MDB DAY MTL

# Atlas Workshop

## Hands-On Introduction

**Alek Demko**
Mongo At Morgan

# github.com/alek-mdb/AtlasWorkshop

# github.com/alek-mdb/AtlasWorkshop

Requirements:

- Personal Laptops
- MongoDB Atlas account with personal email
- Connect to personal-wifi

github.com/alek-mdb/AtlasWorkshop

# Exercise 0

Create MongoDB Cluster → M0
Add Sample Data

github.com/alek-mdb/AtlasWorkshop

# Exercise 1

Rich Queries – MQL

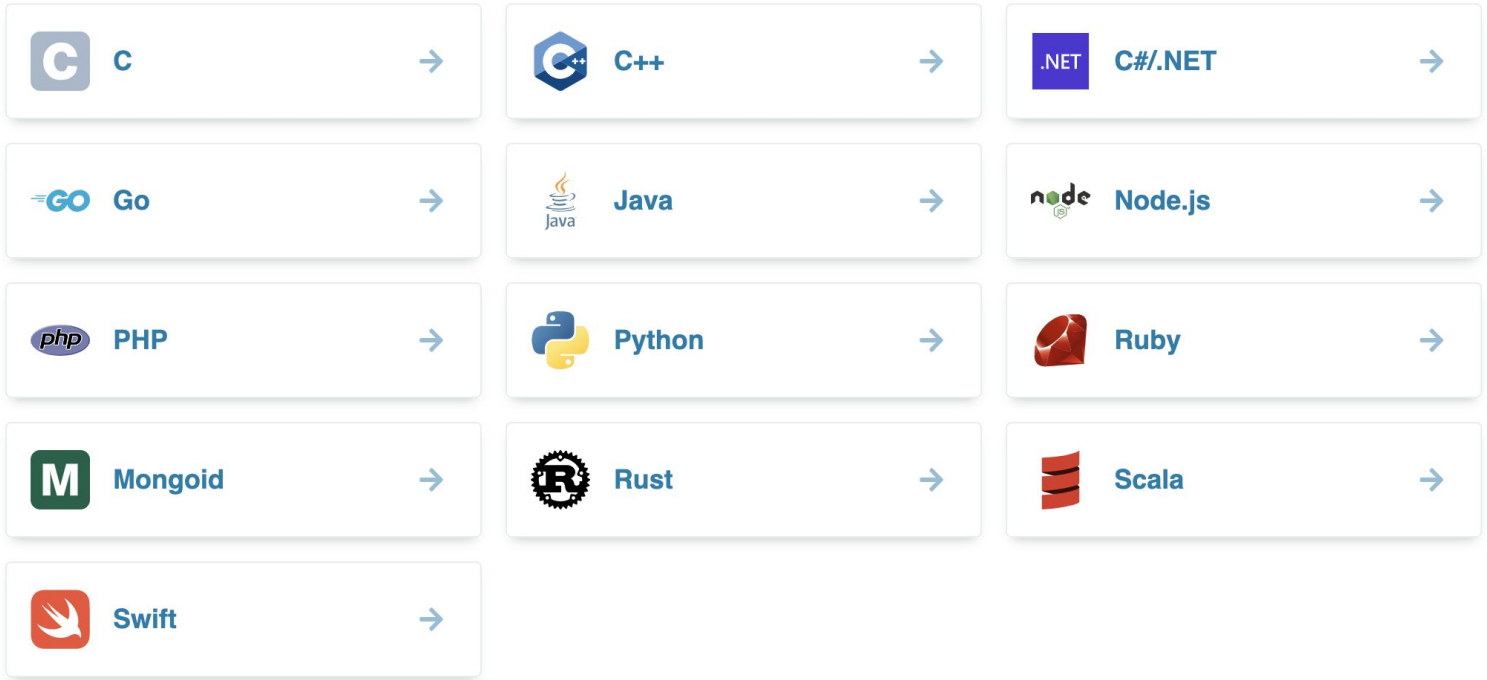
# MongoDB Query API: rich and expressive

| | |
|---|---|
| **Expressive queries** | • Find anyone with phone # “1-212…”<br>• Check if the person with number “555…” is on the “do not call” list |
| **Geospatial** | • Find the best offer for the customer at geo coordinates of 42nd St. and 6th Ave |
| **Text search** | • Find all tweets that mention the firm within the last 2 days |
| **Aggregation** | • Count and sort number of customers by city, compute min, max, and average spend |
| **Native binary JSON support** | • Add an additional phone number to Mark Smith’s record without rewriting the document<br>• Update just 2 phone numbers out of 10<br>• Sort on the modified date |
| **JOIN ($lookup)** | • Query for all San Francisco residences, lookup their transactions, and sum the amount by person |
| **Graph queries ($graphLookup)** | • Query for all people within 3 degrees of separation from Mark |

**MongoDB**

```
{
     customer_id : 1,
       first_name : "Mark",
       last_name : "Smith",
       city : "San Francisco",
       phones: [
         {
               number : "1-212-777-1212",
               type : "work"
          },
          {
               number : "1-212-777-1213",
               type : "cell"
          }
     ]
}
```

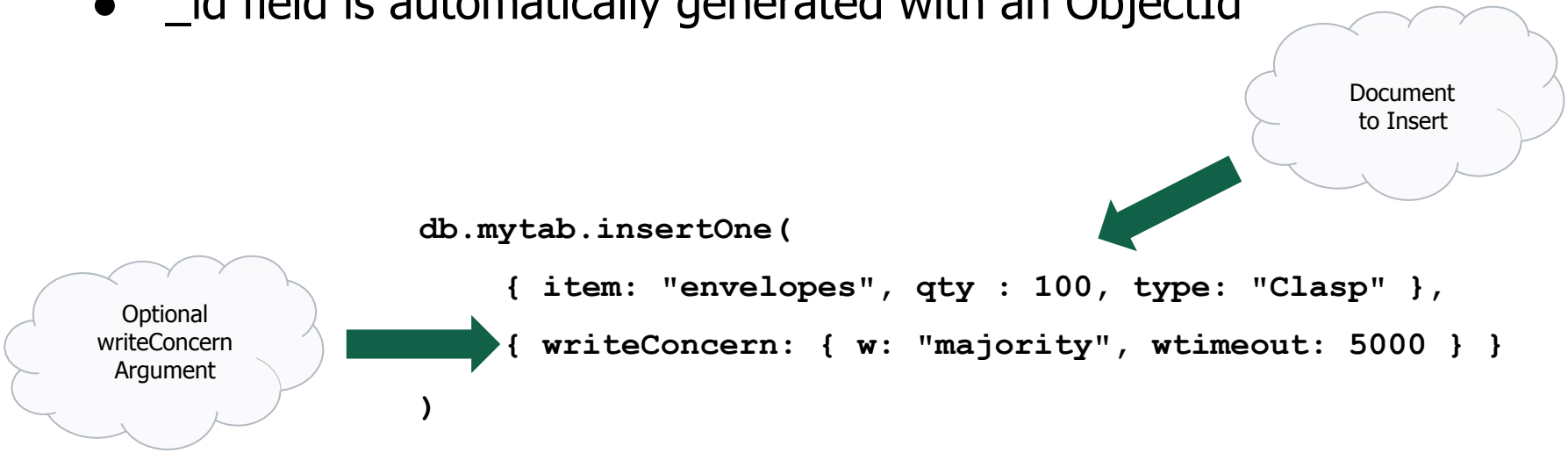
# Intuitive: client drivers




- Common CRUD capabilities but idiomatic to each language
- Uniform HA & Failover capabilities across all

# insertOne()

- Inserts document(s) into a collection
- Can optionally define the writeConcern of the operation
- _id field is automatically generated with an ObjectId



# find() and findOne()

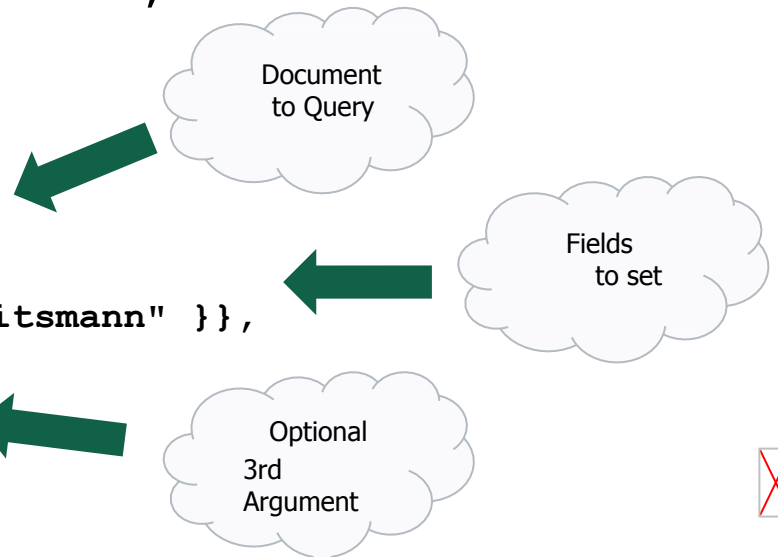
- Get documents from the collection (table)
- Can filter by content (query) and by what is returned (project)
- `findOne()` returns just the first document

```
db.mytab.find({x:1, y:2}, {a:1, b:1, c:1})
```

# updateMany() and updateOne()

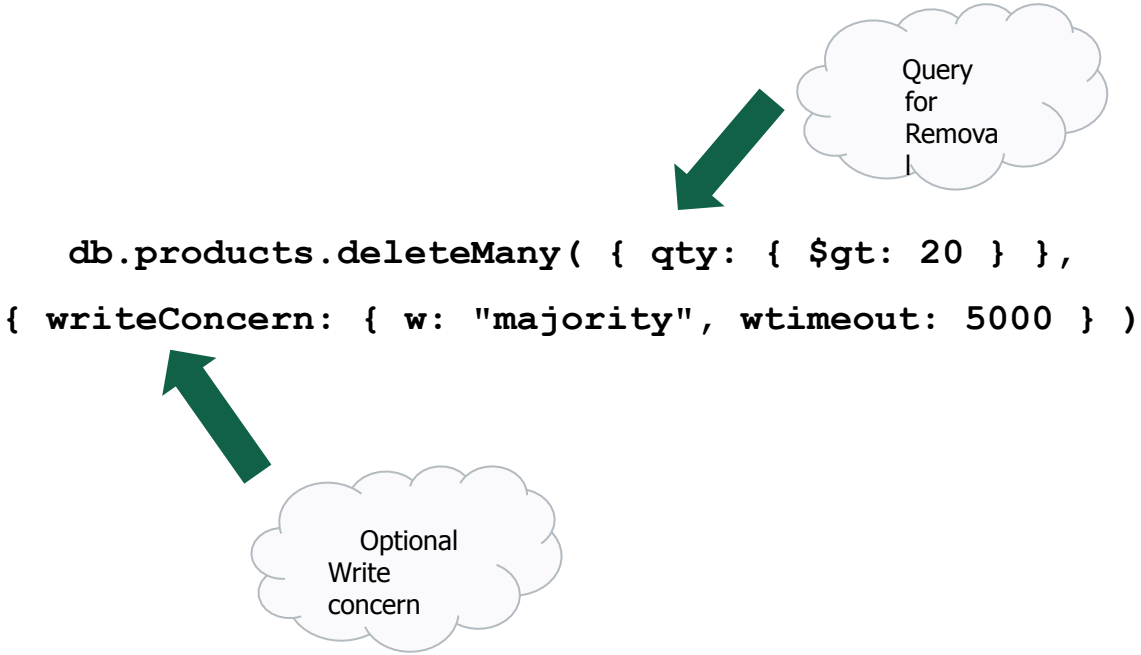
- Modifies existing document(s) in a collection
- Use with a field update operator (MQL)
- Query for the documents you want to update
- Can optionally upsert, define writeConcern, etc.

```
db.books.updateOne(
     { author: "Tom Gleitsmann" },
     { $set: { author: "Thomas Gleitsmann" }},
  { upsert: true }
)
```



# deleteMany()

- Deletes a document from the collection
- Takes an optional writeConcern argument



# Comparison Query Operators

$lt : Exists and less than

$lte: Exists and less than or equal to

$gt : Exists and greater than

$gte : Exists and greater than or equal to

$ne: Does not exist or does but not equal to

$in : Exists and in a set

$nin : Does not exist or not in a set

```
{ age : { $gte : 21 }}
{ temp : { $gt: 10, $lt: 30 }}
```

# Field Update Operators (well some…)

$currentDate : Sets the value of a field to current date

$inc : Increments value

$min : Update if field is less than $min value

$max : Update if field is greater than $max value

$mul : Multiply field value

$set : Set value of field

$unset : Removes field from document

```
{ $set : { bestDB : "MongoDB"}}
{ $mul: { amount : 5, ... } }
```

# github.com/alek-mdb/AtlasWorkshop

## Hands-On: Data Explorer

“Browse Collections”
`sample_airbnb.listingsAndReviews`

Overview | Real Time | Metrics | Collections | Search | Profiler | Performance Advisor | Online Archive | Cmd Line Tools

DATABASES: 9 COLLECTIONS: 23

VISUALIZE YOUR DATA | REFRESH

+ Create Database

Search Namespaces

- sample_airbnb
  - listingsAndReviews
- sample_analytics
- sample_geospatial
- sample_guides
- sample_mflix
- sample_restaurants
- sample_supplies
- sample_training
- sample_weatherdata

## sample_airbnb.listingsAndReviews

STORAGE SIZE: 52.38MB LOGICAL DATA SIZE: 89.99MB TOTAL DOCUMENTS: 5555 INDEXES TOTAL SIZE: 604KB

Find | Indexes | Schema Anti-Patterns 0 | Aggregation | Search Indexes

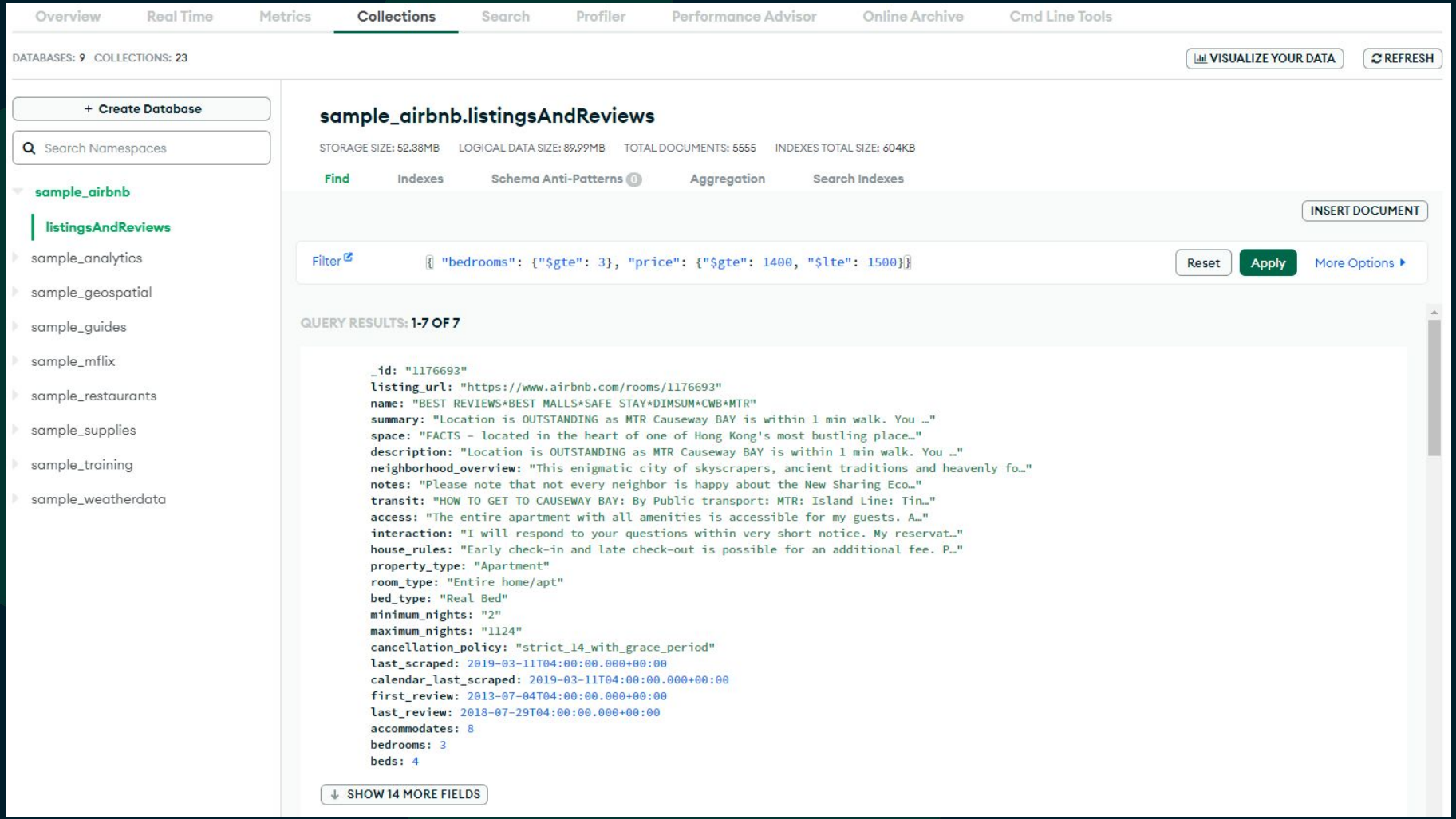
INSERT DOCUMENT

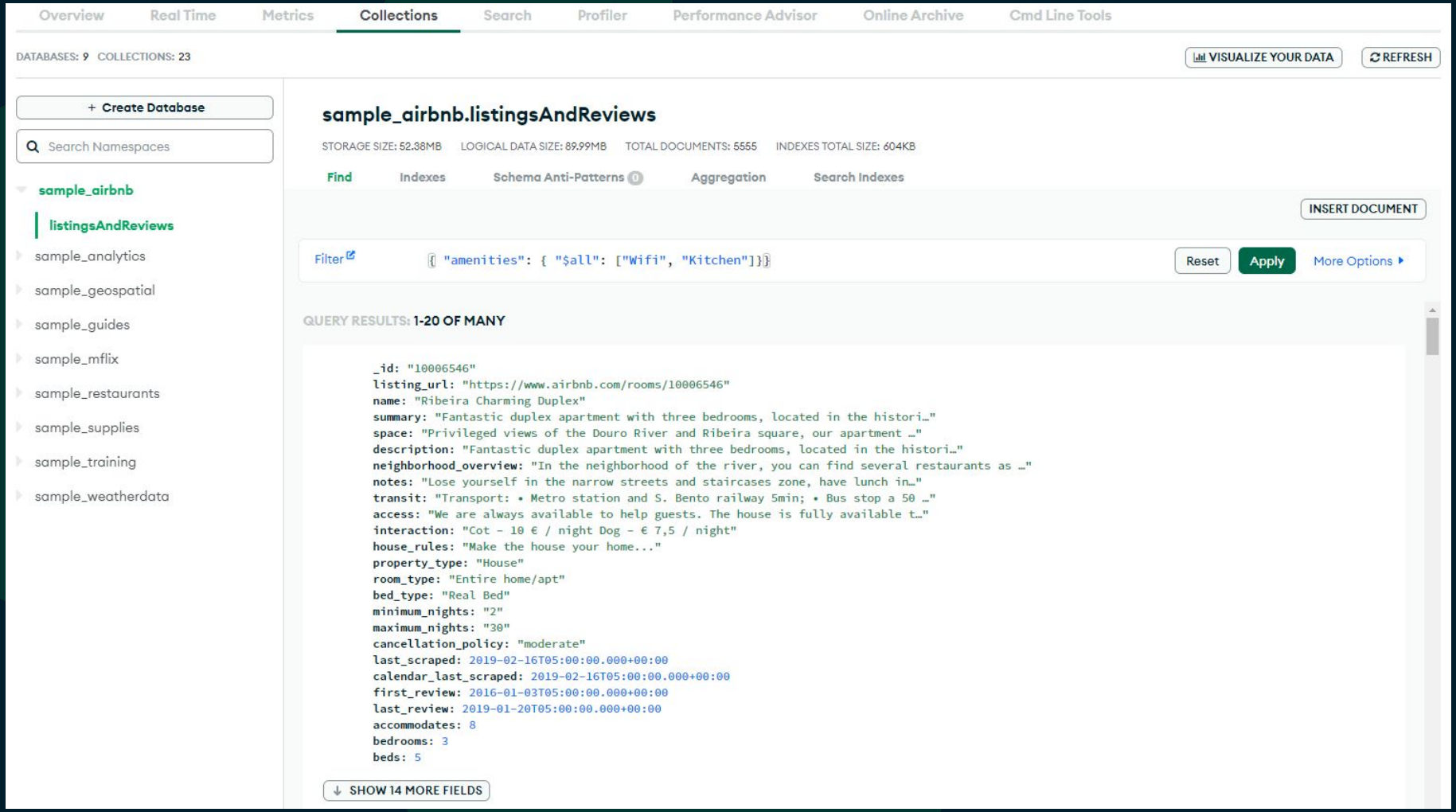
Filter {"address.market":"Hong Kong", "property_type" : "Apartment"} Reset Apply More Options

QUERY RESULTS: 1-20 OF MANY


```
_id: "10141950"
listing_url: "https://www.airbnb.com/rooms/10141950"
name: "Big, Bright & Convenient Sheung Wan"
summary: "Hollywood Rd Studio Only 1 apartment per floor 650sqft! Fast Wifi and …"
space: "Huge Luxury Studio Serviced Apartment 650sqm! That means daily house k…"
description: "Hollywood Rd Studio Only 1 apartment per floor 650sqft! Fast Wifi and …"
neighborhood_overview: "LKF Escalators IFC Mall SOHO food Wellington St Food!"
notes: "I actually live in the studio, so my belongings will be in a wardrobe …"
transit: "3 minutes walk from MTR station"
access: ""
interaction: "I will be on a holiday trip, but you can access me through email and I…"
house_rules: "Please treat it as you would your own home!"
property_type: "Apartment"
room_type: "Entire home/apt"
bed_type: "Real Bed"
minimum_nights: "2"
maximum_nights: "1125"
cancellation_policy: "moderate"
last_scraped: 2019-03-11T04:00:00.000+00:00
calendar_last_scraped: 2019-03-11T04:00:00.000+00:00
first_review: 2016-01-06T05:00:00.000+00:00
last_review: 2016-01-06T05:00:00.000+00:00
accommodates: 2
bedrooms: 0
beds: 1
```


SHOW 13 MORE FIELDS


Overview
Real Time
Metrics
Collections
Search
Profiler
Performance Advisor
Online Archive
Cmd Line Tools
DATABASES: 9 COLLECTIONS: 23
VISUALIZE YOUR DATA
REFRESH
+ Create Database
Search Namespaces
sample_airbnb
listingsAndReviews
sample_analytics
sample_geospatial
sample_guides
sample_mflix
sample_restaurants
sample_supplies
sample_training
sample_weatherdata
sample_airbnb.listingsAndReviews
STORAGE SIZE: 52.38MB LOGICAL DATA SIZE: 89.99MB TOTAL DOCUMENTS: 5555 INDEXES TOTAL SIZE: 604KB
Find
Indexes
Schema Anti-Patterns 0
Aggregation
Search Indexes
INSERT DOCUMENT
Filter
{ "bedrooms": {"$gte": 3}, "price": {"$gte": 1400, "$lte": 1500}}
Reset
Apply
More Options
QUERY RESULTS: 1-7 OF 7
_id: "1176693"
listing_url: "https://www.airbnb.com/rooms/1176693"
name: "BEST REVIEWS*BEST MALLS*SAFE STAY*DIMSUM*CWB*MTR"
summary: "Location is OUTSTANDING as MTR Causeway BAY is within 1 min walk. You …"
space: "FACTS - located in the heart of one of Hong Kong's most bustling place…"
description: "Location is OUTSTANDING as MTR Causeway BAY is within 1 min walk. You …"
neighborhood_overview: "This enigmatic city of skyscrapers, ancient traditions and heavenly fo…"
notes: "Please note that not every neighbor is happy about the New Sharing Eco…"
transit: "HOW TO GET TO CAUSEWAY BAY: By Public transport: MTR: Island Line: Tin…"
access: "The entire apartment with all amenities is accessible for my guests. A…"
interaction: "I will respond to your questions within very short notice. My reservat…"
house_rules: "Early check-in and late check-out is possible for an additional fee. P…"
property_type: "Apartment"
room_type: "Entire home/apt"
bed_type: "Real Bed"
minimum_nights: "2"
maximum_nights: "1124"
cancellation_policy: "strict_14_with_grace_period"
last_scraped: 2019-03-11T04:00:00.000+00:00
calendar_last_scraped: 2019-03-11T04:00:00.000+00:00
first_review: 2013-07-04T04:00:00.000+00:00
last_review: 2018-07-29T04:00:00.000+00:00
accommodates: 8
bedrooms: 3
beds: 4
SHOW 14 MORE FIELDS


Overview
Real Time
Metrics
Collections
Search
Profiler
Performance Advisor
Online Archive
Cmd Line Tools
DATABASES: 9 COLLECTIONS: 23
VISUALIZE YOUR DATA
REFRESH
+ Create Database
Search Namespaces
sample_airbnb
listingsAndReviews
sample_analytics
sample_geospatial
sample_guides
sample_mflix
sample_restaurants
sample_supplies
sample_training
sample_weatherdata
sample_airbnb.listingsAndReviews
STORAGE SIZE: 52.38MB LOGICAL DATA SIZE: 89.99MB TOTAL DOCUMENTS: 5555 INDEXES TOTAL SIZE: 604KB
Find
Indexes
Schema Anti-Patterns 0
Aggregation
Search Indexes
INSERT DOCUMENT
Filter
{ "amenities": { "$all": ["Wifi", "Kitchen"]}}
Reset
Apply
More Options
QUERY RESULTS: 1-20 OF MANY
_id: "10006546"
listing_url: "https://www.airbnb.com/rooms/10006546"
name: "Ribeira Charming Duplex"
summary: "Fantastic duplex apartment with three bedrooms, located in the histori…"
space: "Privileged views of the Douro River and Ribeira square, our apartment …"
description: "Fantastic duplex apartment with three bedrooms, located in the histori…"
neighborhood_overview: "In the neighborhood of the river, you can find several restaurants as …"
notes: "Lose yourself in the narrow streets and staircases zone, have lunch in…"
transit: "Transport: • Metro station and S. Bento railway 5min; • Bus stop a 50 …"
access: "We are always available to help guests. The house is fully available t…"
interaction: "Cot - 10 € / night Dog - € 7,5 / night"
house_rules: "Make the house your home..."
property_type: "House"
room_type: "Entire home/apt"
bed_type: "Real Bed"
minimum_nights: "2"
maximum_nights: "30"
cancellation_policy: "moderate"
last_scraped: 2019-02-16T05:00:00.000+00:00
calendar_last_scraped: 2019-02-16T05:00:00.000+00:00
first_review: 2016-01-03T05:00:00.000+00:00
last_review: 2019-01-20T05:00:00.000+00:00
accommodates: 8
bedrooms: 3
beds: 5
SHOW 14 MORE FIELDS

Overview | Real Time | Metrics | **Collections** | Search | Profiler | Performance Advisor | Online Archive | Cmd Line Tools

DATABASES: 9 COLLECTIONS: 23

VISUALIZE YOUR DATA | REFRESH

+ Create Database

Search Namespaces

- sample_airbnb
  - listingsAndReviews
- sample_analytics
- sample_geospatial
- sample_guides
- sample_mflix
- sample_restaurants
- sample_supplies
- sample_training
- sample_weatherdata

## sample_airbnb.listingsAndReviews

STORAGE SIZE: 52.38MB LOGICAL DATA SIZE: 89.99MB TOTAL DOCUMENTS: 5555 INDEXES TOTAL SIZE: 604KB

Find | Indexes | Schema Anti-Patterns 0 | Aggregation | Search Indexes

INSERT DOCUMENT

Filter {"address.market":"Hong Kong", "property_type" : "Apartment", "bedrooms": {"$gte": 3}, "price": {"$ Reset Apply More Options

QUERY RESULTS: 1-4 OF 4

```
_id: "1176693"
listing_url: "https://www.airbnb.com/rooms/1176693"
name: "BEST REVIEWS*BEST MALLS*SAFE STAY*DIMSUM*CWB*MTR"
summary: "Location is OUTSTANDING as MTR Causeway BAY is within 1 min walk. You …"
space: "FACTS - located in the heart of one of Hong Kong's most bustling place…"
description: "Location is OUTSTANDING as MTR Causeway BAY is within 1 min walk. You …"
neighborhood_overview: "This enigmatic city of skyscrapers, ancient traditions and heavenly fo…"
notes: "Please note that not every neighbor is happy about the New Sharing Eco…"
transit: "HOW TO GET TO CAUSEWAY BAY: By Public transport: MTR: Island Line: Tin…"
access: "The entire apartment with all amenities is accessible for my guests. A…"
interaction: "I will respond to your questions within very short notice. My reservat…"
house_rules: "Early check-in and late check-out is possible for an additional fee. P…"
property_type: "Apartment"
room_type: "Entire home/apt"
bed_type: "Real Bed"
minimum_nights: "2"
maximum_nights: "1124"
cancellation_policy: "strict_14_with_grace_period"
last_scraped: 2019-03-11T04:00:00.000+00:00
calendar_last_scraped: 2019-03-11T04:00:00.000+00:00
first_review: 2013-07-04T04:00:00.000+00:00
last_review: 2018-07-29T04:00:00.000+00:00
accommodates: 8
bedrooms: 3
beds: 4
```

SHOW 14 MORE FIELDS

sample_airbnb.listingsAndReviews

Not in Atlas UI

5.6k DOCUMENTS 4 INDEXES

Documents Aggregations Schema **Explain Plan** Indexes Validation

Filter {"address.market":"Hong Kong", "property_type" : "Apartment", "bedrooms": {"$gte": 3}, ' Reset Explain </> More Options

VIEW VISUAL TREE RAW JSON

## Query Performance Summary Learn more

Documents Returned: **4**

Index Keys Examined: **3626**

Documents Examined: **3626**

Actual Query Execution Time (ms): **14**

Sorted in Memory: **no**

Query used the following index: PROPERTY_TYPE ↑ ROOM_TYPE ↑ BEDS ↑

### FETCH

nReturned 4 Execution Time 2 ms

DETAILS

### IXSCAN

github.com/alek-mdb/AtlasWorkshop

# 5 Minute Break

github.com/alek-mdb/AtlasWorkshop

# Exercise 2

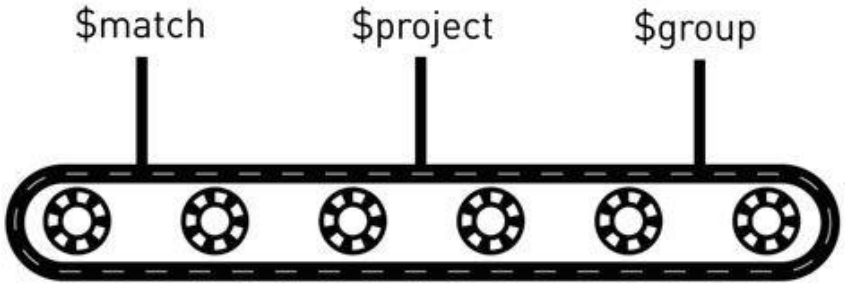
Aggregation Framework

# Aggregations 101

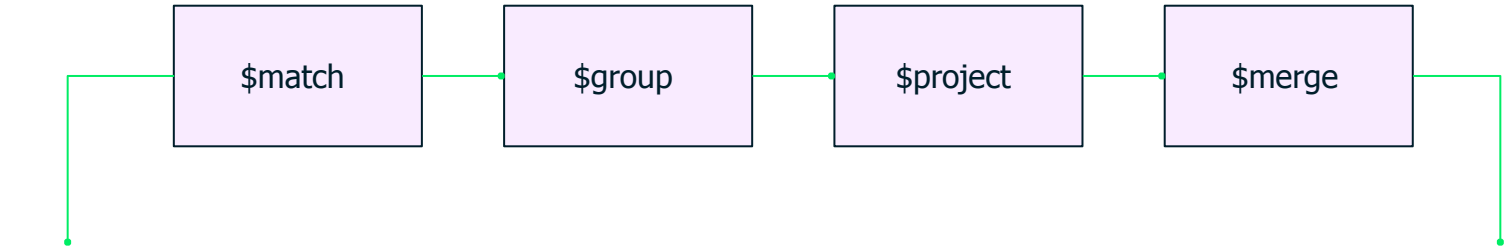
- Until now, we have only filtered data
  - What Documents
  - What Fields
  - What Order
- Aggregation allows us to compute new data
  - Calculated fields
  - Summarized and grouped values
  - Reshape documents

# Aggregation Stages

- Each transformation is a single step known as a stage.
- Compared to one huge SQL style statement this is
  - Easier to understand
  - Easier to debug
  - Easier for MongoDB to rewrite and optimize

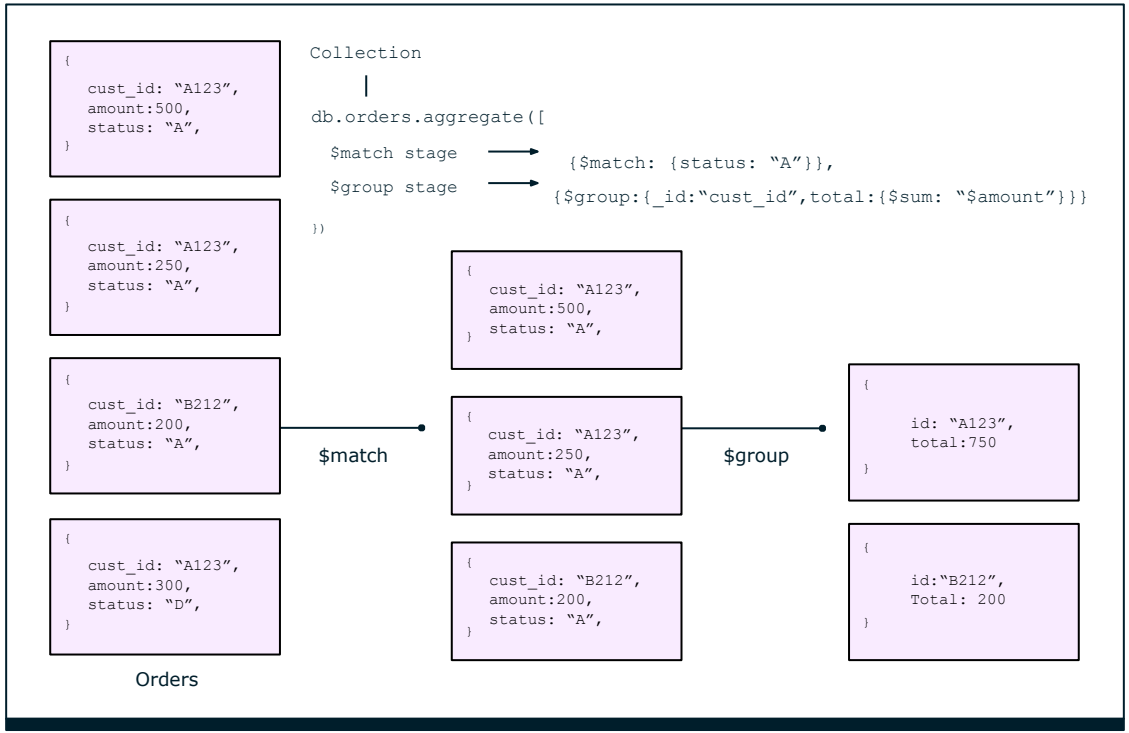


# Aggregation Pipelines




Allows developers to extend MongoDB functionality by creating their own expressions and have them execute as part of an aggregation pipeline

# Aggregation pipelines



# Stages we know

- $match equivalent to find(query)
- $project equivalent to find({},projection)
- $sort equivalent to find().sort(order)
- $limit equivalent to find().limit(num)
- $skip equivalent to find().skip(num)
- $count equivalent to find().count()

When these are used at the start of a pipeline, they are transformed to a find() by the query optimizer.

# Comparing Aggregation Syntax

## Find the name of the host in Canada with the most "total listings":

**Using find()**

```
db.listingsAndReviews.find(
    {"address.country":"Canada"},
    {"_id":0, "host.host_total_listings_count":1, "host.host_name":1}
).sort({"host.host_total_listings_count":-1}).limit(1)
```

**Using aggregate()**

```
db.listingsAndReviews.aggregate([
    {$match:{"address.country":"Canada"}},
    {$sort: {"host.host_total_listings_count":-1 }},
    {$limit:1},
    {$project:{"_id":0, "host.host_total_listings_count":1, "host.host_name":1}}
])
```

# Dollar Overloading

```
{$match: {a: 5}}
```

Dollar on left means a stage name - in this case a $match stage

```
{$set: {b: "$a"}}
```

Dollar on right of colon "$a" refers to the value of field a

```
{$set: {area: {$multiply: [5,10]}}
```

$multiply is an expression name left of colon

```
$map:{input: "$quizzes", as: "grade",in: { $add: [ "$$grade", 2 ] }}}
```

$$grade refers to a variable in $map statements

Think as a programmer - not as a database shell.

Define variables. Doing this helps you keep track of brackets.

It also helps you really understand the Object concept.

```
//Do it THIS way for ease of testing and debugging

> no_celebs = {$match:{"user.followers_count":{$lt:200000}}}

> name_only = {$project:{"user.name":1, "user.folowers_count":1, id:0}}

> most_popular = {$sort: {"user.followers_count":-1}}

> first_in_list = {$limit:1}

> pipeline = [no_celebs,name_only,most_popular,first_in_list]

> db.twitter.aggregate(pipeline)
```

# github.com/alek-mdb/AtlasWorkshop

## Hands-On: Data Explorer Pt. 2

“Browse Collections”
`sample_airbnb.listingsAndReviews`
“Aggregations”

# Exercise 1

Using the Atlas Aggregation Builder and the Airbnb listings data (`sample_airbnb.listingsAndReviews`) read the following question and output the answer. You will need to use a `$set` stage and a `$project` stage with expressions.

To keep in mind…

1. The price for the basic rental is in the **$price** field; for that price, you can have the number of guests provided in **$guests_included** field.
2. The property may take more guests in total (maximum guests is in **$accommodates** property).
3. You have to pay **$extra_people** dollars per person for every person more than **$guests_included**

Requirement:

- Calculate how many extra guests you can have for each property and add that as a field using `$set`
- Calculate how much it would cost with these extra guests. `$project` the basic price and the price if fully occupied with `$accommodates` people.

# ANSWER: Exercise 1

Using the Airbnb listings data:

- Calculate how much it would cost for these extra guests - they cost "$extra_people" each and $project the basic price and the price if fully populated

```
[{$set: {numExtraGuests: {
        $subtract: ["$accommodates","$guests_included",],},},},
 {$set: {extraGuestCost: {
        $multiply: ["$extra_people","$numExtraGuests",],},},},
 {$project: {
      price: 1,
      maxprice: {
        $add: ["$price", "$extraGuestCost"],},},},]
```

# Exercise 2

Calculate how many Airbnb properties there are in each country, ordered by count, list the one with the largest count first. You should continue working with namespace `sample_airbnb.listingsAndReviews`.

- **Hint:** to count things, you add the explicit value 1 to an accumulator using $sum

# ANSWER: Exercise 2

Calculate how many Airbnb properties there are in each country, ordered by count, list the one with the largest count first.

```
[{$group: {
     _id: "$address.country",
     count: { $sum: 1 },},},
 {$sort: { count: -1 },},]
```

# Exercise 3

What are the top 10 most popular amenities across all listings?

# ANSWER: Exercise 3

What are the top 10 most popular amenities across all listings?

```
[{$unwind: { path: "$amenities" },},
 {$group: {
     _id: "$amenities",
      count: {$sum: 1},},},
 {$sort: {count: -1,},},]
```

github.com/alek-mdb/AtlasWorkshop

# 5 Minute Break

# github.com/alek-mdb/AtlasWorkshop

## Atlas Search

# What is Search?

Search, or “Full-Text Search” is the ability to search across all of your data and efficiently return a list of results, ranked based on how well they matched to the search term

**Usually defined as a Search bar**

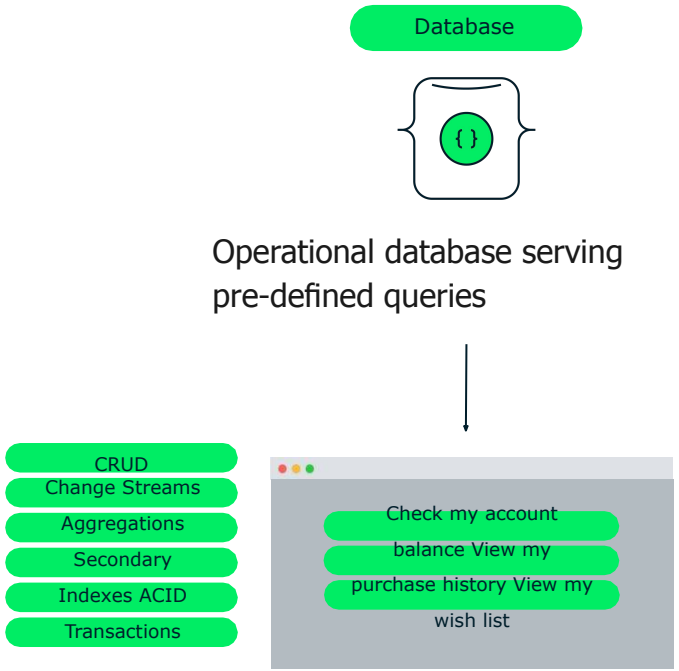

**Not an exact match, but a LIST OF RESULTS BASED ON RELEVANCE**

**Results based on a score**

- **Result 1**
- **Result 2**
- **Result 3**
- **Result 4**

# How developers build app search today

Stand up a database to provide the application's persistence layer



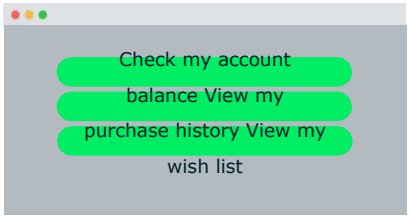
# How developers build app search today

Bolt-on a search engine to power search across application data

Database

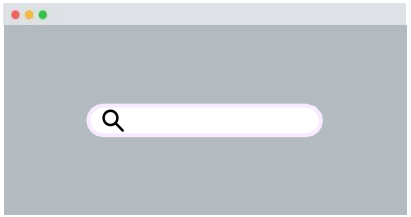
Operational database serving pre-defined queries

CRUD
Change Streams
Aggregations
Secondary
Indexes ACID
Transactions
...




Search Engine

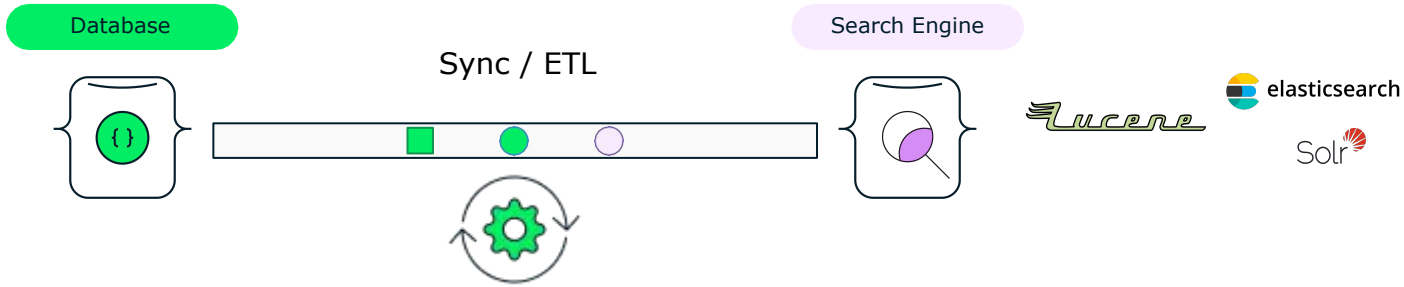
Search engine inferring intent from free form, natural language queries



Fuzzy matching
Autocomplete
Facets and
counts
Synonyms
Scoring /
Weighting
moreLikeThis

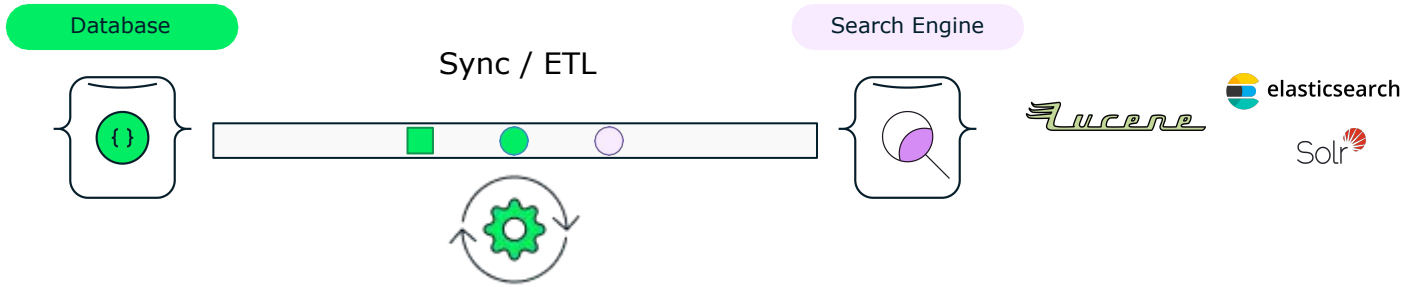
# How developers build app search today

Stand up a replication mechanism to keep the systems in sync



# How developers build app search today

## Architectural complexity in our application stack




### Developer friction

Different query APIs and drivers for database and search, coordinate schema changes

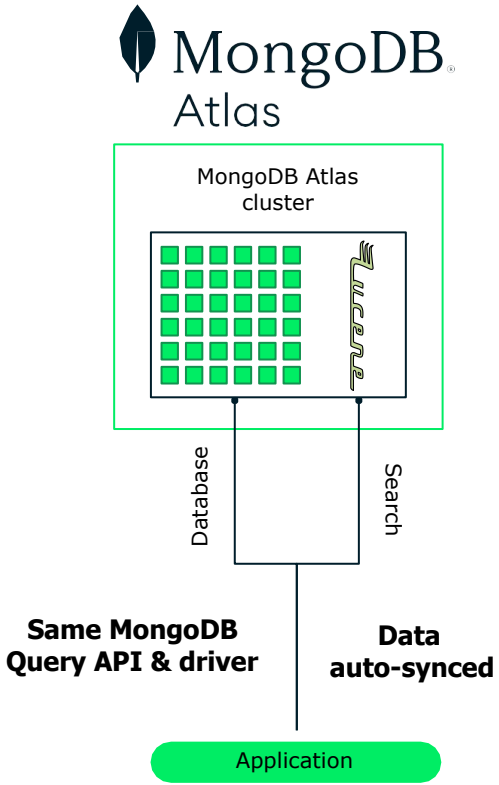
### Pay the sync tax

Requires its own systems and skills. Recovering sync errors can consume 10% of a developer's time

### Operational overhead

More to provision, secure, upgrade, patch, back up, monitor, scale, etc.

# How does Atlas Search address that pain?

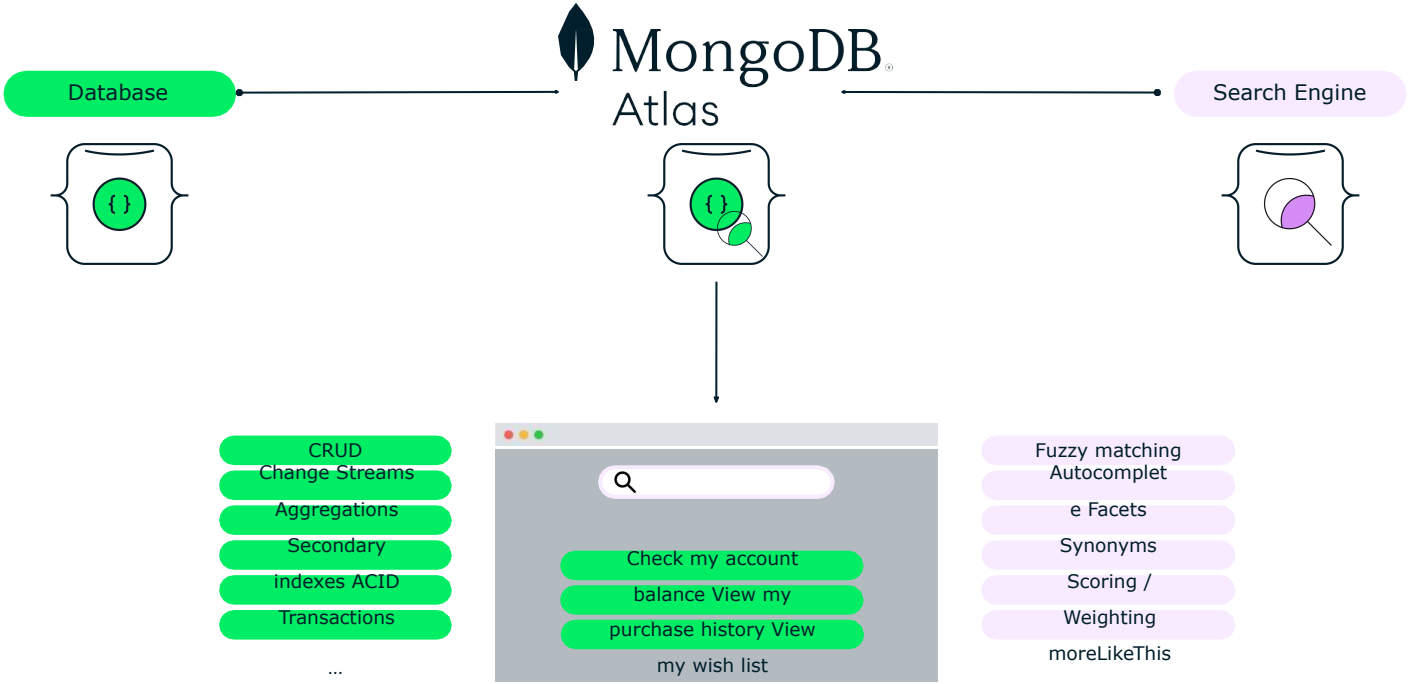



## Embeds a fully managed Apache Lucene index right alongside the database

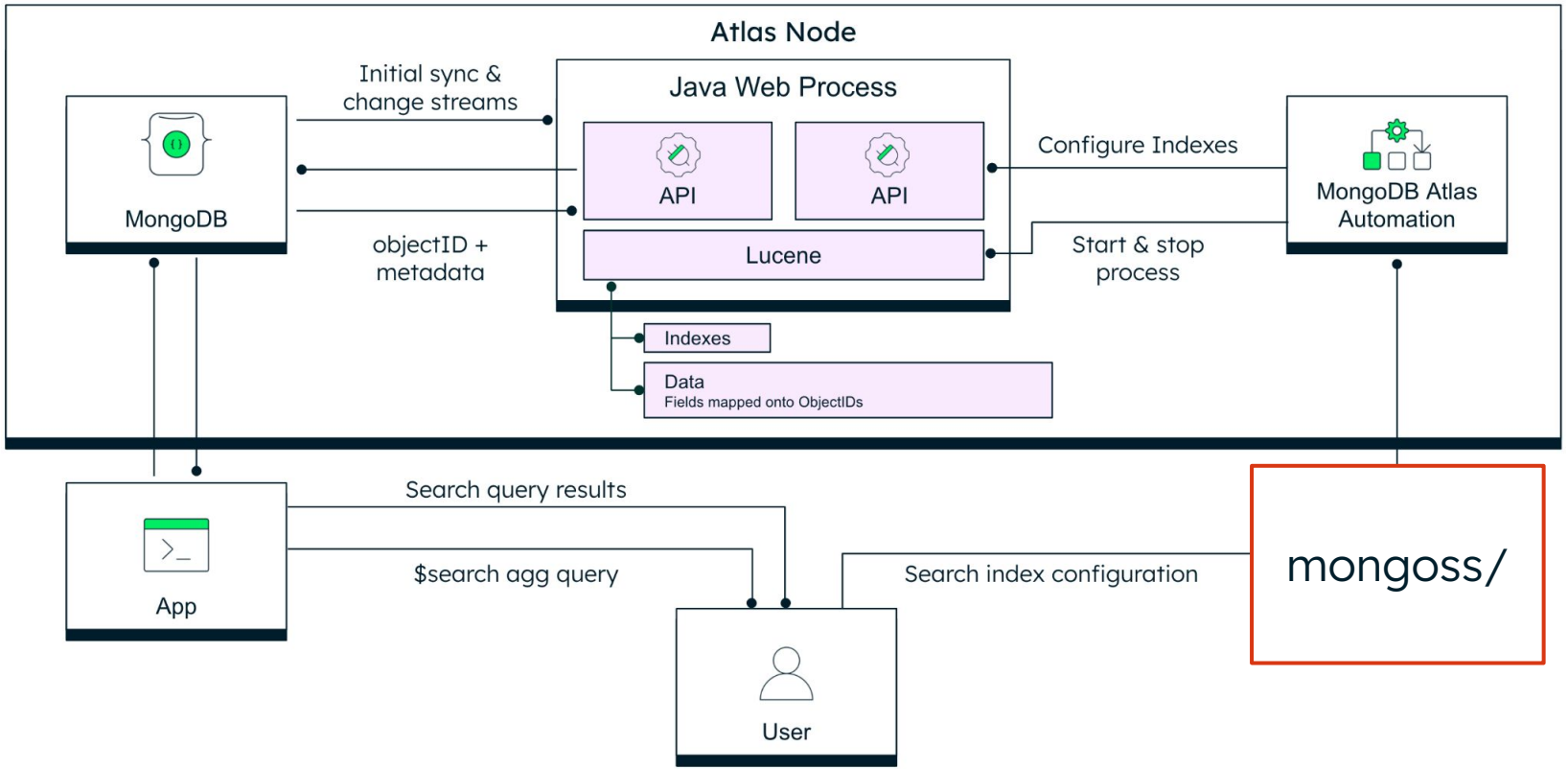
- **Improved developer productivity**
  Build database and search features using the same query API and driver
- **Simplified data architecture**
  Automatic data synchronization, even as your data and schema changes
- **Fully managed platform**
  Get the security, performance, and reliability of Atlas

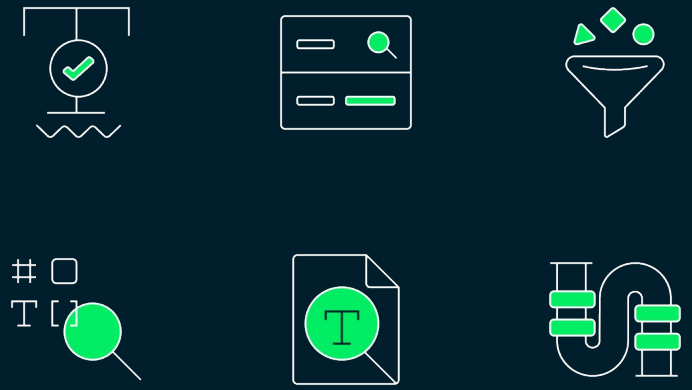
# Unifying database and search

Compress 3 systems into 1, ship search features **30%-50% faster**



# Atlas Search Architecture





# Rich Search Features

**Atlas Search** offers many features to fine-tune your search results to help users find what they need, including:

- ✓ Rich query language
- ✓ Autocomplete
- ✓ Filters and facets
- ✓ Fuzzy search
- ✓ Multi-data type support
- ✓ Multi-language support (40+)
- ✓ Highlighting
- ✓ Custom scoring expressions/functions
- ✓ Synonyms
- ✓ More-like-this similarity
- ✓ Vector Search

# Analyzers: Tokenization

We start with an original query

**Analyzers** apply parsing and language rules to the query

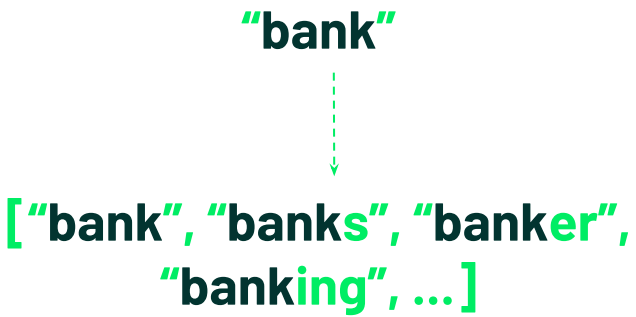
**Tokens** are the individual terms that are queried; think of these as the words that matter in a search query

# Analyzers: Stemming

Analyzers also support **language rules**

A search for...

...can return results for related words



# Indexing Capabilities

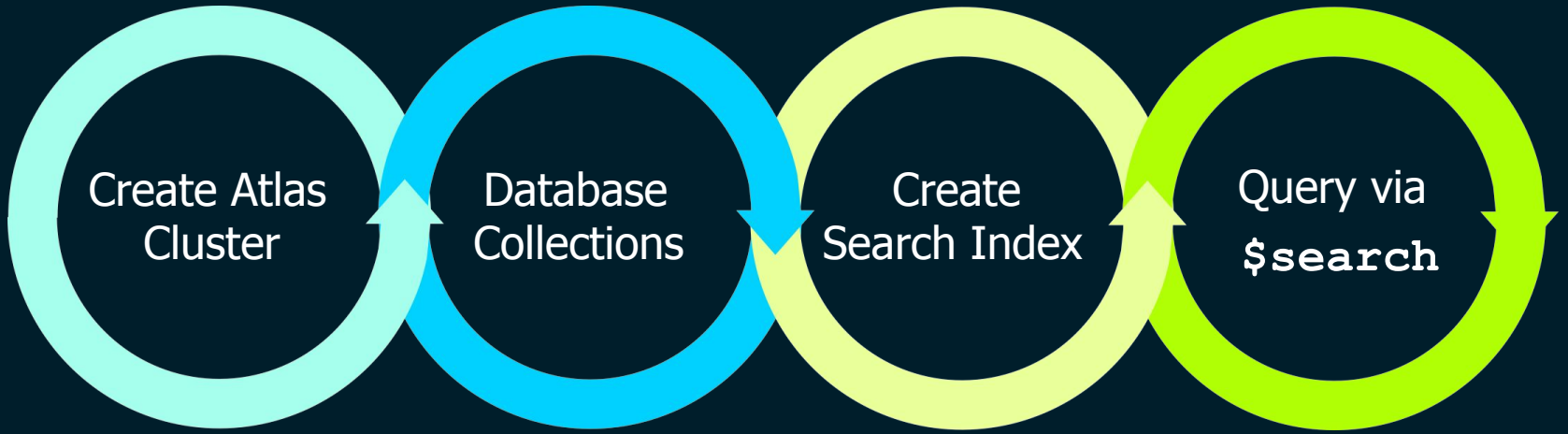
Search features are all about helping customers make their data more **discoverable** and **relevant** for end users

- **Support for 40+ languages**
- Multiple data types
- Custom Analyzers
- Automated ETL
- Autocomplete
- Faceting
- Stored source
- Semantic search

| | | | | |
|---|---|---|---|---|
| `lucene.arabic` | `lucene.armenian` | `lucene.basque` | `lucene.bengali` | `lucene.brazilian` |
| `lucene.bulgarian` | `lucene.catalan` | `lucene.cjk` | `lucene.czech` | `lucene.danish` |
| `lucene.dutch` | `lucene.english` | `lucene.finnish` | `lucene.french` | `lucene.galician` |
| `lucene.german` | `lucene.greek` | `lucene.hindi` | `lucene.hungarian` | `lucene.indonesian` |
| `lucene.irish` | `lucene.italian` | `lucene.latvian` | `lucene.lithuanian` | `lucene.norwegian` |
| `lucene.persian` | `lucene.portuguese` | `lucene.romanian` | `lucene.russian` | `lucene.sorani` |
| `lucene.spanish` | `lucene.swedish` | `lucene.turkish` | `lucene.thai` | |

Each language analyzer has built-in rules based on the language's usage pattern

# How do I search data in atlas?



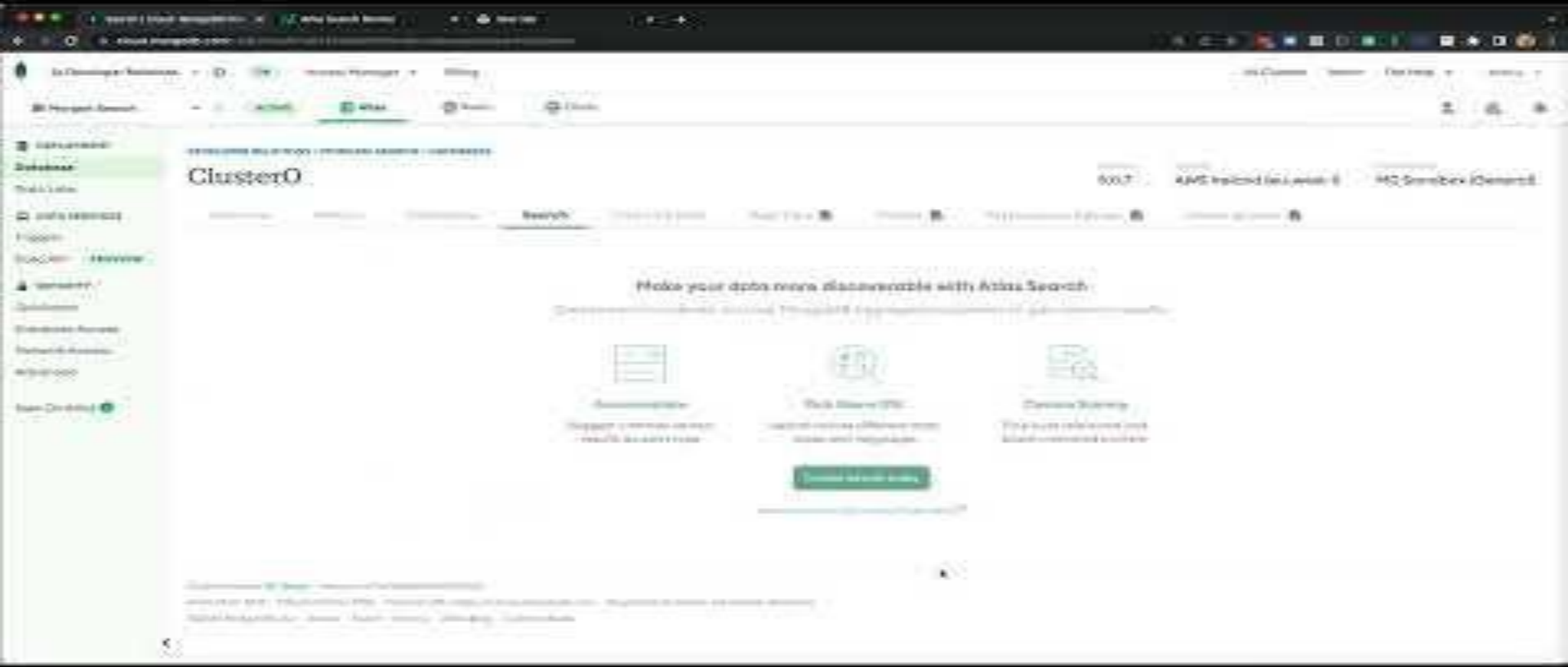

**github.com/alek-mdb/AtlasWorkshop**

github.com/alek-mdb/AtlasWorkshop

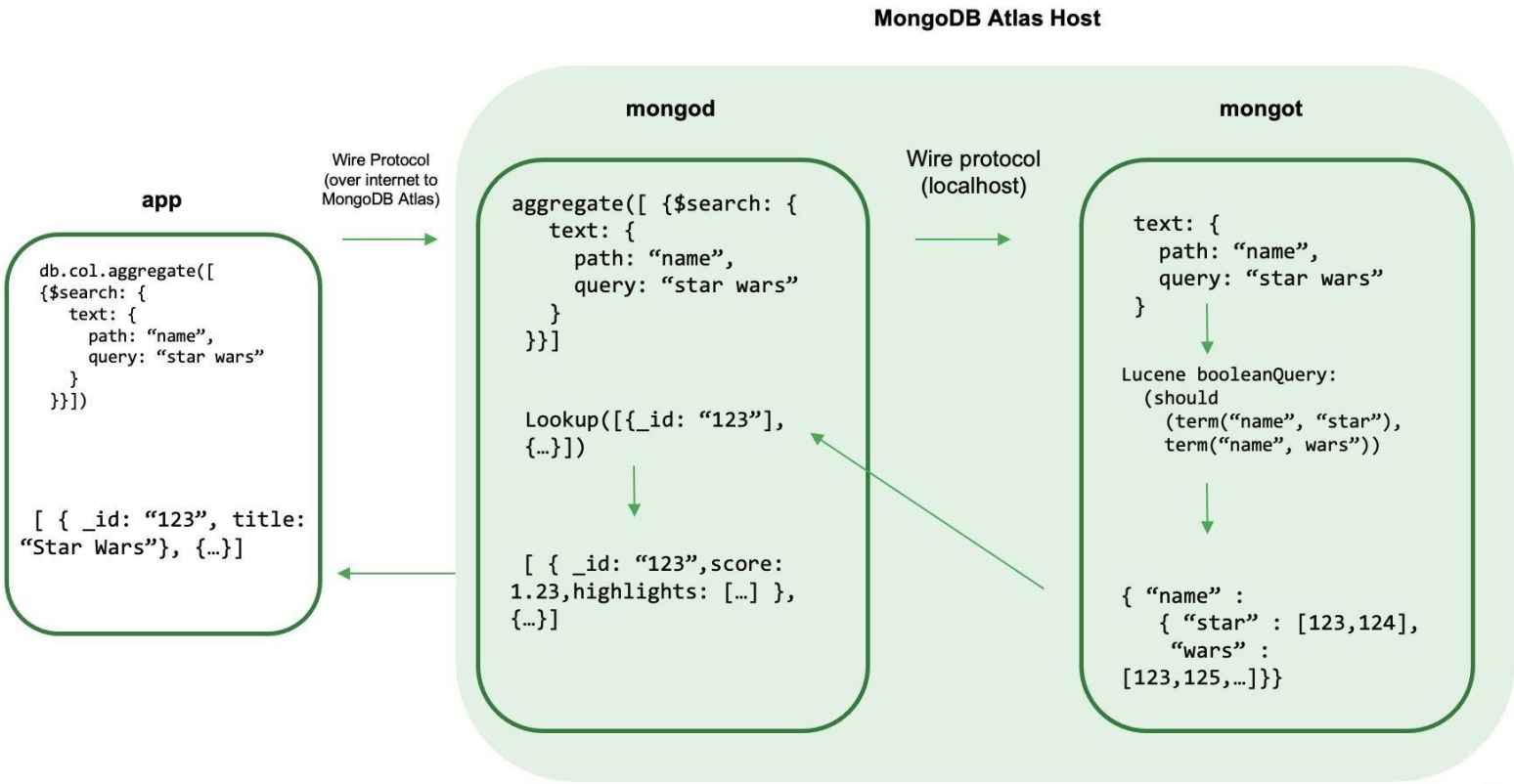
# Search Exercise 1

Create Indexes:

1. Default Search Index
2. Autocomplete on Title


Cluster0
Make your data more discoverable with Atlas Search

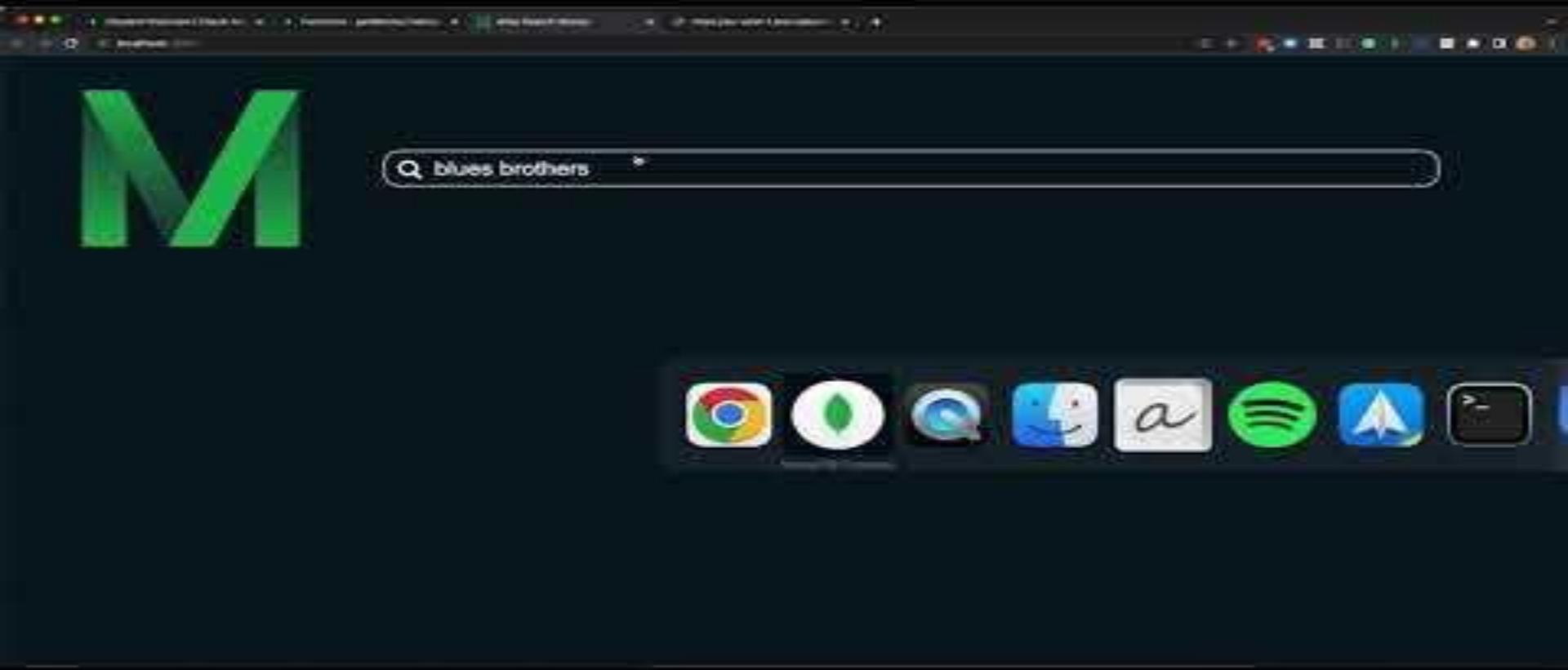
# Query lifecycle



github.com/alek-mdb/AtlasWorkshop

# Search Exercise 2

Search Aggregations


blues brothers

github.com/alek-mdb/AtlasWorkshop

# Search Exercise 3

Vector Search Overview

# Token / Lexical Search

What?

- TF-IDF/BM25(f)
- Keyword Search
- Tokenization is applied during indexing

When?

- Your text corpus closely matches how users search
- Most traditional text search use cases.
- First pass at text-based relevancy.

# Vector Search

What?

- ANN/kNN
- Contextual Similarities
- Search supporting multiple modalities.

When?

- Ambiguous user input.
- There is a ‘vocabulary gap’ between corpus and how users search
- Non-text searches.

# Vector Search powers a number of key use cases

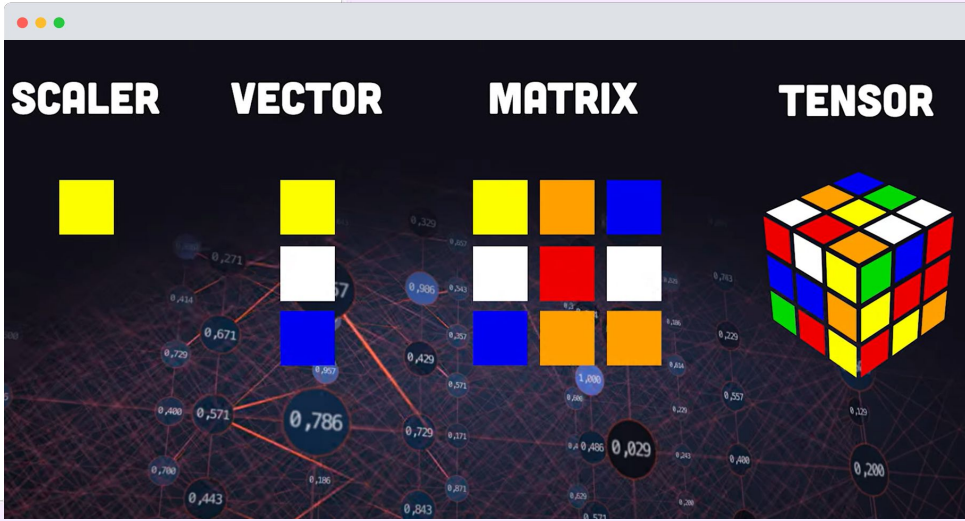
- Semantic Search
- Similarity Search
- Recommendation Engines
- Long-term memory for LLMs
- Q & A Systems
- Image, Audio, Multimedia Search

# What is a vector?

In math terms, it is just a data structure - an array

We store a list of points that make up a representation of a word/term in multidimensional space, then compare how close other points (terms) are to calculate similarity

The devil is in the details of how we convert a word/term/phrase into this vector

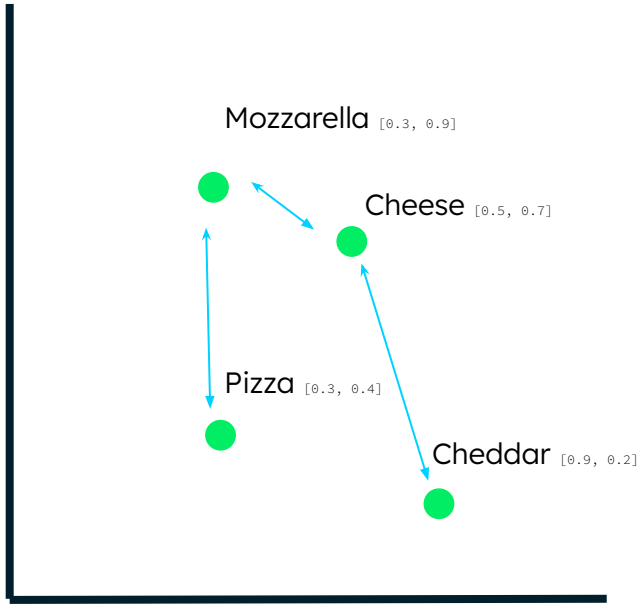



Source: Fireship.io - Video

SEMANTIC SIMILARITY

# Demonstrated via simplified vector

(in two dimensions)



# Vector Search - Data Structure

**Sample Document**


```
"plot":"Three unemployed parapsychology professors set up shop as a unique gho…"
"title":"Ghostbusters"
"year":1984
"plot_embedding": Array
0:-0.03442629799246788
1:-0.013609294779598713
2:-0.01283580157905817
3:-0.044231850653886795
4:-0.02906322292983532
5:-0.021946318447589874
6:-0.001014339504763484
7:-0.11677482724189758
8:-0.03159894421696663
9:-0.05249767750501633
10:0.07232994586229324
...
```


embeddings generated by Atlas trigger invoking an encoding API call.

More dimensions = more floating points

# Semantic Search - Index Definition

**Index Definition**

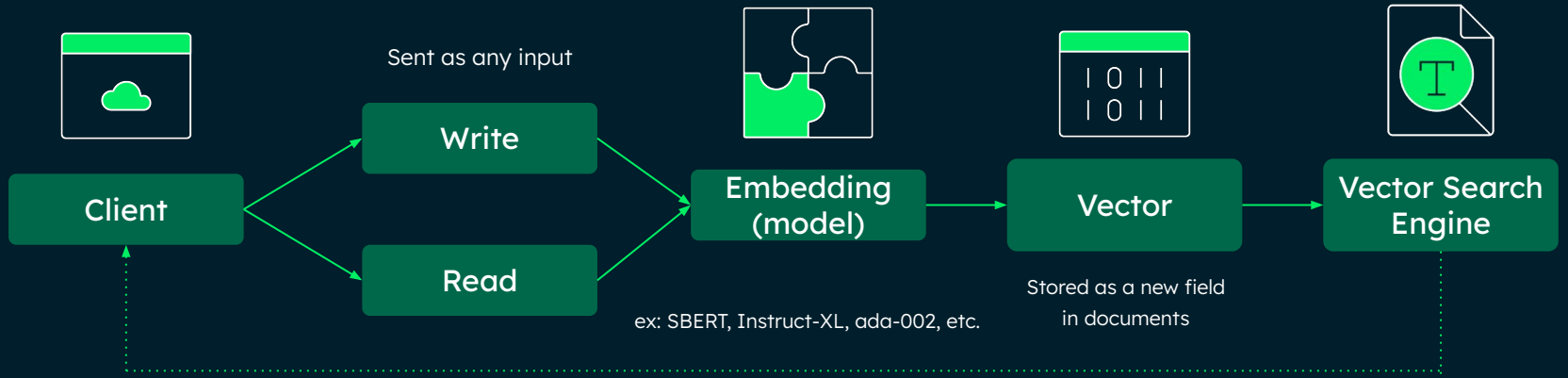
```
{
  "mappings": {
    "fields": {
      "plot_embedding": [
        {
          "type": "knnVector",
          "dimensions": 1536,  // currently limited to 2048
          "similarity": "<euclidean | dotProduct | cosine>",
        }
      ]
      "year": { type: "number" },
      "content": { type: "string" }
    }
  }
}
```

# Semantic Search - Query

**Sample Query**

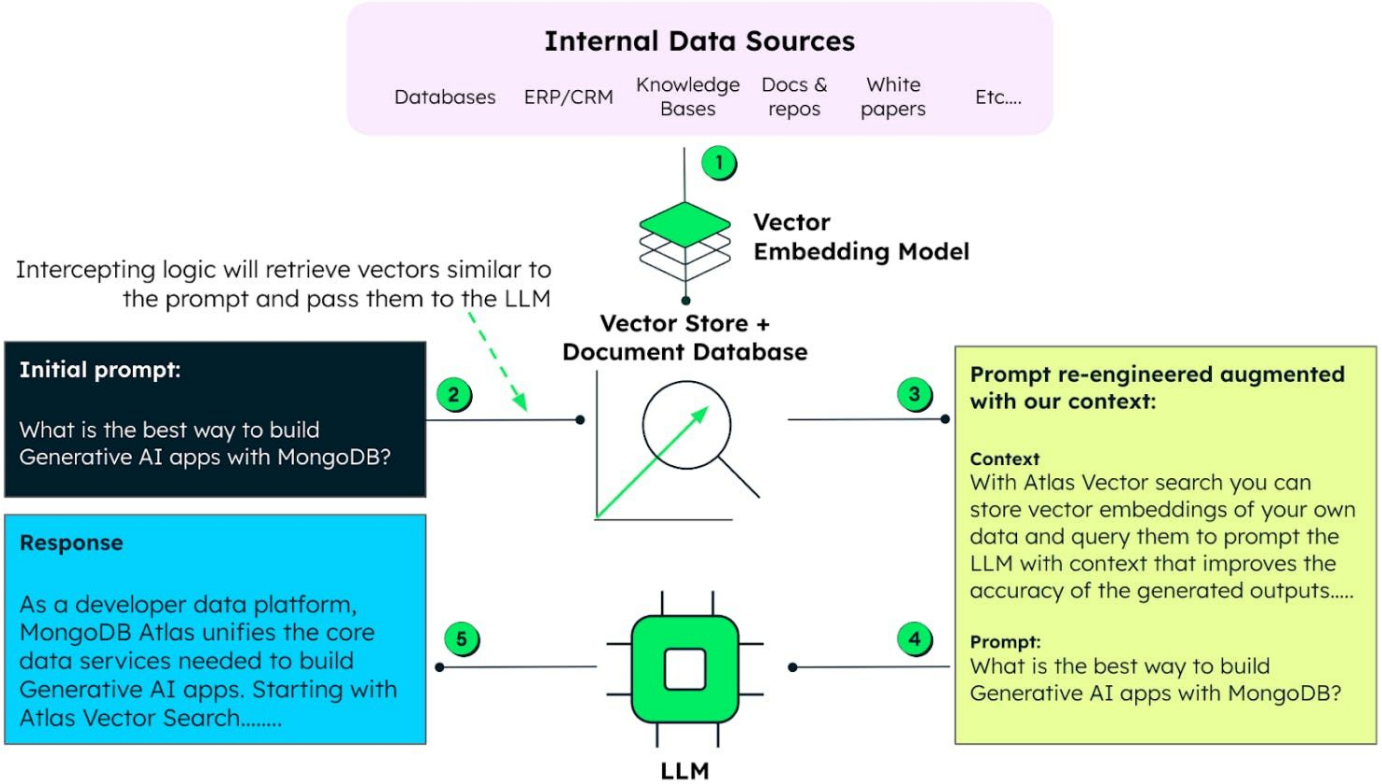
```
[{
    "$vectorSearch": {
      "index": "default",
      "path": "plot_embedding",
      "filter": {
        "$and": [{
          "year": {"$gt": 1955},
          "year": {"$lt": 1975},
          “Tenant_id: {“$eq”: 23}
        }]
      },
      "queryVector": [-0.0072121937,-0.030757688,0.014948666,
                      ...
                      0.015741806,-0.020635074,-0.012945653],
      "numCandidates": 150,
      "limit": 10
    }
  }, {“$project”: {...}},...
]
```

# Vector Search Data Workflow




Documents returned as they're stored, typically remove vectors via $project.

# Vector Search enhanced Chatbots



github.com/alek-mdb/AtlasWorkshop

# Search Exercise 4 (Optional)

Self Serve Demo:

atlassearchrestaurants.com

github.com/mongodb-developer/WhatsCooking