Terrascope

Engineering Manager Case Challenge

Context: Carbon accounting platform

Companies are increasingly required to report on their current carbon emissions and set emission targets as part of the global effort to address climate change. In order to quantify how their businesses contribute to climate change, they go through the carbon accounting process to translate business activities into emissions measurements.

For many companies, supply chain emissions form a majority of their emissions (i.e., Scope 3 emissions). Thus, it is important for companies to be able to systematically collect and analyze emissions data from their suppliers to gain visibility into their supply chain emissions and engage their suppliers in decarbonisation initiatives.

As an Engineering Manager at a SaaS carbon accounting platform, you will lead the development of dynamic and responsive web applications. Your responsibilities include optimizing application performance, ensuring scalability, and delivering seamless user interactions. A strong understanding of frontend (React), backend (Python/Node.js), and database technologies is essential.

Challenge:

Design a platform enabling users to upload their business activities. The system will process these activities using AI models to identify and apply relevant emission factors, subsequently calculating the associated emission volumes. Users should be able to view or download these processed results.

- Emission volume calculation involves multiplying the business activity unit and volume by the emission factor's unit and value per unit.
- The system should be scalable and polyglot, able to perform compute-heavy tasks written in multiple languages (NodeJs & Python), aggregate their results, and display them in a paginated, grouped table on the frontend.
- Tasks must be run in parallel, isolated by resource profiles, retried on failure, and monitored in real time.

System Design

Key system components to address:

- **File Storage and Activity Processing:** Handling and storing large volumes of files and activities.
- **AI Processing:** Integrate an external AI Matching Service API (capable of handling up to 10 concurrent requests, each taking approximately 10 minutes) to match business activities with appropriate emission factors.
- **Post Matching Calculation:** Calculate the emission volumes for each activity based on the matched emission factor.
- **Scalability:** Design the system to accommodate a high volume of concurrent uploads and processing tasks.
- **Latency:** Ensure swift file processing and emission factor matching.
- **Result Storage:** Store processed results for future retrieval and analysis.
- **User Management:** Implement robust authentication and authorization mechanisms.

Design Requirements:

- **System Architecture Diagram:** A high-level diagram illustrating the main components (frontend, backend, database, AI processing, storage) and their interactions.
- **API Endpoints:** Define the API endpoints, including request and response formats for key functionalities.
- **Data Schema:** Specify the database schema, including tables/collections and their relationships.
- **File Processing Workflow:** Describe the end-to-end workflow from file upload to processing and result retrieval.
- **Activity Emission Factor Matching Workflow:** Describe the end-to-end workflow from activities ingestion completion after File Processing Workflow to AI matching service to processing and result retrieval.
- **Storage Solution:** Describe the chosen storage solution for handling and storing the uploaded files and processed data.
- **Scalability Considerations:** Outline strategies for scaling the system to handle increased user loads and data volumes, including identifying and addressing potential bottlenecks.
- **Security Measures:** Describe the security measures implemented to protect user data and uploaded files, including authentication, authorization, and data encryption.
- **Technology Stack:** Specify the technologies, frameworks, and tools selected for the frontend, backend, processing, and database.
- **Deployment Strategy:** Explain the proposed deployment approach, including considerations for containerization, orchestration, and CI/CD pipelines.

Deliverables

You may present your solution using a presentation tool of your choice (e.g., Google Slides, Notion document, etc.). The presentation should cover design requirements, and include an implementation and delivery plan - e.g. resourcing, timeline, project management, risks and mitigation measures - to ensure speed and quality of delivery.

You will have 1.5 hours for this session, so as to cover both the technical and non-technical aspects of the case challenge adequately.

As part of our commitment to staying at the forefront of technology, we encourage you to demonstrate how you integrated AI tools into your solutioning process and leveraged them to accelerate innovation and/or elevate engineering decision-making.