CLOUD ARCHITECTURE DESIGN – JustGo (A Ride-Hailing Global Firm)

TABLE OF CONTENTS

Introduction

In this article, I will outline the process of designing a robust cloud architecture for ***JustGo***, a global ride-hailing company that is transitioning from on-premises to the cloud. The goal is to enhance our current data strategy to accommodate scalability, optimization, efficiency, and cost-effectiveness. Data is our most crucial asset for tackling business challenges, enhancing customer satisfaction, developing advanced models to predict future outcomes, and bolstering our marketing strategies. This process involves collaborating with various customer groups to determine the type of data we need to collect, how to collect and store it and the processes the data will undergo. This collaboration will aid in building a comprehensive cloud architecture and data pipeline to ensure the success and growth of JustGo.

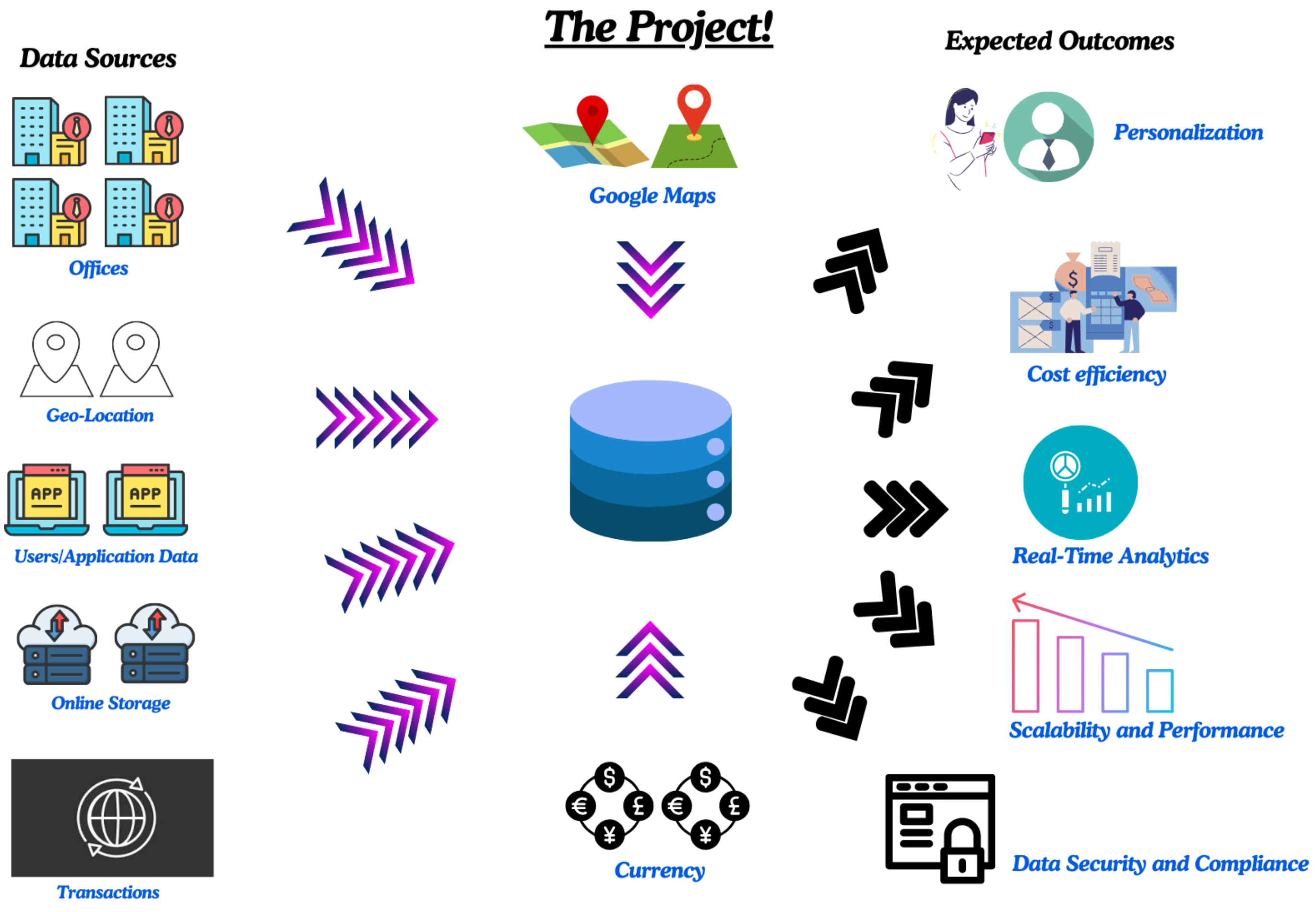
Vision and Objectives

The team at JustGo has conducted a series of meetings and discovery sessions to create a clear overview and intention behind our new robust cloud solution. The data engineering department has actively involved each product owner and customer group in multiple discussion sessions to ensure the right information retrieval, essential for the overall success of the project. With the discussions now complete, the company has established the vision for the project, which will serve as the foundation for our migration from our current on-premises solution to the cloud.

The vision statement is defined below:

"*We want to design and maintain a robust cloud architecture and data flow that ensures seamless scalability, data security, and real-time analytics, empowering our global ride-hailing platform to deliver reliable and personalized experiences to users improving customer satisfaction while adapting to dynamic market demands and technological advancements*"

The vision for our cloud solution is also depicted in the diagram below.



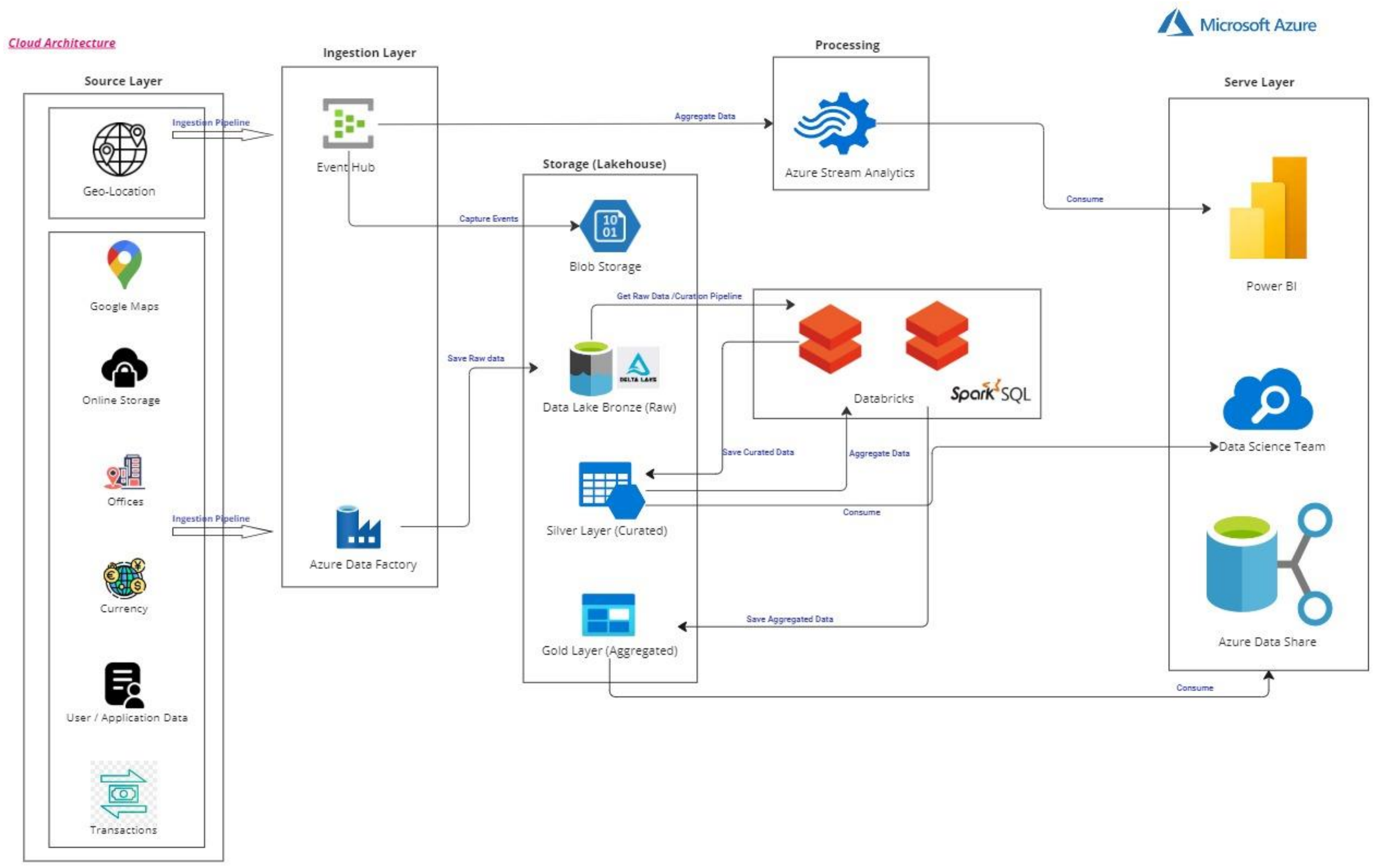
The diagram illustrates the various data sources we will collect and the expected project outcomes from this solution. It explains our expectations from this project and how it will help improve our current processes and boost operational efficiencies.

The Architectural Design

Now that our proposed cloud solution's vision and purpose are clear. It is time to focus on the necessary documentation and build a data architecture that will predict the flow of data from ingestion (how data will be collected) to storage (how data will be stored) to transformation (what and what we will be doing on the data) and finally presentation (how and where data will be used).

Before we delve into the documentation aspects of this project, let us look at the proposed architecture we have designed to implement our cloud solution.

The [proposed] architecture



The Pipeline Design

We have made significant progress in developing our Cloud solution and now have a better understanding of the direction it will take. The proposed architecture design simplifies the systematic process our cloud solution will follow from start to finish. However, we still need a driver mechanism to launch each process of our cloud architecture, starting from phase 1 (usually ingesting data from our sources) until the data is fully transformed and ready for consumption. This implementation is known as the Pipeline design.

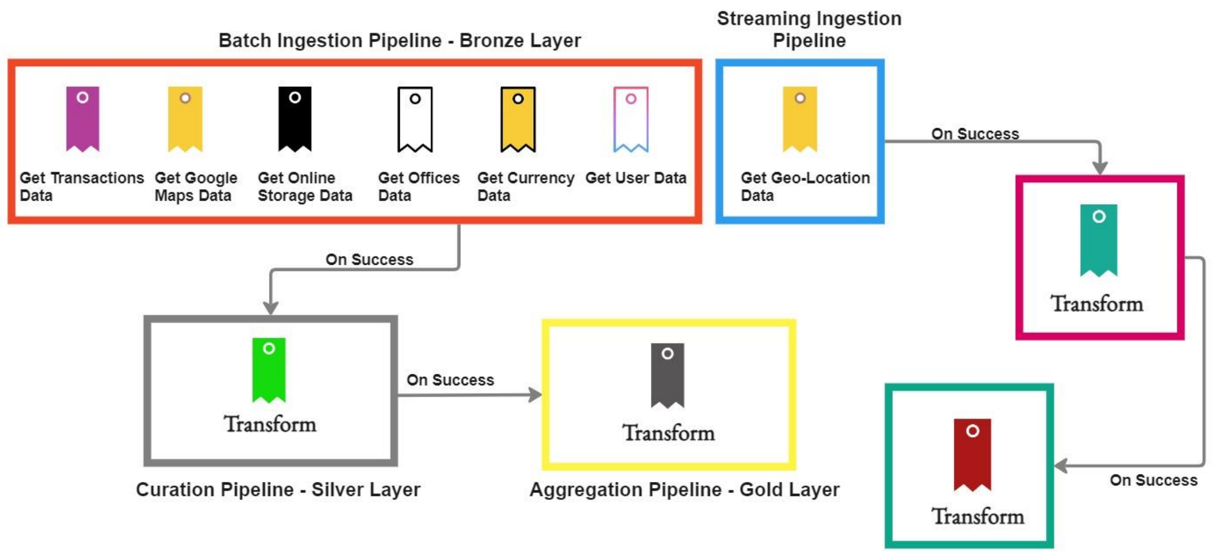
The pipeline design in our cloud architecture is crucial for handling data efficiently from ingestion through transformation and finally to the serve layer. This design ensures that data flows seamlessly across various stages, maintaining high availability, scalability, and performance. Typically, the pipeline design is usually in three (3) stages which we will review shortly.

The three stages of a pipeline design are as follows:

1. **Data Ingestion**: The pipeline begins with data ingestion, where raw data is collected from multiple sources, for example, our data sources at JustGo: Google Maps, Geo-location data, Transactions, Offices, Currency, etc. This data is ingested in real-time or in batches, using scalable cloud services that can handle diverse data types and large volumes.
2. **Data Transformation**: Once ingested, the data enters the transformation stage, where it is cleaned, enriched, and structured to meet the requirements of downstream applications. This stage leverages cloud-based ETL (Extract, Transform, Load) tools and serverless functions to process the data efficiently, ensuring it is ready for analytics, machine learning, or further processing.
3. **Serve Layer**: Finally, the processed data moves to the serve layer, where it is made available to end-users and applications. This layer is designed to support real-time querying and analytics, utilizing data warehouses, databases, or data lakes optimized for quick retrieval and scalability.

The diagram below shows the design of our data pipeline for our cloud architecture which will facilitate the smooth and efficient handling of data from ingestion to its final consumption.

The Pipeline Design



The Deployment Phase

So far everything looks great! Based on our cloud architecture and pipeline design, we will now explain our implementation strategies for this cloud solution from start to finish. This stage is called our deployment or development stage.

In accordance with best practices, the deployment of our data pipeline within our cloud architecture will involve implementing a Lakehouse storage system using CI/CD (continuous integration and continuous development) pipelines. We will utilize Azure Pipelines and Azure Event Hubs for the deployments. But first, let's delve into the concept of Lakehouse storage.

What is a Lakehouse Storage?

A Lakehouse storage combines data lakes and data warehouses, enabling flexible storage and efficient processing of both raw and structured data within a single system. It typically consists of three storage layers: bronze, silver, and gold.

The Journey of Data.

It will be very helpful if we clearly understand what the [bronze, silver, and gold layers] mean in a Lakehouse architecture. This is the backbone of our entire data pipeline and the journey of our data.

A. **Bronze Layer** - Inside a Lakehouse, the bronze layer stores raw data exactly in the same shape, form, and format as it was collected from the data sources. This helps us retain data in its original state for future analysis, ensuring maximum security and other benefits.
B. **Silver Layer** - The silver layer in a data Lakehouse architecture is typically used for storing processed and cleaned data that has undergone initial transformation and enrichment. This layer acts as an intermediate stage between the raw, unprocessed data (bronze layer) and the highly curated, business-ready data (gold layer).
C. **Gold Layer** - The gold layer in a Data Lakehouse architecture is the final stage where data is highly curated, aggregated, and optimized for business intelligence, reporting, and

advanced analytics. This layer contains the most refined and business-ready data, providing critical insights for decision-making.

Now that the storage system is clear. What's NEXT?

The first step in our data pipeline is to identify our data sources. At **JustGo**, we currently have seven (7) different sources:

I. **Geo-location** - Geo-location data is necessary to map the live addresses of drivers to customers based on country of origin. The data is in a *semi-structured format*. These are pushed into Azure Event Hub.
II. **Google Maps** - This download is from Google Maps source.
III. **User/Application Data (mobile & web)** - Customer Data
IV. **Online Storage** - This is stored in a Microsoft SQL Server (3rd Party data).
V. **Offices** – This is stored in a Microsoft SQL Server.
VI. **Currency** - The currency conversion data is required to convert all transactions into USD. The currency conversion data is in a *semi-structured format*. This is downloaded from a REST API.
VII. **Transactions** - Details of customer payments, including payment method, amount, and transaction status.

Step 1: Building our Ingestion Pipelines

Since we now understand the different distribution and format of our data, it's time to define how we will collect data from these multiple sources and move them into our cloud solution.

We created two separate ingestion pipelines:

1. Batch Ingestion Pipeline
2. Stream Ingestion Pipeline

Each pipeline will be created using different Azure services.

- For batch ingestion, we will use **Azure Data Factory**, and
- for streaming ingestion, we will use **Azure Event Hubs Capture**.

Batch Ingestion [Raw Data – Bronze Layer]

We have some data sources where the records change frequently, with some updating on an hourly or daily basis, such as Google Maps, Transactions, Currency, etc.

To consistently have the latest data, we've established a pipeline using **Azure Data Factory** to fetch data from our sources every hour, with a maximum delay of 1 hour. This data pipeline will collect data from the following sources: Google Maps, Transactions, Currency, Online Storage, User/Application Data, and Offices. Incorporating Delta Lake into the bronze layer of our Lakehouse will allow us to carry out ACID (Atomicity, Consistency, Isolation, Durability) transactions, ensuring the accuracy, consistency, and reliability of all ingested data. This means that even in the event of failures, our data remains uncorrupted and intact. By utilizing Delta Lake, we can improve data integrity and fortify the reliability of our data processing pipeline from the outset. The collected data is then stored in **Azure Data Lake Storage Gen2** (bronze layer).

Stream Ingestion [Raw Data – Blob Storage]

During our brainstorming session, we discovered that we could collect coordinates from devices and sensors in our Geo-location data and use them for real-time analysis.

Consequently, we have decided to gather and capture this data using **Azure Event Hubs** for ingestion. We have established a 2-minute time window to capture events, allowing our business to monitor moving assets, track deliveries, and provide location-based services immediately. This will lead to timely decision-making and operational efficiency. The captured events are being successfully stored in **Azure Blob Storage**.

We have stored the ingested data in Azure Blob Storage. However, we will not be reading or streaming the data from this storage. Instead, the data received through Azure Event Hubs will be directly sent to **Azure Stream Analytic**s for processing and analysis of the fast-moving streams of data. Within Azure Stream Analytics, the incoming data will be filtered, aggregated, and transformed using a SQL-like query language.

Finally, the real-time data will be visualized using Power BI. More detailed information will be provided later.

Step 2: Curate Data [This time – Silver Layer]

Once data has been successfully collected from our sources. It is a good practice to read and cross-check the current structure of data. This is called inspection. Our data curation process begins by examining the data we received from our sources. We were looking for data that was not **standardized, invalid, inconsistent, non-uniform, duplicates, or insecure**. We did this by creating a read-only view of the files using programming codes.

This allowed us to understand the current condition of our data.

Clean Data.

Once we identified the current form of our data, we brainstormed and documented the best set of curation procedures to carry out on the data. We chose Azure Databricks to provide an environment to run **Apache Spark** in parallel and **Spark SQL** as our main transformation tools to query, curate, and transform our data.

Once the curation logic was established, we implemented this code within our curation pipeline. The pipeline extracts the data from the bronze layer, curates it, and stores it within **Azure Data Lake Storage Gen2** (silver layer).

Step 3: Aggregate Data [This time – Gold Layer]

Why Do We Want to Aggregate Data?

We did a few transformations in the previous storage layer [**curation layer**], like removing duplicates, standardization, dealing with nulls, etc. However, the purpose of our data, which is hugely based on fast-tracking analysis, requires us to carry out more advanced processing on the data. This layer of data remains our finest data to date.

Aggregating and summarizing our data is a major priority, and that's why we've organized our data to follow a **Lakehouse architecture**. This approach will help us achieve our goals for deploying our cloud solution and data pipeline, which we'll explain more about shortly. **To gain valuable insights and improve business intelligence, we must be able to identify patterns and behaviors in the data by analyzing it against specific variables**.

This is the rationale behind our focus on aggregating and summarizing our data.

Before discussing the type of aggregation, we want to perform on the refined data at this stage, let's review the company's vision for implementing the Lakehouse storage in the Cloud Architecture.

These are the expected outcomes of implementing our data solution at JustGo:

I. Real-time Analytics
II. Personalization
III. Cost Efficiency
IV. Data Security and Compliance
V. Scalability and Performance

What Types of Aggregations Do We Want on the Data?

Now is the time to focus on the type of metrics and aggregations we want to include in the transformed data so far. During our discovery and brainstorming section with all stakeholders, particularly the customer team on the data. We highlighted four (4) key metrics for which we will further summarize and aggregate the data in the silver layer. This was done by **observing the datasets** to understand which ones would be included in the aggregation.

These **KPIs** will help us meet the five key business goals and expectations mentioned earlier.

The Four Key Metrics for Our Aggregation

1. **Total ride completed** – (Number of rides completed within a specific time frame: Daily, weekly, Monthly, City. region?).

2. **Average ride duration** – (Average time taken to complete a ride: Daily, Weekly, Monthly, City, Region?).

3. **Total revenue generated from rides.**

4. **Driver utilization rate** – (Percentage of time drivers are engaged in rides versus idle: Daily, Weekly, Monthly, City, Region?)

Aggregating Data

Now that we're clear on the reason for Aggregating data with our KPIs well defined, we provisioned Azure Databricks to develop an aggregation logic using Spark and SQL codes.

Once the logic has been developed, we then integrate it into our **aggregation pipeline** which runs on a scheduled basis. The Aggregated data is stored within **Azure Data Lake Storage Gen2 (Gold layer).**

Understanding Data Consumption

We're getting close to the end of our cloud architecture layout and strategy. Now that we're done with our data storage, curation, and aggregation, Let's talk about how the data in our Lakehouse will be used.

How will the data be used?

Now that we have data in three (3) different forms (raw, curated, and aggregated data). We need to document how data in each layer of our storage will be used in line with business rules to help us achieve our business goals and objectives.

1. **Build Report and Display Data using Power BI –** The data in the gold layer [*cleanest form of data*] will be pushed to Power BI to create meaningful reports for business groups, customers apps, and executives. Additionally, we'll stream live data from Azure Stream Analytics [*streaming data*] directly to Power BI for real-time analytics and decision-making.

2. **Azure Data Share –** We are authorized to publish data from the gold and silver layers because they store clean and secure data. We will share the aggregated data from the Gold Layer with the public or third parties through APIs for their use. This will allow us to monetize our data.

3. **Data Science Team –** One significant advantage of the Lakehouse and the data transformation that has occurred so far is the ease of querying and sharing data with other end users, such as data or business analysts, and the data science team. This enables them to carry out ad-hoc queries or develop advanced models that meet their needs. We will provide the aforementioned end users with access to the data in the silver layer for advanced querying capabilities and analysis.

Conclusion & Best Practices

In transitioning to a cloud architecture, JustGo is poised to harness the full potential of its data, driving scalability, efficiency, and innovation. By implementing a robust data pipeline and adopting a Lakehouse architecture, we ensure that our data journey—from ingestion to transformation and finally to consumption—is seamless and secure. This strategic move not only strengthens our data infrastructure but also enhances our ability to deliver personalized, real-time experiences to our customers, positioning JustGo for sustained growth and success in an increasingly competitive market.

Best Practices

1. **Data Integrity and Quality**: Implement ACID-compliant systems like Delta Lake in the bronze layer to maintain data accuracy and consistency, even in the event of failures.

2. **Scalability and Flexibility**: Design your cloud architecture to be scalable, accommodating future growth and evolving business needs without compromising performance.

3. **Real-Time Processing**: Utilize stream processing tools, such as Azure Stream Analytics, to enable real-time data analysis and decision-making, enhancing operational efficiency.

4. **Data Security**: Ensure that data at all stages, from raw ingestion to final aggregation, is securely stored and handled, adhering to compliance standards.

5. **Documentation and Collaboration**: Maintain clear and comprehensive documentation throughout the pipeline to facilitate collaboration among teams and ensure alignment with business objectives.

By following these best practices, JustGo can achieve a resilient, high-performing cloud architecture that supports its mission to deliver exceptional, data-driven services to its global user base.