Couchbase

Coding With AI: Vector Search and RAG

Matthew Groves, DevRel Engineer

Tyler Mitchell, Product Marketing Manager

June 2024

Agenda

Introduction to AI in development

Vector Search & Vector Embeddings

RAG

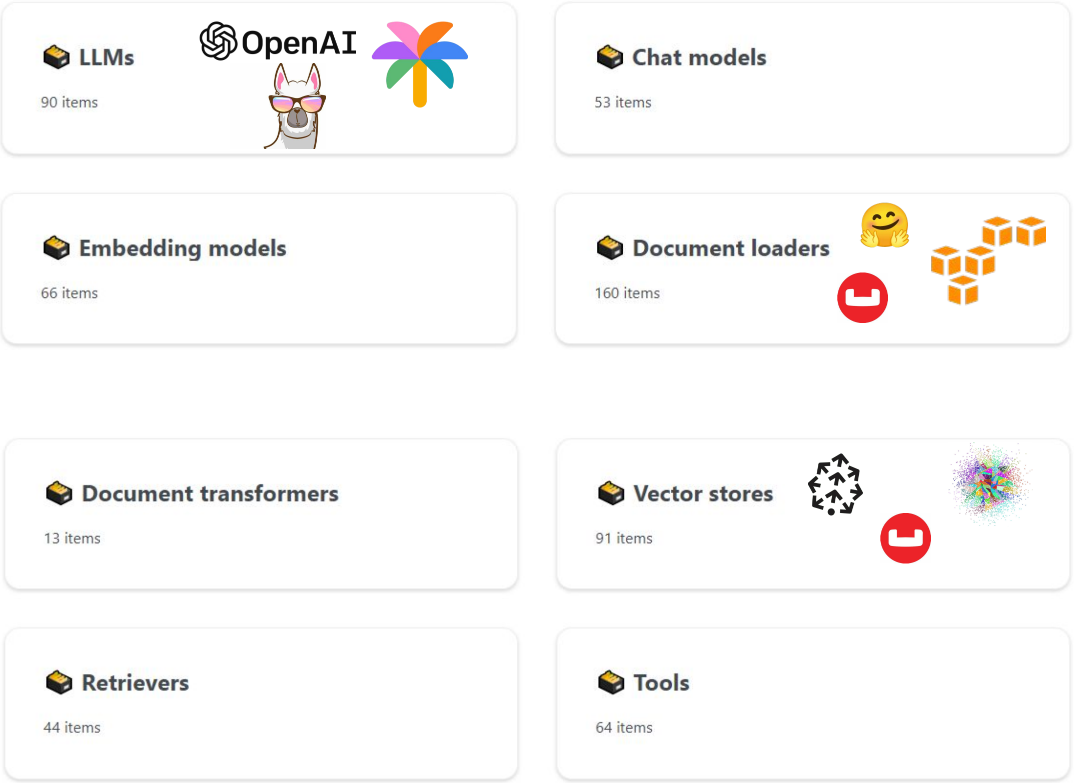
Hybrid Search

Q&A

Introduction to AI in Development

AI Landscape for Developers

Explosion of tools, models, frameworks



LLMs

90 items

Chat models

53 items

Embedding models

66 items

Document loaders

160 items

Document transformers

13 items

Vector stores

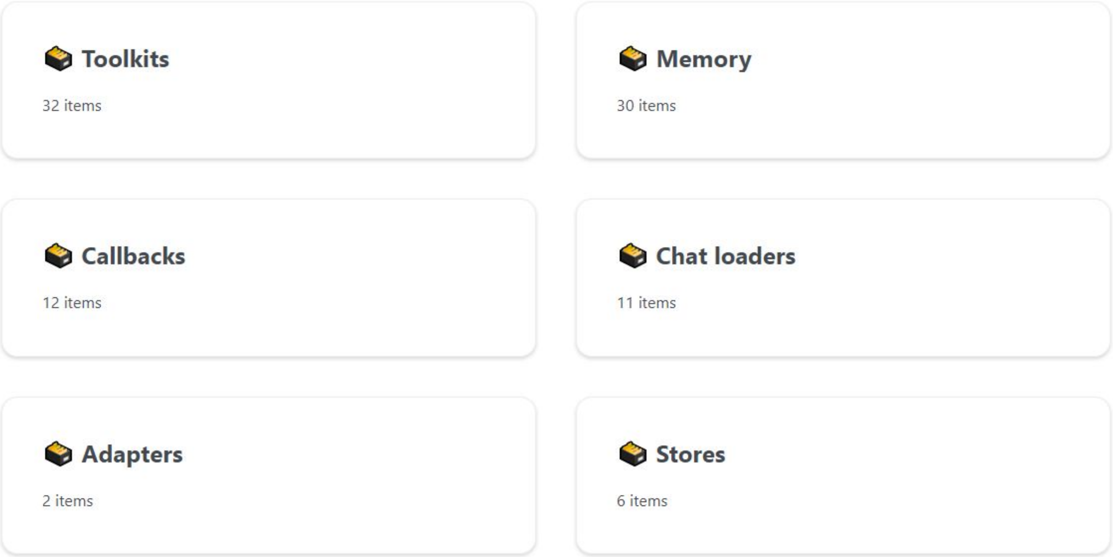
91 items

Retrievers

44 items

Tools

64 items



Toolkits

32 items

Memory

30 items

Callbacks

12 items

Chat loaders

11 items

Adapters

2 items

Stores

6 items

Common Development Challenges

What challenges are involved with AI?

Training LLMs

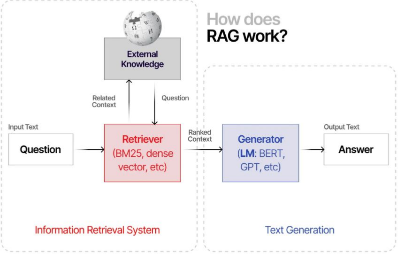
- Expensive, time-consuming, requires expertise, results not guaranteed

Content Generation

- Summaries, rewording, images, audio, video, code

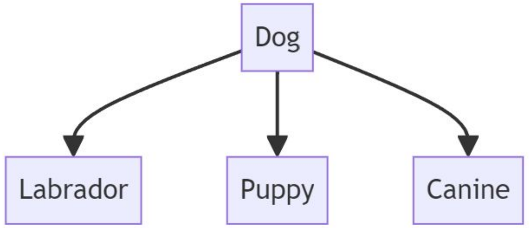
Chat

- How do I build in my own data into an LLM chat bot?



Semantic Search

- Find relevant results by meaning, not just matching letters



How can AI help?

Solutions and techniques

Vector Search

"Nearest neighbor" or "knn"

- Vector search can find the closest vectors to a given vector
- Your data gets embeddings
- Search text gets an embedding
- The nearest data to the search term is semantically similar

RAG

Retrieval-Augmented Generation

- Retrieve useful contextual data from your own enterprise
- Instead of building an LLM, use an existing one.
- Supply context along with question/command to an LLM to get more accurate, up-to-date results

LangChain + Couchbase

Chain together AI components

- Marshall data from multiple sources
- Store vectors in Couchbase, load documents from Couchbase
- Use LangChain as a central integration point for all AI components

Vector Embeddings & Vector Search

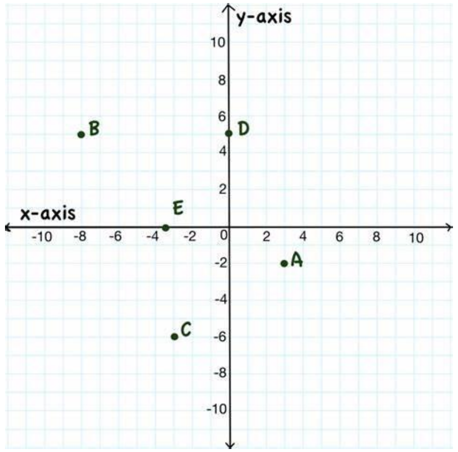
What is a Vector?

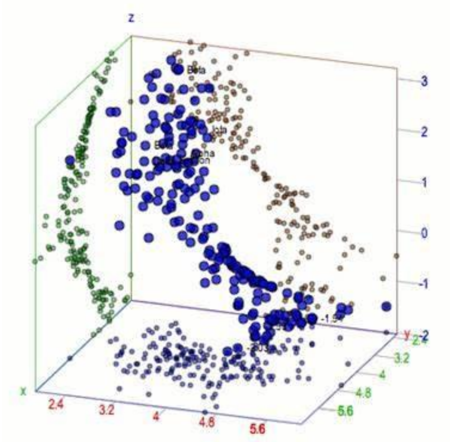
It's numbers correspond to data for which other nearby numbers can be located

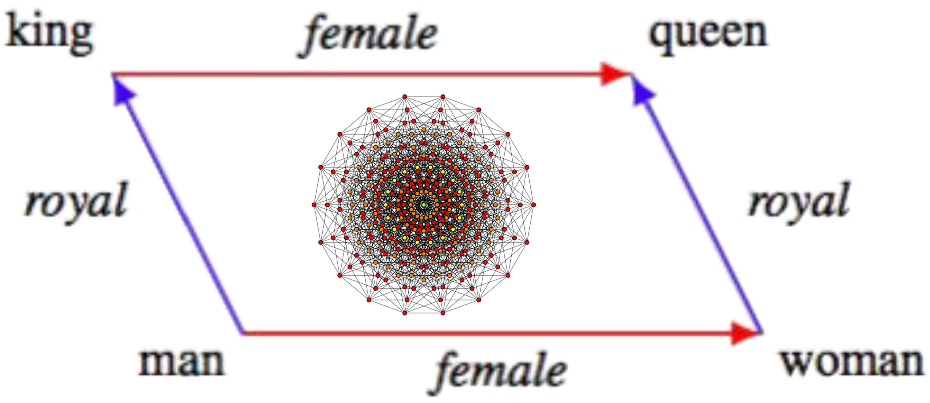
- `[48.51, 2.21]` is a lat/long vector for Paris

- `[124, 237, 246]` is a RGB value for light blue

- `[0.4532313, -1.243144345, 0.6781252319, 0.81234912, -0.33412356, ...]` is a GPT value for "Luke Skywalker joins forces with a Jedi Knight, a cocky pilot, a Wookiee and..."





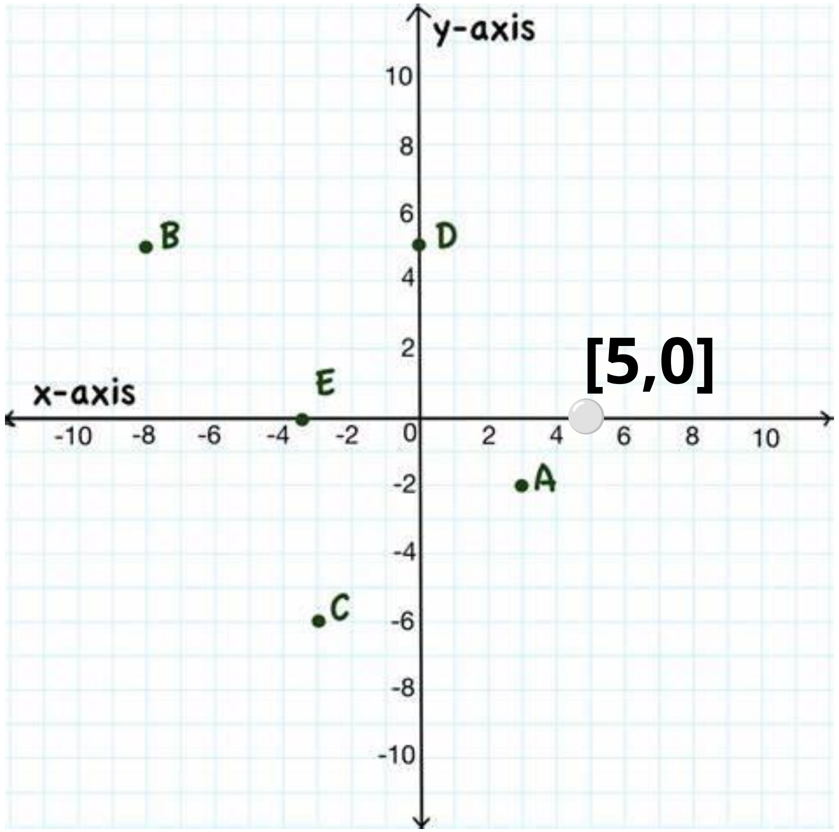


Vector Search: Nearest Neighbor

Sometimes called "knn"

- If we have a whole bunch of vectors in a database, and a vector generated from a user input, how do we find the most relevant data to their input?

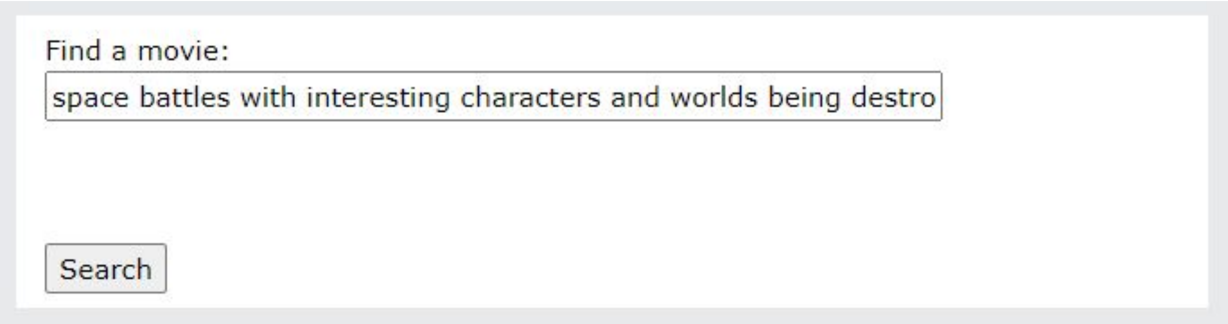
- Answer: Nearest Neighbor



Vectors in your database

Couchbase JSON example

```
{
    "movie": "Star Wars",
    "synopsis": "Luke Skywalker joins forces with a Jedi Knight, a cocky pilot, a
Wookiee and two droids to save the galaxy from the Empire's world-destroying battle
station, while also attempting to rescue Princess Leia from the mysterious Darth
Vader.",
    "synopsis_vector": [0.4532313, -1.243144345, 0.6781252319, 0.81234912, . . .]
}
```



Find a movie:

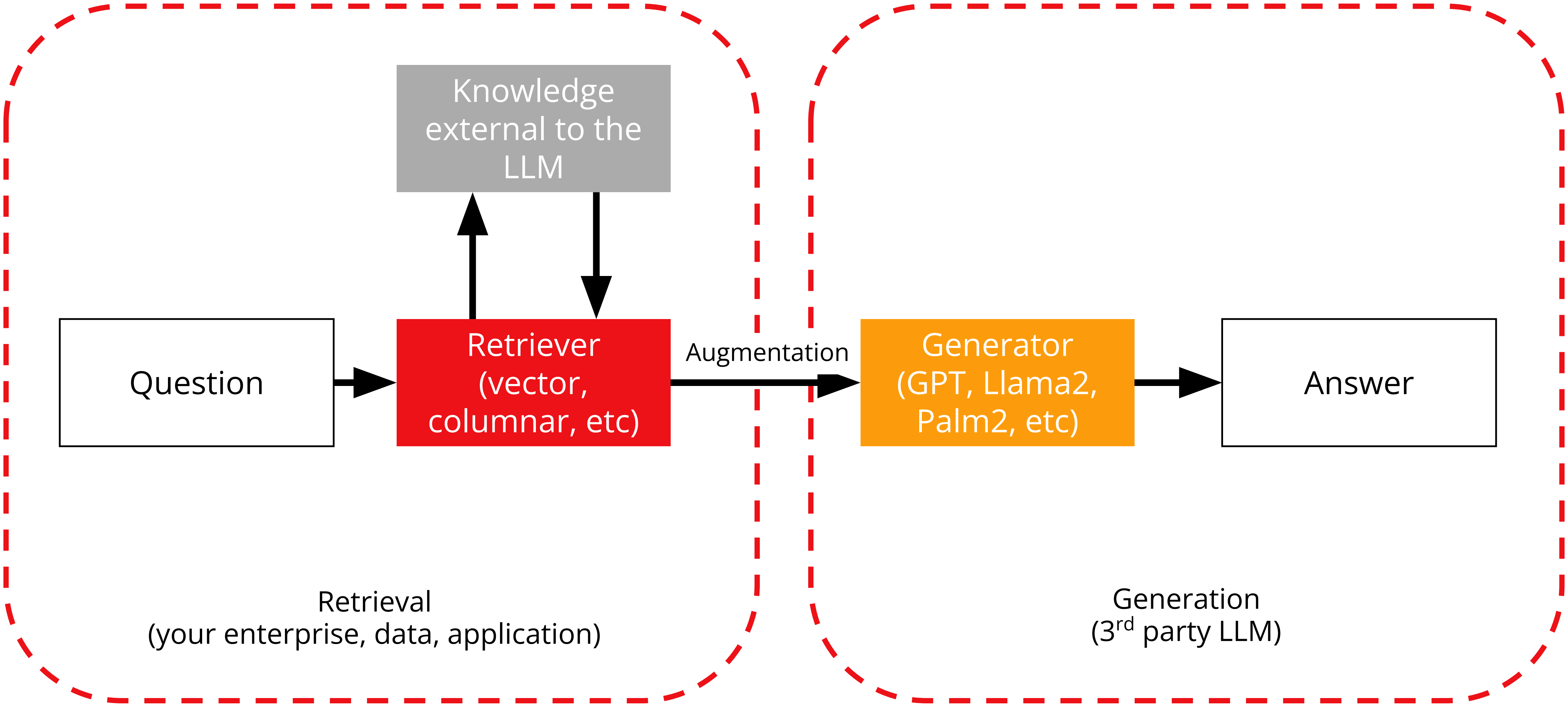
space battles with interesting characters and worlds being destro

RAG

Retrieval-Augmented Generation

What is Retrieval-Augmented Generation?

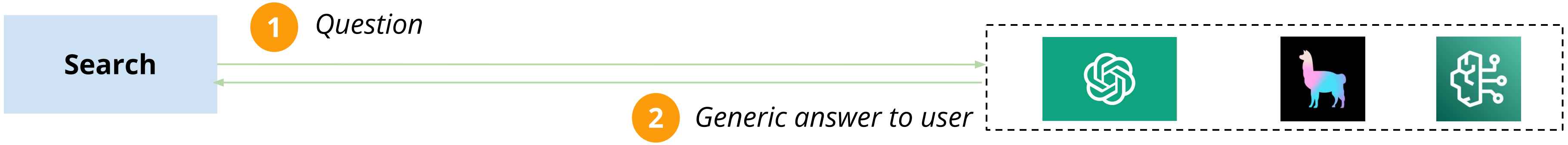
Supplying an existing LLM with relevant context

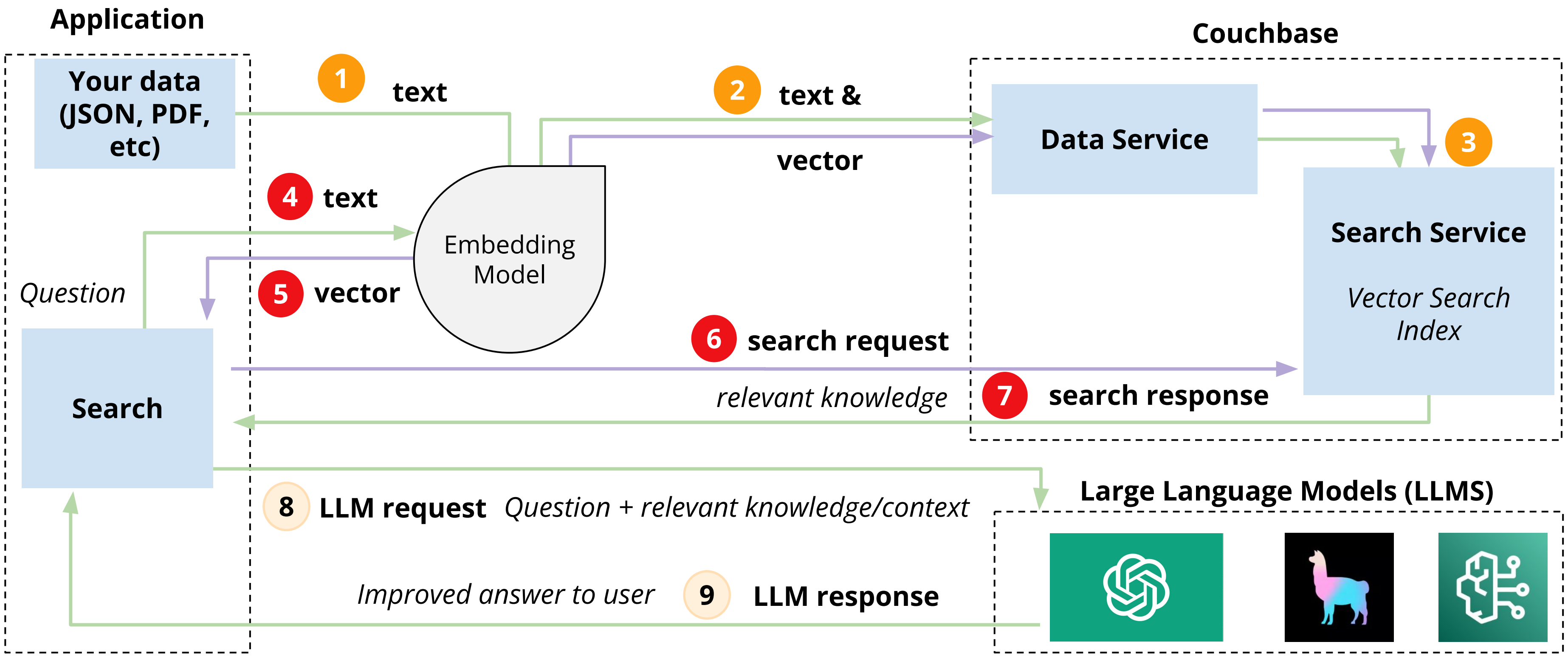

Knowledge external to the LLM
Question
Retriever (vector, columnar, etc)
Augmentation
Generator (GPT, Llama2, Palm2, etc)
Answer
Retrieval
(your enterprise, data, application)
Generation
(3rd party LLM)

3

RAG Application: Chatbot for Auto Parts Supplier Chain

***Question**: What is the best way to reduce muffler noise on my 1967 Ford Bronco?*





1. Application sends the relevant knowledge base or corpus of documents to Embedding Model adds vectors to JSON documents
2. Uploads JSON documents (text with vector) to the Data Service to load corpus.
3. Search Service receives mutations (upsert or delete) JSON document. maintains indexes across partitions & nodes.
4. Application sends “question” to Embedding Model
5. Embedding Model returns embedding Vector
6. Application calls Search Service with a Vector search with a “question”.
7. Search Service returns top k-NN results of query with requested fields typical text or semantically similar documents. This is the relevant knowledge base.
8. Application combines the initial question with the relevant knowledge and sends the combined information as a prompt to a LLM
9. Foundation Models create factual answer based on relevant knowledge for the original question.

Demo: RAG

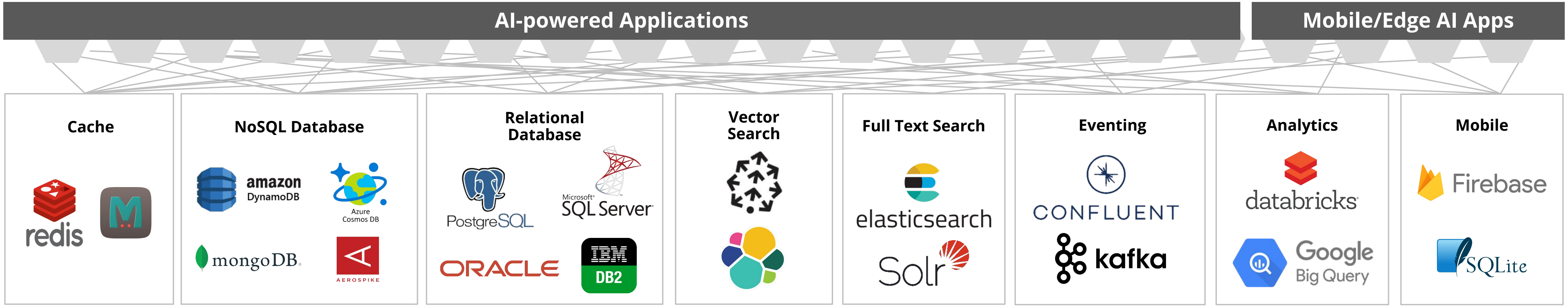
Implementing RAG with Couchbase vector search and LangChain

Hybrid Search

3

Data Sprawl and Vectors

Architectures built on purpose-built databases: Complexity & data sprawl



AI-powered Applications

Mobile/Edge AI Apps

Cache

NoSQL Database

Relational Database

Vector Search

Full Text Search

Eventing

Analytics

Mobile

Separate platforms with multiple interfaces

1. Introducing latency and AI confusion
2. Independent deployment and management
3. Different data model and programming interfaces
4. Integration between multiple products
5. Debugging and data redundancy challenges

Per product factors (Financial, time, & effort)

1. License & agreement
   - Sourcing for renewals
   - Legal for agreements
2. Training
   - Developers
   - Operations
3. Support

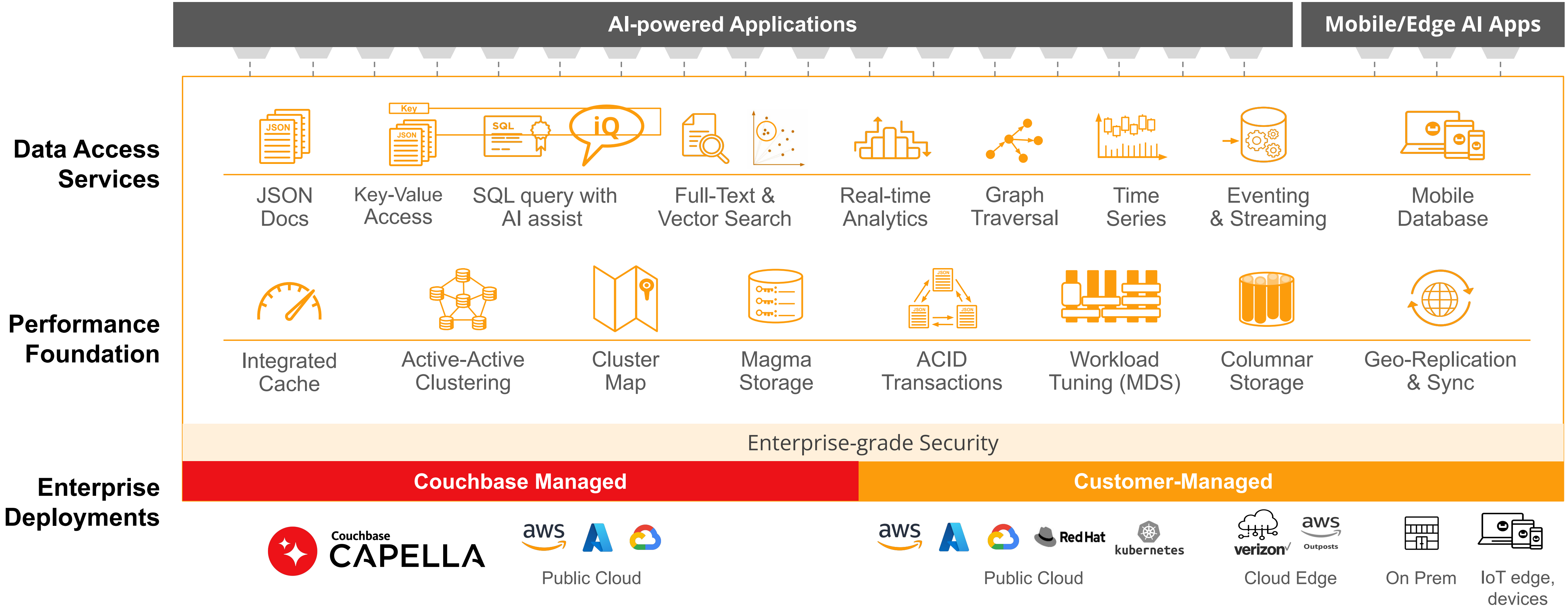
4. Build API or connector to database
5. Security
6. Purchase infrastructure

COST

- Infrastructure
- Licenses
- Integration
- Training
- Operational
- Support costs

Reducing Data Sprawl with Couchbase

Innovate Faster with Couchbase



AI-powered Applications

Mobile/Edge AI Apps

Data Access Services

JSON Docs

Key-Value Access

SQL query with AI assist

Full-Text & Vector Search

Real-time Analytics

Graph Traversal

Time Series

Eventing & Streaming

Mobile Database

Performance Foundation

Integrated Cache

Active-Active Clustering

Cluster Map

Magma Storage

ACID Transactions

Workload Tuning (MDS)

Columnar Storage

Geo-Replication & Sync

Enterprise-grade Security

Enterprise Deployments

Couchbase Managed

Customer-Managed

Couchbase CAPELLA

aws

Public Cloud

aws

Red Hat

kubernetes

Public Cloud

verizon

aws Outposts

Cloud Edge

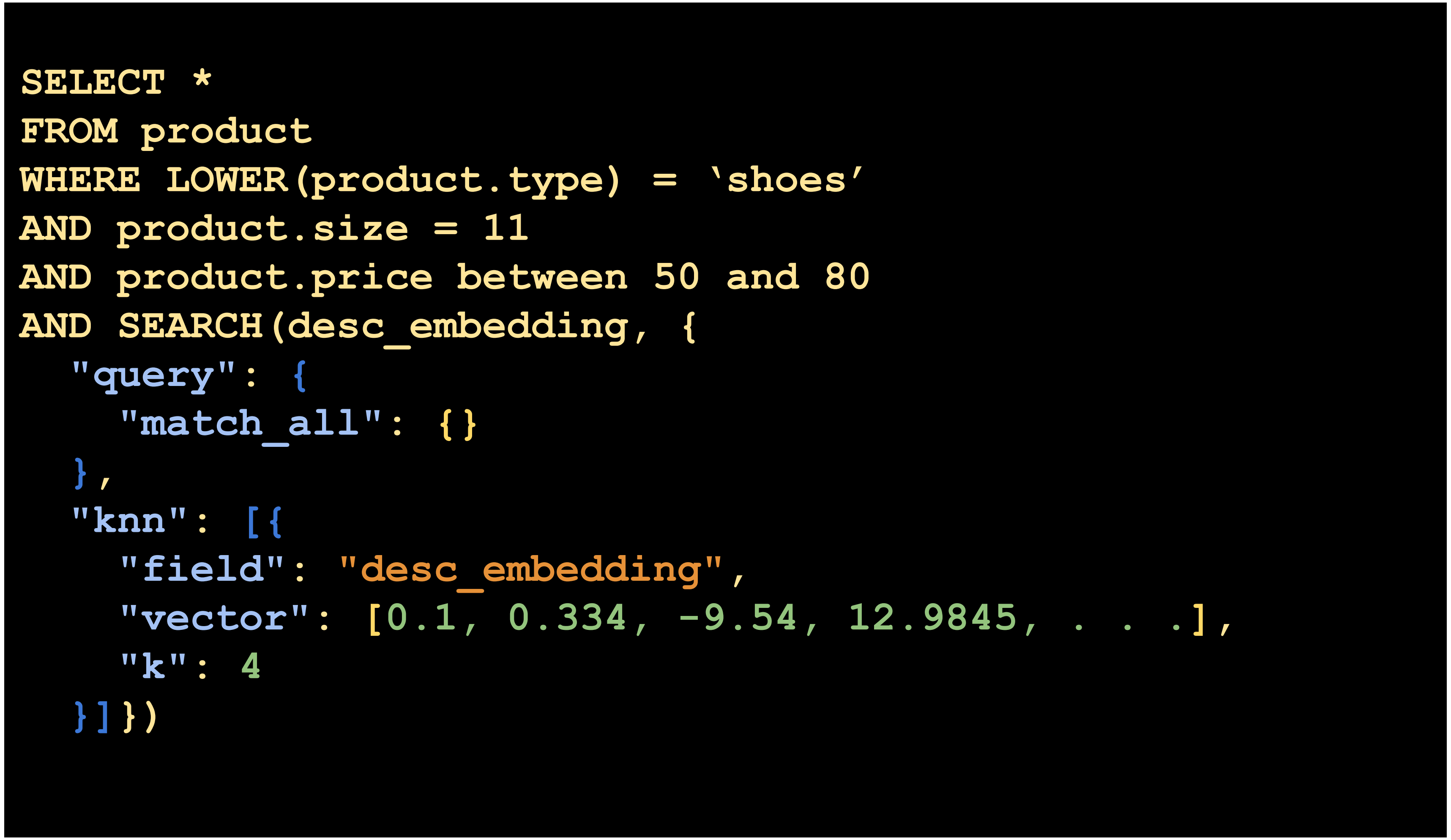
On Prem

IoT edge, devices

Hybrid Search with SQL++

```
SELECT *
FROM product
WHERE LOWER(product.type) = ‘shoes’
AND product.size = 11
AND product.price between 50 and 80
```

Hybrid Search with SQL++



```
SELECT *
FROM product
WHERE LOWER(product.type) = ‘shoes’
AND product.size = 11
AND product.price between 50 and 80
AND SEARCH(desc_embedding, {
  "query": {
    "match_all": {}
  },
  "knn": [{
    "field": "desc_embedding",
    "vector": [0.1, 0.334, -9.54, 12.9845, . . .],
    "k": 4
  }]})
```

Beyond Vector Search: Hybrid Search

Building real-world use cases

Customer wants new shoes

- Match color and style of an object (semantic/vector)
- Description to mention “casual” (text/fuzzy)
- Price between $100 and $200 (range)
- With rating over 4.5 stars (range)
- Within 15 miles (geospatial)
- Available today in stores (inventory)

Your next steps

- Sign up for Couchbase Capella
  - https://www.couchbase.com/products/capella
  - test drive CTA

- Check out demos
  - Chat with PDF
    - Python: https://github.com/couchbase-examples/rag-demo
    - Node.JS: https://github.com/couchbase-examples/vector-search-nodejs
  - Hybrid Movies Search: https://github.com/couchbase-examples/hybrid-search-demo
  - Q&A Chatbot: https://github.com/couchbase-examples/qa-bot-demo

- Vector Search for mobile
  - https://www.couchbase.com/blog/vector-search-at-the-edge-with-couchbase-mobile/

Q&A

Thank you!

tyler.mitchell@couchbase.com

https://www.linkedin.com/in/tylermitchell

@1tylermitchell

matthew.groves@couchbase.com

https://www.linkedin.com/in/mgroves/

@mgroves

Couchbase