Case Study - Software

Background

Statista is a German online platform specialized in market data and statistics. It provides over a million statistics on more than 80,000 topics worldwide, covering content such as advertising, purchasing behaviour, and various industry trends. The company's data is aimed at business customers, researchers, and educators, using it to inform market research and reporting.

Statista is seeing increased demand from customers wanting to access this data in LLM-powered applications, and this new team is being established to build developer-friendly solutions for these customers.

For this case-study, imagine Statista wants to expose its catalogue of text-data and statistics in order to serve RAG-like applications, with Statista providing a REST API as a source for these applications to ingest additional context from.

Data

Consider the following TypeScript schema and have an LLM generate some mock data (~100 datapoints) for the purpose of this exercise:

```
interface MockDataSchema {
id: number;
title: string;
subject: string;
description: string;
link: string;
date: Date;
}
```

Coding Task

Candidates should build a small demo HTTP service (in Python, Typescript or Golang) that does the following:

1. **Exposes a GET /find** Endpoint that takes any natural language query and returns an array of the top-5 most relevant items with a similarity score.
2. **Exposes a /stream/find** Endpoint that returns top-10 most relevant items progressively via an HTTP stream.

Preparation time: Please do not spend more than 1-2 hours in preparing this assignment. We are not interested in seeing a sophisticated solution but are rather interested in seeing how you approach the problem.

Before the interview: Please send over the solution via a public github repo 12hrs prior to the interview.

During the interview: Be ready to demo the application and explain limitations in approach at a larger scale. We'll spend 15-20 minutes on this during the interview.

Reflection Task

Assume that you're now working with a more sophisticated iteration of the application above. Assume that we're working with significantly more data which needs to be ingested regularly - and that we have multiple clients using this solution.

During the interview we'll touch upon questions like the ones listed below. Reflect a bit on these beforehand and be prepared to discuss - no need to draft a solution.

- How would you think about testing the application?
- How would you approach observability?
- How would you think about designing a crediting/token system?
- How would you approach client-facing documentation?
- What would you suggest for ensuring a rapid development pace?