

# Knowledge Vectorization: The Way RAG Actually Adds Value



**Serge Zakharov**

**Partner, Rebels.ai ML Lab**

Founder, ASOdesk.com & Keggly.beer  
MIPT alumni, ex-ABBY

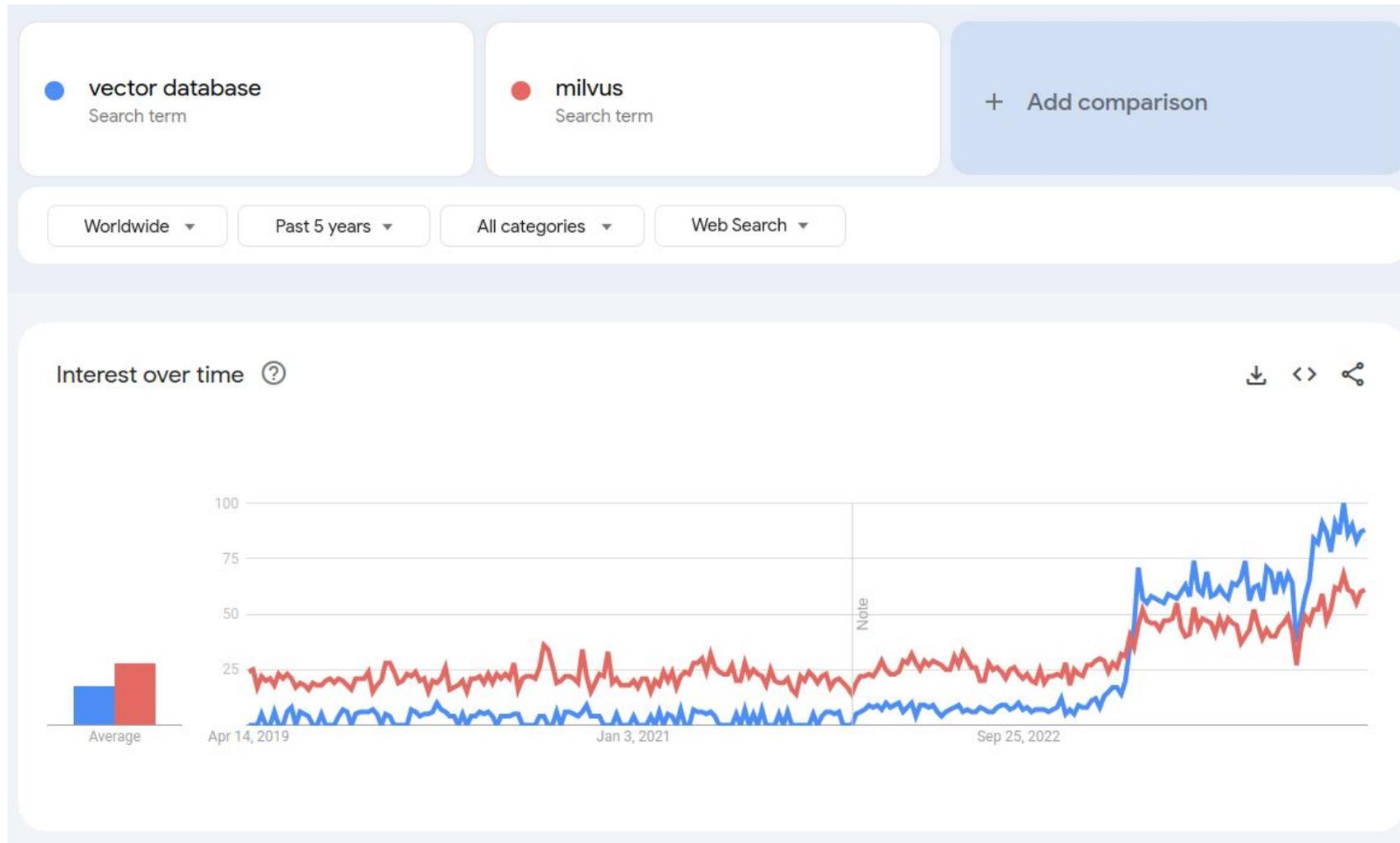


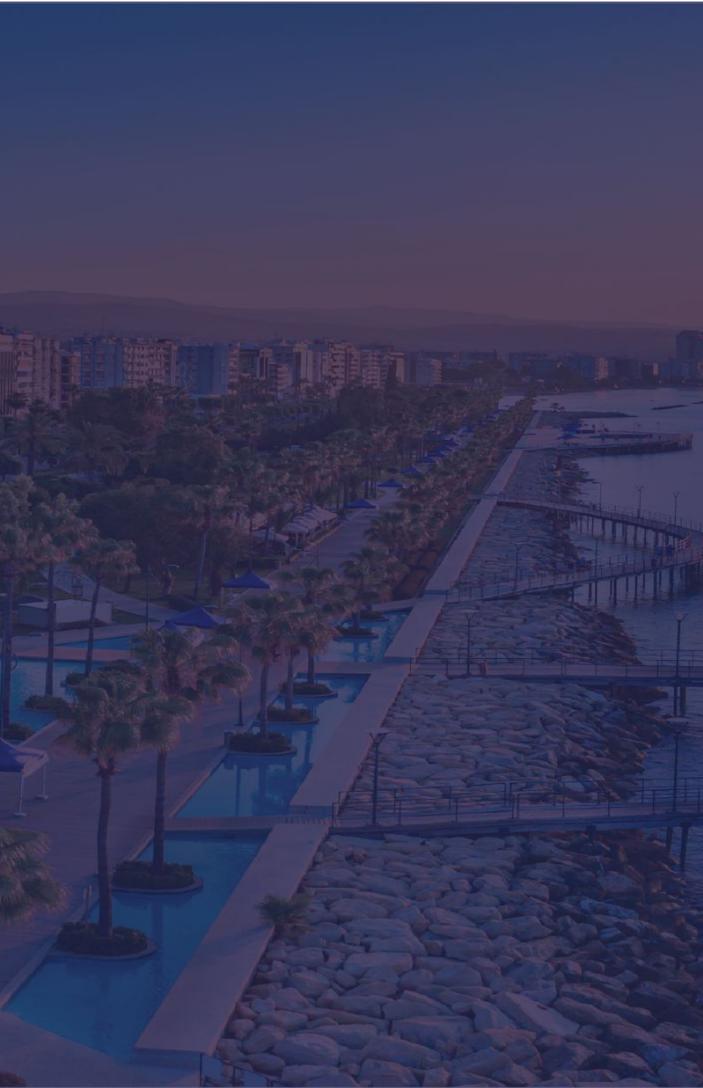
**Rebels.ai**



Limassol, Cyprus 2024

**PERCONA  
UNIVERSITY**





**2019**

Company established

**25**

Employees  
including  
5 PhD in ML and  
computer science

**50+**

Clients

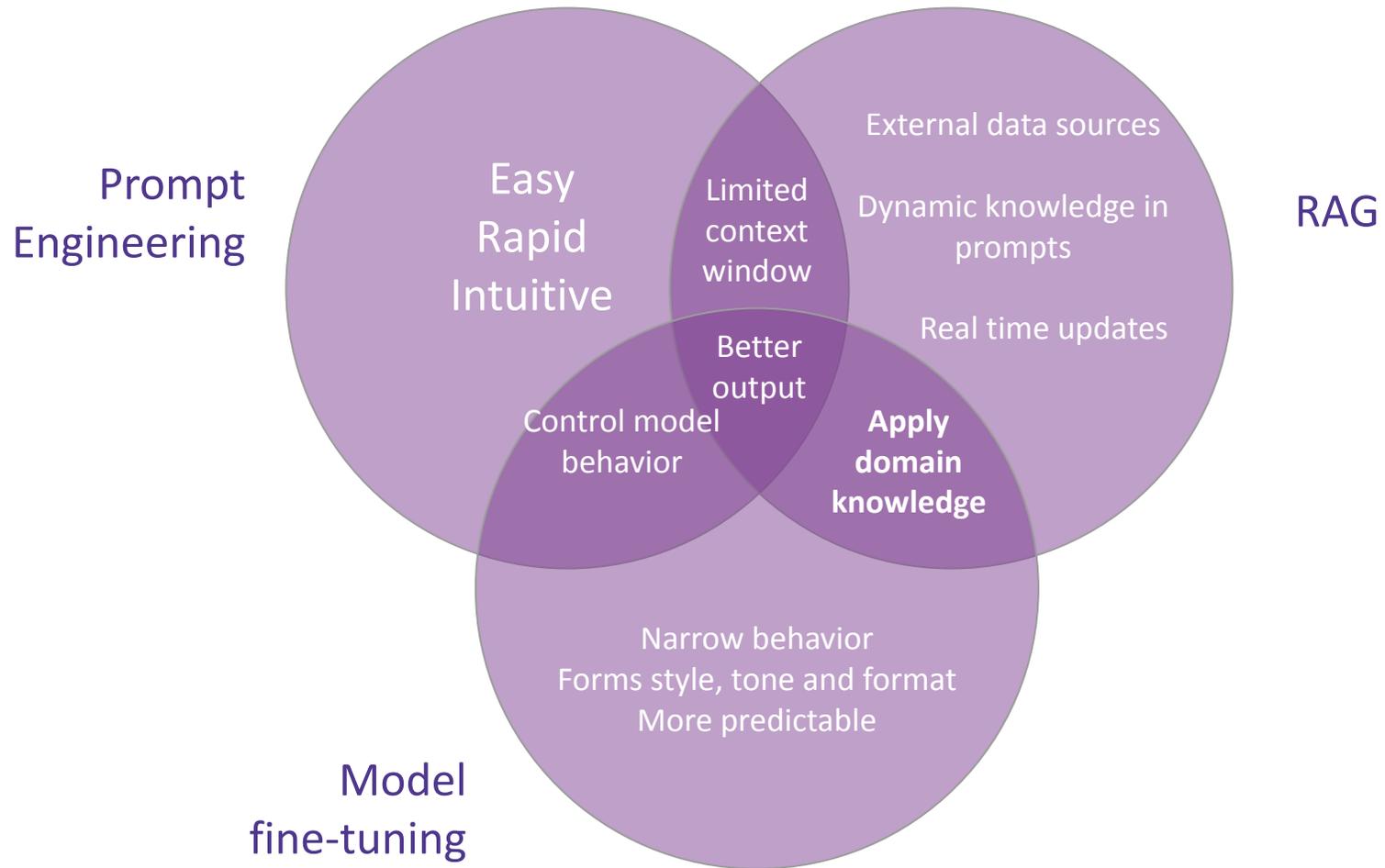
more than  
**100 solutions**  
implemented

## Machine Learning & Data Science

- Conversational agents, LLM applications
  - knowledge base localization for digital integrator in western Europe
  - Copywriting adviser for Top-3 telecom operator in one of Eastern Europe countries
- Natural language processing (NLP)
  - sentiment analysis, hate speech detection, user behaviour analysis for Blended learning platform
- Geographic Information Systems (GIS)
- User data analysis
- Immersive reality
- Fusion Power Reactor control with RL ⚡

*and much more*

# LLM and Techniques for Better output

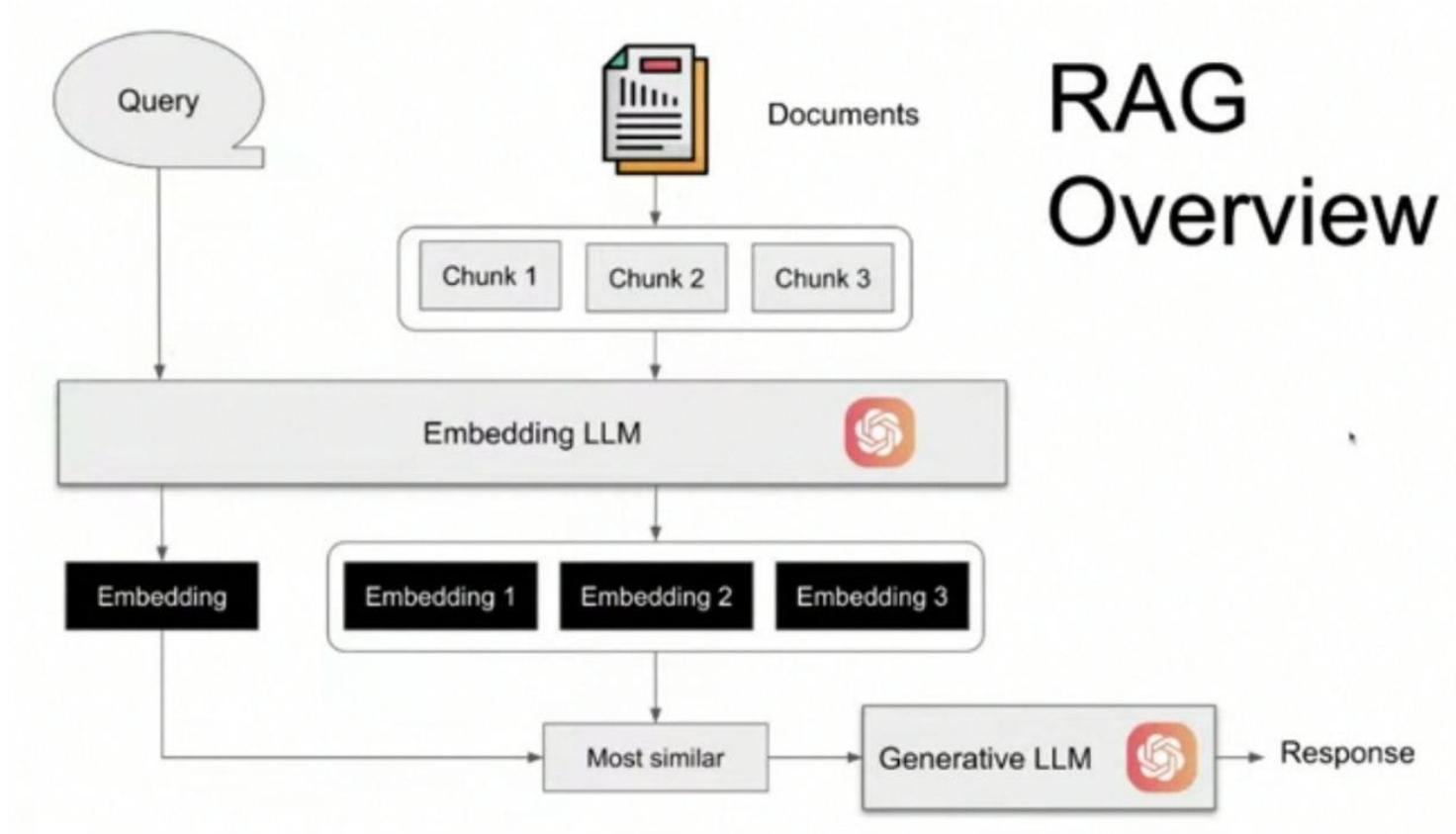


to DELVE - to dig in

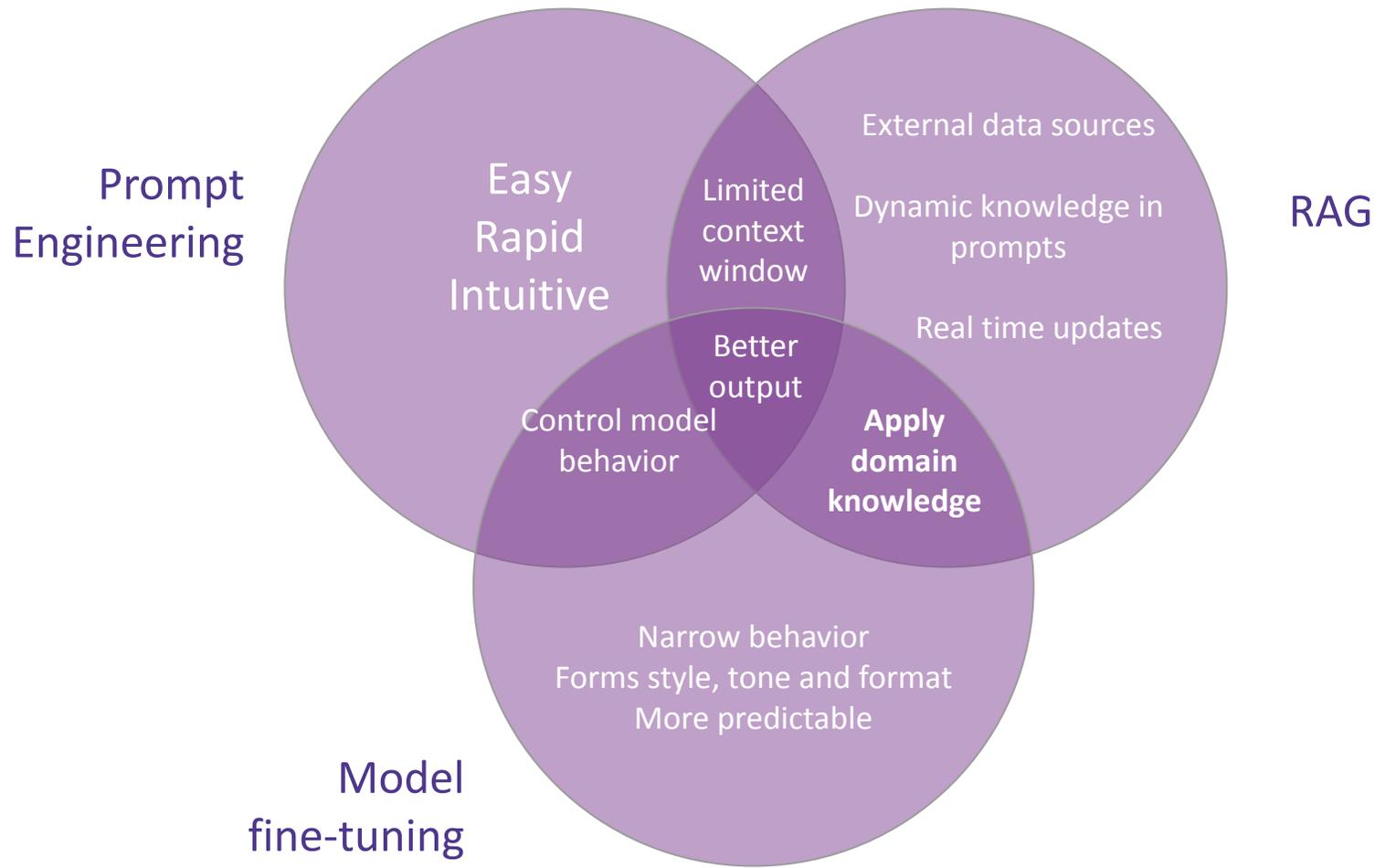
# What is Retrieval-Augmented Generation?

The way to build **Responsible AI**

Technique for **enhancing the accuracy and reliability** of generative AI models with facts **fetches from external sources**



# LLM and Techniques for Better output



# RAG, How to Brew it Right

Knowledge base

Chunking

Retrieval

Evaluation



# RAG Knowledge Preparations

- Remove irrelevant texts and documents, noise data, special characters, stop words
- Identify and correct errors: spelling, typos, and grammar
- Replace pronouns with names!  
*she -> Kate, them -> the viewers*



# RAG Chunking

Model	Context length	Number of English pages*
GPT 3.5	4,096	6
GPT 4	8,192	12
GPT 4-32k	32,768	49
Llama 1	2,048	3
Llama 2	4,096	6

Context length comparison. (\*Assuming 500 words per page.)

Dive into the joy of chunking, where each piece is a puzzle of its own. As you assemble them, a mosaic of understanding takes shape. This engaging mental exercise sparks creativity and hones analytical skills. It's like solving a puzzle, finding satisfaction in each arrangement. Approach chunking with curiosity and a playful spirit. Let it be an intellectual playground, making the process not only enjoyable but deeply satisfying. Happy chunking!

**Fixed Size Chunking** Character, Words, Tokens

**Recursive Chunking**

**Document Based Chunking**

**Proposition-led Chunking!**

**Chunk enrichment by custom features** Tone of Voice, Scene, Context, ...

# RAG Embedding and Retrieval

## Vector databases



Postgresql  
+ pg\_vector

Ecosystems  
Langchain  
llama index

# RAG Evaluation

## Translation inherited

BLEU  
METEOR  
ROUGE

## Sample prediction

Perplexity

## Semantic similarity

BERTScore

## User feedback

### Human Evaluation

Likert Scales, Open feedback  
Pairwise Comparisons

### Task-Based Evaluation

ChatBot Arena

### Interactive Evaluation

A/B, Wizard of Oz Tests

## LLM monitoring, evaluation and observability

open source tools

[athina.ai](https://athina.ai)

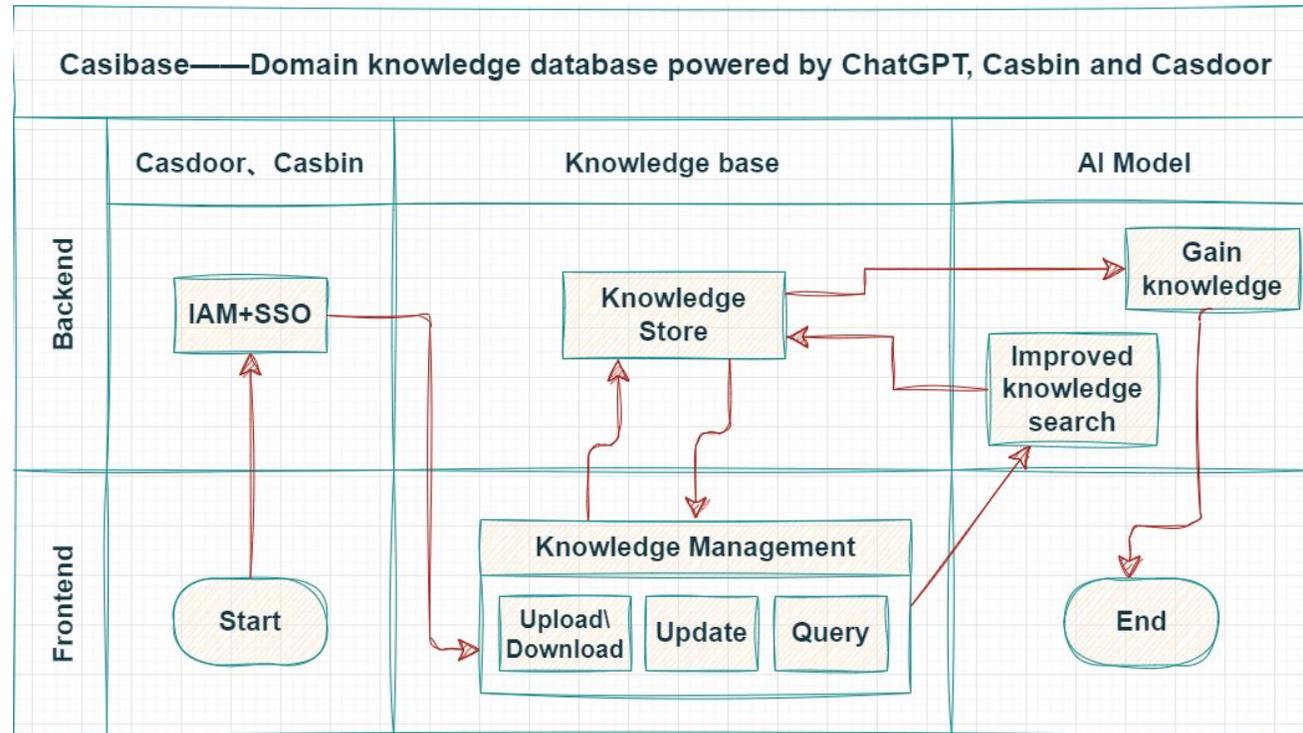
[confident-ai](https://confident-ai.com)

[Deepchecks](https://deepchecks.com)

[EvidentlyAI](https://evidentlyai.com) 

[Arize Phoenix](https://arize.com)

# Open source tools for RAG: Casibase



<https://github.com/casibase/casibase>

# Open source tools for RAG: Verba by Weaviate

The screenshot displays the Verba web interface. At the top left is the Verba logo, a yellow square with a cartoon dog, and the text "Verba The Golden RAGriever". To the right are buttons for "+ Add Document...", "ADAEmbedder", "WindowRetrie...", and "GPT4Generat...". Below these are several blog post thumbnails with titles like "65 blog-graphql-api-de...", "63 developers-weaviate", "61 developers-weaviate", "75 developers-weaviate", and "73 blog-crud-support-...".

The main content area is split into two panels. The left panel is titled "RAGriever Chat" and shows a chat window with the text: "Weaviate is an open-source vector database that stores both objects and vectors. It allows for the storage and retrieval of data objects based on their semantic properties by indexing them with vectors. Weaviate can be used stand-alone or with a variety of modules that can do the vectorization for you and extend the core capabilities. Weaviate has a GraphQL-API to access your data easily and is built for speed. It is a low-latency vector database with out-of-the-box support for different media types (text, images, etc.). It offers Semantic Search, Question-Answer Extraction, Classification, Customizable Models (PyTorch/TensorFlow/Keras), etc. Built from scratch in Go, Weaviate stores both objects and vectors, allowing for combining vector search with structured filtering and the fault tolerance of a cloud-native database. It is all accessible through GraphQL, REST, and various client-side programming languages. Weaviate is used by Software Engineers as an ML-first database for their applications and by Data Engineers as a vector database that is built up from the ground with ANN at its core. It is cloud-native, distributed, runs well on Kubernetes, and scales with one's workloads." Below the text is an input field containing the question "What is a vector database?".

The right panel shows a blog article preview titled "blog-graphql-api-design". The main heading is "What is Weaviate and why does it need an API?". The text below reads: "Weaviate is an open-source Vector Database: for understandable knowledge representation, enabling semantic search and automatic classification. Weaviate does not only store data, but also its (automatically derived) context, consisting of linguistic information and relations to other concepts. The result is a network of knowledge, or a graph of data. One core question is: How do we interact with this new type of data storage? Interacting with big data, enriched with contextual information, might sound even more overwhelming than interacting with a traditional, relational database. Data needs to be added, retrieved and manipulated, all controlled by the user but enabled by the underlying database interface. Here's where APIs jump in. Because of Weaviate's graph-based architecture, an alternative to traditional RESTful APIs was what we were looking for." Below this is another heading: "What is GraphQL, and why does Weaviate use it?".

At the bottom of the interface are three tabs: "Search", "Documents", and "Status".

## Extra links

[Seven Failure Points When Engineering a Retrieval Augmented Generation System + Guide](#)

[Vector DBs Compared](#)

[Data Frameworks for LLM: Llama Index vs LangChain](#)  
[Casibase new alternative to LangChain](#)

['Awesome' list of LLM RAG systems and components](#)



**Subscribe Rebels.ai News**  
[linkedin.com/company/rebels-ai](https://www.linkedin.com/company/rebels-ai)  
[t.me/rebels\\_ai](https://t.me/rebels_ai)



# Thank you!



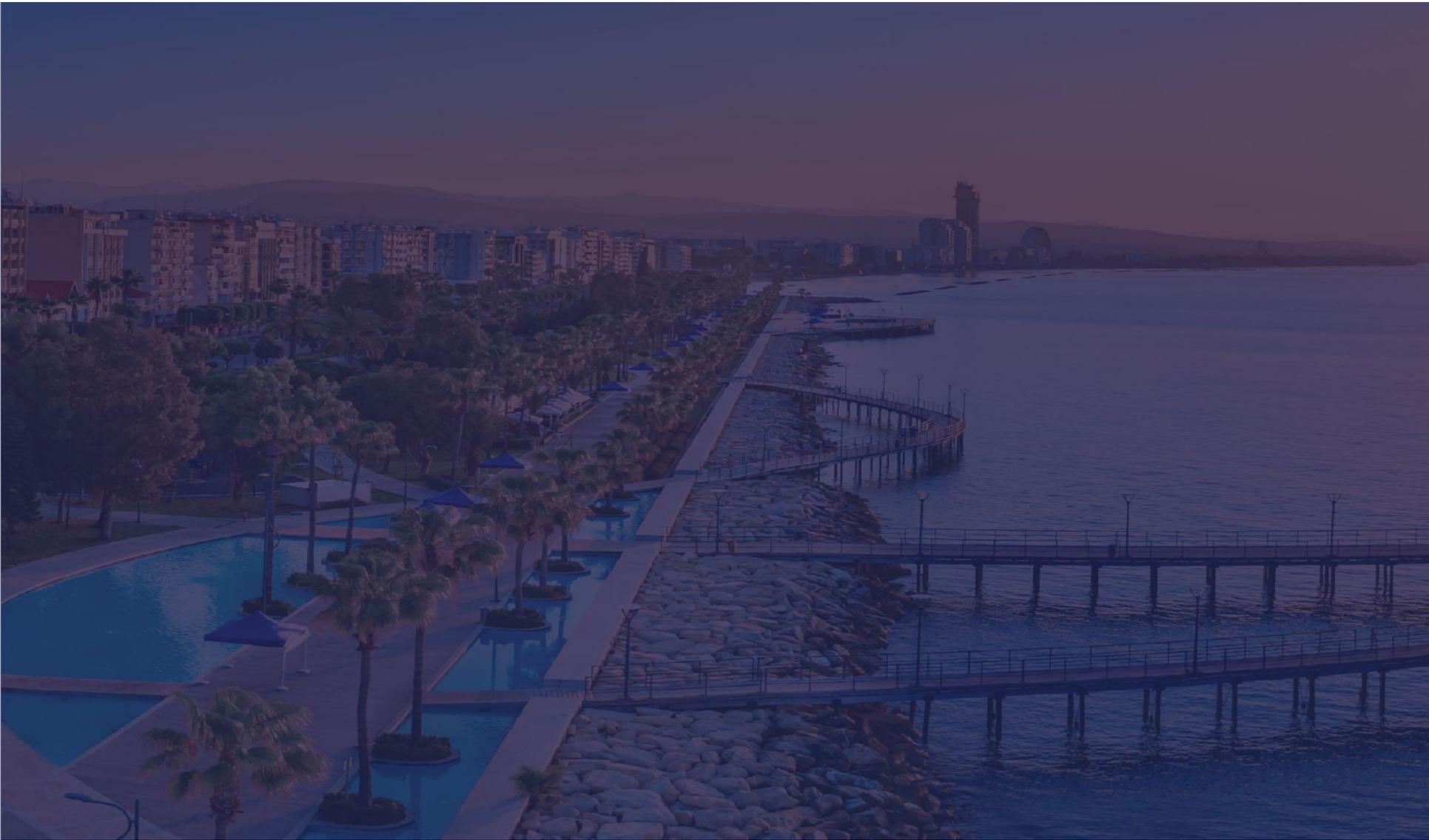
## Serge Zakharov

Partner, Rebels.ai ML Lab  
Founder, Asodesk.com & Keggly.beer  
MIPT alumni, ex-ABBYY



Rebels.ai





Limassol, Cyprus 2024

**PERCONA**  
UNIVERSITY