Introduction

Welcome to the second round of interviews for the Lead Full Stack Engineer at Concord Compliance, Inc. We are an IT compliance firm that specializes in streamlining how our clients achieve statutory, regulatory, and contractual requirements according to industry and government regulation.

We appreciate your participation and look forward to a constructive and critical conversation regarding your leadership, creativity, and communication skills.

The following challenge describes a hypothetical scenario and is wholly unrelated to our products, goods, and services. Should you have questions or concerns please contact our hiring manager.

Prompt

Vendor A specializes in moving cargo shipments from the Port of Baltimore to various warehouses in the U.S. via a railway system. The railway system offers an API that allows vendors to subscribe to asynchronous push notifications about their cargo shipments by sending tracking numbers. Vendor A wants to build a real-time analytics, alerts, and reporting system for internal staff and customers using this API. Your challenge is to design this system, keeping in mind the business needs and technical constraints.

Railway API Specifics:

1. Vendors can subscribe to asynchronous push notifications by sending tracking numbers to the railway API.
2. Example notifications include:
    - DEPARTED - Cargo has left the Port of Baltimore
    - INTRANSIT - Cargo has arrived at an intermediate station
    - DELAYED - Cargo has been delayed due to specific reasons
    - ARRIVED - Cargo has arrived at the destination warehouse
3. Test case inclusive of all notifications provided on Page 4.
4. Sample API endpoints provided on Page 5.

Challenge Guidelines

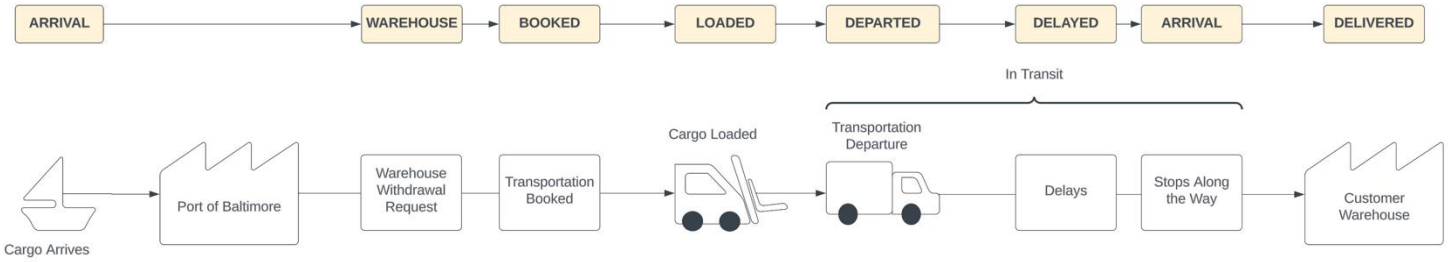
- Objectives:
    1. Demonstrate your ability to achieve executive buy-in by summarizing solutions and costs for achieving customer value and revenue.
    2. Demonstrate your technical understanding of systems design as it applies to a full stack of technology including infrastructure, API, database, and UI.
- Deliverables:
    1. A network diagram illustrating the system's architecture.
    2. A linear business process detailing the flow of information and tasks.
    3. A budget estimation for system setup and ongoing costs, factoring in scaling scenarios.
- Second Round Interview:
    1. Pitch your solution as though you are interacting with an executive panel.
    2. Participate in a Q&A session to discuss your approach to planning, implementing, maintaining, and improving the system over time, especially considering the scaling factors.

Guidelines for Deliverables

1. Network Diagram:
    a. Describe the cloud infrastructure (AWS preferable) that supports your solution.
    b. Consider how you would deploy and maintain ALL aspects of your solution.
    c. Prepare strategies for horizontal and vertical scaling.
    d. Factor in accommodations for high availability, resilience, and recovery.
2. Linear Business Process:
    a. Using a flowchart, describe a series of tasks and activities from start to finish that demonstrate how users will interact with the system to track their cargo.
    b. Factor in business processes and consider batch processing, queuing, or other methods for efficiently handling larger volumes of data.
3. Budget Estimation:
    a. Use a spreadsheet to express your budget.
    b. Provide an estimate of the infrastructure costs based on solutions expressed in your network diagram.
    c. Prepare a timeline describing how to deliver your solution according to team size.

Test Case

The following describes a scenario along with the possible notification codes (CODES) that are sent by the railway system.



1. (ARRIVAL) Vendor A's cargo arrives at Port of Baltimore.
2. (WAREHOUSE) The cargo is entered into a warehouse.
3. (BOOKED) Vendor A submits a request to retrieve their cargo from the warehouse.
    i. Vendor A books the cargo for transport on the railway system.
    ii. An estimated time of arrival (ETA) is issued once the cargo is booked.
4. (LOADED) Vendor A's cargo is loaded onto the transport at the port.
5. (INTRANSIT) Once the transport begins to move, it is in transit until delivered. ETA are issued with every update from the Railway.
    a. (DEPARTED) The transport begins moving along the railway. ETA is unchanged.
    b. (DELAY) The transport is delayed for one day. New ETA is issued.
    c. (ARRIVAL) The transport makes a secondary delivery. Previous ETA unchanged.
6. (DELIVERED) Vendor A receives their cargo at their warehouse.

Scaling Factors

Product Cost Model

- Customer subscription to track shipments;
    - $200/month per 20 shipments tracked
    - $10 for every additional shipment tracked above 20

Assumed Product Growth

- **3 Months**
    - **Customer Base**: 100-200 active customers
    - **Number of Cargo Shipments**: Approximately 1000 shipments to track monthly
- **6 Months**
    - **Customer Base**: 300-500 active customers
    - **Number of Cargo Shipments**: Approximately 5000 shipments to track monthly
- **1 Year**
    - **Customer Base**: 800-1200 active customers
    - **Number of Cargo Shipments**: Approximately 15,000 shipments to track monthly

Possible Team Sizes

- **Team A**: 2-3 members
    - 1 Solutions Architect (coding) - $120,000/year
    - 1 Back-end Developer - $110,000/year
      1 Front-end Developer - $100,000/year
- **Team B**: 5-7 members
    - Additional roles might include:
        - 1 DevOps Engineer - $120,000/year
        - 1 QA Engineer - $90,000/year
        - 1 Additional Back-end - $110,000/year
        - Additional Front-end Developers - $110,000/year
- **Team C**: ~10 members
    - Additions might include:
        - 1 Security Analyst - $120,000/year
        - 1 Project Manager - $90,000/year
        - 2 Additional Data Analysts - $80,000/year

Sample API Endpoints

Below you will find hypothetical APIs to get you started.

Summary

- Subscribe: POST /subscriptions
- Cancel Subscription: DELETE /subscriptions/{subscription_id}
- Get Status: GET /status/{tracking_number}
- Get All Subscriptions: GET /subscriptions?vendor_id={vendor_id}

API Base URL
https://api.railwaytracking.com/v1

Subscribe to Notifications for a Tracking Number

Endpoint:
POST /subscriptions

Request Body:

a. { "tracking_number": "RWT123456789", "vendor_id": "V12345", "callback_url": "https://vendor.com/notify" }

Response:

- **Status Code**: **201 Created**
  a. { "subscription_id": "S123456", "tracking_number": "RWT123456789", "vendor_id": "V12345", "status": "Subscribed" }
- **Status Code**: **400 Bad Request**
  b. { "error": "Invalid parameters." }

Cancel Subscription

Endpoint:
DELETE /subscriptions/{subscription_id}

Response:

- **Status Code**: **200 OK**
  a. { "subscription_id": "S123456", "status": "Cancelled" }
- **Status Code**: **404 Not Found**
  b. { "error": "Subscription not found." }

Get Status of a Tracking Number

Endpoint:
GET /status/{tracking_number}

Response:

- **Status Code**: **200 OK**
    a. { "tracking_number": "RWT123456789", "current_status": "INTRANSIT", "location": "New York Station", "ETA": "2023-08-23T14:30:00Z" }
- **Status Code**: **404 Not Found**
    a. { "error": "Tracking number not found." }

Get All Subscriptions for a Vendor

Endpoint:
GET /subscriptions?vendor_id=V12345

Response:

- **Status Code**: **200 OK**
    a. { "vendor_id": "V12345", "subscriptions": [{ "subscription_id": "S123456", "tracking_number": "RWT123456789", "status": "Subscribed" }, { "subscription_id": "S123457", "tracking_number": "RWT987654321", "status": "Subscribed" }] }