

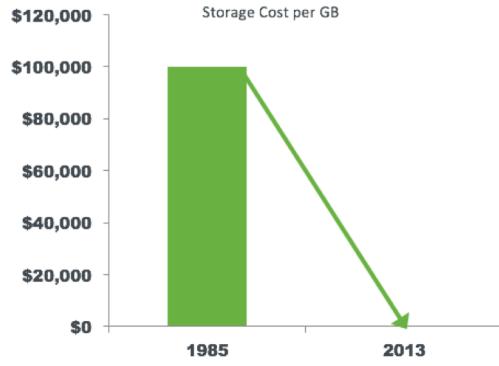
CONFERENCE: Nov 7-9, 2016
WORKSHOPS: Nov 10-11, 2016

The Rise of the Multi-Model Database

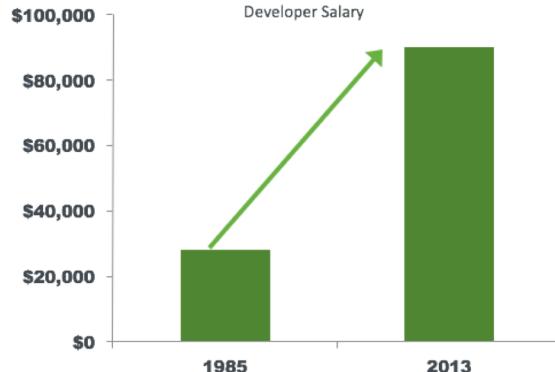
Sig Narváez
Sr. Solution Architect, MongoDB
sig@mongodb.com
@SigNarvaez

Brief History

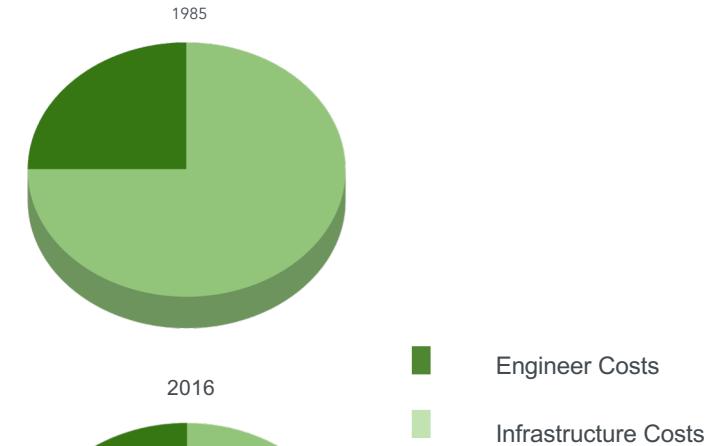
Developer Costs on the Rise



Developer Salary



Optimize for Engineering Productivity





MUGifier

MUGifier Required Capabilities

The screenshot shows the Amazon search results for "coffee mugs". At the top, there's a search bar with the query "coffee mugs". Below it, a sidebar on the left provides filtering options like "Prime", "AmazonFresh", and "fresh". The main content area displays several product cards. One card for "Francois et Mimi Set of 6 Large-sized 16 Ounce Ceramic Coffee Mugs (Solid Colorful)" is highlighted. Another card for "Primrose Colorful Mugs by Mader Kitchen - Set of 6 Ceramic Coffee Mugs Small Mouth 14oz - 100%..." is also visible. The page includes sections for "Best Selling" and "Top Rated" products.

Images
Spec sheets
Blobs

Attribute dimensions
Ranges (price, years, ratings)
Faceted Search / Navigation

Orders
Shipping
RMA's
Relational

Keywords
Multi-language
Text Search

User sessions
Cache
Key/Value

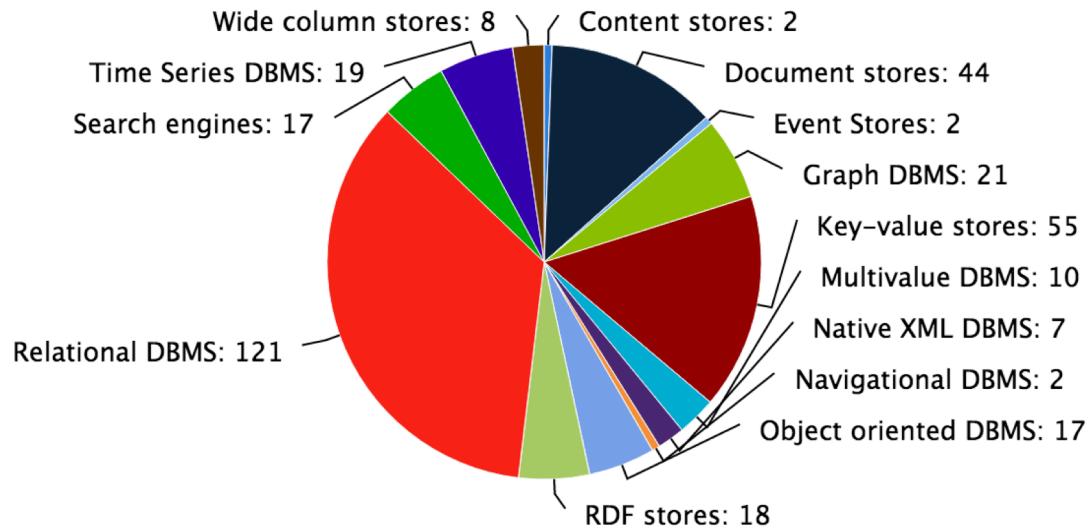
Product Catalog
Content Management
Flexible Schema

Product Categories
Recommendations
Related Products
Graphs & Hierarchies

The Database Landscape

DBMS popularity broken down by database model

Number of systems per category, November 2016



Too many tools

Tools to
manage tools!



Less Tools – More Functions



Slot
Head



Phillips
Head



Pozidriv



Torx



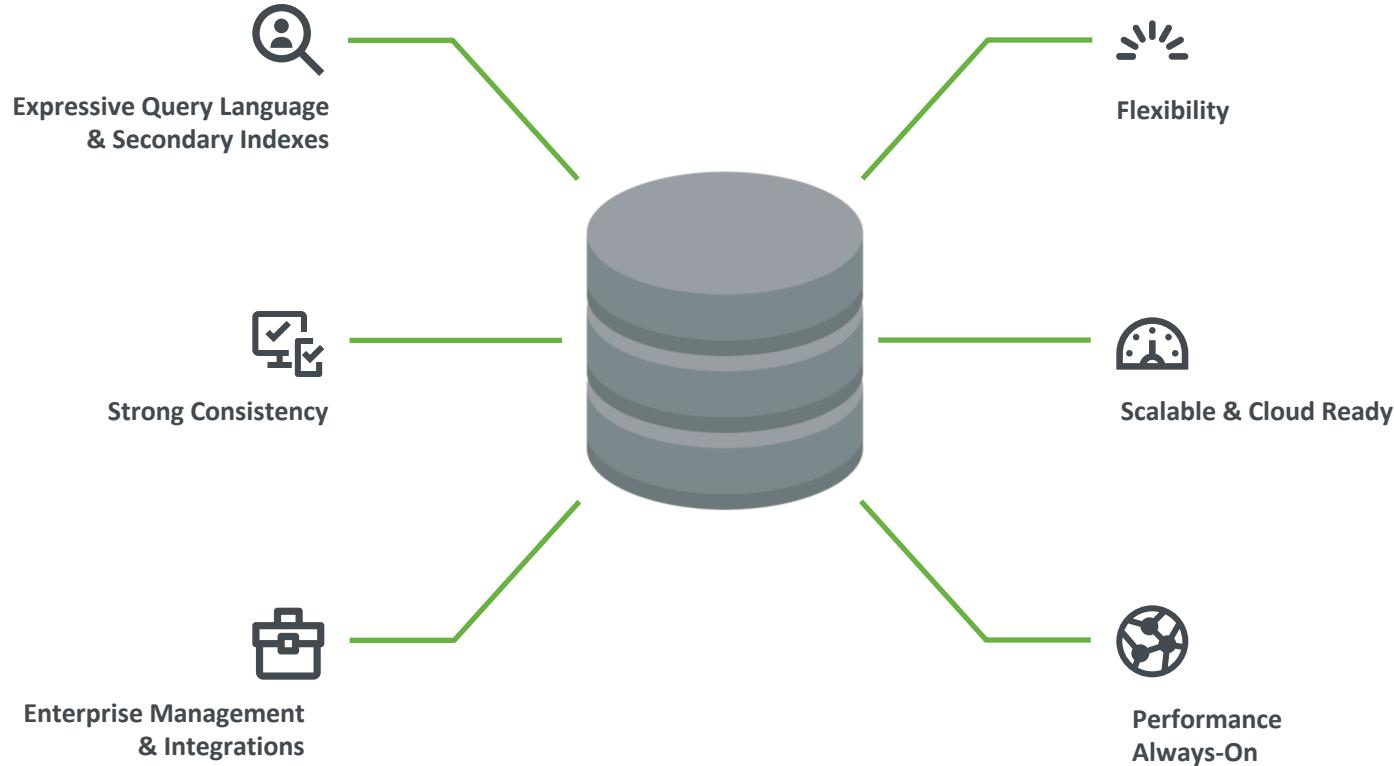
Secure
Torx



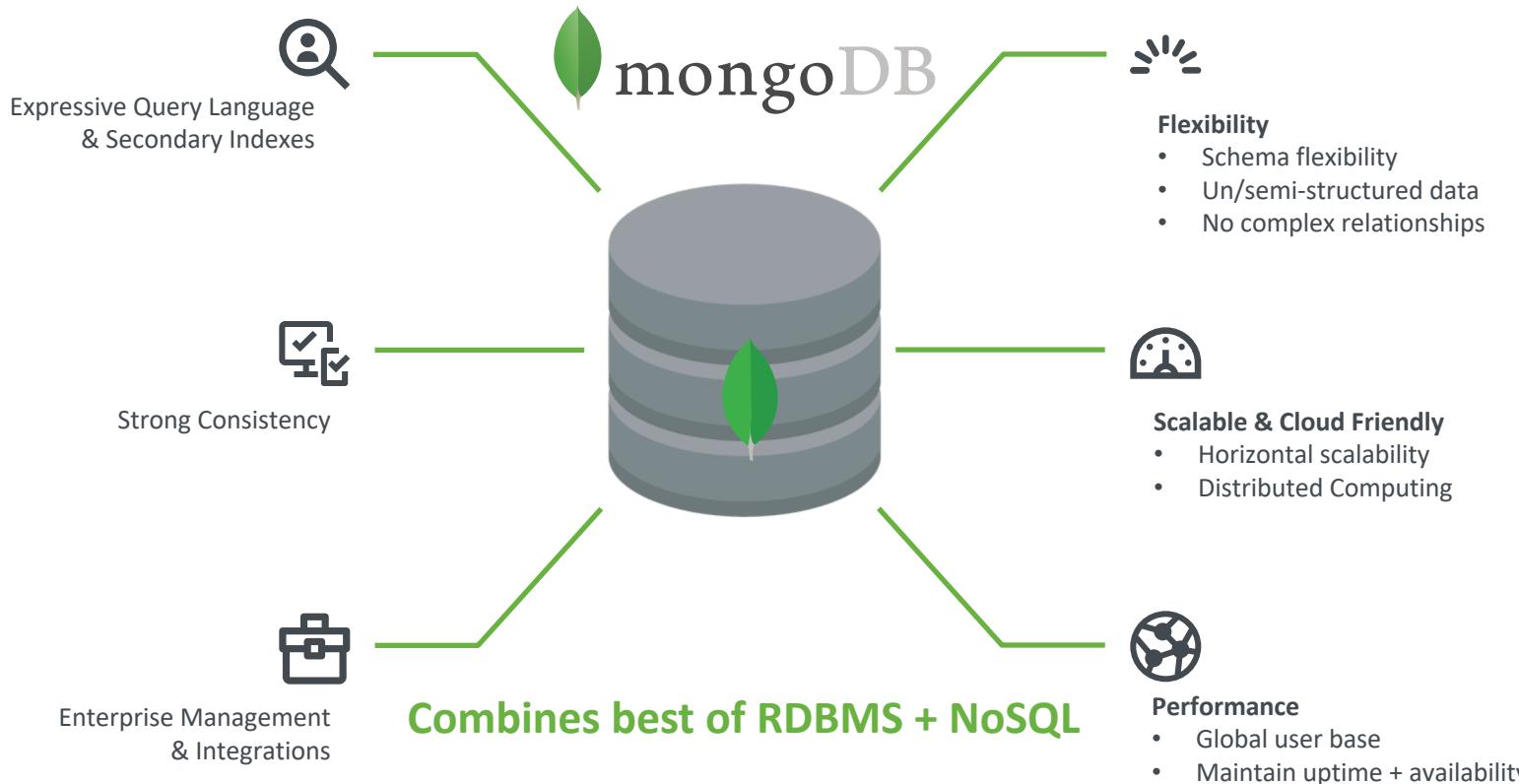
Hex
Head



Essential Data Needs ... Today



MongoDB Nexus Architecture



MongoDB 3.4 - Multi-Model Database

Document

Rich JSON Data Structures
Flexible Schema
Global Scale

Relational

Left-Outer Join
Views
Schema Validation

Key/Value

Horizontal Scale
In-Memory

Graph

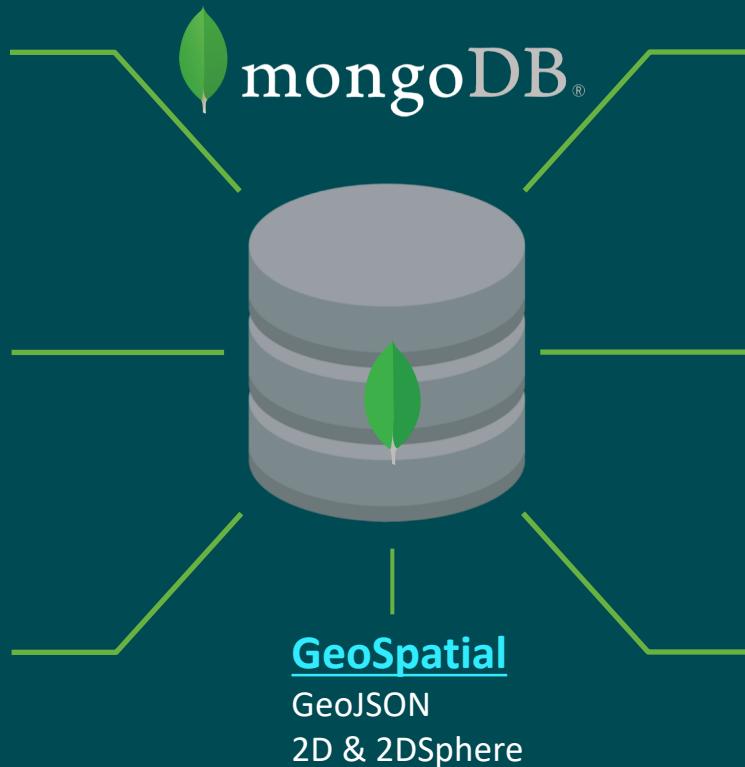
Graph & Hierarchical
Recursive Lookups

Search

Text Search
Multiple Languages
Faceted Search

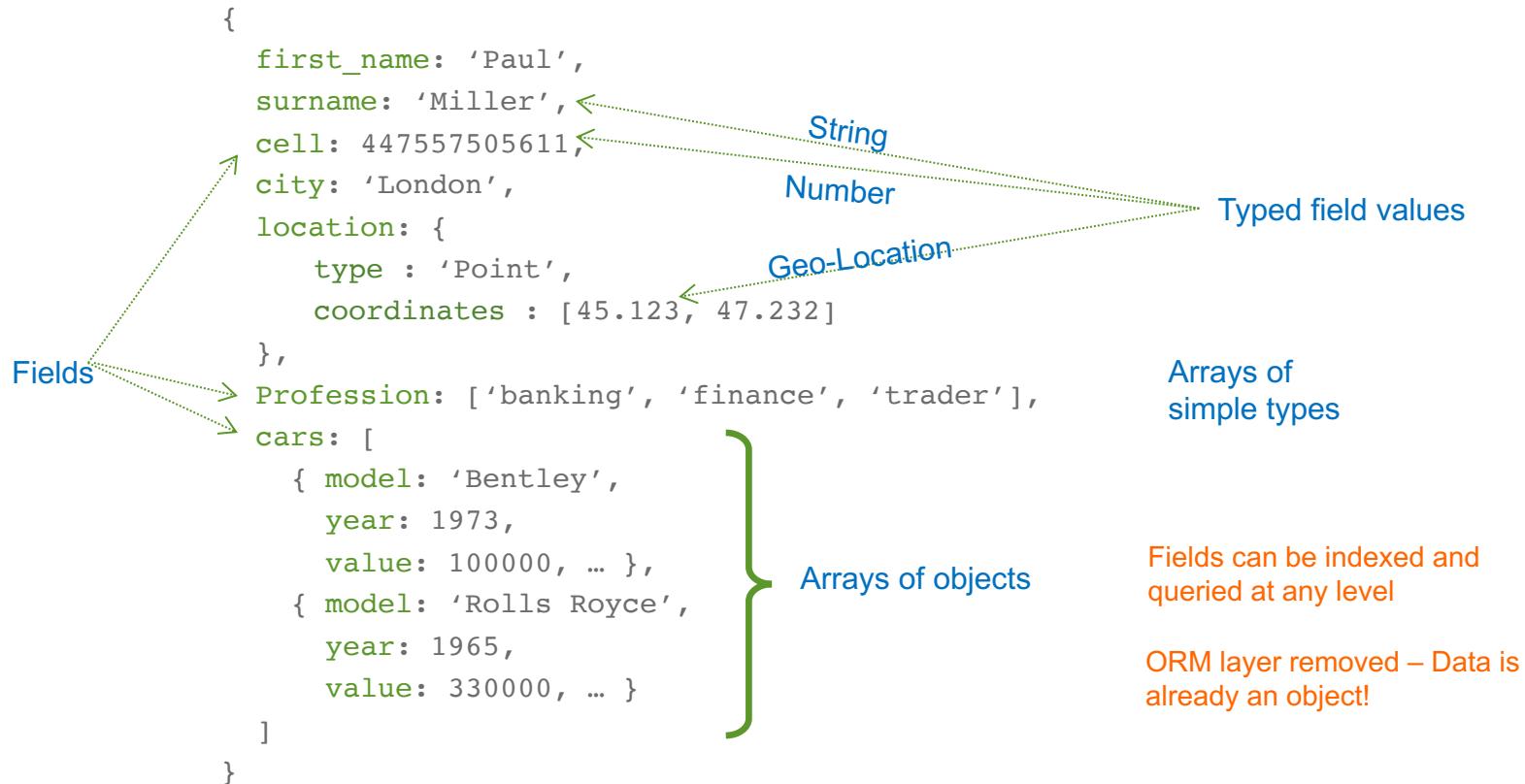
Binaries

Files & Metadata
Encrypted



The JSON Document Model as the superset

Documents are Rich Data Structures



Documents are Flexible

Documents in the same Product catalog collection in MongoDB

Paint products

```
{  
  sku: 'PAINTZXC123',  
  product_name: 'Acme Paint',  
  colors: ['Red', 'Green'],  
  size_oz: [8, 32],  
  finish: ['satin', 'eggshell']  
}
```

T-Shirt products

```
{  
  sku: 'TSHRTASD43546',  
  product_name: 'T-shirt',  
  size: ['S', 'M', 'L', 'XL'],  
  colors: ['Heather Gray' ... ],  
  material: '100% cotton',  
  wash: 'cold',  
  dry: 'tumble dry low'  
}
```

Cycling products

```
{  
  sku: 'CYCLBVCX6543',  
  product_name: 'Mountain Bike',  
  brake_style: 'mechanical disc',  
  color: 'grey',  
  frame_material: 'aluminum',  
  no_speeds: 21,  
  package_height: '7.5x32.9x55',  
  weight_lbs: 44.05,  
  suspension_type: 'dual',  
  wheel_size_in: 26  
}
```

Relational

orders

```
{  
  orderID: "ORD123",  
  total: "3450.45",  
  ...  
  customer: {  
    name: "Sig Narvaez",  
    customerID: "CUST456",  
    ...  
  },  
  items: [{ sku: "ITEM123",  
            name: "MUG 123",  
            qty: 3 },  
          { sku: "ITEM456",  
            name: "MUG 456",  
            qty: 5 }],  
  payment: {  
    method: "visa",  
    txID: "123123XXXDD11DD"  
  }  
}
```

customers

```
{  
  customerID: "CUST456",  
  name: "Sig Narvaez",  
  ...  
  addresses: [  
    { type: "home", address: "123 MyStreet, MyCity, CA, 123456"},  
    { type: "mail", address: "456 MyMail, MyCity, CA, 123456"},  
  ],  
  ...  
}
```

items

```
{  
  sku: "ITEM123",  
  name: "MUG 123",  
  unit_cost: 15.95,  
  images: [  
    { "front-view": "..."},  
    { "left-view": "..." }  
  ],  
  ...  
}
```

Relational – joins & views

“left-outer join” - \$lookup

```
db.orders.aggregate([
  { "$unwind" : "$items",
  { "$lookup" : {
      "from" : "items",
      "localField" : "items.sku",
      "foreignField": "sku",
      "as" : "items_full" }},
  { "$unwind": "items_full" },
  { "$group" : { "_id" : "$orderID",
      "total" : { $sum: {
          $multiply: [ "$items.qty",
                      "$items_full.unit_cost"]}}}}
]);

```

```
{ "_id" : "ORD123", "total" : 147.6 }
```

views

```
db.createView("vwRecalcOrderTotals", "orders",
[ { "$unwind" : "$items",
  { "$lookup" : {
      "from" : "items",
      "localField" : "items.sku",
      "foreignField": "sku",
      "as" : "items_full" }},
  { "$unwind": "items_full" },
  { "$group" : { "_id" : "$orderID",
      "total" : { $sum: {
          $multiply: [ "$items.qty",
                      "$items_full.unit_cost"]}}}})
]);

```

Query against the view

```
MongoDB > db.vwRecalcOrderTotals.find({"_id":"ORD123"});
```

```
{ "_id" : "ORD123", "total" : 147.6 }
```

Relational – schema validation

```
db.runCommand({collMod: "orders",
  validator : {
    "$and" : [
      {"orderID" : {"$type" : "string"}},
      {"customer.customerID" : {"$type": "string"}},
      {"items.0" : {"$exists" : true}}
    ],
    "validationLevel": "strict",
    "validationAction": "error"
  }
});
```

Rule

"An order must have at least one item"

Not possible with SQL DDL

Level: off | moderate | strict

Action: error | warn

```
db.orders.insert({
  "orderID": "order1",
  "customer": {"customerID": "customer1"},
  "items": []
});
WriteResult({
  "nInserted" : 0,
  "writeError" : {
    "code" : 121,
    "errmsg" : "Document failed validation"
  }
})
```

Schema on-write

Faceted Search / Navigation





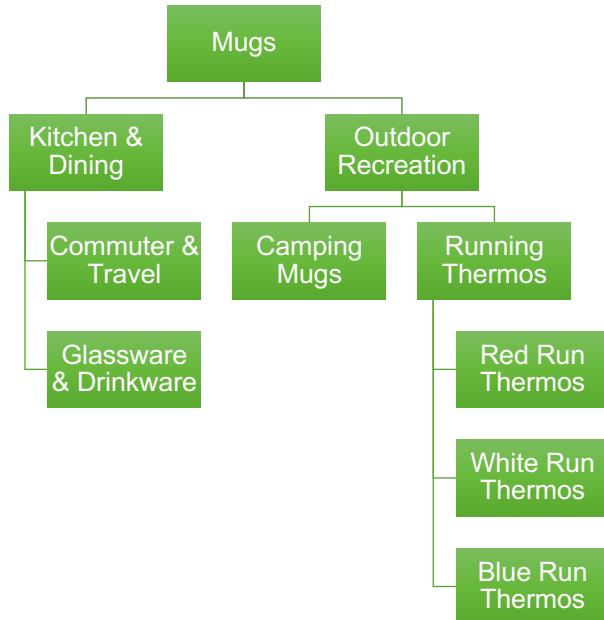
The screenshot shows the Amazon search interface for "coffee mugs". On the left, a sidebar displays various facets: Prime delivery, AmazonFresh, Any Category, Home & Kitchen, Kitchen & Dining (with sub-options like Glassware & Drinkware, Novelty Coffee Mugs, Dining & Entertaining, Cups, Mugs & Saucers, Coffee Cups & Mugs, Mug Sets, Commuter & Travel Mugs), and a "See more" link. Below this is a "Refine by" section for Subscribe & Save. The main content area shows search results for "coffee mugs" with sections for "Best Selling" (featuring a stack of colorful mugs) and "Top Rated" (featuring a set of mugs from Madero Kitchen). Each product card includes an image, title, price, delivery information, and a star rating.

Faceted Search / Navigation

```
db.itemsFacets.aggregate( [  
    {  
        $facet: {  
            "categorizedByTags": [ { $unwind: "$tags" }, { $sortByCount: "$tags" } ],  
            "categorizedByYears": [  
                { $match: { year: { $exists: 1 } } },  
                { $bucket: { groupBy: "$year", boundaries: [ 1970, 1980, 1990 ] } }  
            ],  
            "categorizedByYears(Auto)": [ { $bucketAuto: { groupBy: "$year", buckets: 4 } } ]  
        }  
    }  
];
```

```
{  
    "categorizedByTags": [ { "_id": "Geek", "count": 6 },  
                          { "_id": "Characters", "count": 5 },  
                          { "_id": "Sci-Fi", "count": 4 },  
                          { "_id": "TV", "count": 4 },  
                          { "_id": "Transformers", "count": 2 },  
                          { "_id": "Star Trek", "count": 2 },  
                          { "_id": "Movies", "count": 2 },  
                          { "_id": "Cartoons", "count": 2 },  
                          { "_id": "Star Wars", "count": 2 } ],  
  
    "categorizedByYears": [ { "_id": 1970, "count": 2 },  
                           { "_id": 1980, "count": 2 } ],  
  
    "categorizedByYears(Auto)": [ { "_id": { "min": null, "max": 1977 }, "count": 2 },  
                                { "_id": { "min": 1977, "max": 1987 }, "count": 2 },  
                                { "_id": { "min": 1987, "max": 1987 }, "count": 2 } ]  
}
```

Graph or Recursive Lookup



categoryGraph

```
[{"_id": 1, "name": "Mugs"}, {"_id": 2, "name": "Kitchen & Dining", "parentId": 1}, {"_id": 3, "name": "Commuter & Travel", "parentId": 2}, {"_id": 4, "name": "Glassware & Drinkware", "parentId": 2}, {"_id": 5, "name": "Outdoor Recreation", "parentId": 1}, {"_id": 6, "name": "Camping Mugs", "parentId": 5}, {"_id": 7, "name": "Running Thermos", "parentId": 5}, {"_id": 8, "name": "Red Run Thermos", "parentId": 7}, {"_id": 9, "name": "White Run Thermos", "parentId": 7}, {"_id": 10, "name": "Blue Run Thermos", "parentId": 7}]
```

Graph or Recursive Lookup

Get all children two levels deep

```
db.categoryGraph.aggregate([
  { $match: { _id: 1 }},
  { $graphLookup: {
      from: "categoryGraph",
      startWith: "$_id",
      connectFromField: "_id",
      connectToField: "parentId",
      depthField: "depth",
      maxDepth: 1,
      as: "Children" }}]
);
```

```
{
  "_id": 1,
  "name": "Mugs",
  "Children": [
    { "_id": 7, "name": "Running Thermos", "parentId": 5, "depth": 1 },
    { "_id": 6, "name": "Camping Mugs", "parentId": 5, "depth": 1 },
    { "_id": 4, "name": "Glassware & Drinkware", "parentId": 2, "depth": 1 },
    { "_id": 3, "name": "Commuter & Travel", "parentId": 2, "depth": 1 },
    { "_id": 5, "name": "Outdoor Recreation", "parentId": 1, "depth": 0 },
    { "_id": 2, "name": "Kitchen & Dining", "parentId": 1, "depth": 0 }
  ]
}
```

Graph or Recursive Lookup

Get all children two levels deep

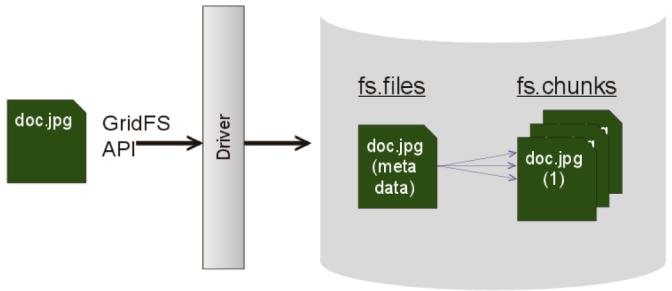
```
db.createView("vwCategoryDepth1", "categoryGraph", [
  { $graphLookup: {
      from: "categoryGraph",
      startWith: "$_id",
      connectFromField: "_id",
      connectToField: "parentId",
      maxDepth: 0,
      as: "Children" }}]
);

db.categoryGraph.aggregate([
  { $match: { _id: 1 }},
  { $graphLookup: {
      from: "vwCategoryDepth1",
      startWith: "$_id",
      connectFromField: "_id",
      connectToField: "parentId",
      maxDepth: 0,
      as: "Children" }}]
);
```



```
{
  "_id" : 1,
  "name" : "Mugs",
  "Children" : [
    {
      "_id" : 5,
      "name" : "Outdoor Recreation",
      "parentId" : 1,
      "Children" : [
        { "_id" : 7, "name" : "Running Thermos", "parentId" : 5 },
        { "_id" : 6, "name" : "Camping Mugs", "parentId" : 5 }
      ]
    },
    {
      "_id" : 2,
      "name" : "Kitchen & Dining",
      "parentId" : 1,
      "Children" : [
        { "_id" : 4, "name" : "Glassware & Drinkware", "parentId" : 2 },
        { "_id" : 3, "name" : "Commuter & Travel", "parentId" : 2 }
      ]
    }
  ]
}
```

Blobs



Benefits

- Replicated
- Metadata / search

Optional

- Encrypted At-Rest

.files

```
{
    "_id": ObjectId("546fb304f9781507293f9891"),
    "length": NumberLong("834284"),
    "content-type": "video/quicktime",
    "uploadDate": ISODate("2014-11-21T21:47:48Z"),
    "md5": "8eb3e2a9de0ad7c17dcef99110e2463d",
    "chunkSize": NumberLong("261120"),
    "filename": "IMG_0425.MOV",
    "metadata": {
        "ExternalID": 1,
        "Keywords": "summer pools cousins yosemite 2016 july road trip"
    }
}
```

.chunks

```
{
    "_id": ObjectId("546fb304f9781507293f98a6"),
    "files_id": ObjectId("546fb304f9781507293f9891"),
    "data": BinData(0, "+QWU+Pwf1u03d....."),
    "n": NumberLong("0")
}
{
    "_id": ObjectId("546fb304f9781507293f9898"),
    "files_id": ObjectId("546fb304f9781507293f9891"),
    "data": BinData(0, "EgKwOoOAOYEWFoi..."),
    "n": NumberLong("1")
}
```

Key / Value

Benefits

- Replicated
- Secondary reads for scale
- Sharding for horizontal scale & local writes
- TTL Indexes
- Capped Collections

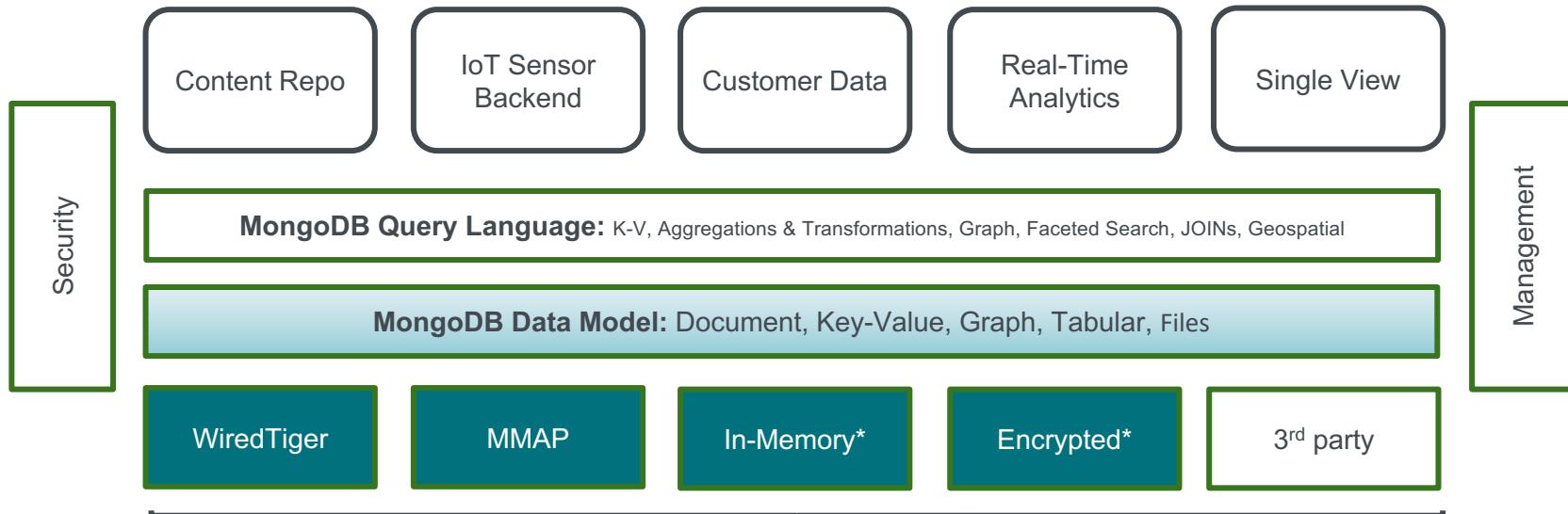
Optional

- In-Memory

Key	Value
k1	v1
k2	v2
k3	v3
k4	v4
k5	v5
k6	v6
k7	v7
k8	v8
k9	v9
k10	v10
k11	v11
k12	v12
k13	v13
k14	v14
k15	v15
k16	v16
k17	v17
k18	v18
k19	v19
k20	v20
k21	v21

```
{ _id: "k1", v: "v1" },  
{ _id: "k2", v: "v2" },  
{ _id: "k3", v: "v3" },  
{ _id: "k4", v: "v4" },  
{ _id: "k5", v: "v5" }, ...
```

Pluggable Storage Engines



Available in MongoDB 3.4

*Enterprise Advanced Only

Private to Public Cloud



 mongoDB
Ops Manager

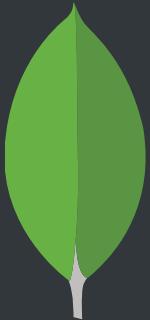
```
{  
  engine: "Enterprise",  
  data: "On-Prem | Cloud",  
  ops: "On-Prem | Cloud"  
}
```

 mongoDB
Cloud Manager

```
{  
  engine: "Enterprise | Community",  
  data: "On-Prem | Cloud",  
  ops: "Cloud"  
}
```

 Atlas

```
{  
  engine: "Community",  
  data: "Cloud",  
  ops: "Cloud"  
}
```



Atlas

A weight off your shoulders

Operations Burden

PATCHES

UPGRADES

SECURITY

BACKUPS

RECOVERY

99.999%

UPTIME

UPSCALE

DOWNSCALE

PERFORMANCE

UAT

STAGING

MONITORING

ALERTS

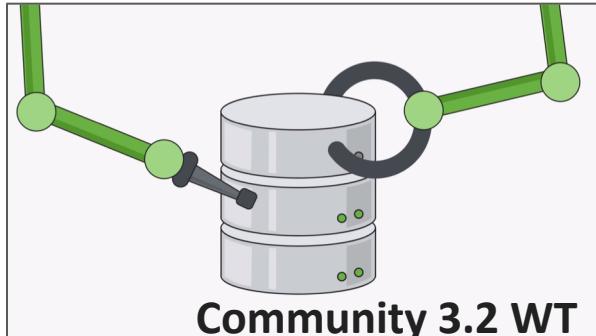
PROVISION

CONFIGURE

INSTALL



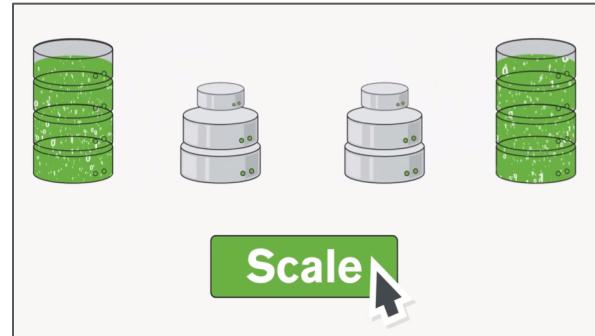
MongoDB Atlas



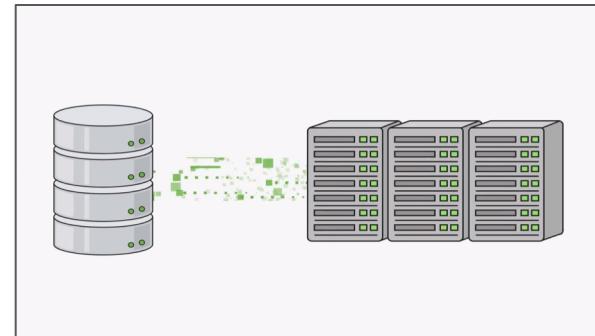
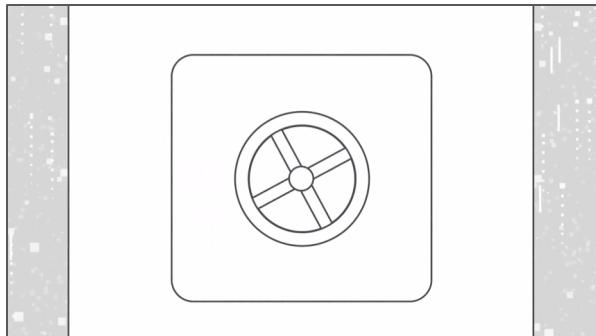
Automated Service



On-demand DBaaS



Elastic Scalability



Automation

Under the hood

- Rolling Upgrades
 - Minor:** Automatic
 - Major:** User Approved
- Monitoring and Alerts - full metrics
- 100's of Notifications

Always up to date

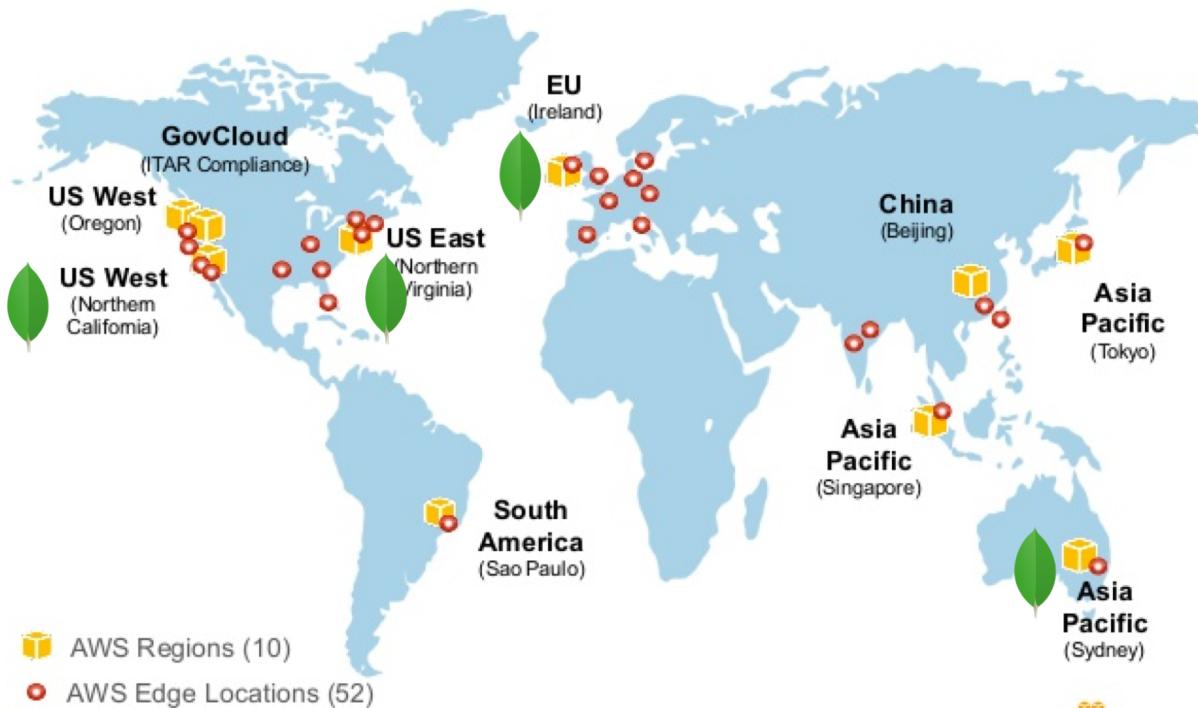
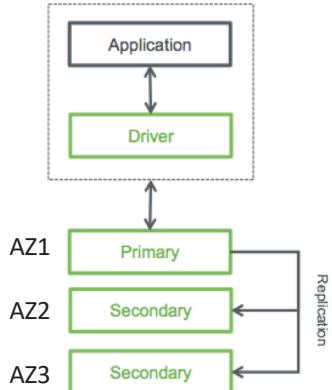


High Availability

Under the hood

4 AWS Regions with 3 AZ zones each

- us-east-1 (N. Virginia)
- us-west-1 (N. California)
- eu-west-1 (Dublin)
- ap-southeast-2 (Sydney)

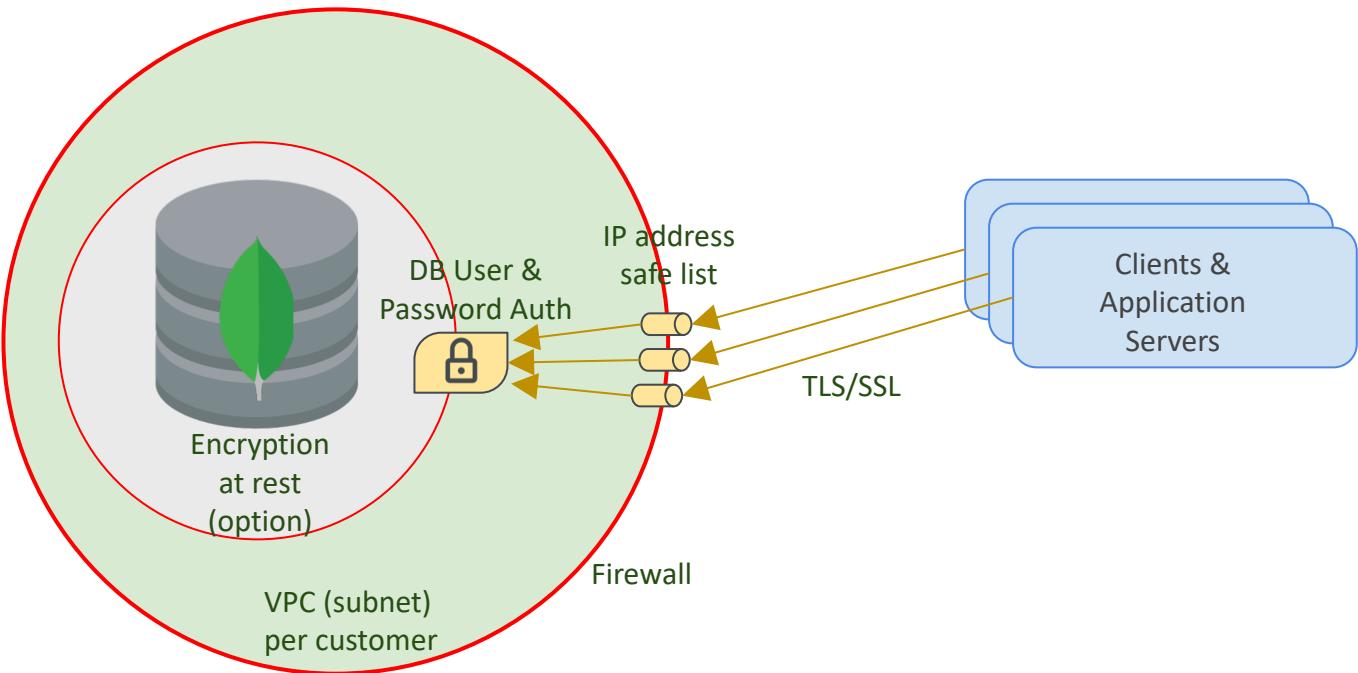


More AWS Regions and Azure and Google Cloud coming soon



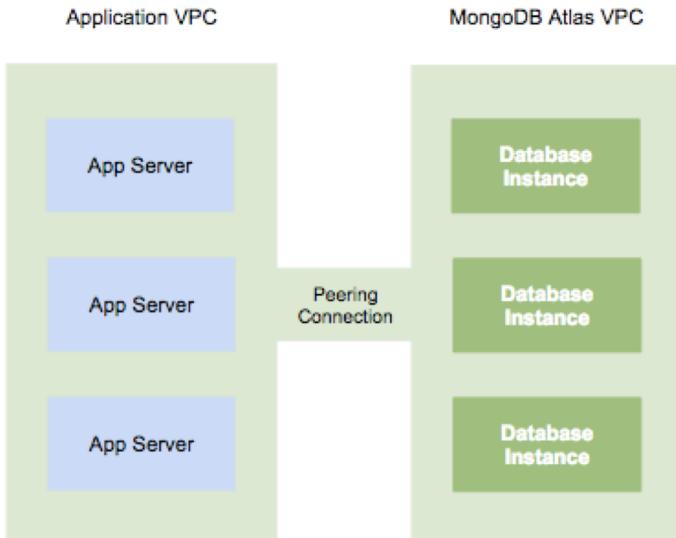
Security

Under the hood



Security – VPC Peering

Under the hood



- Connect App Server and Dev Environments to MongoDB Atlas
 - isolated from public networks
- Peer VPCs in each AWS region to each MongoDB Atlas group
- Use VPC vs. IP-whitelist when scaling app servers

Operations Burden

PATCHES

UPGRADES

SECURITY

BACKUPS

RECOVERY

99.999%

UPTIME

UPSCALE

DOWNSCALE

PERFORMANCE

UAT

STAGING

MONITORING

ALERTS

PROVISION

CONFIGURE

INSTALL



Don't forget, these ...

SCHEMA DESIGN

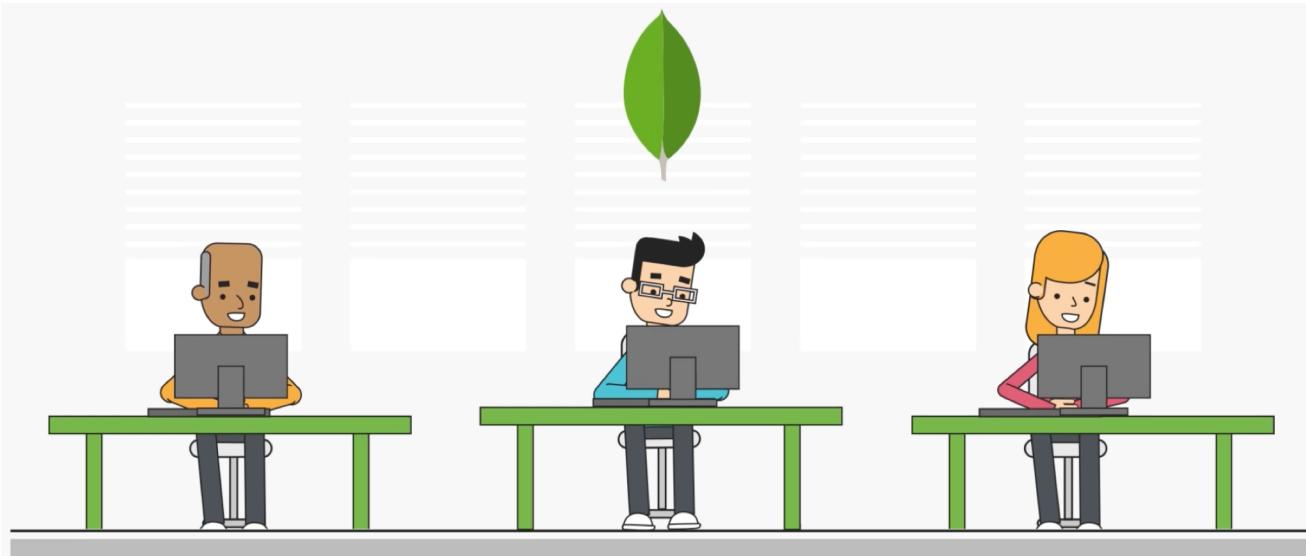
QUERY

SHARD KEY

PROFILING

INDEX

OPTIMIZATION



SERVER SIZE

CPU, RAM, IOPS

CAPACITY

PLANNING

REPLICAS

SHARDS

WHEN TO

RESTORE

Are still your responsibility



{ Code: “GoAtlas25” }

\$25 credit!



Sig Narváez
Sr. Solution Architect
sig@mongodb.com
@SigNarvaez