Hyper-ADS (HLD)

Background

PRD: This HLD is in fulfillment of this PRD
This project focuses on providing forecasts and recommendations for businesses looking to optimize their offerings and advertising strategies based on specific event data and weather conditions. The primary objective is to analyze incoming schedules, predict the impact of certain environmental factors, and present actionable suggestions in a user-friendly format.

Problem

Many businesses, especially in the food and retail sectors, struggle to make informed decisions about marketing and promotions when factoring in weather patterns, local events, and evolving customer behavior. Existing solutions often require complex integrations or manual data collation, leading to missed opportunities or late campaign launches. Additionally, concerns around performance (e.g., timeouts in serverless functions), maintaining code across multiple platforms, and overall system complexity can complicate the process further.

By centralizing event data ingestion, forecast modeling, and recommendation output into a single streamlined pipeline, we aim to reduce complexity and latency. This ensures that decision-makers receive timely, concise suggestions without manually parsing large amounts of data or juggling multiple front-end and back-end components.

The main **customer use cases** for this are:

1. Provide small businesses with clear, actionable advice on whether and how to advertise today based on local events and weather.
2. Help businesses understand which specific local events represent the best advertising opportunities.
3. Enable businesses to adapt their advertising messaging based on current and forecasted weather conditions.
4. Provide businesses with customized advertising approaches based on their specific type and location.
5. Alert businesses to sudden, time-sensitive advertising opportunities.

High Level Proposal

3.1 Overview of Architecture

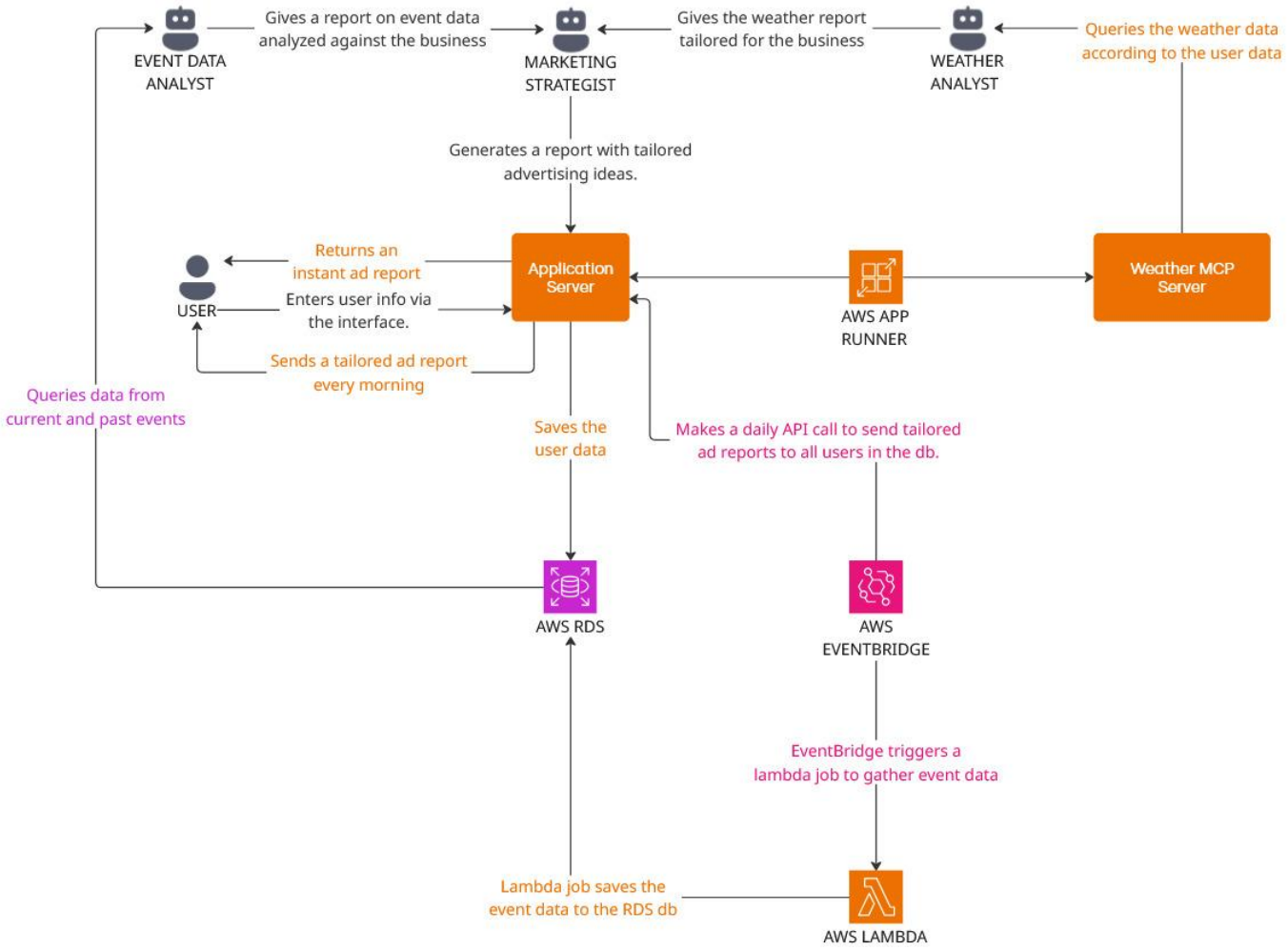
The Hyper ADS Advertising Advisor employs three-tier architecture designed for scalability, reliability, and real-time processing capabilities:

1. Data Ingestion Layer:
   - Event Data Collection: A scheduled AWS Lambda function scrapes EventBrite daily at 1 AM Central Time, extracting local event information including event type, venue location, expected attendance, and timing details for specific postal codes
   - Weather Data Integration: A dedicated Model Context Protocol (MCP) server deployed on AWS App Runner provides real-time weather forecasts and historical weather patterns via RESTful APIs
   - Data Storage: PostgreSQL database hosted on AWS RDS with serverless auto-scaling capabilities stores both event data and user business profiles with automatic cleanup of stale records
2. Processing Layer:
   - AI Agent Framework: Three specialized AI agents powered by Google Gemini models work in sequential coordination:
   - Data Analyst Agent: Analyzes event patterns, attendance trends, and business relevance scoring
   - Weather Analyst Agent: Processes meteorological data to predict consumer behavior impacts
   - Marketing Strategist Agent: Synthesizes insights from both agents to generate actionable advertising recommendations

- Business Rules Engine: Applies business-specific logic based on industry type, location demographics, and historical performance patterns
- Real-time Processing: Server-Sent Events (SSE) enable live progress updates during recommendation generation

3. Output Delivery Layer:

- Web Interface: Flask-based responsive web application with real-time agent workflow visualization
- Email Distribution: Automated daily email delivery system sending personalized recommendations to registered business owners
- API Endpoints: RESTful APIs for programmatic access to recommendations and user management
- The architecture follows microservices principles with clear separation of concerns, enabling independent scaling and maintenance of each component.



3.2 Example Event Data and Forecast

Suppose we gather the following event dataset for a single day:

Event Type: Local festival
Expected Turnout: 500 attendees
Weather: Very sunny and hot outside
Forecast Analysis Output could be something like:

- High midday temperature is likely to increase demand for cold beverages
- Afternoon foot traffic likely elevated due to festival popularity

3.3 Gist of the Recommendation Output (Shortened Example)

"Should we advertise now?
Yes, because today's hot weather and event schedule present a strong opportunity to draw crowds.

Suggested Tactics:

1. Focus on refreshing food and drink specials in the afternoon.
2. Offer brief, targeted discounts (e.g., 'Show event ticket for 10% off').
3. Emphasize benefits of a cool, comfortable place to relax or dine."

This means the application would generate a simple, readable directive combining weather cues and event details to guide marketing or operational decisions.

Justification for Using App Runner

- Instead of deploying a small static webpage on GitHub Pages and coupling it with a Lambda function, we have chosen an AWS App Runner-based server for several key reasons:
- Bias for Action and Simplification: A single service deployment reduces overall complexity by eliminating separate hosting and function coordination.
- Avoiding Lambda Timeouts: Given the potential for heavier data processing, Lambda's execution time limits might be exceeded. A continuously running App Runner service circumvents these constraints.
- Manageable Cost: While there is a small ongoing cost for App Runner compared to free static hosting, this is acceptable to ensure reliable performance and streamlined operations.

Conclusion

This design unifies event data ingestion, forecasting, and actionable recommendations into a single, manageable flow. By leveraging AWS App Runner for hosting and prioritizing simplified architecture, we can deliver timely, data-driven insights even under peak loads or complex scenarios. The forthcoming detailed diagram will illustrate our data flow and integration points, ensuring both technical and non-technical stakeholders can easily understand and support the project.