PepsiCo: Advanced Software Engineering Virtual Program

Cole Dombrowski - University of North Texas

TASK 1:

The objective involves formulating a comprehensive design for Augur, PepsiCo's forthcoming predictive supply chain management system, encompassing dashboard metrics, integration specifics, Azure deployment, and scalability considerations. This plan will culminate in a visual depiction of service-level interactions, demonstrating the ability to independently craft and communicate a detailed blueprint for the system's development.

System Architecture Overview:

User Interface (UI) Layer:
- Browser-Based Dashboard: Allows users to interact with the system.

Backend Services:
- Data Ingestion Service: Receives monthly shipment quantities and forecasts for distributors.
- Data Processing Service: Analyzes and processes the received data.
- Storage Service: Stores historical and current data for analysis and retrieval.
- Prediction Service: Utilizes predictive algorithms to forecast future shipment quantities.
- Dashboard API: Provides access to relevant data for the dashboard.

Azure Cloud Deployment:
- Utilizes Azure services for hosting and scaling the system.

Service Interactions:

Data Ingestion Service:
- Listens to messaging queues for monthly shipment quantities and forecasts.
- Upon receiving new data, forwards it to the Data Processing Service.

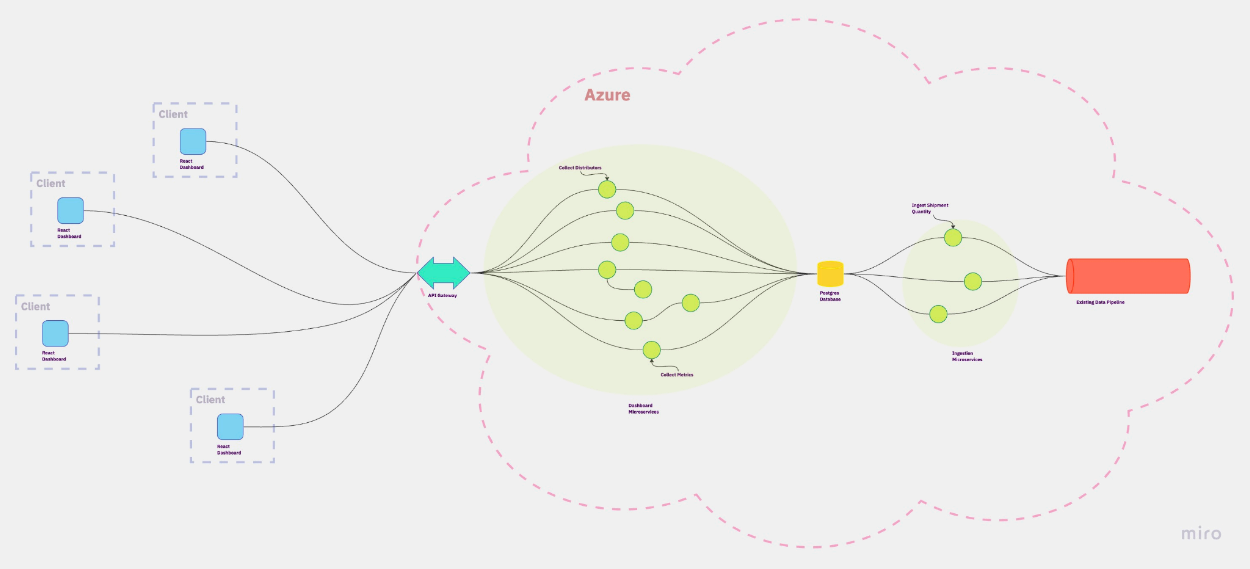
Data Processing Service:
- Receives data from the Ingestion Service.
- Processes and stores the data in the Storage Service.
- Triggers the Prediction Service for forecasting next month's shipments.

Storage Service:
- Stores historical shipment quantities, forecasts, and other relevant data.
- Provides data retrieval to the Dashboard API.

Prediction Service:
- Utilizes historical data and predictive algorithms to forecast future shipments.
- Sends forecasted quantities to the Data Processing Service for storage.

Dashboard API:
- Retrieves data from the Storage Service.
- Provides the necessary metrics for the dashboard.

Visualization from “Forage”:



TASK 2:

The task requires breaking down a system's design into actionable engineering tickets for the team to build Augur, emphasizing the importance of clear delegation and proficiency in Agile workflows at PepsiCo.

Breakdown of engineering tickets required to implement the Augur system:

Sprint 1: Initial Setup and Infrastructure

Ticket 1: Set up Azure Environment
- Create Azure account
- Set up required services (compute, storage, messaging queues)
- Grant necessary access permissions

Ticket 2: Define Data Schema
- Design database schema for storing shipment quantities and forecasts
- Define data models for distributors and shipment data

Ticket 3: Implement Data Ingestion Service
- Set up messaging queue listeners
- Create service to receive and process incoming shipment data
- Validate and store received data in Azure storage

Sprint 2: Backend Services Development

Ticket 4: Develop Data Processing Service
- Build service to process and analyze incoming data
- Implement logic to calculate average quantity of goods shipped

- Store processed data in the defined schema

Ticket 5: Prediction Service Implementation

- Develop service to utilize historical data for forecasting
- Integrate predictive algorithms to forecast next month's shipment quantities
- Validate and test prediction accuracy

Sprint 3: Dashboard and API Development

Ticket 6: Design Dashboard Interface

- Create wireframes/mockups for the dashboard UI
- Define user interactions and visual elements

Ticket 7: Dashboard API Development

- Build API endpoints to retrieve data from storage service
- Implement endpoints to fetch metrics for each distributor
- Ensure API security and data validation

Sprint 4: Integration and Testing

Ticket 8: Integrate Backend Services with Dashboard

- Connect Dashboard API with data processing and prediction services
- Validate data retrieval and accuracy in the dashboard UI

Ticket 9: Testing and Quality Assurance

- Conduct unit tests for each service and API endpoint
- Perform integration testing to ensure seamless communication
- Address any bugs or issues identified during testing

Sprint 5: Documentation and Deployment

Ticket 10: Create System Documentation

- Document system architecture, service interactions, and APIs
- Write user guides and developer documentation

Ticket 11: Deploy Augur to Azure

- Prepare deployment scripts and configurations
- Deploy the system to Azure environment
- Monitor and ensure system stability post-deployment

Sprint 6: Enhancements and Future Development (Stretch Goals)

Ticket 12: Explore Extendibility

- Research and plan for additional functionalities
- Identify areas for future enhancements and scalability