# **CASE STUDY –** Cloud Engineer

This assignment is designed to give you a glimpse of some of the challenges you will be facing in this role. Please be aware there are no perfect solutions - for us, it's more important to see how you find solutions, process your ideas, structure your thoughts and how you make your decision paths.

Be creative but realistic about what's possible. We are thrilled to get to know a bit more about the way you solve tasks.

Are you up for the challenge?

**SUBMISSION DEADLINE:** As indicated in the email.

## THE CHALLENGE –

The accommodation catalog in trivago comes from different sources. Partners and other contributors deliver data periodically to us and in different formats, that we call imports.

This data is very heterogeneous. For example, some partners specify the facilities and amenities that their hotels have, others do the same in another shape, others simply don’t provide them at all.

Our challenge is to collect all these imports, assure that we receive the data as we aligned with them, funnel this data into a common shape, consolidate every bit of information and deliver the output to the rest of the company, ready to be consumed.

## Scenario

A new project is requested to our team. We need to process hotel data coming from different partners. Our Product Owner has agreed on a JSON schema with them for the incoming and outgoing data. Our team is responsible to come up with a design and development of a new pipeline that ingests the partner’s import data, group them together and consolidate them.

The solution needs to be built on GCP or AWS. We will continuously receive data files in jsonl format(each line in the file is a json object) in an Google Cloud Storage bucket. This data needs to be picked up, consolidated and delivered as JSONL files in an output storage bucket. It’s up to you to pick the technologies for the purpose, taking into account that the maintainability, the performance and the costs are well balanced.

### Consolidation process

Consolidation is the process of grouping together records from multiple partners.

ex: partner B's record would have the accommodation name as "The Grand Budapest Hotel", whereas partner C's record would have the accommodation name as "The Budapest Hotel". We need to pick ONE name as per a defined consolidation logic.

The consolidation logic we use for this assignment is called "priority". We define an ordered list of partners.

```
1. Partner_A
2. Partner_B
3. Partner_C
```

We try to pick the data from the highest ranked partner. In our case, the correct accommodation name would be "The Grand Budapest Hotel". Since we do not have any records from `Partner_A`, we pick the accommodation name from `Partner_B` .

## Tech Details

- The input storage follows these rules:
  - prefix: [root]/import/year=%Y/month=%m/day=%d/hour=%H/[input_object-999].jsonl
  - The received JSONL files will be between 0KB and 2MB
  - The input comes from another team and we don't have control over their input generation process
- The volume of data received will be between 5GB and 20GB daily, spread through the day.
- The number of records will be around 10k to 10 million.
- The output files **must** follow these rules:
  - The output files need to be in JSONL format.
  - Each output file size should not exceed 500KB.
  - The files need to be delivered in a way that the team should easily be able to retrieve the data for each day of a month.
- The pipeline and its components must not be triggered manually

Task 1 - Design the pipeline

We would like you to come up with the design for the data pipeline that picks up the records from partners and consolidates them.

- A design of your approach.
- Designed to run on GCP or AWS.
- Along with the architecture, please explain how would you orchestrate pipeline.
- Which cloud components would you use and why this choice.
- How these components will be triggered.
- How would you monitor the pipeline.
- How would you deploy the pipeline.

**Deliverables (in a zip file)**

- A document in Markdown (.md) explaining the design
- Additional images for diagrams if needed

Task 2 - Build a consolidator

Now that you have the pipeline designed, it's time to write some code. Remember the "priority" consolidation which was explained earlier, we would like you create a component which does this.

This task is only about building a single component which can run on your local machine, and not the entire pipeline which was described in Task 1.

**Notes**

- The code should be readable and extendable
- Add sufficient documentation on how to execute your consolidator
- Consolidate only the field accommodation_name, other fields can be ignored
- The input array of partner records can contain between 1 and 1000 records
- The input of priority list can contain between 1 and 100 partner names

**Deliverables**

- A software component (ex: python module, jar file, docker image) which can consolidate partner records
- The component should be readable, extendable and testable
- Documentation on how to execute the component

## Sample input and output

Input #1 - an array of partner records

```
[
  {
  "partner_name": "Partner_B",
  "accommodation_id" "101",
  "accommodation_data":{
      "accommodation_name":"The Grand Budapest Hotel",
      "accommodation_address":"Budastrasse 71",
      "accommodation_geocodes":"101,102",
      "accommodation_city":"Budapest"
    }
  },
  {
  "partner_name": "Partner_C",
  "accommodation_id" "101",
  "accommodation_data":{
      "accommodation_name":"The Budapest Hotel",
      "accommodation_address":"Budastrasse 88",
      "accommodation_geocodes":"101,103",
```

```
      "accommodation_city":"Budapest"
    }
  }
]
```

Input #2 - priority list

```
"Partner_A"
"Partner_B"
"Partner_C"
```

Output #1 - consolidated accommodation object

```
{
  "accommodation_id" "101",
  "accommodation_data":{
      "accommodation_name":"The Grand Budapest Hotel"
    }
  }
```

Good luck! 😊