



TravelMate

Technical presentation
Ettore Ricci, Paolo Palumbo, Francesco Boldrini

Project overview

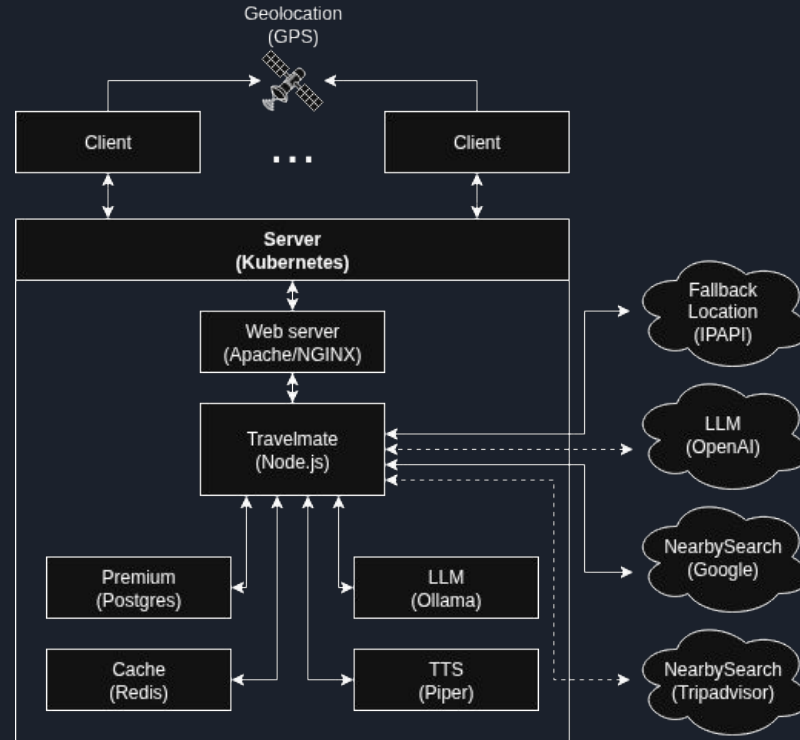
- Discover nearby points of interest
- Filter POI by type
- Driver friendly commands
- Nearby area AI description read by TTS
- Automatic update with use movement



User Interface



System architecture





Microservices

We developed a microservices-oriented application

This paradigm allows for high scalability and modularity

The app is easily deployable on Kubernetes



Microservices TravelMate

We used the Next.js React framework to develop the application

The resulting application is platform independent and easy to maintain and develop

We created a reverse proxy for all the APIs that we use to control API access and keep the keys secure



Microservices Web server

The demo used Apache but any HTTPS-reverse-proxy-capable web server is adequate for the job

Ideally we would use NGINX because it's easily configurable and deployable




Microservices LLM

We chose Ollama for self hosting our LLM because it's fast and requires practically no configuration to work

We configured Ollama to always keep the model loaded to save time on the inference

The LLM we chose is Llama3.2:3b because it's relatively small but smart enough to follow the prompt correctly

This microservice could be swapped with the OpenAI API if a more powerful model is necessary



Microservices TTS

Piper and FastAPI were chosen for the job because it was the easiest way to provide a TTS server without resorting to external APIs

Piper is the model that translates text to raw PCM frames while FastAPI serves them on an HTTP stream

Unfortunately Piper is poorly documented and does not have incredible performance, but it's the model with the lowest latency among the ones we considered



Microservices Premium

For the demo we used a placeholder service implemented with FastAPI

In a real use-case we would use Postgres to provide the necessary information about the sponsored businesses



Microservices Cache

In the demo this was disabled

In a real use-case we could use Redis to provide a fast cache that avoids repeated calls to the more costly APIs like Google and LLM

App flowchart

