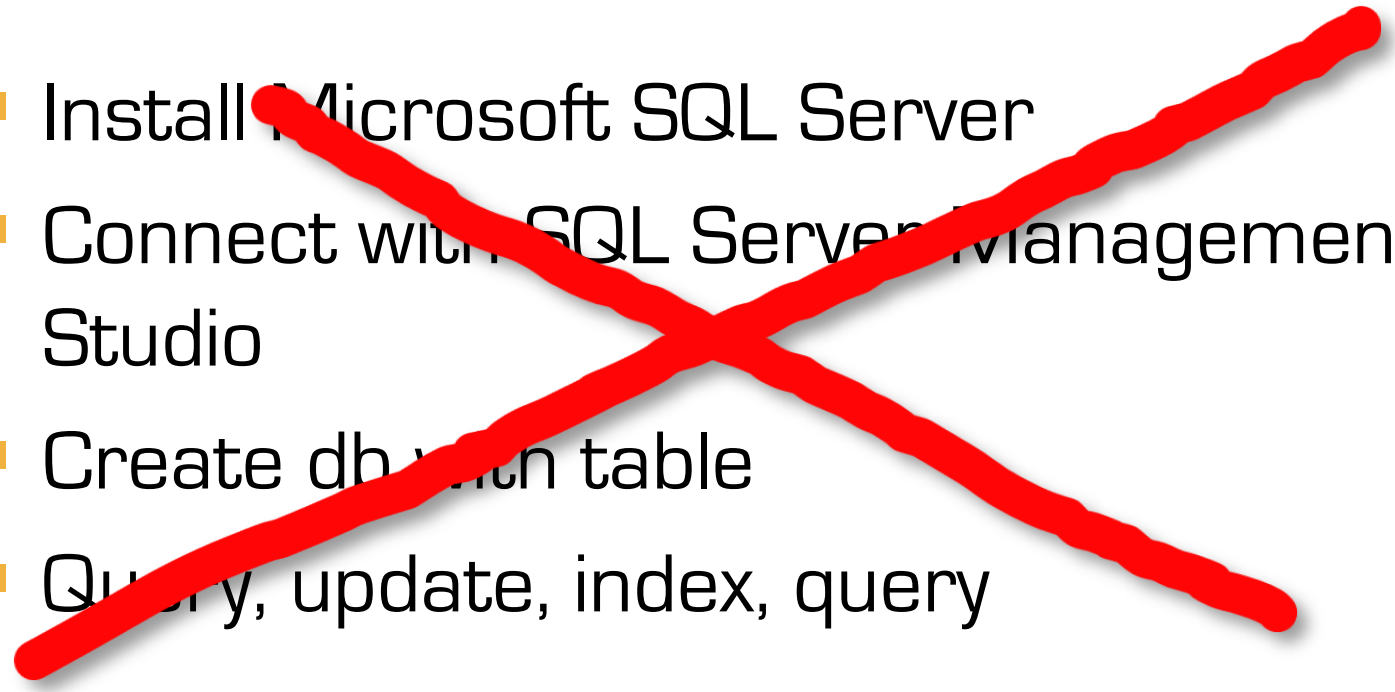
DEMO

- Install MongoDB
- Connect with Mongo Shell
- Create db with collection
- Query, update, query
- Query, explain, index, explain

# Let's try and compare...

DEMO

- Install Microsoft SQL Server
- Connect with SQL Server Management Studio
- Create db with table
- Query, update, index, query



*just kidding* ☺


Now let's take a look at

# HOW TO DO IT WITH C#

# MongoDB and .NET

- Available .NET drivers
  - Official
    - mongo-csharp
  - Community
    - NoRM
    - mongodb-csharp
    - simple-mongodb

    (...)

# Official driver: mongo-csharp

- Two parts:
  - MongoDB.Driver
    - Connect and manage the db
    - Interact with ”raw documents”
  - MongoDB.Bson
    - BSON-serialization in general
    - Serialize complex datatypes

# Let's see...

DEMO

- Connect with mongo-csharp
- Insert, query, update, query
- Insert complex model


Now let's take a look at a

# TINY WEB APP SAMPLE

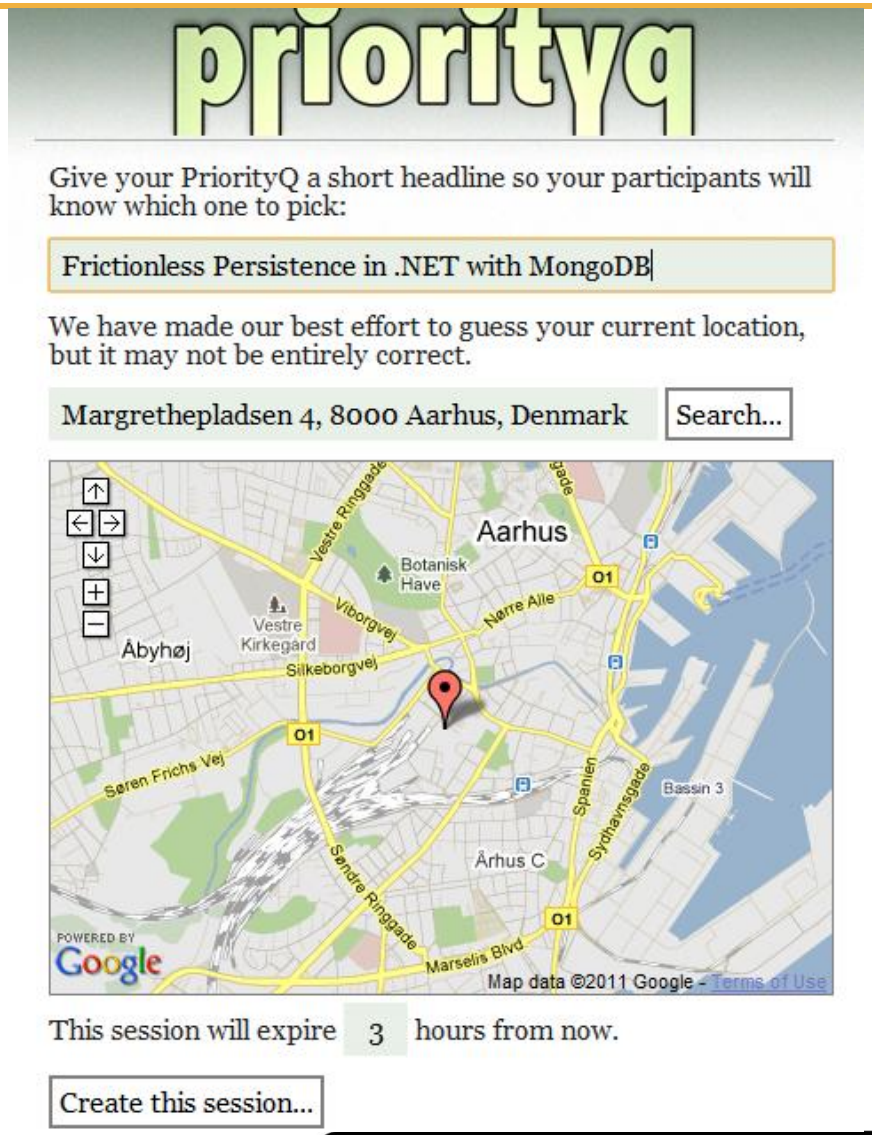
# Purpose


- Collect questions from audience during presentations
- Allow audience to vote
- Should be easy to find the session you're at

# Required functionality


- Session has title, location, duration and questions
- Question has text and votes
- Create new session
- Add question to session
  - Only if question does not already exist
- Increment votes on question
  - Only if person has not voted on that question before
- Search for ongoing sessions near current location
- Show all ongoing sessions
- Be fairly accessible from both desktop & mobile

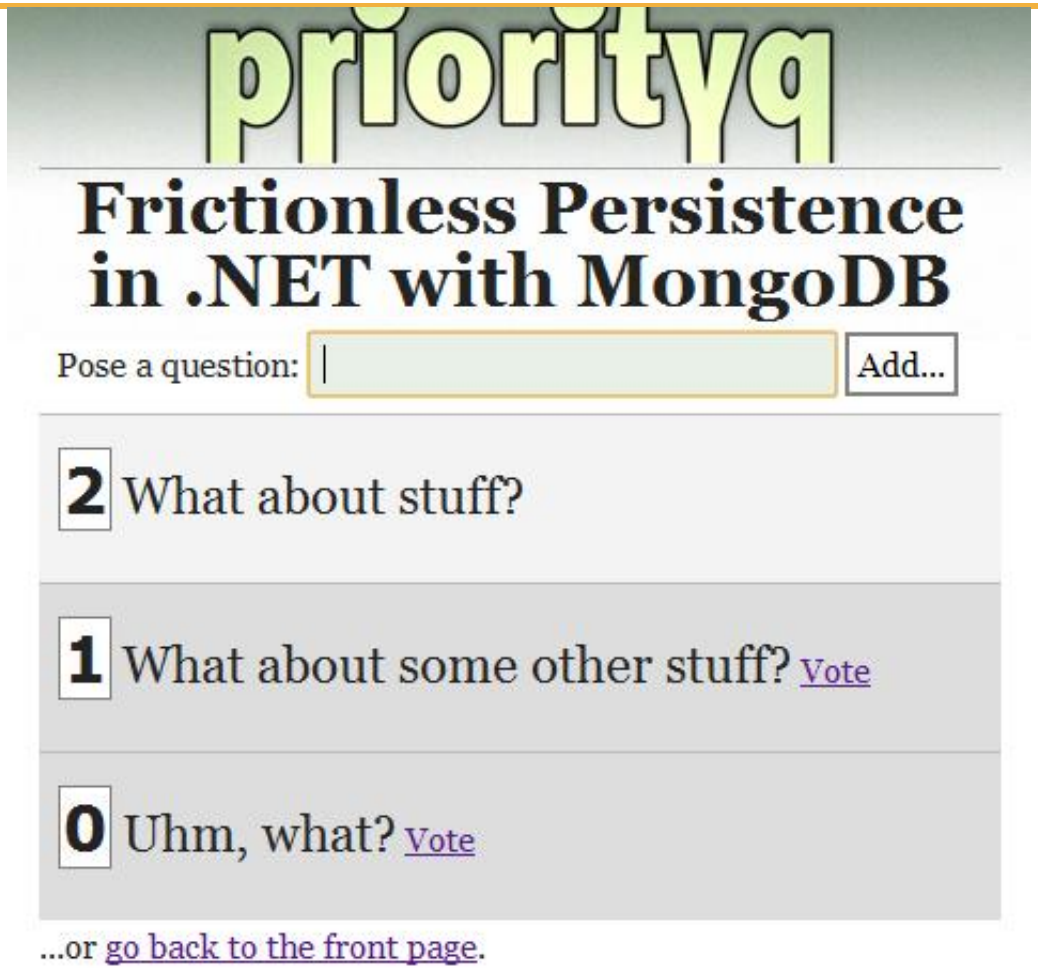
# PriorityQ


prioritYq
Give your PriorityQ a short headline so your participants will know which one to pick:
Frictionless Persistence in .NET with MongoDB
We have made our best effort to guess your current location, but it may not be entirely correct.
Margrethepladsen 4, 8000 Aarhus, Denmark
Search...
Aarhus
Botanisk Have
Vestre Ringgade
Viborgvej
Nørre Alle
Vestre Kirkegård
Åbyhøj
Silkeborgvej
Søren Frichs Vej
Spanien
Bassin 3
Sydhavnsgade
Århus C
Søndre Ringgade
Marselis Blvd
O1
POWERED BY Google
Map data ©2011 Google - Terms of Use
This session will expire 3 hours from now.
Create this session...

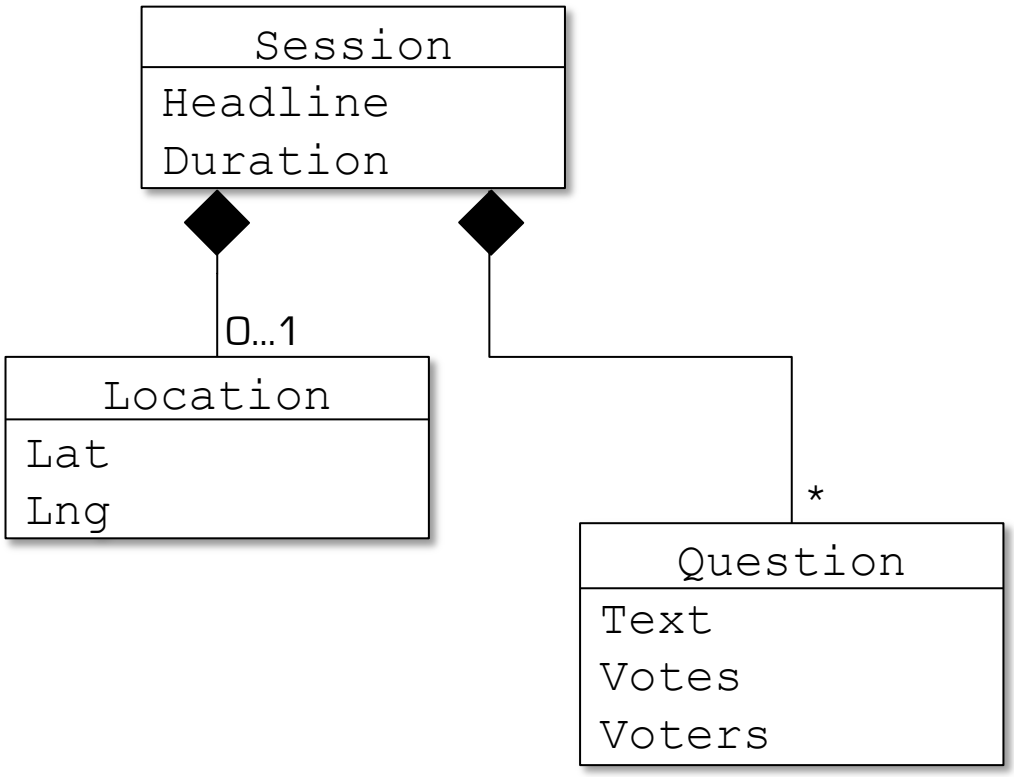
# PriorityQ


prioritYq
Enter one of the ongoing sessions:
Frictionless Persistence in .NET with MongoDB
Some other event in Aarhus
Didn't find what you were looking for? Either go to the index of ongoing sessions or create a new session...

# PriorityQ


priorityq
Frictionless Persistence in .NET with MongoDB
Pose a question:
Add...
2 What about stuff?
1 What about some other stuff? Vote
0 Uhm, what? Vote
...or go back to the front page.

# The model (roughly)



# Questions within a session

- Session is at the document level, and among other things it has an array of questions

```
{
        // (...)
        Questions: [
                {Text: ”blah blah?”, Votes: 0},
        ]
}
```

# Add question to session

C#:

```
var sessions = database.GetCollection<Session>("Session");
var text = "blah?";

var criteria = Query.And(Query.EQ("_id", id),
                         Query.NE("Questions.Text", text));

var update = Update.AddToSet("Questions",
                         new Question {Text = text});

sessions.Update(criteria, update);
```

```
{       // (...)
        Questions: [
                {Text: "blah blah?", Votes: 0},
                {Text: "blah?", Votes: 0}
        ]
}
```

# Increment votes on question

C#:

```
var sessions = database.GetCollection<Session>("Session");

var criteria = Query.And(Query.EQ("_id", id),
                         Query.NE("Questions.1.Voters", "joe"));

var update = Update.Inc("Questions.1.Votes", 1)
                   .Push("Questions.1.Voters", "joe"),

sessions.Update(criteria, update);
```

```
{       // (...)
        Questions: [
                {Text: "blah blah?", Votes: 0},
                {Text: "blah?", Votes: 1, Voters: ["joe"]},
        ]
}
```

# Location within session

- Each Session can have a location object containing latitude and longitude coordinates

```
{       // (...)
        Location: {
                Lat: -40.232556,
                Lng: -45.2535353
        }
}
```

# Create geospatial index

- Geospatial index can be added to any object containing exactly two floating point values (at most one geospatial index per collection though)

**C#:**

```
var sessions = database.GetCollection<Session>("Session");

sessions.CreateIndex(IndexKeys.GeoSpatial("Location"));
```

**Mongo shell:**

```
use PriorityQ

db.Session.ensureIndex({Location: "2d"})
```

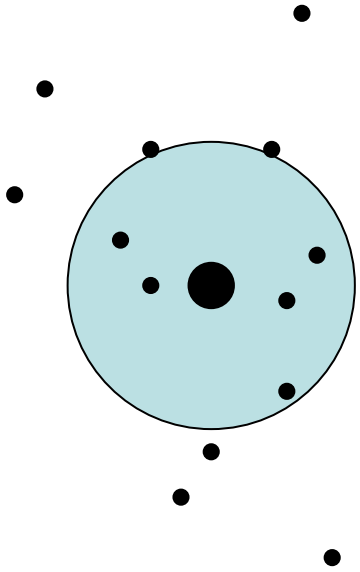
# Query geospatial index

C#:

```
var sessions = database.GetCollection<Session>("Session");

var query = Query.Near("Location", lat, lng, maxDistance);

sessions.Find(query);
```



# Conclusion

- Was very easy to
  - Get started persisting some data
  - Satisfy invariants
  - Extend the model with geospatial capabilities
  - Look at and modify data while developing
  - Run tests on local MongoDB
  - Deploy app to AppHarbor using MongoHQ for hosting my MongoDB instance

By now, it's obvious that MongoDB is pretty cool and easy to get started with, and it has a pretty sweet spot in backends for e.g. mobile, possibly location-aware apps... but how about

# SOME (GRANDER) USAGE SCENARIOS

# CQRS

- Event store
  - Persist batches of events in one document

    (NServiceBus-committer Jonathan Oliver has MongoDB support in his EventStore project: https://github.com/joliver/EventStore)
- Query store
  - Quick to add new views
  - Take advantage of ad-hoc querying

# Caching

- Example on implementing a distributed cache by Peter Bromberg: http://www.eggheadcafe.com/tutorials/aspnet/93206c89-09c9-40fc-9296-7d74bb7996ad/a-mongodb-cache-utility.aspx

# Central logging database

- There's a Log4net appender for MongoDB: Log4Mongo: https://github.com/log4mongo/log4mongo-net

  (just punch in a few lines of XML, and then you're good to go....)

# Storing aggregate roots

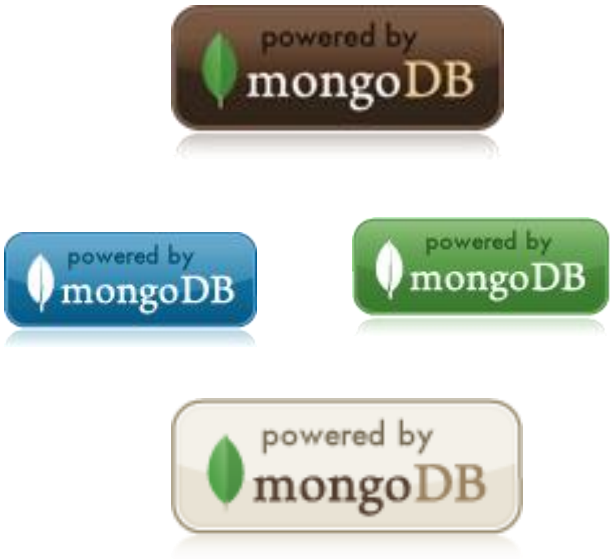
- Document == DDD ”aggregate root”
  - Objects beneath the root live & die with the root
  - A root represents a transactional boundary

  (no more ”how do I load an entire aggregate root in one query with NHibernate?)..... ☺

# Sour spots

- Heavily interrelated stuff
  - Like e.g. social network relations
- Relational and transactional stuff
  - Some data IS in fact relational in nature and should be transactional across tables/rows

(...and there's probably many other sour spots...)




A quick mention of some someones

# WHO'S USING IT

# Who uses it?

Boxed Ice

- Great blog on how Boxed Ice use MongoDB as the backend of their server monitoring application, Server Density: http://blog.boxedice.com/

# Who uses it?

theguardian

- Nice presentation about why The Guardian chose MongoDB over CouchDB and other NoSQL type dbs: http://www.slideshare.net/tackers/why-we-chose-mongodb-for-guardiancouk

# Who uses it?


- Nice blog post on how Attachments.me followed 10Gen's guidance and tuned their MongoDB performance: http://attachmentsme.tumblr.com/post/5168114317/tips-from-a-production-mongodb-deployment

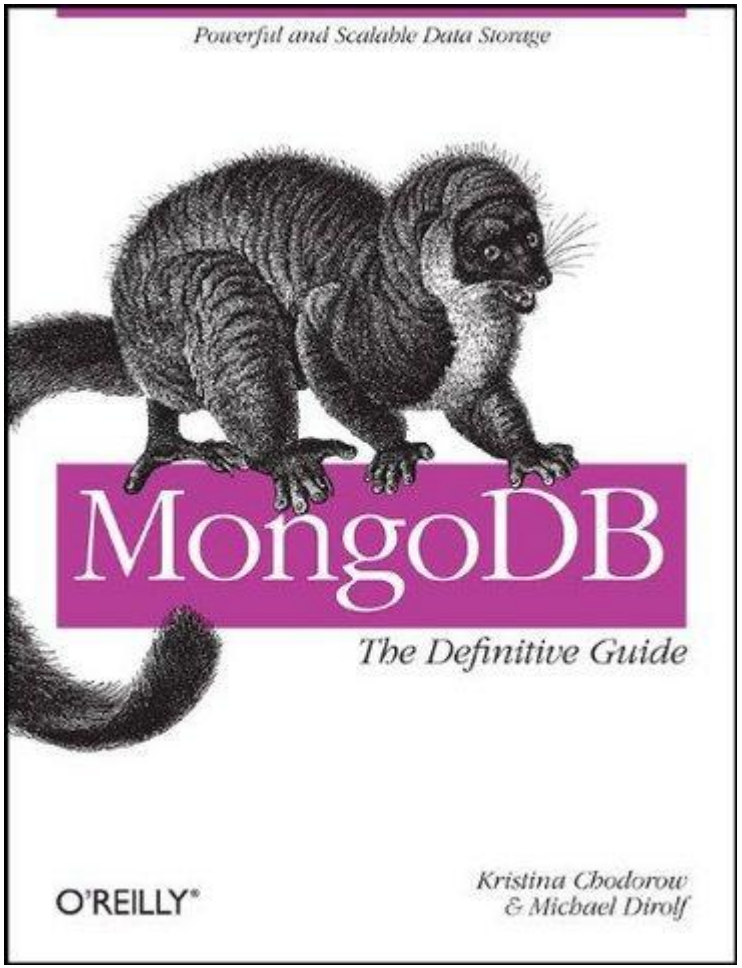
# Who uses it?

- Short interview on why MTV Networks's CMS is based on MongoDB: http://blog.mongodb.org/post/536000734/mongodb-powering-mtvs-web-properties

# Who uses it?


- Presentation by FourSquare on why they moved from Postgres to Mongo: https://docs.google.com/present/view?id=dhkkqm6q_13gm6jq5fv&pli=1

# Litterature




MongoDB – The Definitive Guide

by Kristina Chodorow and Mike Dirolf from 10Gen

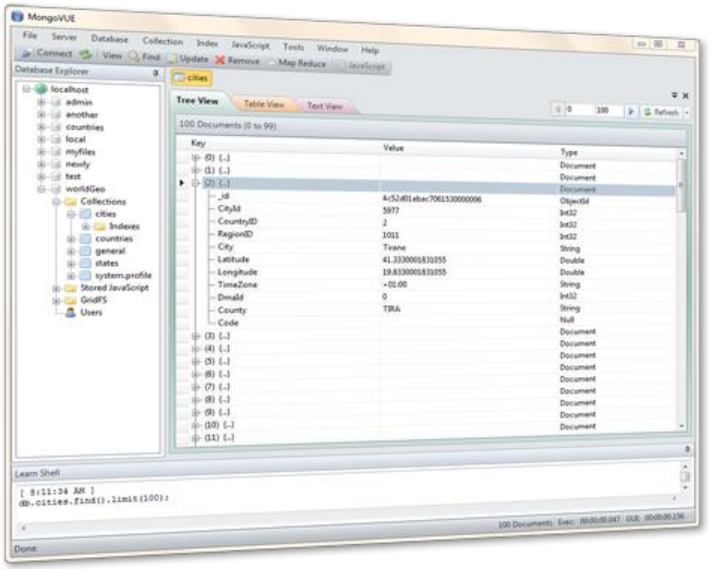
# Some cool resources

- http://www.mongodb.org
  - Binaries, official drivers, great documentation
- http://mongly.com
  - ”The Little MongoDB Book”
  - Interactive MongoDB tutorials in the browser!
- http://www.snailinaturtleneck.com/blog/
  - Kritstina Chodorow’s blog

# Some cool resources

- http://www.mongohq.com
  - Hosted MongoDB – free for instances with less than 16 MB of data!


- http://www.mongovue.com
  - Management GUI (written in C#... completely free until recently, now it's $35... still free evaluation though)

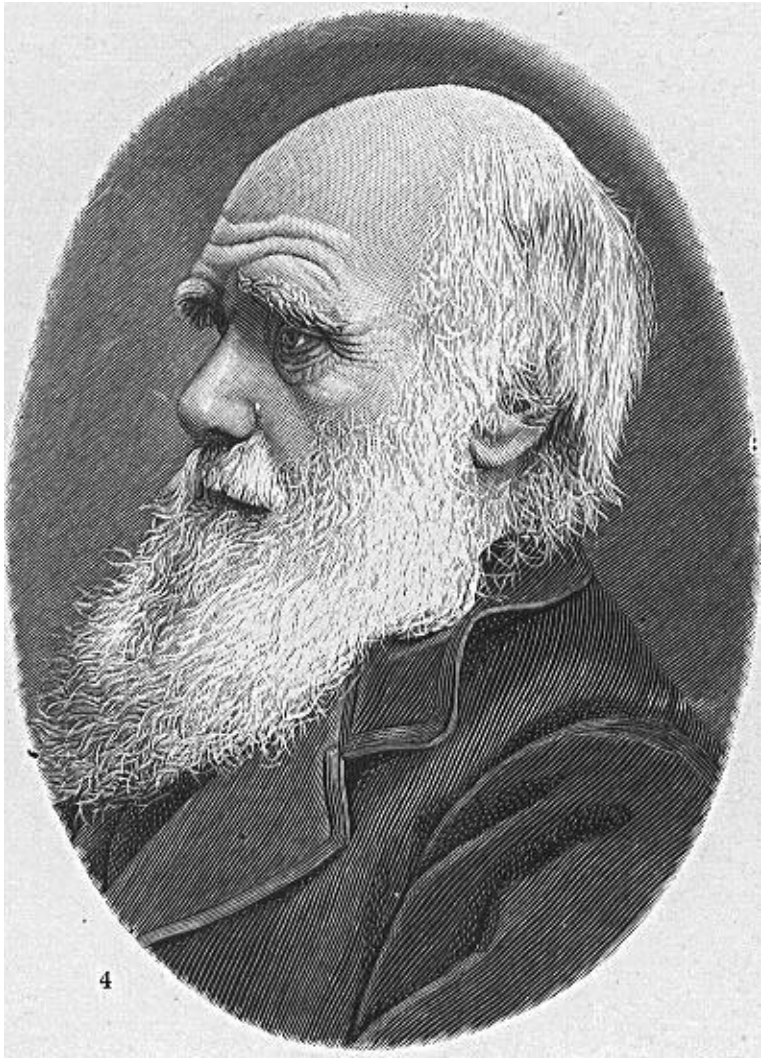


# What's in the queue?

# Remember this...

"It is not the strongest of the species that survives, nor the most intelligent that survives. It is the one that is the most adaptable to change."

- Charles Darwin



# Thank you for listening!

- Mogens Heller Grabe
- twitter://mookid8000
- mhg@trifork.com
- http://mookid.dk/oncode

- You can leave personal feedback here:

  http://speakerrate.com/talks/7415

# Some image credits

1. "I speak JavaScript": http://www.cafepress.com/+i_speak_javascript_mug,55296885
2. Bison: http://toyandgift.co.uk/acatalog/14349%20american%20bison.jpg
3. Snowflakes: http://italiano.istockphoto.com/stock-illustration-4195149-snowflakes-worn.php
4. Darwin: http://lmgtfy.com/?q=darwin
5. Enterprise: http://www.flickr.com/photos/sgodt/3279515658/
6. Lemon: http://www.flickr.com/photos/darwinbell/258156642/

I'm very grateful I could use your photos ☺ if you ever meet me, I'll buy you a beer (or another cold beverage of your choice…)