WELCOME

How MongoDB and Voyage AI Build Your Agentic System

Ainara Miller-Askar
Solutions Architect

Hello from a full-time working parent

Bring to school

WORK

Pick up from school

Order bigger shoes

Order books & development stuff

Quality time with my daughter

WORK

Clean up home

Cook dinner

WORK

Order bigger outfit for the dance contest

WORK

Schedule pediatrician

Do groceries

1:1 with teachers at school

Quality time with my partner

WORK

Plan school holidays

Good-night story

Rest of the family? Friends?

ME TIME?!?

Hello from a full-time working parent

Bring to school

WORK

Pick up from school

Order bigger shoes

Order books & development stuff

Quality time with my daughter

~~WORK~~

Clean up home

Cook dinner

WORK

Order bigger outfit for the dance contest

~~WORK~~

Schedule pediatrician

Do groceries

1:1 with teachers at school

Quality time with my partner

~~WORK~~

Plan school holidays

Good-night story

Rest of the family? Friends?

ME TIME?!?

Hello from a full-time working parent

Bring to school

WORK

Pick up from school

Order bigger shoes

Order books & development stuff

Quality time with my daughter

WORK

Clean up home

WORK

Cook dinner

Order bigger outfit for the dance contest

WORK

Schedule pediatrician

Do groceries

1:1 with teachers at school

Quality time with my partner

WORK

Plan school holidays

Good-night story

Rest of the family? Friends?

ME TIME?!?

Hello from a full-time working parent

Bring to school

WORK

Pick up from school

Order bigger shoes

Order books & development stuff

Quality time with my daughter

WORK

Clean up home

Cook dinner

WORK

Order bigger outfit for the dance contest

WORK

Schedule pediatrician

Do groceries

1:1 with teachers at school

Quality time with my partner

WORK

Plan school holidays

Good-night story

Rest of the family? Friends?

ME TIME?!?

Hello from a full-time working parent

Bring to school

WORK

Pick up from school

Order bigger shoes

Order books & development stuff

Quality time with my daughter

WORK

Clean up home

Cook dinner

WORK

Order bigger outfit for the dance contest

WORK

Schedule pediatrician

Do groceries

1:1 with teachers at school

Quality time with my partner

WORK

Plan school holidays

Good-night story

Rest of the family? Friends?

ME TIME?!?

Hello from a full-time working parent

Bring to school

WORK

Quality time with my daughter

Pick up from school

Order bigger shoes

Order books & development stuff

WORK

Clean up home

Cook dinner

WORK

Order bigger outfit for the dance contest

WORK

Schedule pediatrician

Do groceries

1:1 with teachers at school

Quality time with my partner

WORK

Plan school holidays

Good-night story

Rest of the family? Friends?

ME TIME?!?

Hello from a full-time working parent

Bring to school

WORK

Quality time with my daughter

Order books & development stuff

Cook dinner

WORK

Schedule pediatrician

1:1 with teachers at school

Quality time with my partner

Good-night story

Pick up fro

Multi-functional family and household Agentic System?

ORK

Do groceries

WORK

Plan school holidays

Rest of the family? Friends?

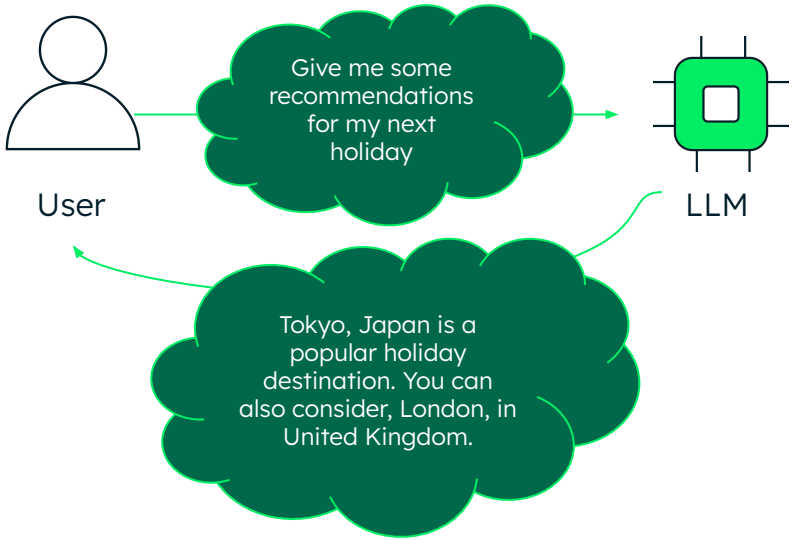
ME TIME?!?

Form Factor Evolution

LLM Powered Chatbots

Knowledge in models used to respond to queries

Traditional Chatbots to LLM Powered Chatbots



Form Factor Evolution

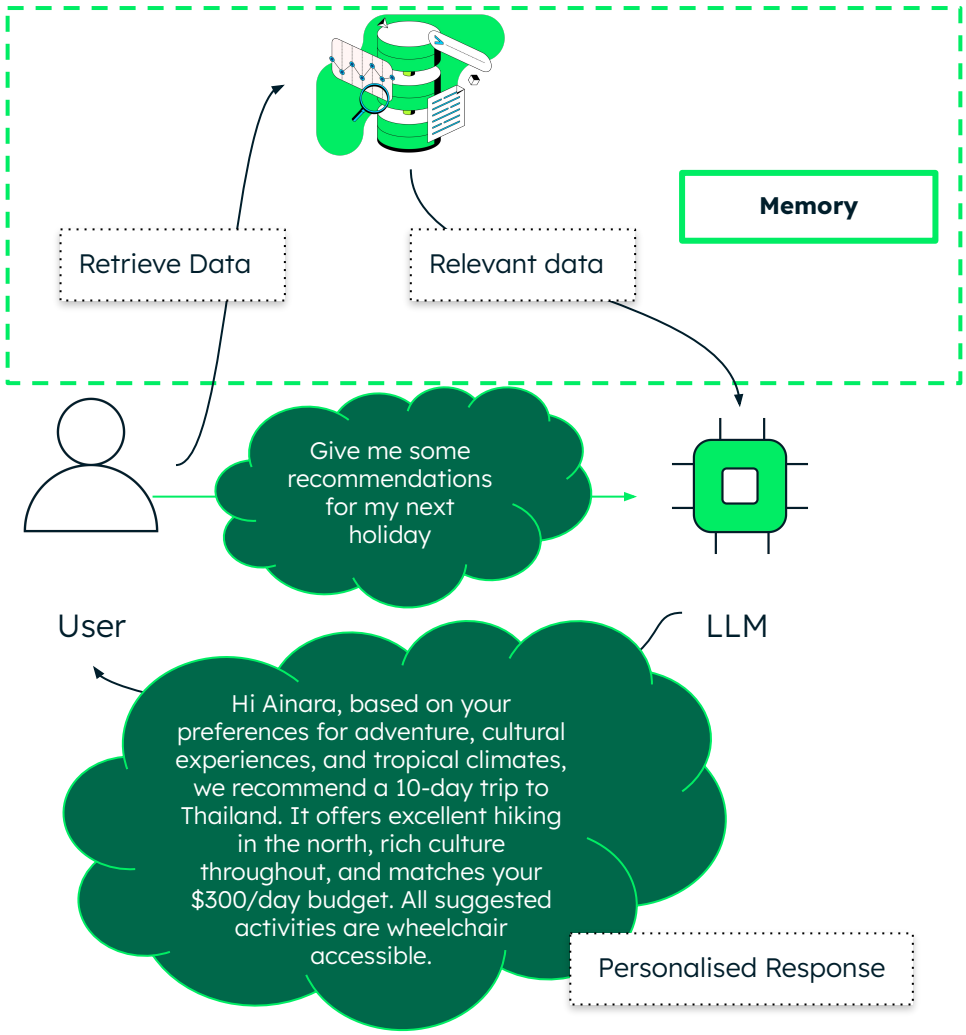
LLM Powered Chatbots

Knowledge in models used to respond to queries

RAG Chatbots

Knowledge in models supplemented with external data

LLM Powered Chatbots to RAG Chatbots



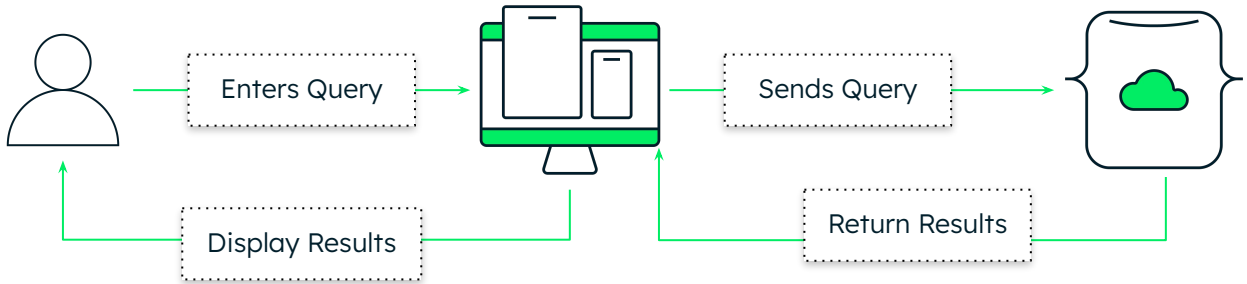
Challenges

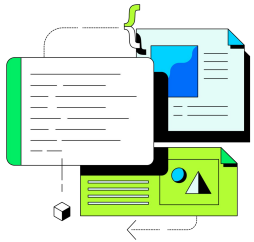
Provide the right information to LLM

MongoDB Atlas Search

$search

“artificial intelligence”

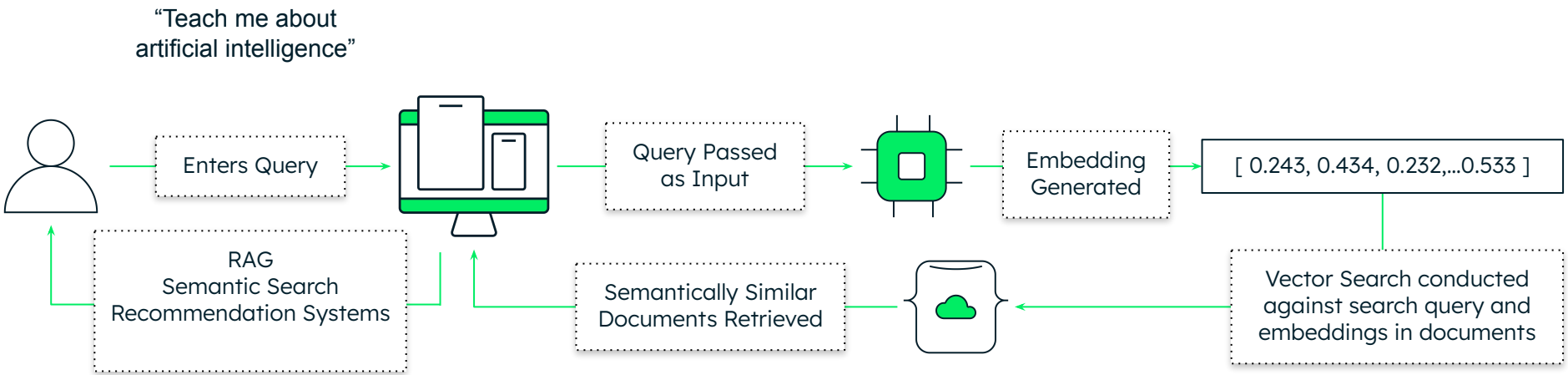




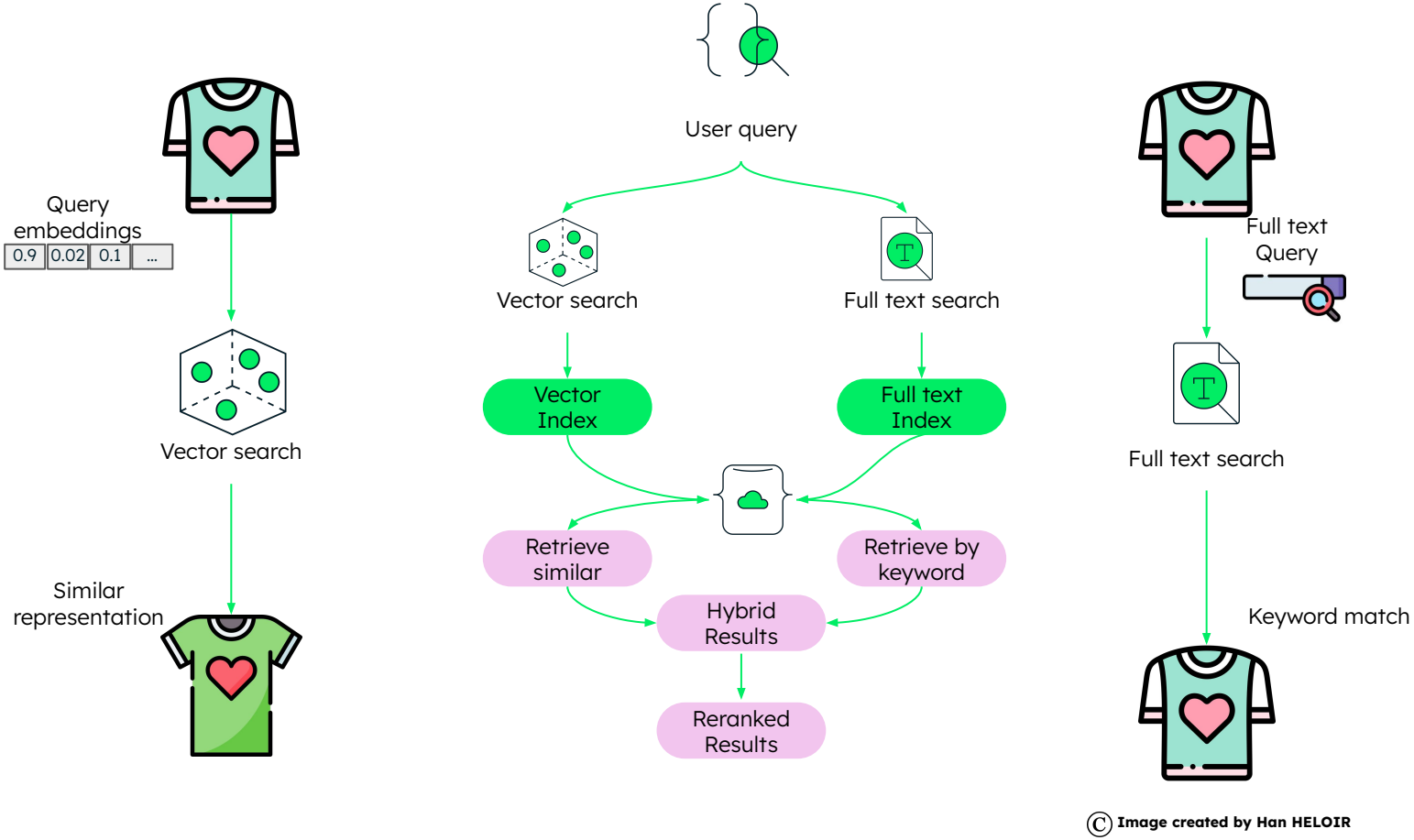
[{ "title": "Advanced Artificial Intelligence: Concepts and Applications", "author": "Sarah Johnson", "year": 2022, "score": 9.8 }, { "title": "Machine Learning and AI in Business", "author": "Michael Chen", "year": 2019, "score": 8.5 }, { "title": "The Future of Intelligence: Artificial and Beyond", "author": "Emily Rodriguez", "year": 2021, "score": 7.9 }, { "title": "AI in Healthcare: Revolutionizing Patient Care", "author": "Dr. Robert Williams", "year": 2020, "score": 7.2 }]

MongoDB Atlas Vector Search

$vectorSearch

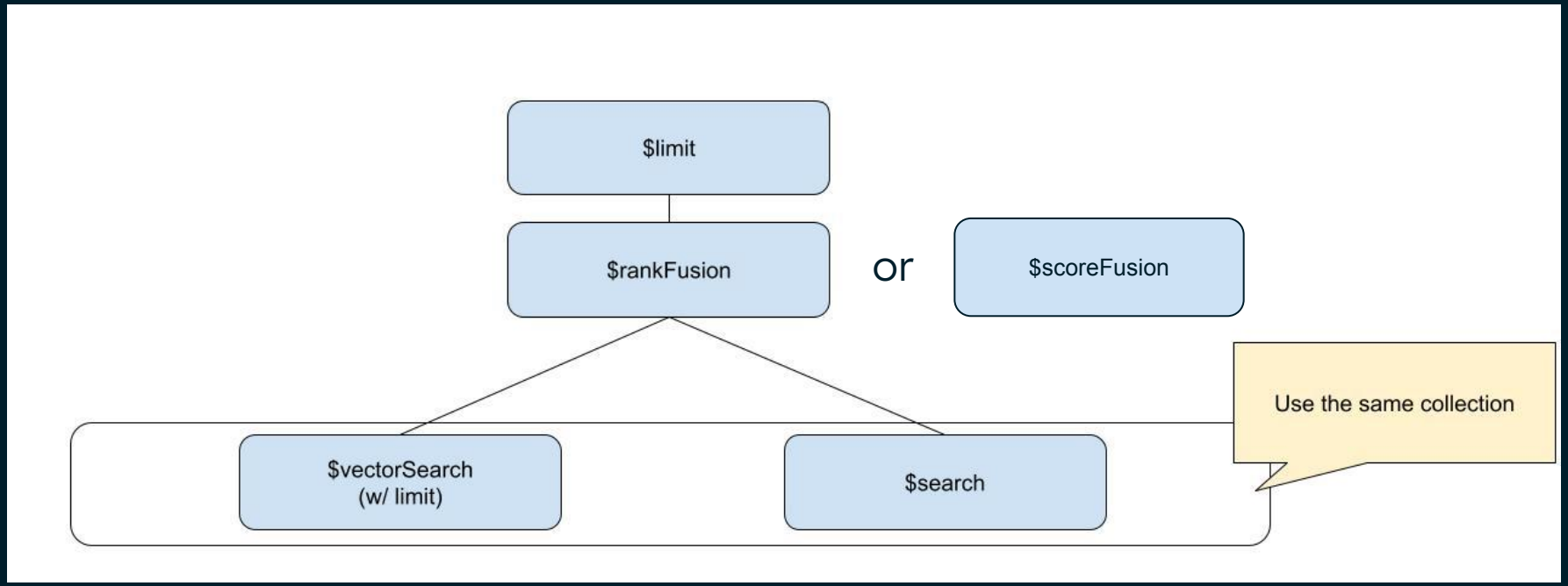


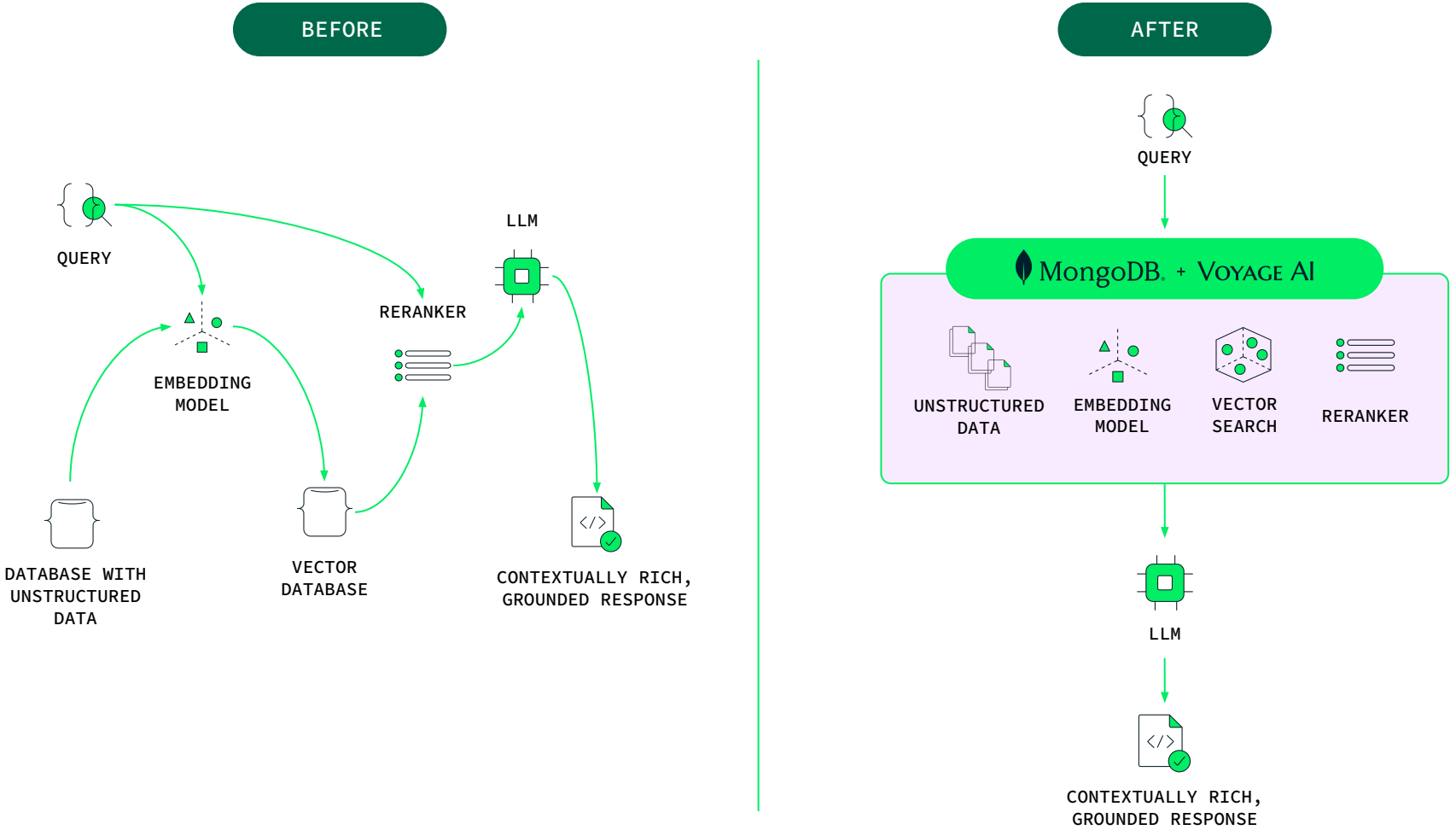
[{ "title": "Advanced Artificial Intelligence: Concepts and Applications", "author": "Sarah Johnson", "year": 2022, "score": 9.8 }, { "title": "Machine Learning and AI in Business", "author": "Michael Chen", "year": 2019, "score": 8.5 }, { "title": "The Future of Intelligence: Artificial and Beyond", "author": "Emily Rodriguez", "year": 2021, "score": 7.9 }, { "title": "AI in Healthcare: Revolutionizing Patient Care", "author": "Dr. Robert Williams", "year": 2020, "score": 7.2 }]



© Image created by Han HELOIR

Hybrid Search via $rankFusion or $scoreFusion




BEFORE
QUERY
LLM
RERANKER
EMBEDDING MODEL
DATABASE WITH UNSTRUCTURED DATA
VECTOR DATABASE
CONTEXTUALLY RICH, GROUNDED RESPONSE
AFTER
QUERY
MongoDB. + Voyage AI
UNSTRUCTURED DATA
EMBEDDING MODEL
VECTOR SEARCH
RERANKER
LLM
CONTEXTUALLY RICH, GROUNDED RESPONSE

Expanding Voyage AI's model offering with contextualized chunk embeddings

Embedding Models

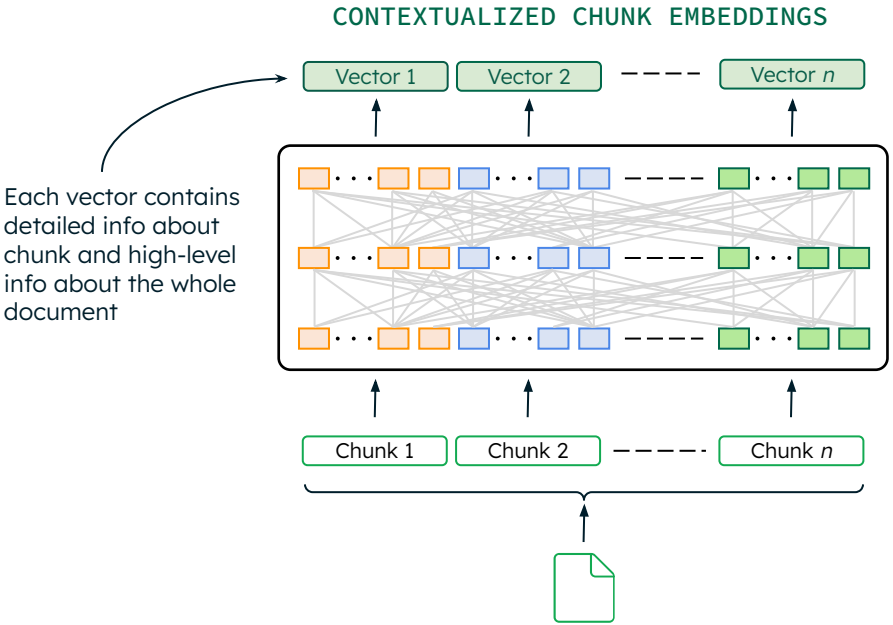
General-Purpose	Domain-Specific
Text Lite | Standard | Large	Code
Multimodal	Legal
Contextualized Chunk	Finance

Rerankers

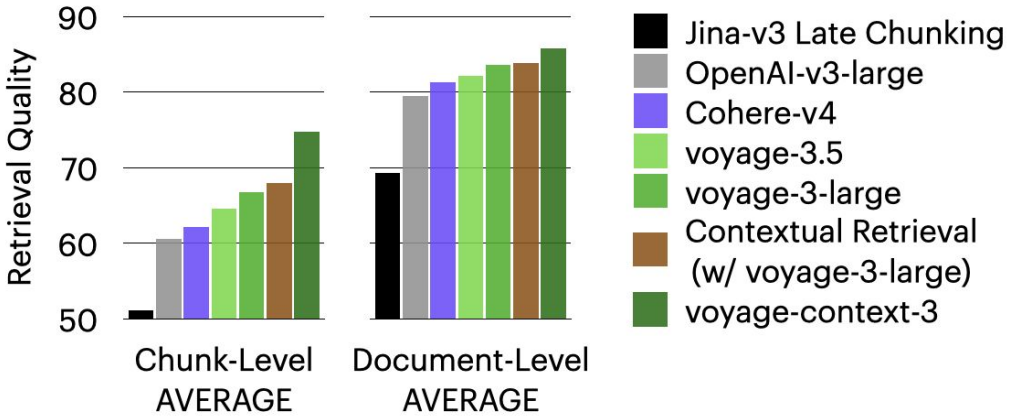
Text
Lite | Standard

voyage-context-3

voyage-context-3: A breakthrough in precise chunk retrieval with global document context

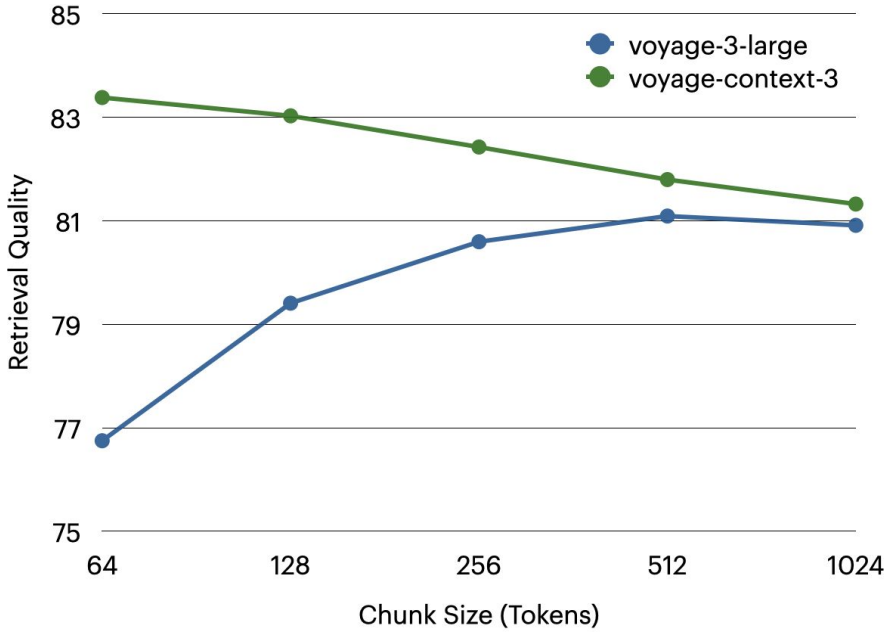


Contextualized chunk embeddings deliver superior retrieval performance



voyage-context-3

Reduced sensitivity to chunking size

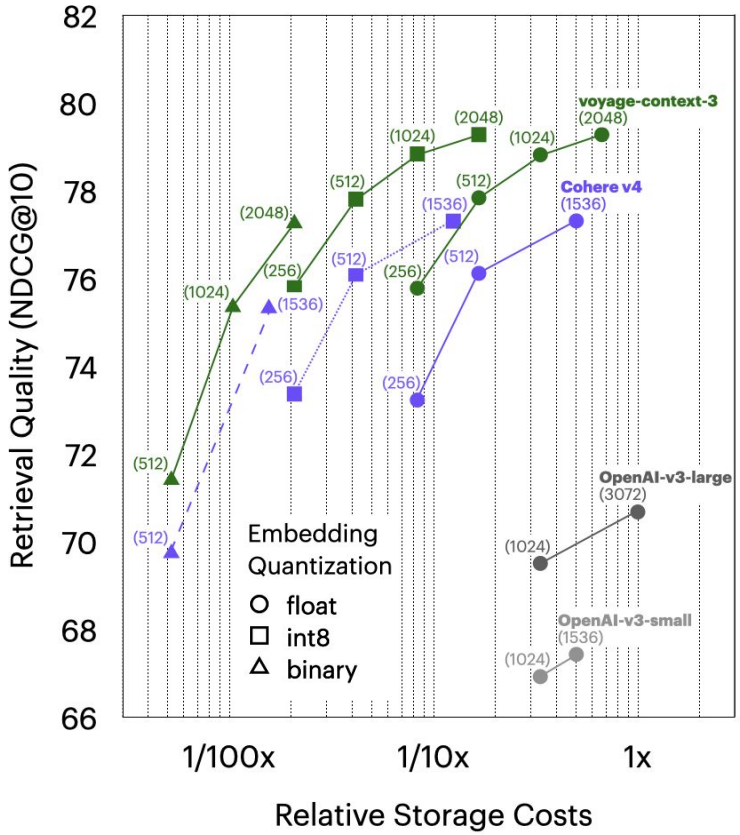


voyage-context-3

Minimizing quality loss across quantization options

Compared to OpenAI-v3-large (float, 3072), **`voyage-context-3`**:

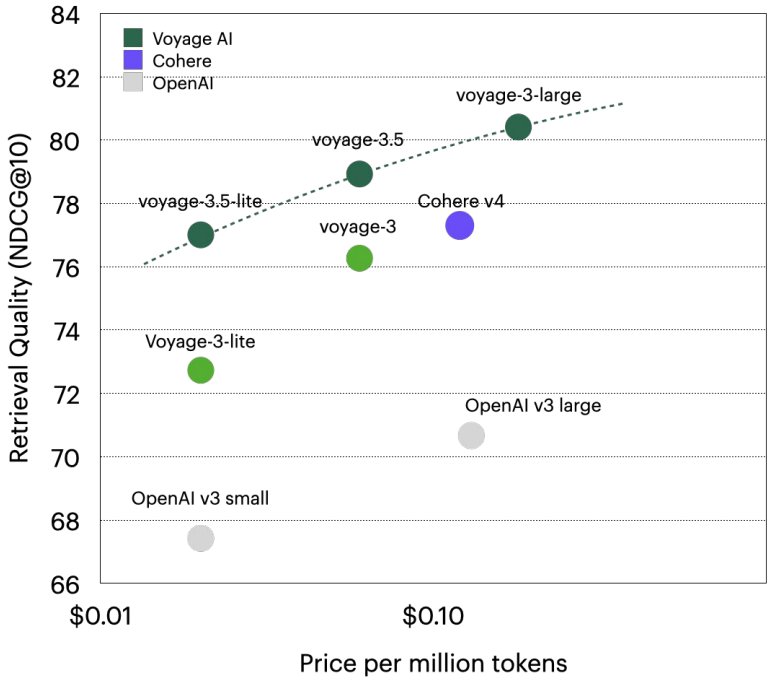
- int8 with 2048 dims: **reduces vector db costs by 83%** with **8.60%** better retrieval quality.
- binary with 512 dims: reduces vector db costs by 99.48% with 0.73% better retrieval quality—that's virtually the **same retrieval performance at 0.5% of the cost.**



`voyage-context-3`

voyage-3.5 and voyage-3.5-lite

voyage-3.5 and voyage-3.5-lite
New frontier for price-performance

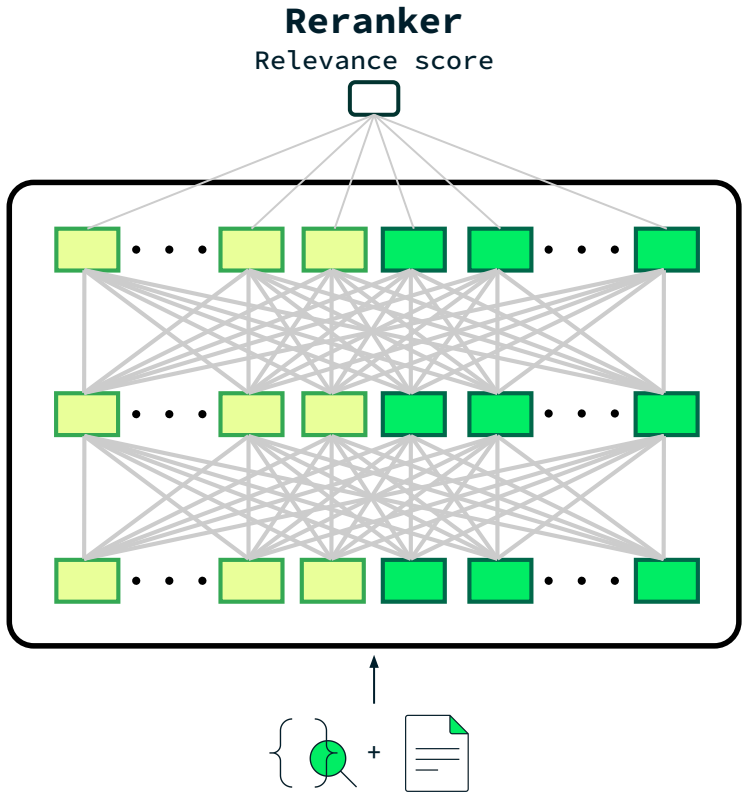


rerank-2.5 & 2.5-lite

What are rerankers and how do they work?

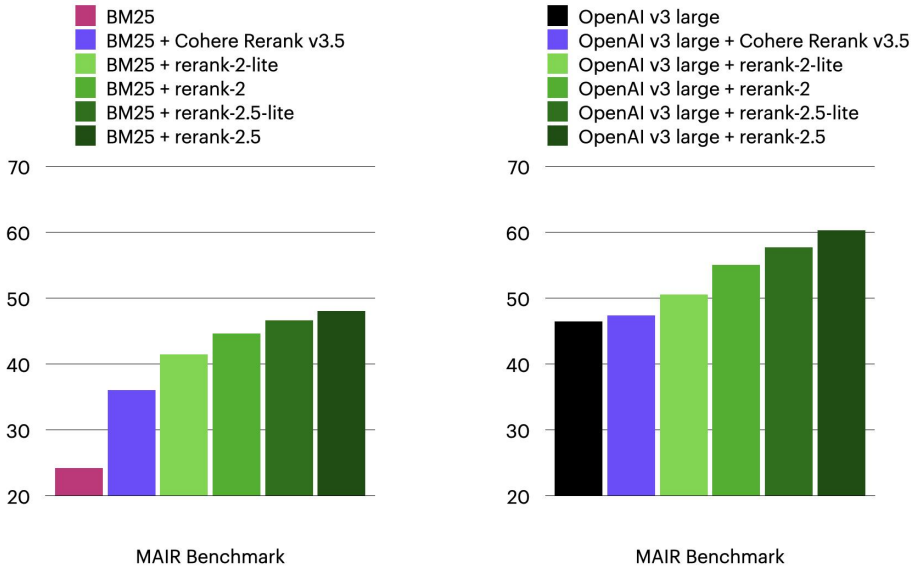
Rerankers (or reranking models) are cross-encoder models that assign relevance scores to a query and retrieved documents.

Given a query and a set of documents, rerankers return a rank for each document per its relevance to the query. This process can help improve the relevance of semantic search results.

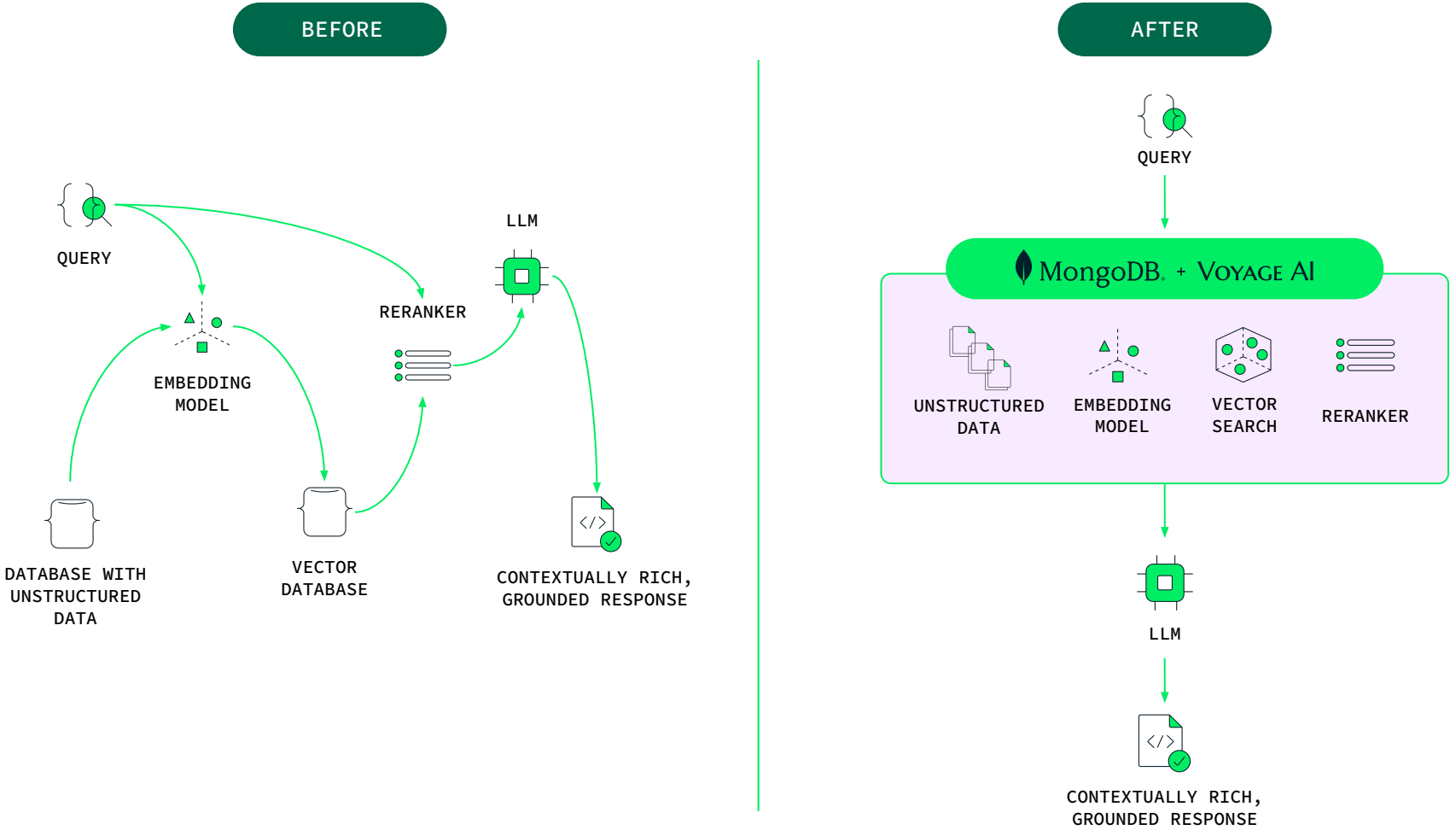


MAIR benchmark performance

rerank-2.5 and rerank-2.5-lite outperform competitors



rerank-2.5 & 2.5-lite


BEFORE
QUERY
LLM
RERANKER
EMBEDDING MODEL
DATABASE WITH UNSTRUCTURED DATA
VECTOR DATABASE
CONTEXTUALLY RICH, GROUNDED RESPONSE
AFTER
QUERY
MongoDB. + Voyage AI
UNSTRUCTURED DATA
EMBEDDING MODEL
VECTOR SEARCH
RERANKER
LLM
CONTEXTUALLY RICH, GROUNDED RESPONSE

Form Factor Evolution

LLM Powered Chatbots

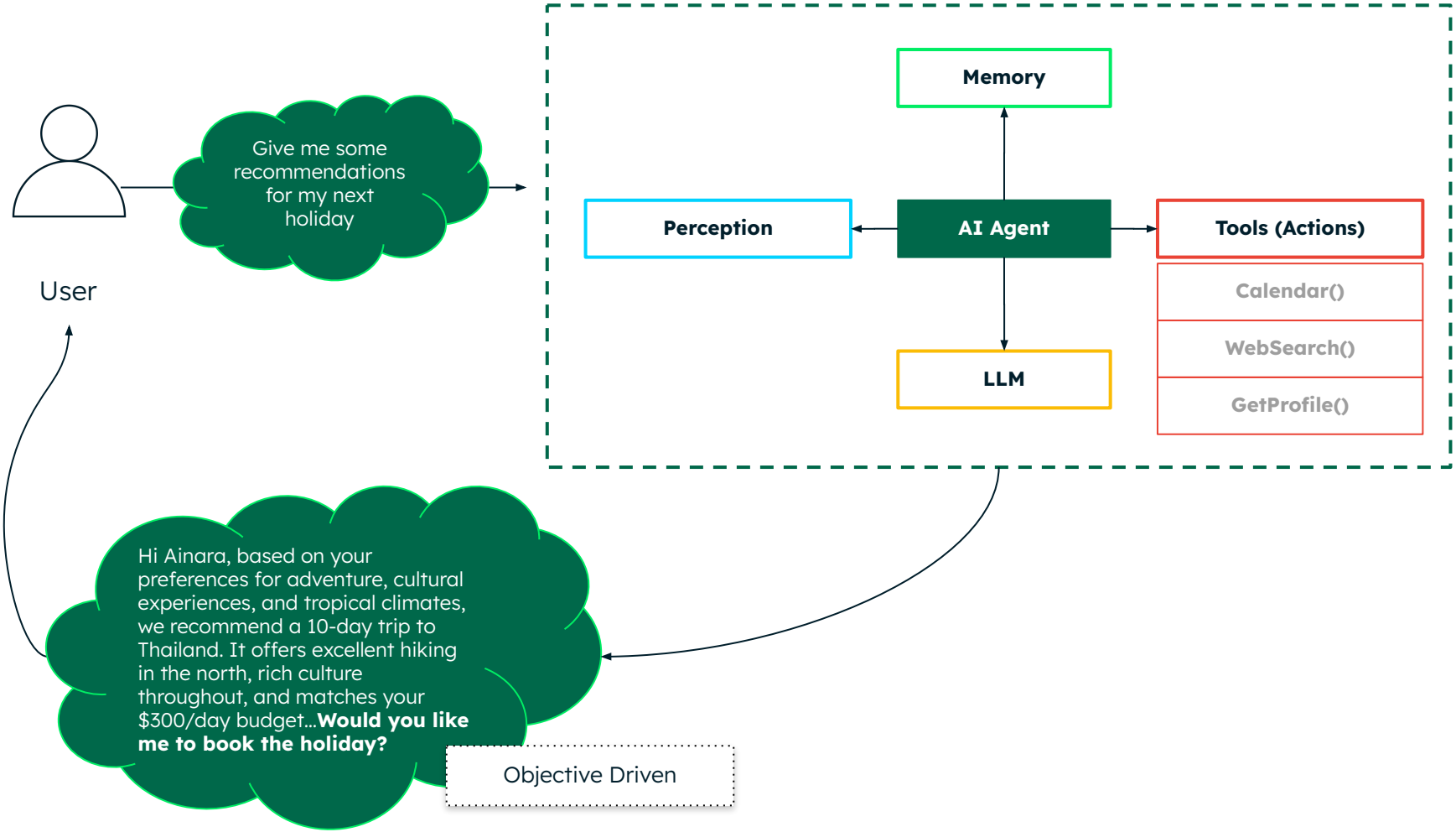
Knowledge in models used to respond to queries

RAG Chatbots

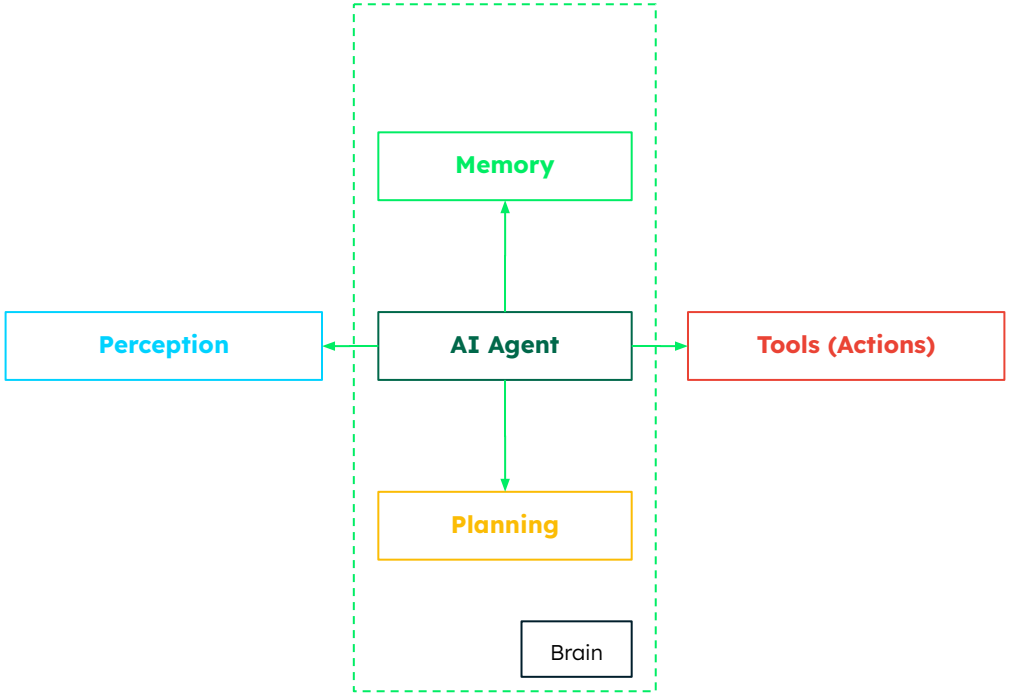
Knowledge in models supplemented with external data

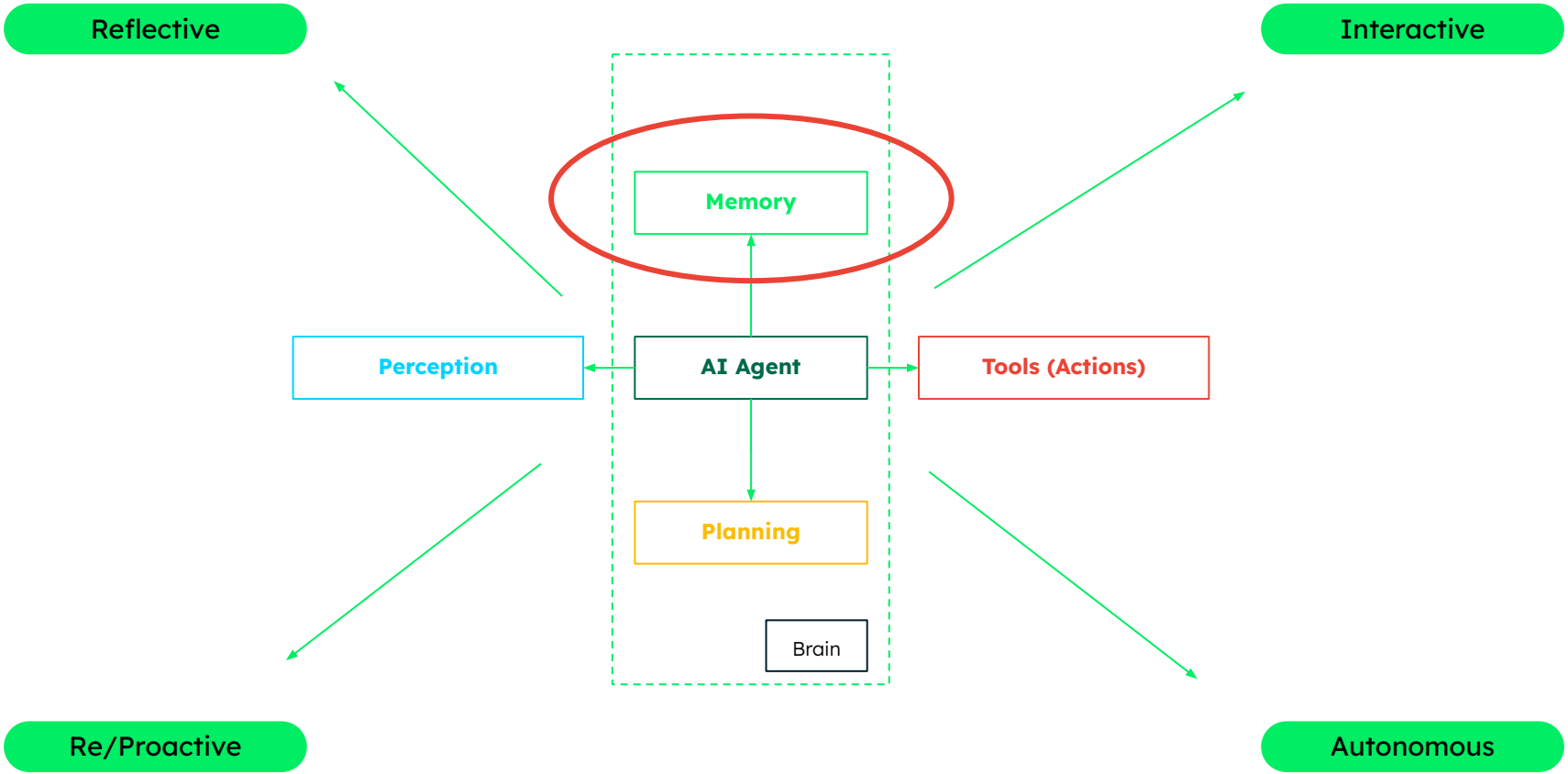
AI Agents

LLMs with tools use capabilities and advanced reasoning and planning capabilities

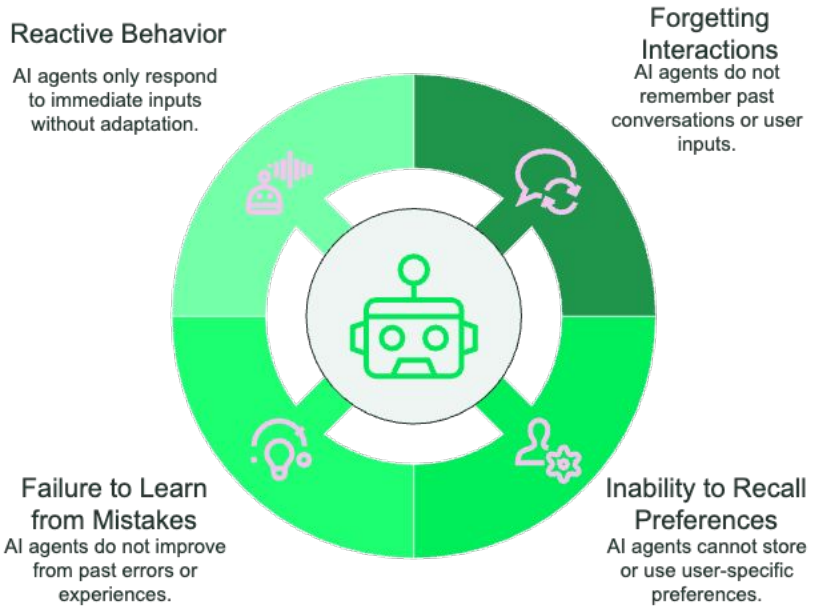

Give me some recommendations for my next holiday
User
Memory
Perception
AI Agent
Tools (Actions)
Calendar()
WebSearch()
GetProfile()
LLM
Hi Ainara, based on your preferences for adventure, cultural experiences, and tropical climates, we recommend a 10-day trip to Thailand. It offers excellent hiking in the north, rich culture throughout, and matches your $300/day budget...**Would you like me to book the holiday?**
Objective Driven

Form Factor Evolution


Memory
Perception
AI Agent
Tools (Actions)
Planning
Brain


Reflective
Interactive
Memory
Perception
AI Agent
Tools (Actions)
Planning
Brain
Re/Proactive
Autonomous

Why Building Truly Intelligent Agents Is So Hard?



Modern AI agents may seem smart, but most are stateless.

The result?

- Repetitive conversations
- High token and compute costs
- No long-term personalization

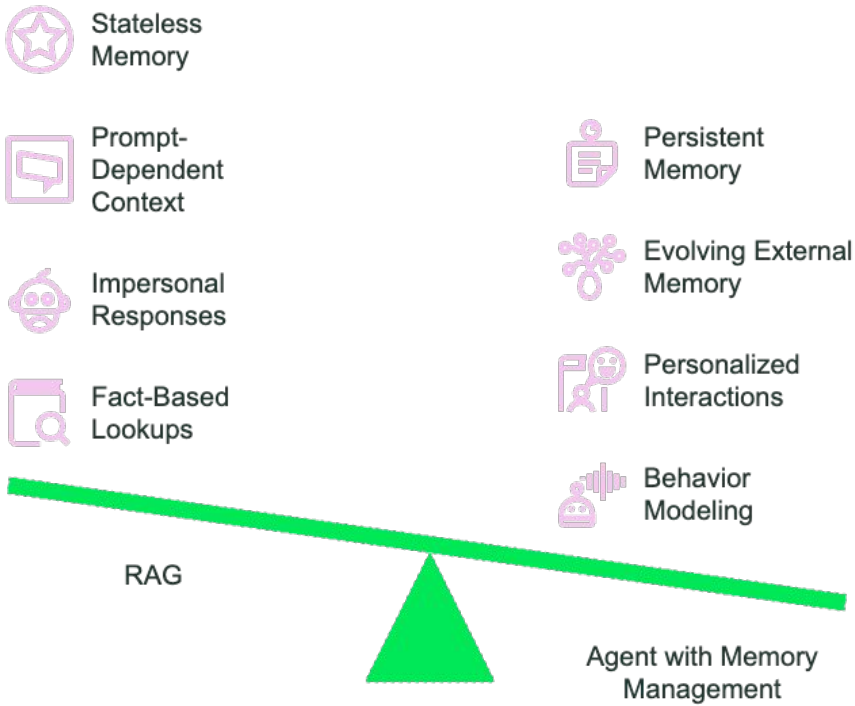
MongoDB Memory + RAG: Complementary Power

RAG (Retrieval-Augmented Generation) is excellent for fetching knowledge from documents, but it is **stateless.**

It lacks memory of **who** the user is, **what** they asked, or **why** something failed last time.

MongoDB Memory solution bridges that gap by enabling:

- **Continuity** across sessions
- **Personalized decision-making**
- **Behavioral memory**, not just factual recall



MongoDB Memory for AI agents

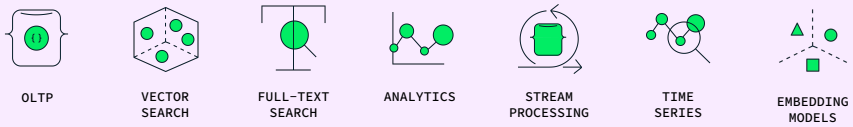
Memory Types

- Short Term
- Long Term
- Procedural
- Taxonomic

Agentic Frameworks

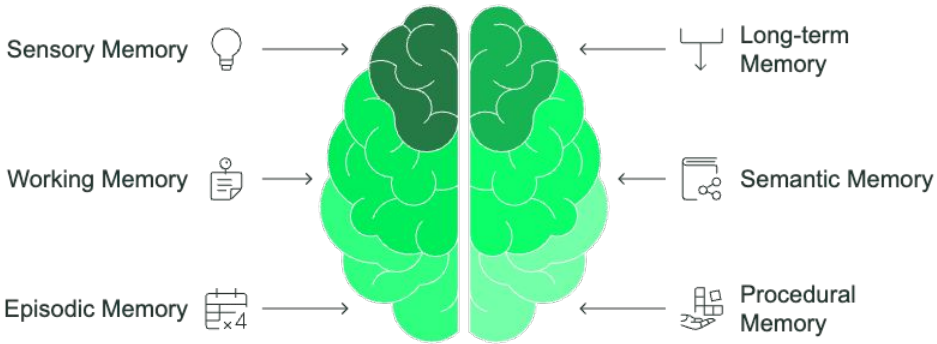
- LangGraph
- CrewAI
- AutoGen
- Custom / Others

MongoDB.



Multi-Cloud Scale, Resilience, Performance, & Security
supporting short-term and long-term memory, and state persistence on multimodal data

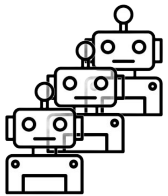
Human Memory



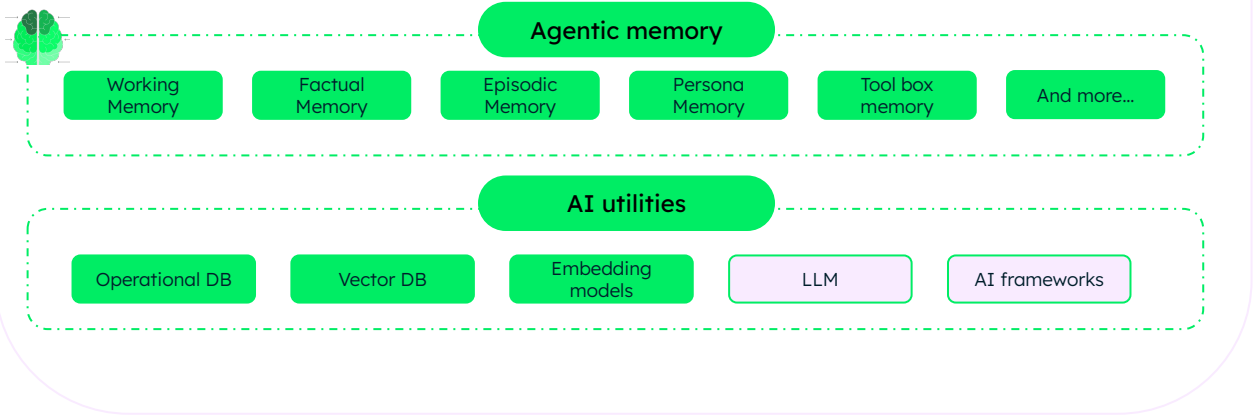
The most effective form of intelligence - for now - is human intelligence, and **human memory capabilities substantially define intelligence**.

How does MongoDB memory solution fit?

User



Agentic system



Hello from a full-time working parent

Bring to school

WORK

Quality time with my daughter

Order books & development stuff

Cook dinner

WORK

Schedule pediatrician

1:1 with teachers at school

Quality time with my partner

Good-night story

Pick up fro

Multi-functional family and household Agentic System?

ORK

Do groceries

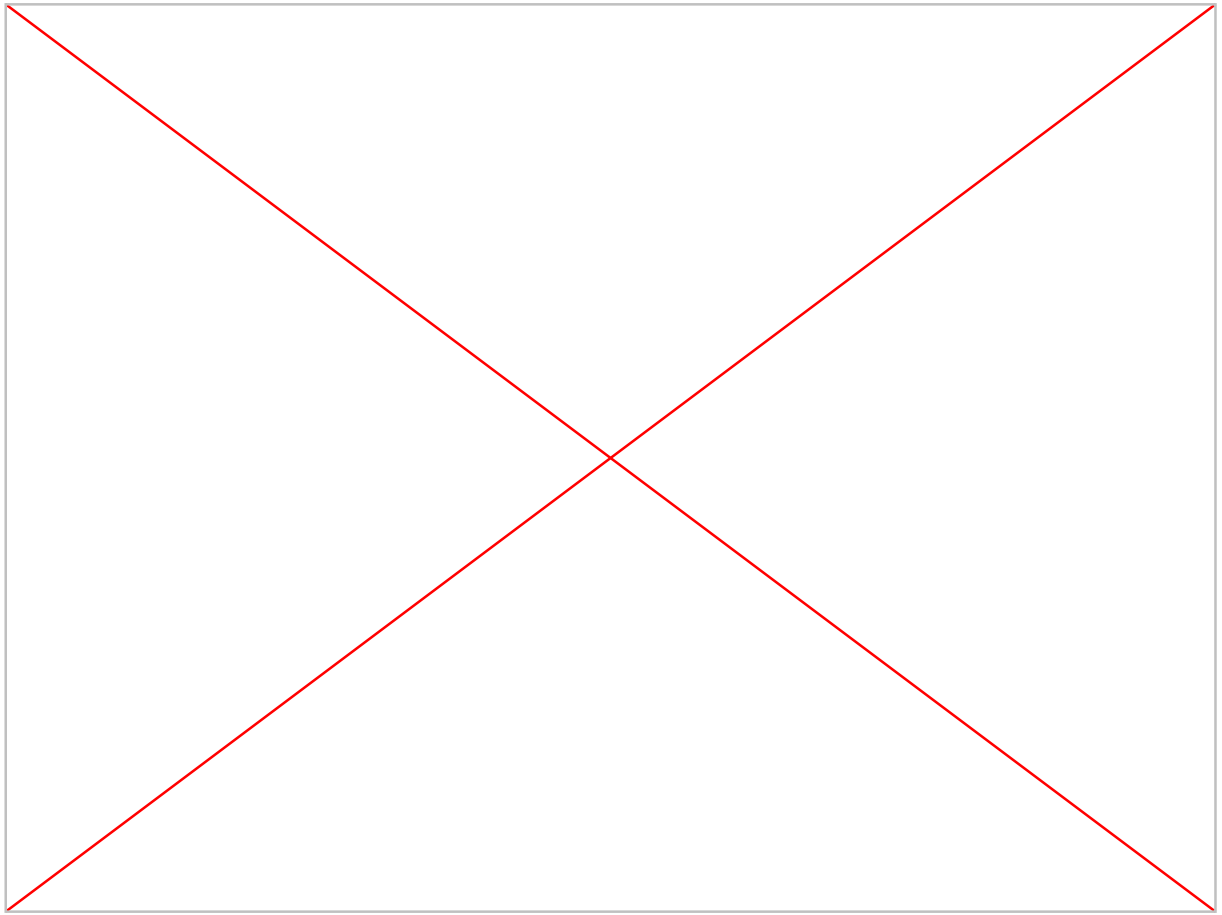
WORK

Plan school holidays

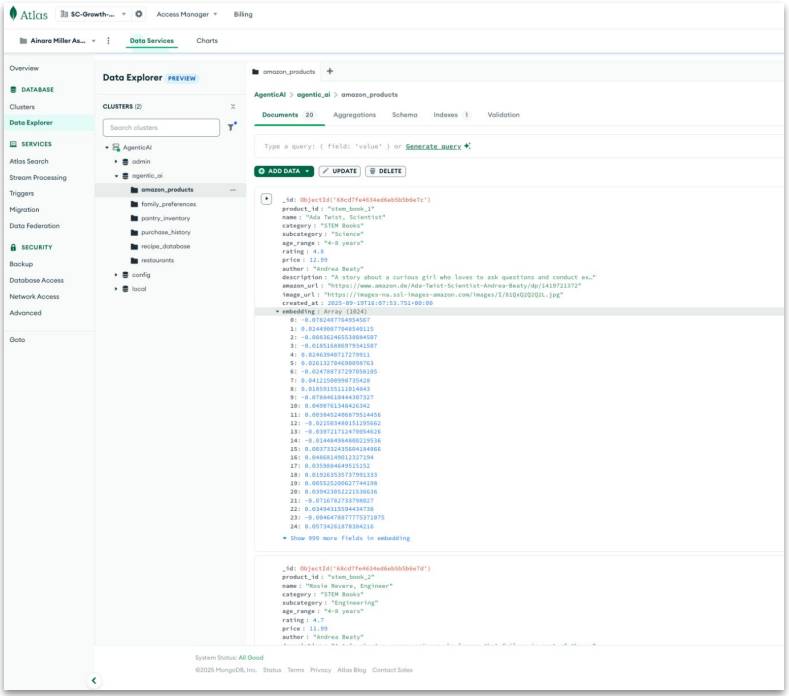
Rest of the family? Friends?

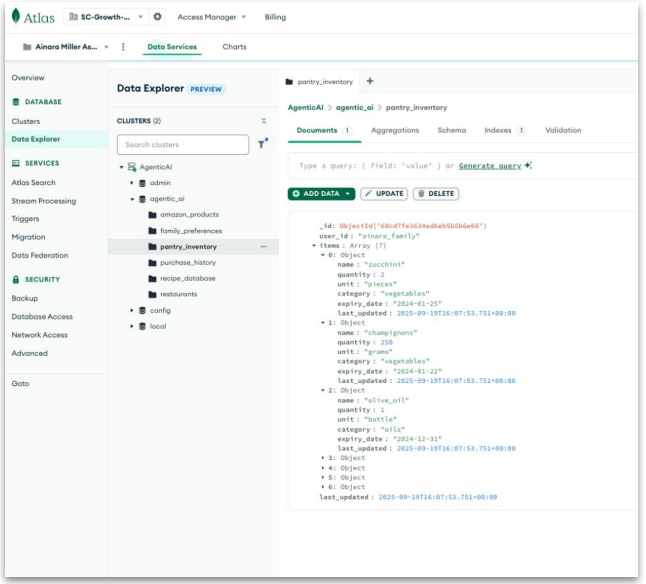
ME TIME?!?

Demo



MongoDB as Agentic Memory





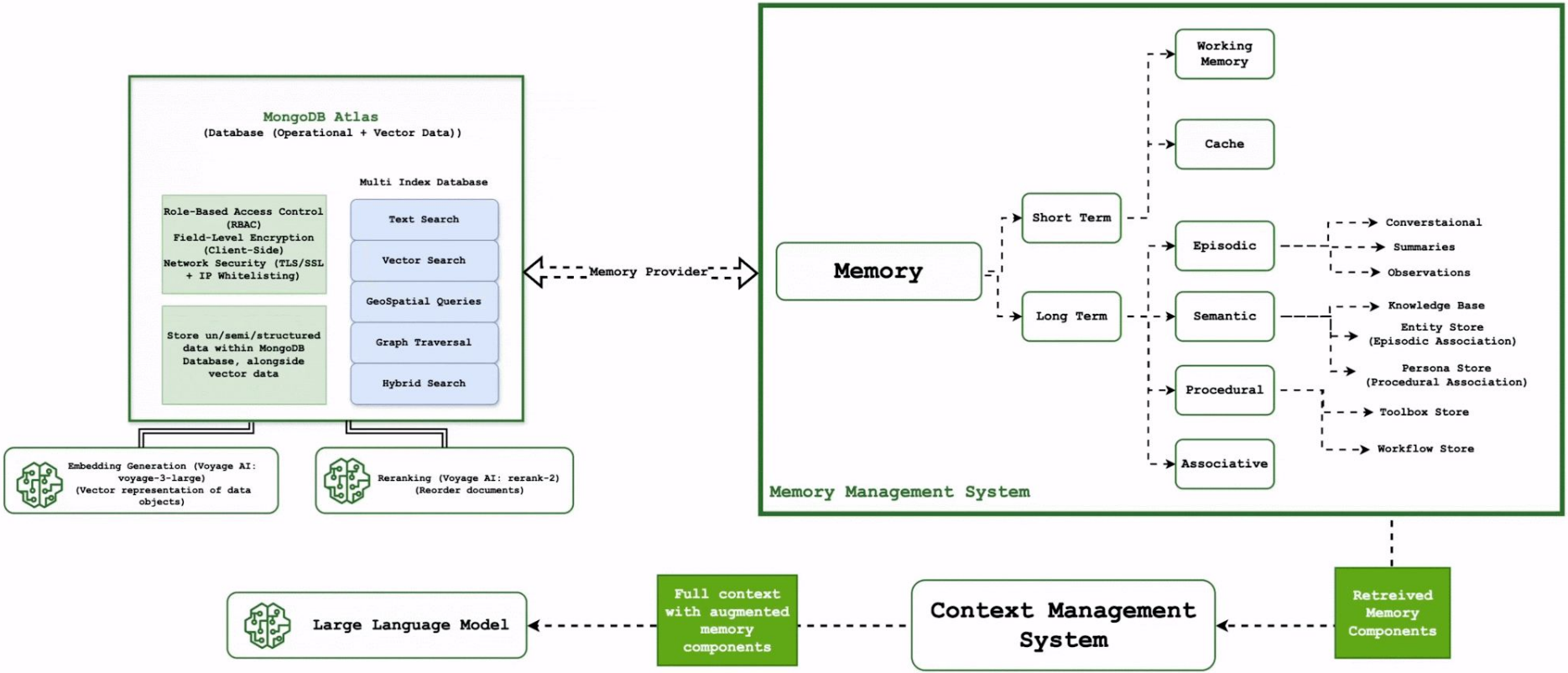
$vectorSearch and custom rerank?

```python
def get_voyage_embedding(self, plot, model='voyage-3-large'):
    response = self.voyage_client.embed([plot], model=model, input_type="document").embeddings
    return response[0]
```

```python
# Search recipes using MongoDB $vectorSearch
if self.mongo.voyage_client:
    query_embedding = self.mongo.get_voyage_embedding(query)

    # Use MongoDB aggregation pipeline with $vectorSearch
    pipeline = [
        {
            "$vectorSearch": {
                "index": "vector_search_index",
                "path": "embedding",
                "queryVector": query_embedding,
                "numCandidates": 20,
                "limit": 10
            }
        },
        {
            "$addFields": {
                "vector_score": {"$meta": "vectorSearchScore"}
            }
        }
    ]

    # Execute aggregation pipeline
    recipes = list(self.mongo.recipe_database.aggregate(pipeline))
```

```python
# Get pantry items
pantry = self.mongo.pantry_inventory.find_one({"user_id": user_id})
pantry_items = [item["name"] for item in pantry.get("items", [])] if pantry else []

# Get family preferences
preferences = self.mongo.family_preferences.find_one({"user_id": user_id})
preferred_cuisines = preferences.get("preferred_cuisines", []) if preferences else []
health_goals = preferences.get("health_goals", []) if preferences else []
```

```python
# Score recipes based on additional factors
scored_recipes = []
for recipe in recipes:
    recipe_ingredients = [ing["name"] for ing in recipe.get("ingredients", [])]

    # Get vector score from MongoDB
    vector_score = recipe.get("vector_score", 0)

    # Calculate pantry match score (capped to prevent domination)
    pantry_matches = len(set(recipe_ingredients) & set(pantry_items))
    pantry_score = min(pantry_matches / len(recipe_ingredients), 0.7) if recipe_ingredients else 0

    # Calculate cuisine preference score
    cuisine_score = 0
    for cuisine in preferred_cuisines:
        if cuisine in recipe.get("tags", []):
            cuisine_score += 0.3

    # Calculate health goal score
    health_score = 0
    for goal in health_goals:
        if goal in recipe.get("tags", []):
            health_score += 0.2

    # Overall score with vector similarity (prioritize semantic matching)
    total_score = (vector_score * 0.9) + (pantry_score * 0.05) + (cuisine_score * 0.03) + (health_score * 0.02)

    scored_recipes.append({
        "recipe_id": recipe["recipe_id"],
        "name": recipe["name"],
        "cooking_time": recipe["cooking_time"],
        "health_score": recipe["health_score"],
        "difficulty": recipe["difficulty"],
        "pantry_matches": pantry_matches,
        "total_ingredients": len(recipe_ingredients),
        "missing_ingredients": list(set(recipe_ingredients) - set(pantry_items)),
        "available_ingredients": list(set(recipe_ingredients) & set(pantry_items)),
        "total_score": total_score,
        "tags": recipe.get("tags", [])
    })

# Sort by total score
scored_recipes.sort(key=lambda x: x["total_score"], reverse=True)
```


MongoDB Atlas
(Database (Operational + Vector Data))
Multi Index Database
Role-Based Access Control (RBAC)
Field-Level Encryption (Client-Side)
Network Security (TLS/SSL + IP Whitelisting)
Store un/semi/structured data within MongoDB Database, alongside vector data
Text Search
Vector Search
GeoSpatial Queries
Graph Traversal
Hybrid Search
Memory Provider
Embedding Generation (Voyage AI: voyage-3-large)
(Vector representation of data objects)
Reranking (Voyage AI: rerank-2)
(Reorder documents)
Memory
Short Term
Long Term
Working Memory
Cache
Episodic
Semantic
Procedural
Associative
Converstaional
Summaries
Observations
Knowledge Base
Entity Store
(Episodic Association)
Persona Store
(Procedural Association)
Toolbox Store
Workflow Store
Memory Management System
Retreived Memory Components
Context Management System
Full context with augmented memory components
Large Language Model

Let’s connect!

Ainara Miller-Askar
Solutions Architect

E-Mail: ainara.milleraskar@mongodb.com
Mobile: +49 157 56 23 45 78

LinkedIn