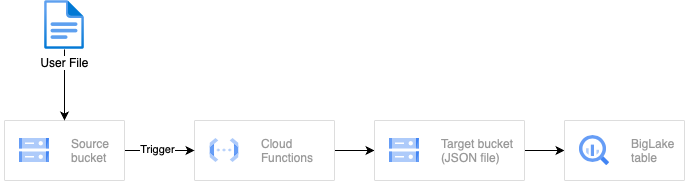
## Overview

This document describes the technical architecture for transferring data from a CSV file dropped in a GCP bucket. The data will be converted to JSON format and placed in a secondary GCP bucket, then loaded into a BigLake table in BigQuery.

## Technical Architecture



**Source Bucket:**

The entry point for the entire workflow is this Google Cloud Storage (GCS) bucket, where a user file is placed via a designated process.

**Google Cloud Functions (Trigger):**

This cloud function is triggered by the event of a new file upload in the source GCS bucket. This cloud function is responsible for following process

* Verify the file type of the dropped file in the source GCS bucket is CSV.
* Convert the source CSV file to JSON format and load it into the Target GCS bucket.
* Create or update the BigLake table using the JSON files in the Target GCS bucket.

**Target Bucket:**

This is the Google Cloud Storage(GCS) bucket where the converted JSON files from the sourced CSV files are stored. This bucket is also source for the BigLake table

**BigLake Table:**

This is a BigQuery external table which sources JSON files located in the Target GCS bucket.

## **Design Considerations**

1. **Scalability**: The architecture leverages Google Cloud Functions' event-driven processing to scale with the volume of CSV files in the source bucket, enabling concurrent handling of multiple files.
2. **Security**: Implementing appropriate IAM roles and permissions for GCS, Cloud Functions, and BigQuery ensures data access and modification is restricted to authorized users and services.
3. **Data Validation**: Checks for file type check and schema compliance to load the data into BigLake table. If the file format or data schema are changed, code changes will be required to absorb the changes.
4. **Cost Management**: Strategies for monitoring usage, optimizing data transfer and queries can be employed to manage costs associated with Google Cloud Storage, Cloud Functions, and BigQuery resource utilization.

## Assumptions

1. The source files, which trigger the conversion process, are in CSV format.
2. The schema, or arrangement of data columns, within the source files remains consistent.
3. The desired output format resulting from the file conversion is JSON.
4. If the source file replaces the previous existing source file, the data in the BigLake table will reflect the new latest data records.
5. The source and target GCS buckets are already created and the Cloud Function has access to the GCS buckets
6. The dataset in the BigQuery is already created and the Cloud Function has access to the dataset to create tables.