

Project Presentation

Instructor: Ngoc Thanh Dat Thai

Team 18

01

Introduction

02

System Design Overview

03

How we did it?

04

Question and answer

Problem Statement

‘Group Travel’ == ‘Planning Headache’

- **Scattered Conversations:** Group chats quickly become messy with fragmented messages, making it hard to track decisions or suggestions.
- **Decision Fatigue:** Planning as a group often leads to indecision due to overload of options and differing preferences.



Photo by Getty Images on Unsplash

Gravel: Smart Travel Planning

Group Chat + Recommendation System for Collaborative Trip Planning



Project Overview:

Gravel is a web-based group chat app enhanced with an AI assistant that suggests travel destinations based on chat content.



Tailored for Travelers:

Designed for friends planning trips together, making decisions simpler and smarter with contextual AI support.

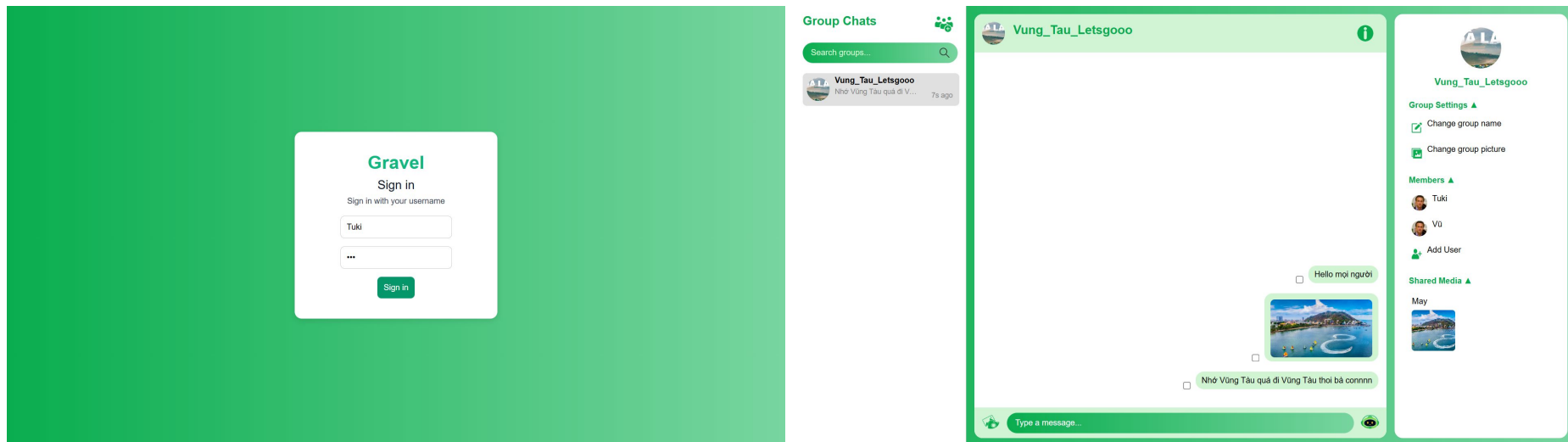


Built with Modern Stack:

Uses NodeJS & Python backend, ReactJS frontend, deployed via Vercel. Powered by latest technologies, with OpenAI + FAISS for intelligent travel insights.

Gravel: Smart Travel Planning

Group Chat + Recommendation System for Collaborative Trip Planning



Demo

Demo here

Gravel: Smart Travel Planning

Core Goals

01

Group chat service

Support complex context understanding from extended conversations

02

Real-time data collection

Real-time data collection and processing at scale, enabling millions of location records to be updated instantly.

03

Location suggestion

Provides multiple suggestions that closely aligned user expectation.

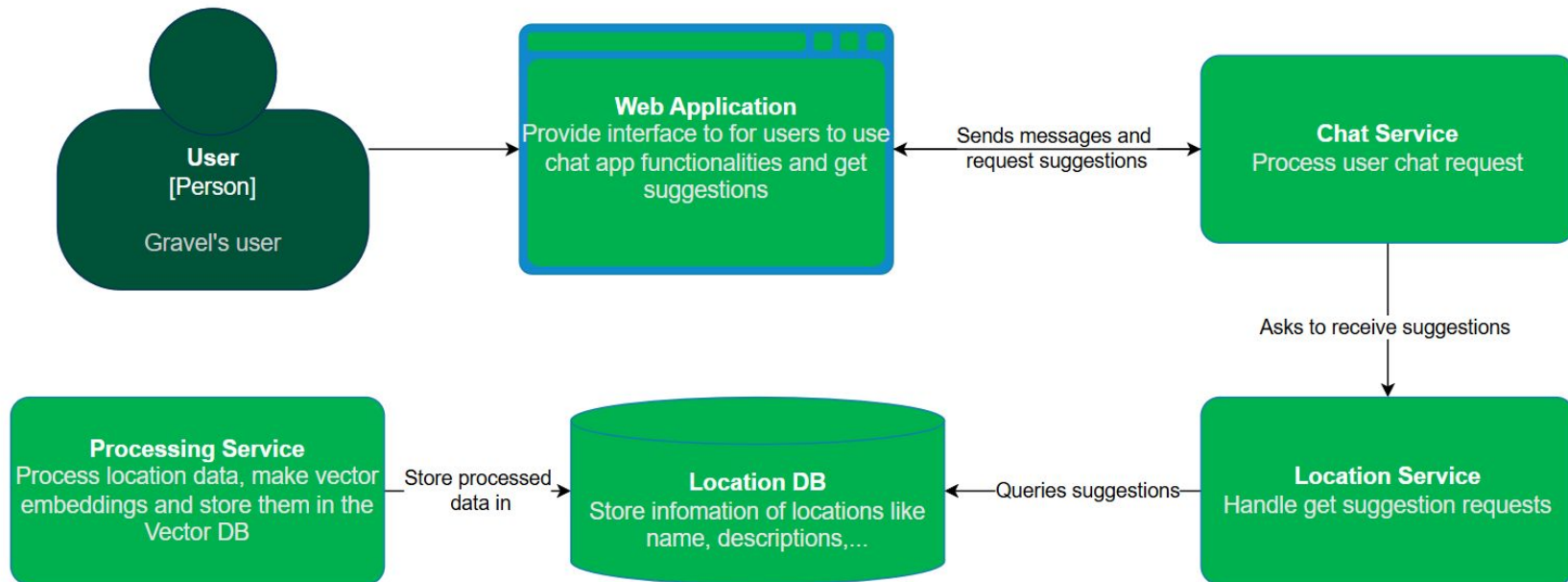
04

Efficient search for minimal latency

Gives responses in a short time, ensure user experience.

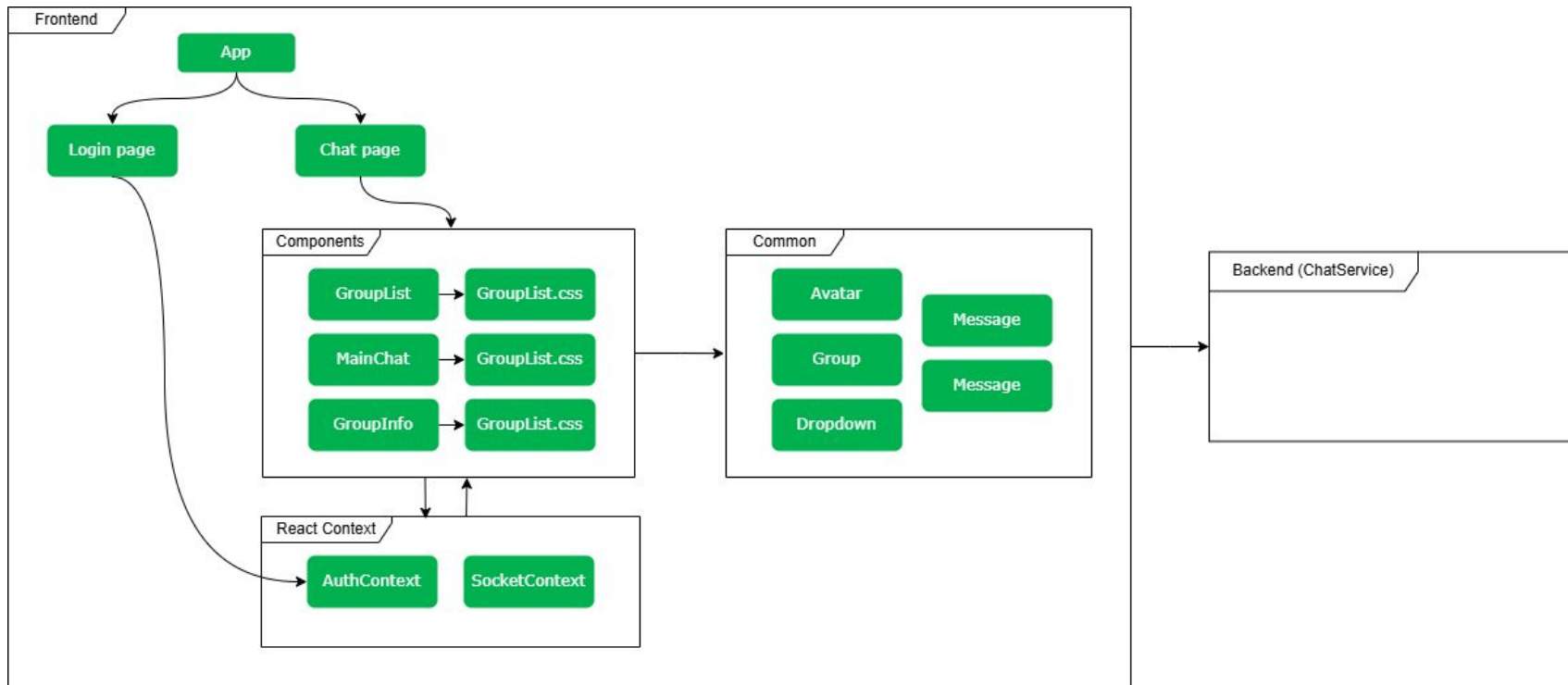
Gravel: User flow

Overview

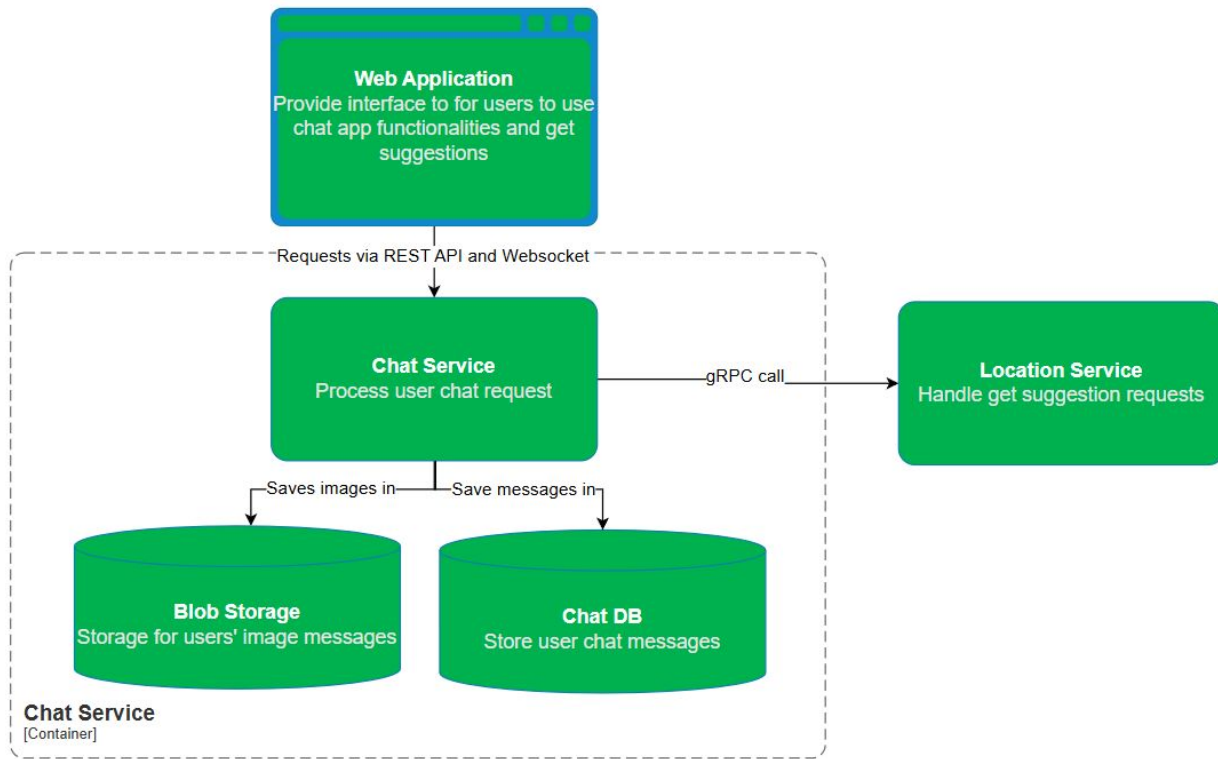


Part 1 - Frontend

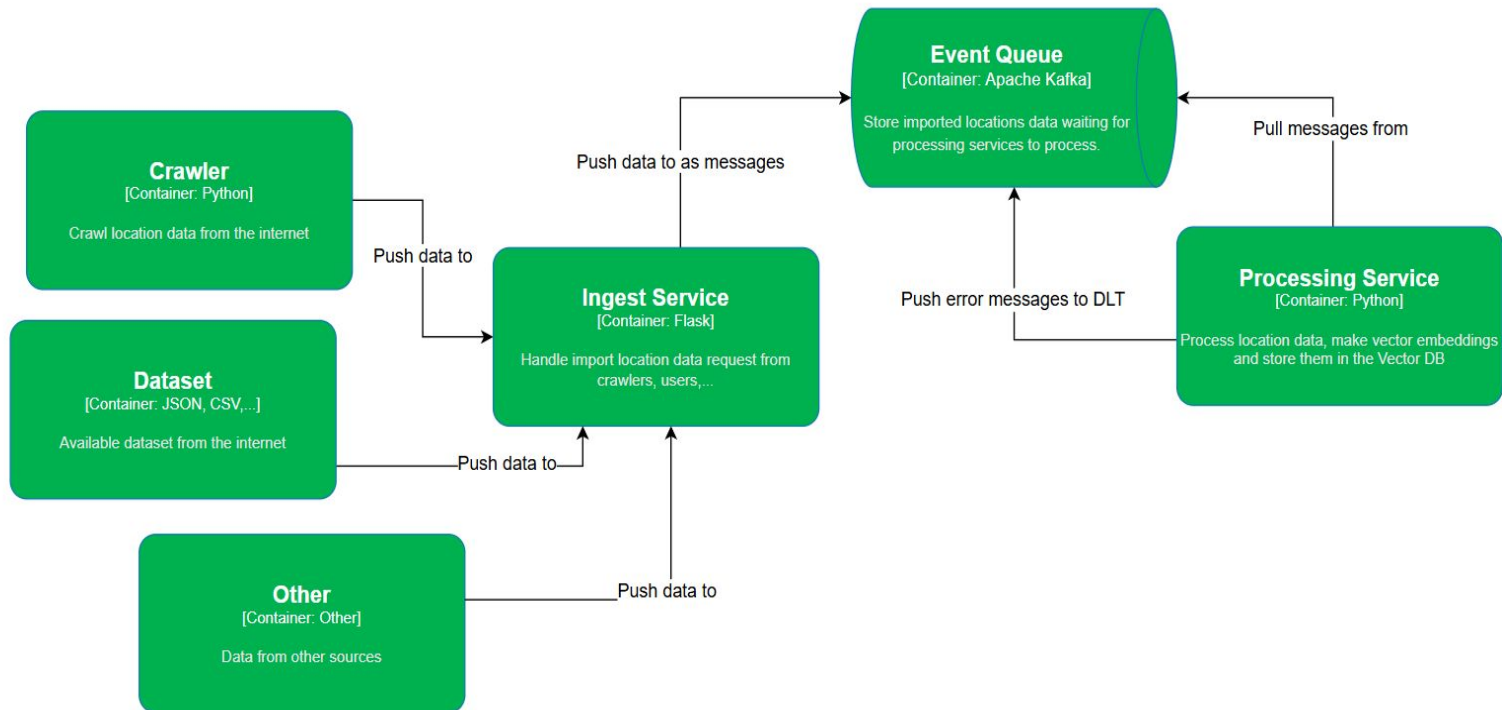
Modular ReactJS Architecture with Real-time Features and Grab Branding



Part 2 - Chat Service Backend

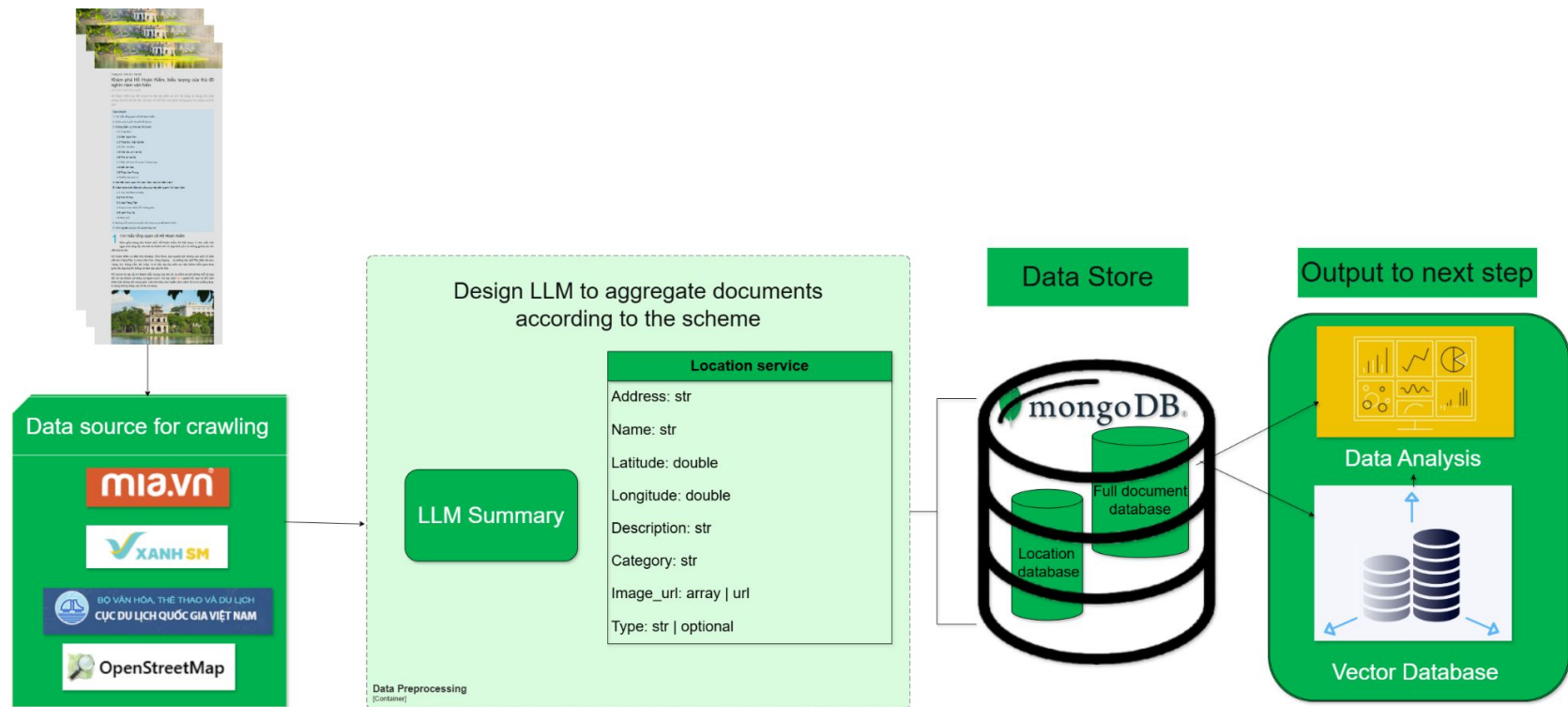


Part 3 - Data Processing Backend



Part 4 - Crawl data

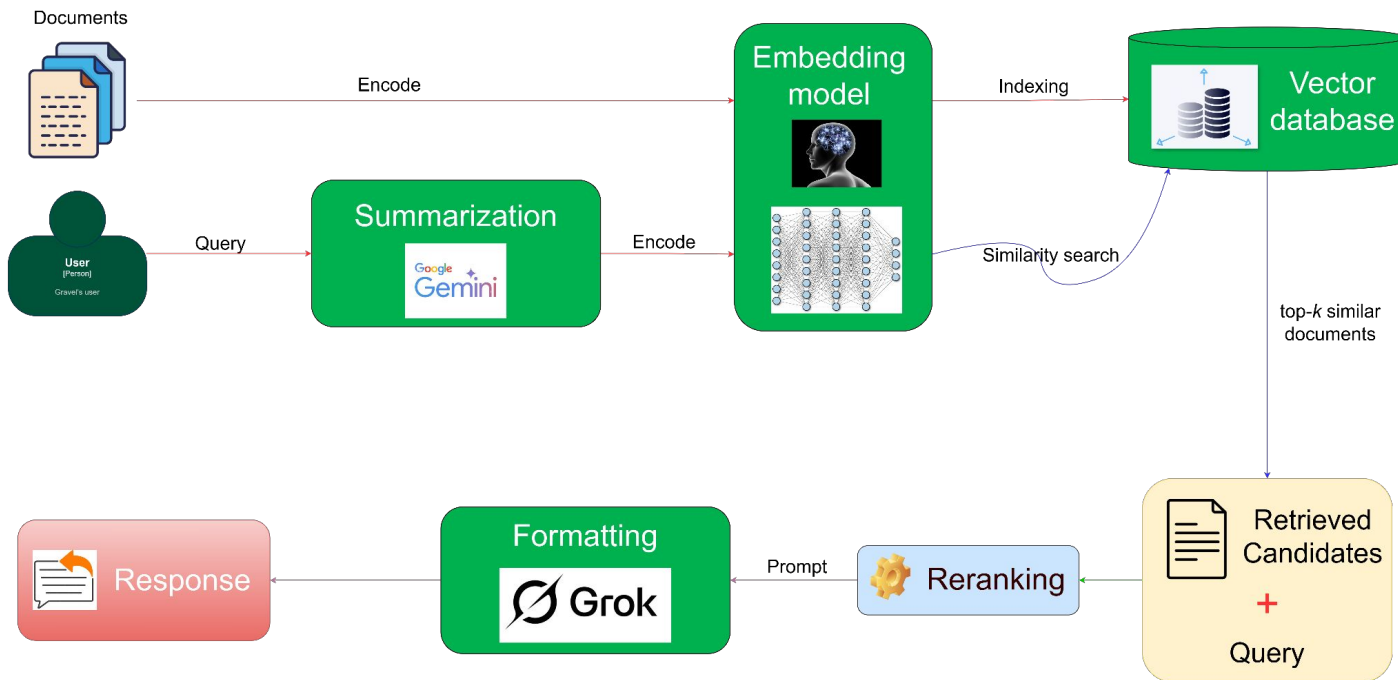
A good schema turns raw data into gold.



Part 5 - AI CORE - Location Service

RESPONSE =

$\text{PostProcessing}(\text{Search}(\text{Process\&Transform}(\text{QUERY}), \text{Process\&Transform}(\text{DOCUMENTS})))$



Part 5 - AI CORE - Location Service

Weights used in our system are **not magic** - hyperparameter optimization

