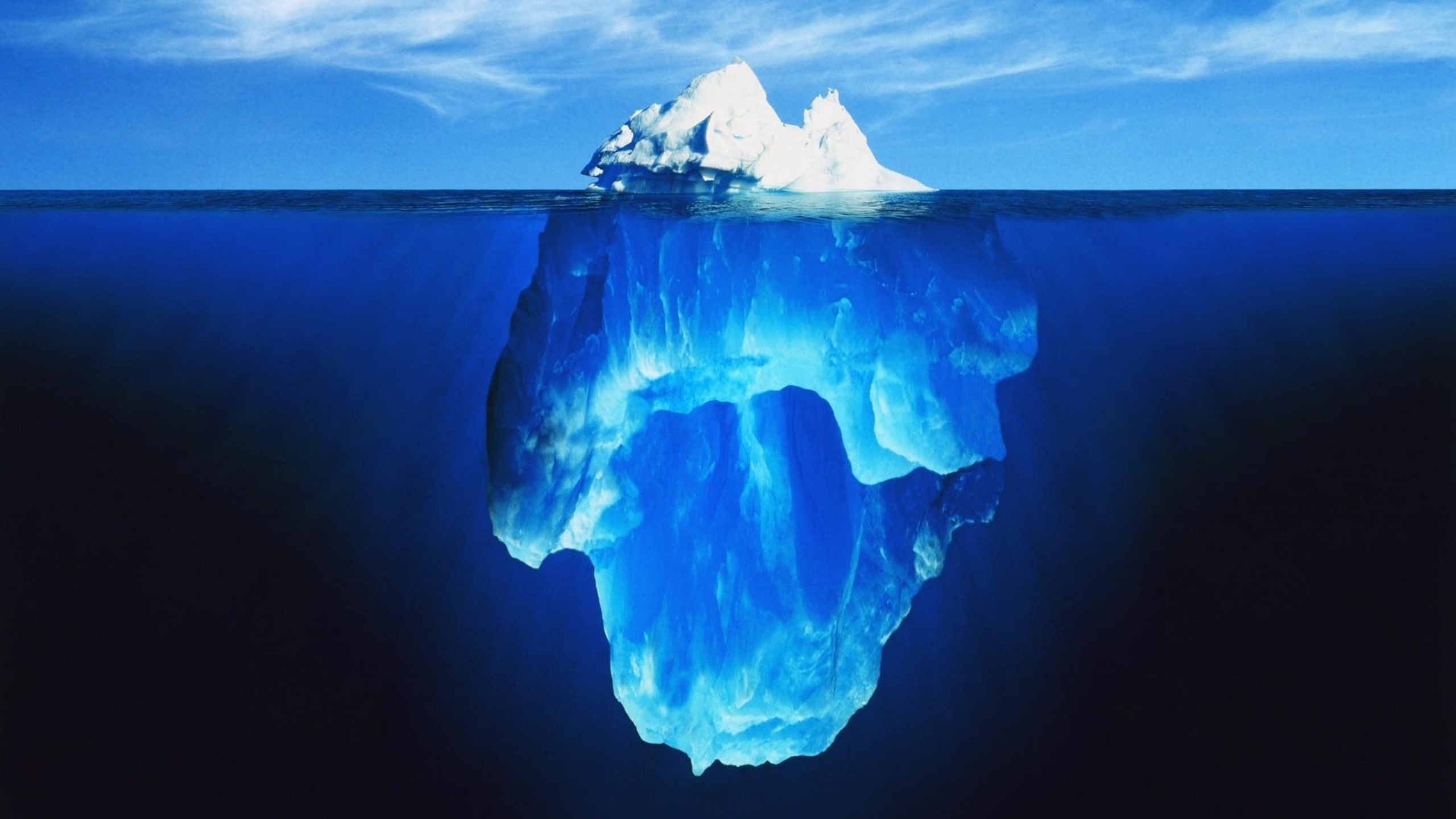
Messing With Mongoose

Michael A. Nowiszewski

3/10/2018

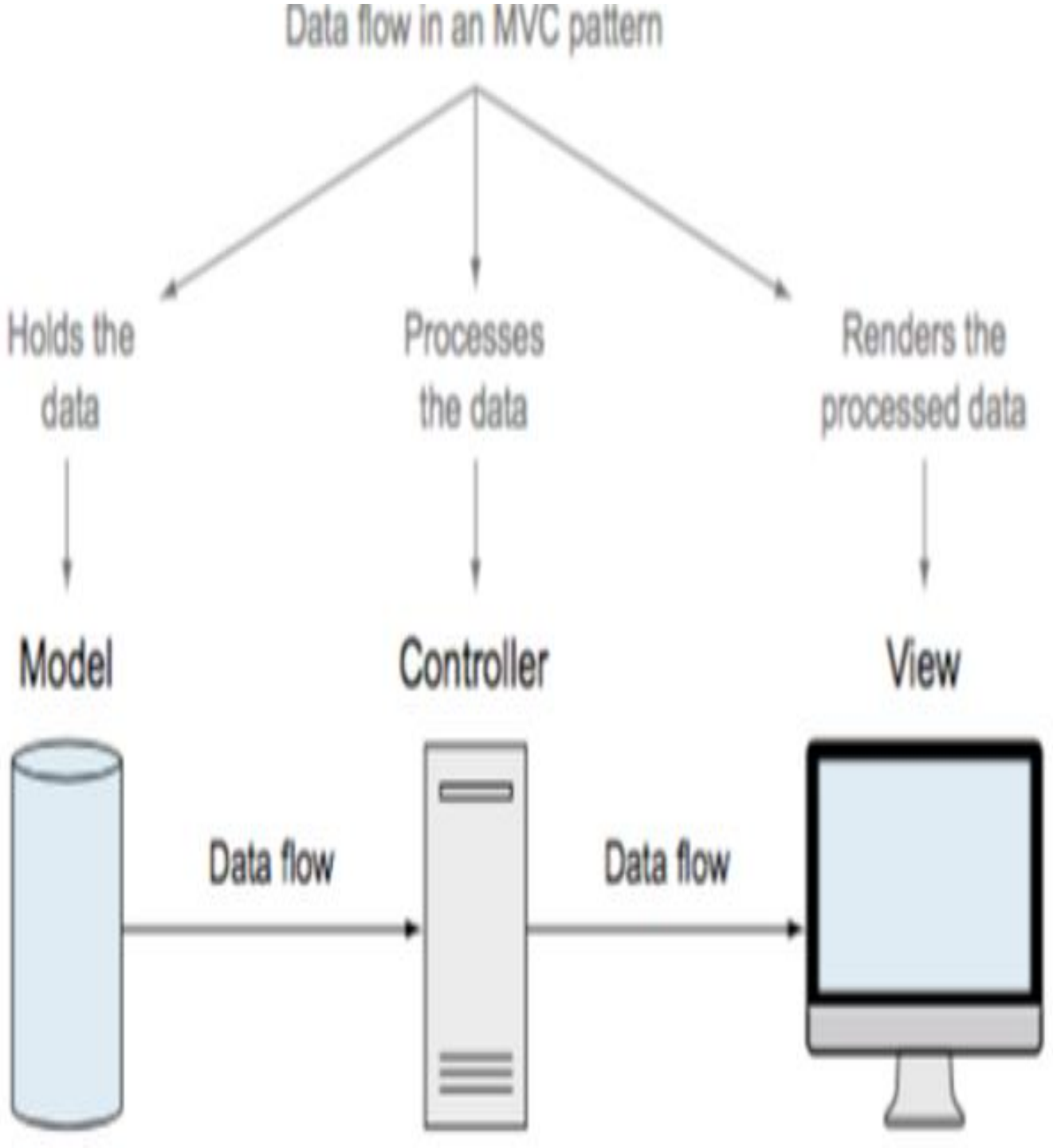


What this presentation will cover…

- What Mongoose is
- Why Mongoose is useful
- How to install Mongoose
- Where Mongoose Lives
- How Mongoose Interacts w/ App
- Setting up connections in M
- Monitoring connections in M
- Discuss dancing w/ MongoDB
- What is a Schema
- Overview of Data Types
- Close w/ some examples of use cases from most recent project!

What Is Mongoose?

- “elegant mongoDB object modeling for node.js”
- Part of our MVC Model! Goals?
- Simultaneous development, code reuse
- Mongoose is an ODM. What’s that?
- A MongoDB object modeling tool
- Designed for use in an asynchronous environment



Why Use It, Anyway?

Mongo DB Native Driver

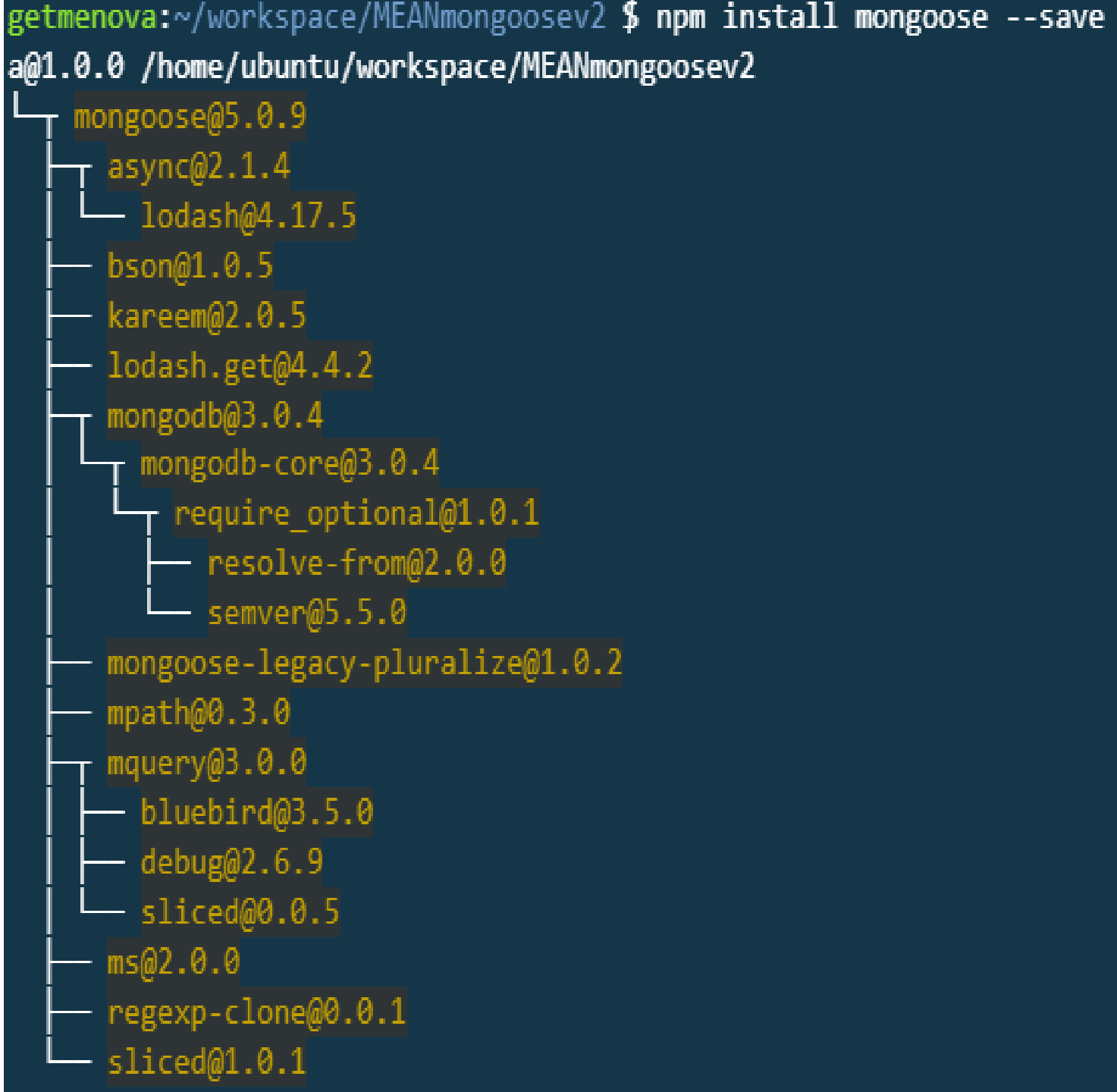
- Not easy to work with without customization, add-ons
- No way to define and maintain data structures out of the box – requires Mongoose and more
- Less Convenient For Users

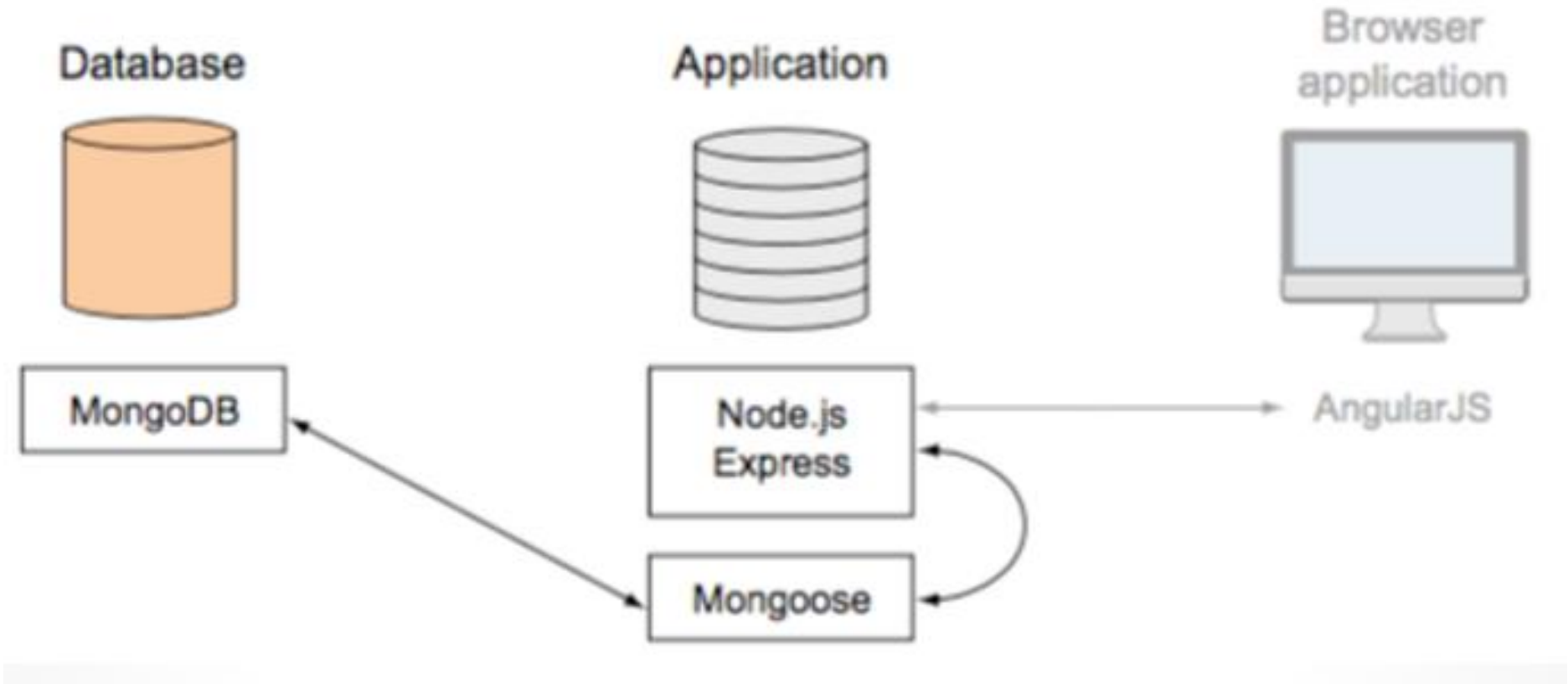
Mongoose

- Ease of Use
- Allows us to Manage Data In-App
- Solutions for a Range of Datatypes
- Uses core Functionality of MongoDB Efficiently
- Allows to define/maintain models
- Allows us to decide how we'd like our new data models/structures to interact with our DB of choice!

Installing Mongoose:

$ npm install mongoose --save





Where Does Mongoose Live?

Mongoose Connections : Interaction with App

- Mongoose will open a connection pool
- Remember: connection pool is shared between all requests
- Also consider the following:
- Opening the connection when you spin up app
- Keep connection open as long as your app continues to run

Mongoose: Setup & Monitoring Connections

Simple Steps to Setup App Connection

- Create db.js
- Use this file in app.js or route it
- Remember ‘require’ it!

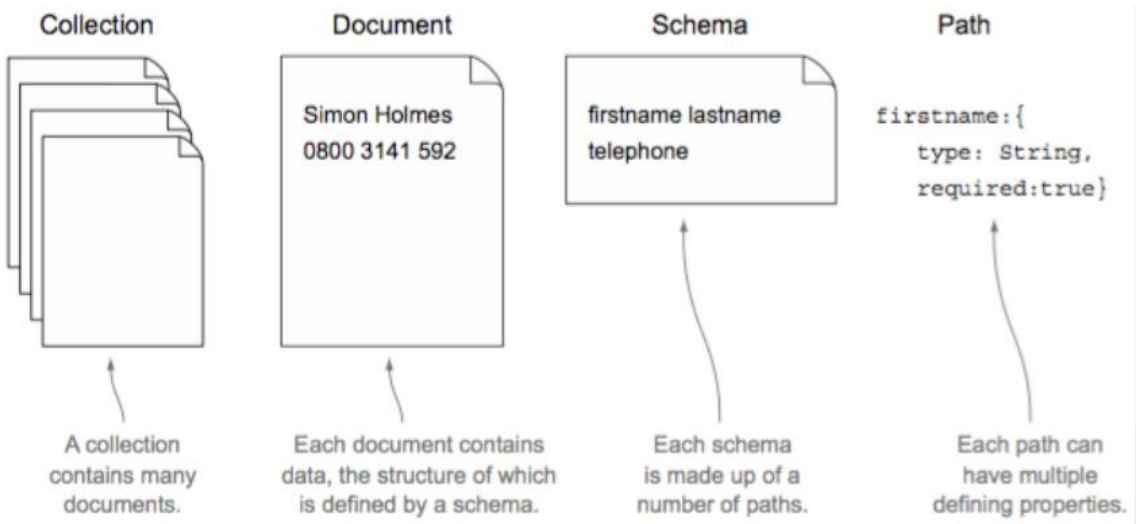
```
var mongoose = require('mongoose');
```

- *appears in db.js in hotel due to routing

Monitoring Connections In-App

- What info are we interested in?
- On connect
- When we encounter an error
- When we disconnect
- In meanhotel, we did it this way:

```
mongoose.connection.on('connected', function() {
  console.log('Mongoose connected to ' + dburl);
});
mongoose.connection.on('error', function(err) {
  console.log('Mongoose connection error: ' + err);
});
mongoose.connection.on('disconnected', function() {
  console.log('Mongoose disconnected');
});
```


Collection
Document
Schema
Path
Simon Holmes
0800 3141 592
firstname lastname
telephone
firstname:{
type: String,
required:true}
A collection contains many documents.
Each document contains data, the structure of which is defined by a schema.
Each schema is made up of a number of paths.
Each path can have multiple defining properties.

Defining your schema

Everything in Mongoose starts with a Schema. Each schema maps to a MongoDB collection and defines the shape of the documents within that collection.

```
var mongoose = require('mongoose');
var Schema = mongoose.Schema;

var blogSchema = new Schema({
  title:  String,
  author: String,
  body:   String,
  comments: [{ body: String, date: Date }],
  date: { type: Date, default: Date.now },
  hidden: Boolean,
  meta: {
    votes: Number,
    favs:  Number
  }
});
```

What's in a Schema?

SchemaTypes Supported: Common Examples

- String
- Number
- Date
- Buffer
- Boolean
- Mixed
- ObjectId
- Array

```
var schema = new Schema({
  name:    String,
  binary:  Buffer,
  living:  Boolean,
  updated: { type: Date, default: Date.now },
  age:     { type: Number, min: 18, max: 65 },
  mixed:   Schema.Types.Mixed,
  _someId: Schema.Types.ObjectId,
  array:      [],
});
```

```
var Thing = mongoose.model('Thing', schema);

var m = new Thing;
m.name = 'Statue of Liberty';
m.age = 125;
m.updated = new Date;
m.binary = new Buffer(0);
m.living = false;
m.mixed = { any: { thing: 'i want' } };
m.markModified('mixed');
m._someId = new mongoose.Types.ObjectId;
m.array.push(1);
```

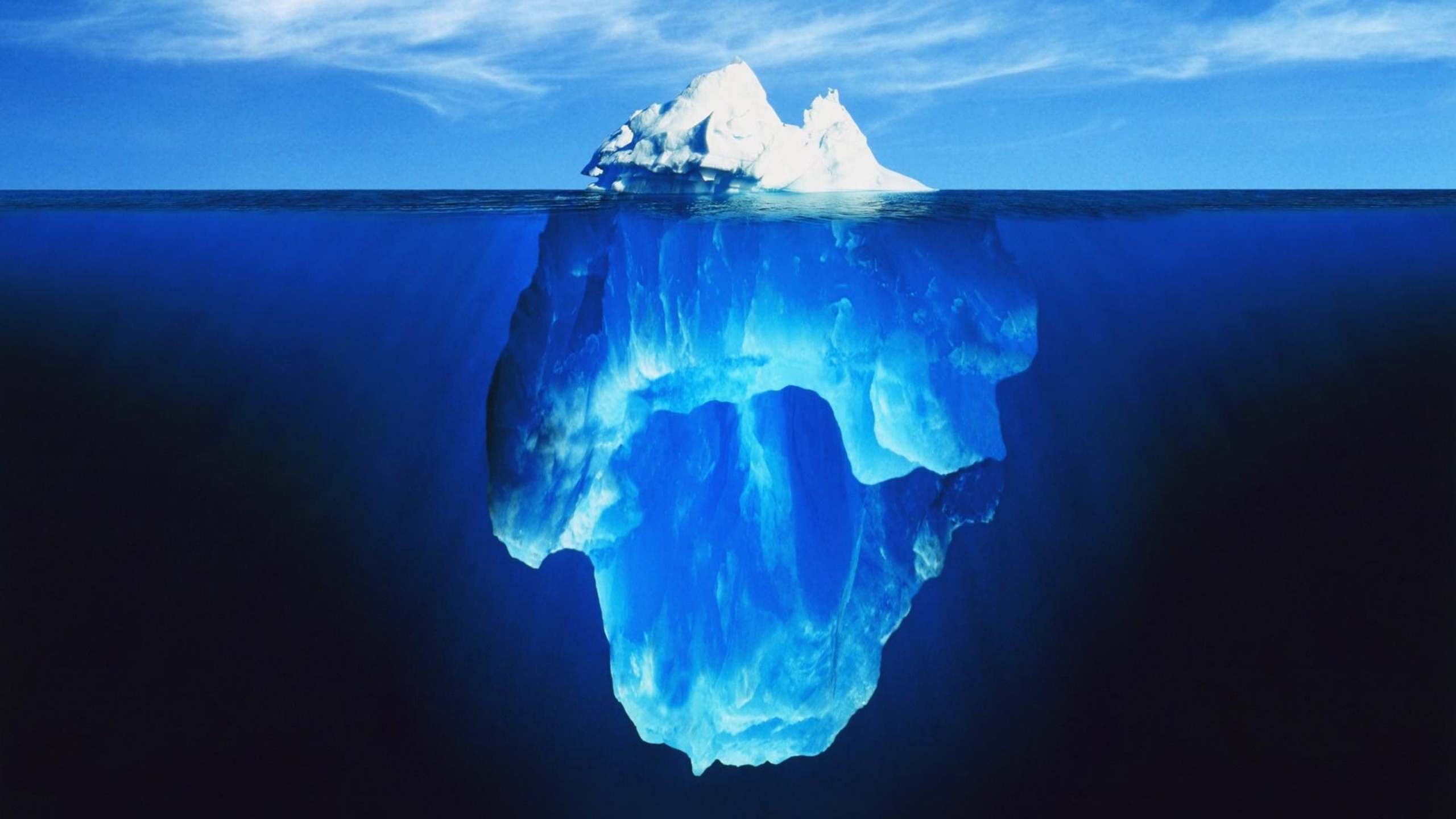
Examples from recent project @/meanhotel/data/hotel.models.js

Simple 'Room' Schema

```
var roomSchema = new mongoose.Schema({
  type : String,
  number : Number,
  description : String,
  photos : [String],
  price : Number
});
```

Adding Parameters in 'Review' Schema

```
var reviewSchema = new mongoose.Schema({
  name : {
    type : String,
    required : true
  },
  rating : {
    type : Number,
    required : true,
    min : 0,
    max : 5
  },
  review : {
    type : String,
    required : true
  },
  createdOn : {
    type : Date,
    "default" : Date.now
  }
});
```

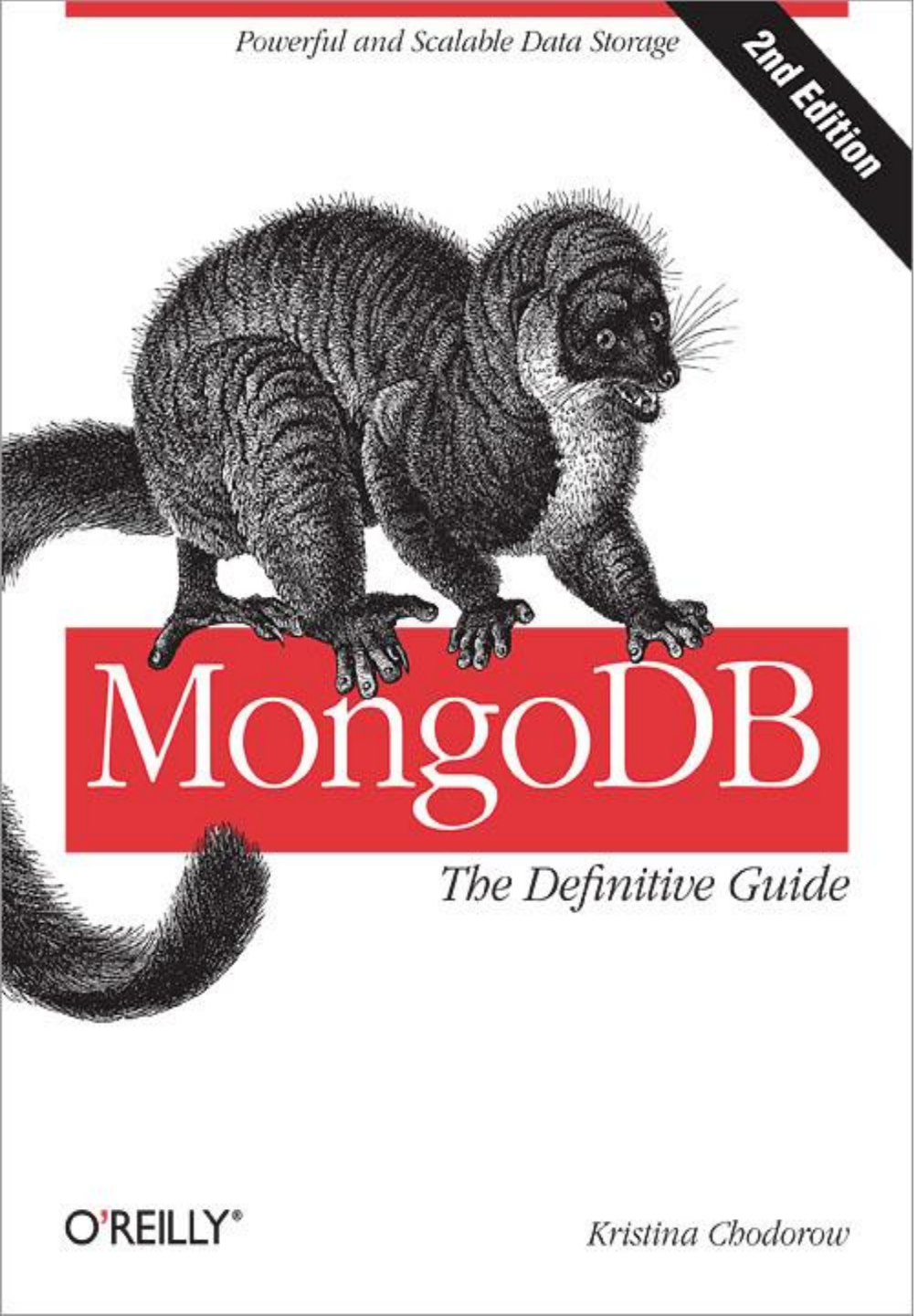


What this presentation covered…

- What Mongoose is
- What an ODM is
- Why Mongoose is useful
- How to install Mongoose
- Where Mongoose Lives
- How Mongoose Interacts w/ App
- Setting up connections in M
- Monitoring connections in M
- Discuss dancing w/ MongoDB
- What is a Schema
- Overview of Data Types
- Close w/ some examples of use cases from most recent project!

Additional Resources

- http://mongoosejs.com/
- https://github.com/Automattic/mongoose
- https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/mongoose
- https://code.tutsplus.com/articles/an-introduction-to-mongoose-for-mongodb-and-nodejs--cms-29527
- https://www.mongodb.com/blog/post/introducing-version-40-mongoose-nodejs-odm
- http://mongodb-tools.com/ to compare other ODM, ORM tools.



The animal on the cover of *MongoDB: The Definitive Guide, Second Edition* is a mongoose lemur, a member of a highly diverse group of primates endemic to Madagascar. Ancestral lemurs are believed to have inadvertently traveled to Madagascar from Africa (a trip of at least 350 miles) by raft some 65 million years ago. Freed from competition with other African species (such as monkeys and squirrels), lemurs adapted to fill a wide variety of ecological niches, branching into the almost 100 species known today. These animals' otherworldly calls, nocturnal activity, and glowing eyes earned them their name, which comes from the lemures (specters) of Roman myth. Malagasy culture also associates lemurs with the supernatural, variously considering them the souls of ancestors, the source of taboo, or spirits bent on revenge. Some villages identify a particular species of lemur as the ancestor of their group.

Thank you kindly for your attention and participation.
Please forward any questions to micski@gmail.com.
Have a nice day!
MN

FIN