NEWSITO INFORMIRO

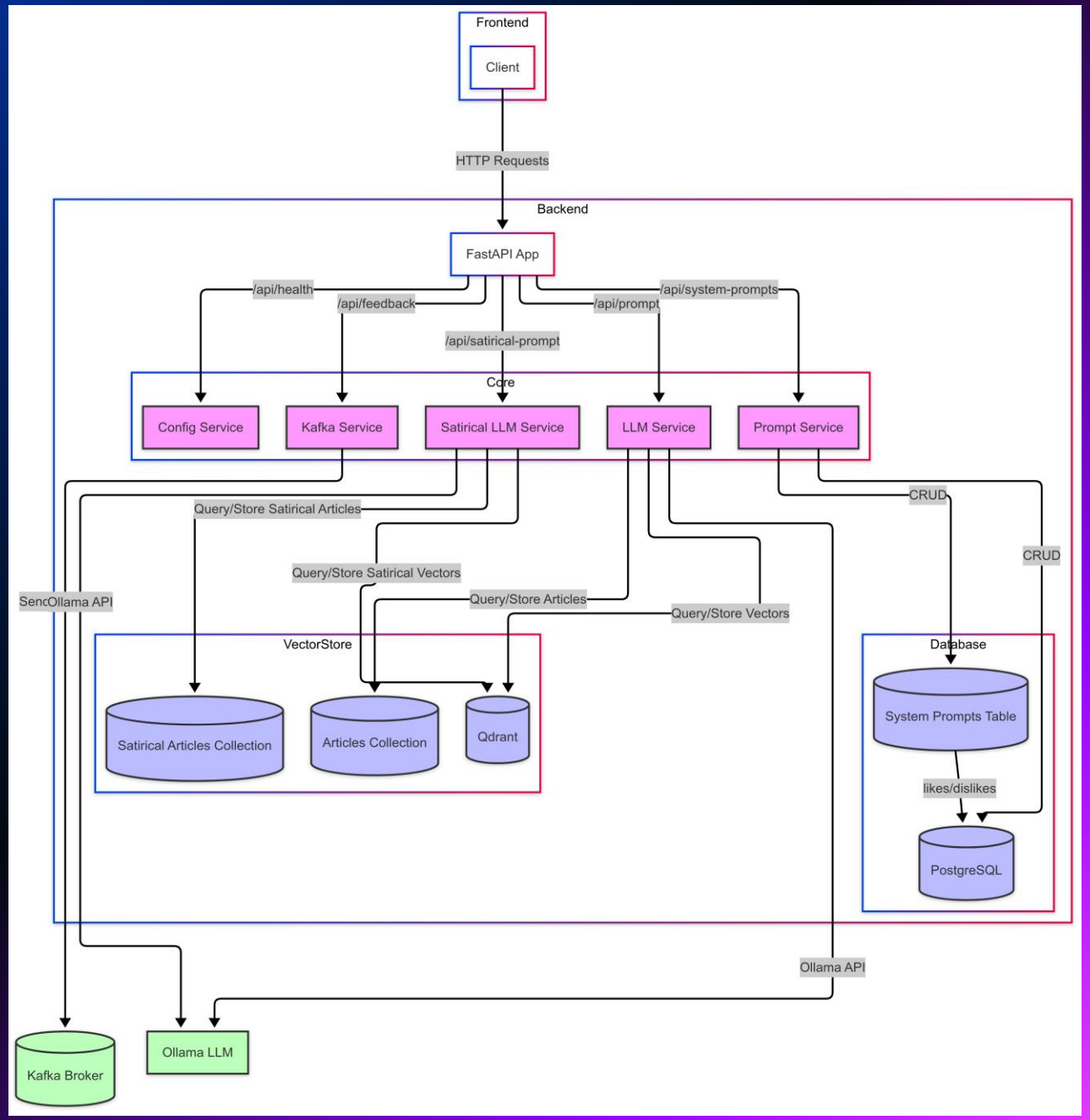
A RAG SYSTEM FOR DYNAMIC NEWS INSIGHTS

Enache Vlad
Gabor Ioana
Ignat Dragoș
Lăcătuș Darius
Măierean Mircea

ARHITECTURE

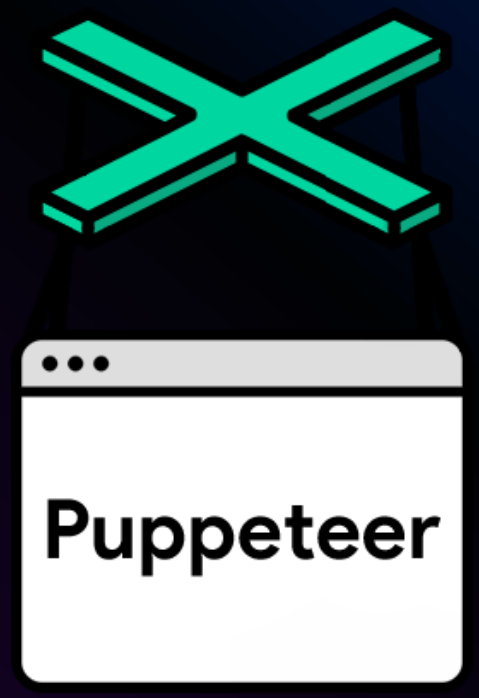
Key Components (Labels):

1. **Frontend (NextJS)**
2. **Backend (FastAPI)**
3. **Vector Store (Qdrant)**
4. **Database (PostgreSQL)**
5. **LLM (Mistral)**



DATA COLLECTION: PUPPETEER

- For up to date information around the world, news represent a valid data source
- Dataset composed of news from accurate news sites with an international opening
- In total, we scraped over 75k sites to create our dataset
- For achieving this, we used puppeteer, which is a scraping library developed by Google

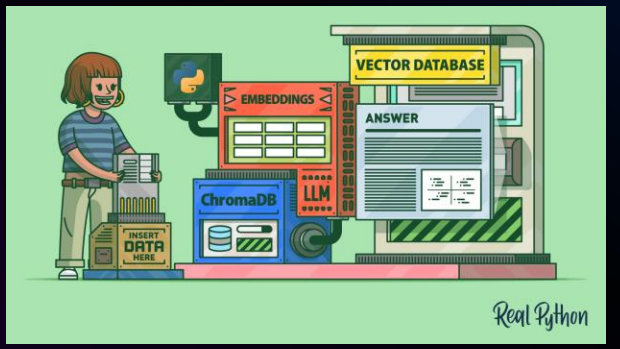


DATA STORAGE: QDRANT

- Qdrant: Blazingly fast Vector DB (written in rust) and accurate vector search using HNSW algorithm
- Supports filtering with metadata
- Easy to use with REST API, Python SDK, and Docker
- Open source and scalable

DATA PROCESSING: EMBEDDINGS

- Convert text into high-dimensional vectors (1024 dimensions)
- Similar content → similar vectors
- Qdrant stores vectors and finds the closest ones using similarity (e.g., cosine)
- Used for search, recommendations, and semantic matching



RAG SYSTEM PROMPTING

- **RAG** combines information retrieval with generative AI.
- Retrieves relevant documents from a vector store, creates context to prompt the LLM for a more accurate, grounded response.
- **System Prompts:** Stored in a database (SystemPrompts table).
- Curated prompts guide the LLM's behavior (e.g., tone, style, focus).
- Users can like/dislike prompts, and the system tracks usage statistics.
- The backend selects a "favorite" or random prompt to prepend to user queries.
- Sometimes, multiple prompts are used to generate diverse responses (based on a configuration of the percentages)

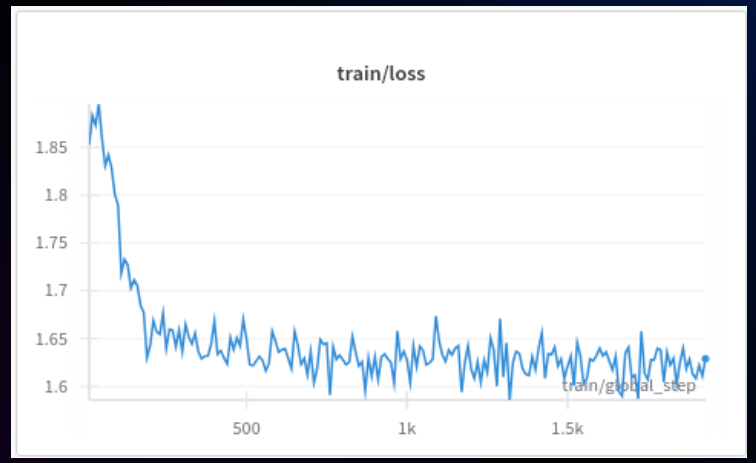
MITIGATION COMPONENT

We employ a dual-layer safety system:
1️⃣ **Toxicity Detection:** Every response is analyzed using Detoxify's AI-powered filters
2️⃣ **Content Sanitization:** Flagged outputs are automatically rewritten to remove harmful elements

This steps is of paramount importance especially for satirical content, as the data from satirical websites is often rougher in language, prone to harmful content

MODEL FINE TUNING USING UNSLOTH AI

- For finetuning on News question-answering, we used the NewsWA dataset from Microsoft
- We used the original training / test split
- Finetuned for 3 epochs (2-3h), using Weights & Biases for logging
- Used Mistral-7B-Instruct-v0.2 as a base model



MODEL EVALUATION USING DEEPEVAL

- Framework providing LLM-as-a-judge evaluation
- Evaluated variations of models, system prompts, with rag strategy and finetuning

- Test metrics:
  - Contextual Precision
  - Contextual Recall
  - Contextual Relevancy
  - Answer Relevancy
  - Faithfulness Metric

DEEPEVAL RESULTS

Contextual metrics – measure the quality of the retrieved context

Scenario	Dataset	Contextual Precision	Contextual Recall	Contextual Relevancy
System prompt 1	custom	0.679	0.501	0.625
System prompt 2	custom	0.679	0.517	0.605
Finetuned	NewsQA	0.847	0.6613	0.487

DEEPEVAL RESULTS

Answer metrics – measure the quality of the answer, in relation to the context

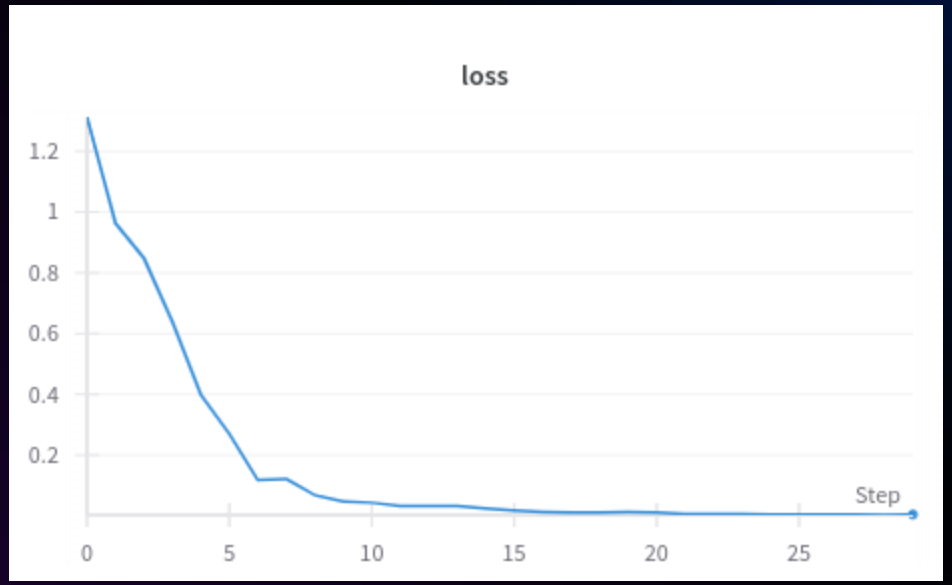
Scenario	Answer Relevancy	Faithfulness
mistralai/Mistral-7B-Instruct-v0.2	0.766	0.631
mistralai/Mistral-7B-Instruct-v0.2 (finetuned on NewsQA)	0.684	0.802
google/gemma-7b	0.807	0.694
meta-llama/Llama-3-8b	0.723	0.664

FEEDBACK SYSTEM
DATA INGESTION USING KAFKA

- As messages are displayed in pairs, users can give feedback to individual messages
- Users can also give feedback to individual messages, by liking or disliking
- Each event of this type is processed on a message queue using Kafka a Kafka Producer
- A microservice consumes using a Kafka Client these events, in batches, as a cronjob
- These events are used for training the model in the future

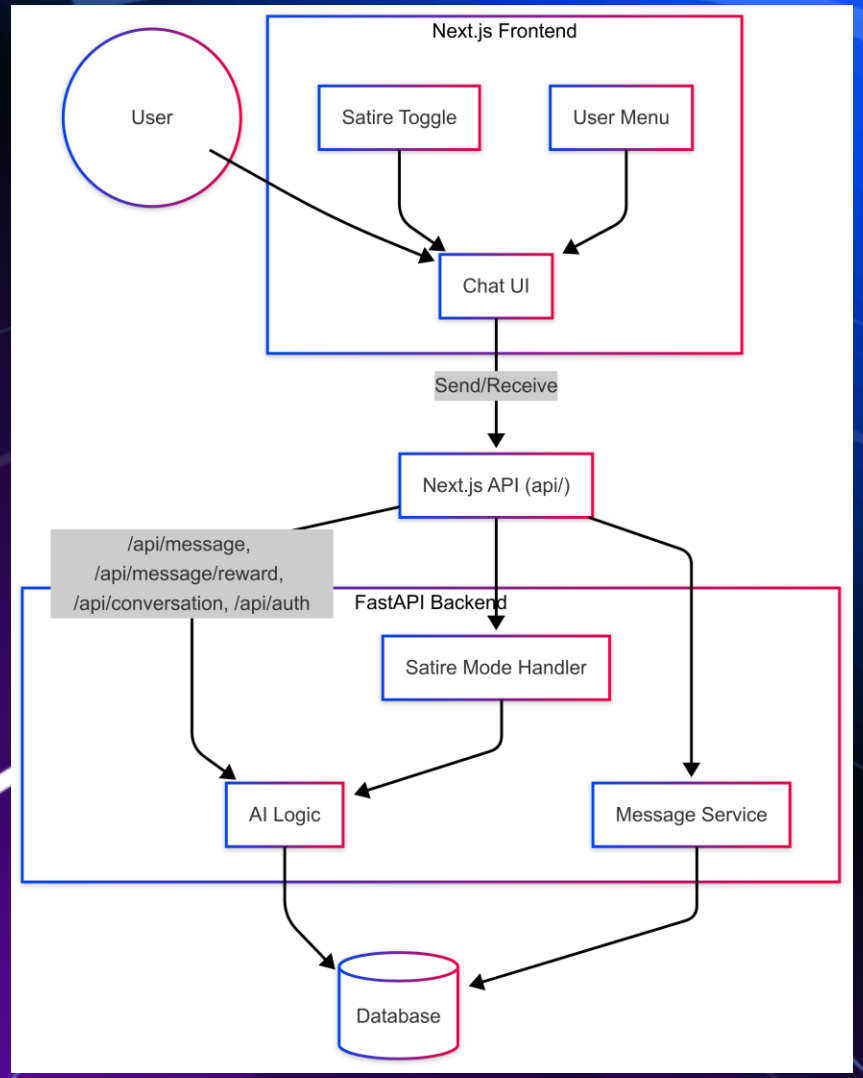
REINFORCEMENT LEARNING FROM HUMAN FEEDBACK STRATEGY

- Reward Model: the messages, together with the user feedback, are used to train a BERT-based reward Model.
- For an appropriate quantity of messages, this reward model can be used in conjunction to the original model and a reference model to adjust based on feedback, following a Proximal Policy Optimization approach (TRL library)



FRONTEND ARCHITECTURE

- Each user has an account, that stores the existing conversations with the assistant
- For each message, user has options to select 🤪 satirical 🤪 mode



THANK YOU