**PART 2**

**Problem Statement:**

In the second part of the assessment, imagine that size of the JSON input has increased significantly, APIs will not work for this approach, and we decided to use a messaging system. Please implement a solution based on messaging if time permits. If not, at least share a design with a short paragraph on how you would solve the issue of scale and timeouts present in APIs.

**Assumptions:**

1. While upload, all the following validations are in place for the uploaded file
   1. File size limit (depending on system capacity and speed) KBs or MBs
   2. File first record must be the headers match with the template in system
   3. Record level columns validation is an optional decision by business while upload (this will slows down the upload API response respectively depending on the file size)

My recommendation is not to have data level validation while upload.

1. Base64 encoded file upload through API

**Solution:**

**Systems to Include in solution:**

1. IBM API Connect
2. Sterling file gateway
3. MW ESB layer for mapping of information based on channels
4. Relational DB
5. BPM layer (optional if B2B is involved in the solution)
6. Target system to consume the uploaded file

**Solution description:**

CQRS based design is suitable for separating writes and reads of the information and a working sample of event based design is provided in shared git repository at a very basic level (which needs running axon system to run without any issues)

COMMAND

In this approach, the API pushes the file to the **UploadFile** queue along with a unique identifier which will be sent back in successful response for future reads/tracking

MQ will have a consumer from MW layer which monitors the queue and process the incoming files and inserts the file entry to the database (**Files**) table along with status at file level (Uploaded, Validated, Failed Validation, Processing, Processed, Failed, etc) as per business needs. Must keep channel, partner timestamps and other details needed.

Here further solution can be implemented in MW layer or a BPM can be introduced based on solution scope is B2B to create subprocesses based on the client partner code to ease the customizations based on organization requirements of processing. Once the file’s records starts processing, the entries need to be kept in the DB (**FileRecords**) with a unique reference for each record of the file associated with file unique ID and a status need to be maintained for the records (Initiated, Processing, Hold, Processed, DataError, etc as per the needs and transactions requirements)

Each record transaction will be fed to target systems (payment or transfers, etc) and the system responses will be stored in respective record status and retries if required as per business needs.

Note: Need to consider the configuration of DBs for cursor counts as per the payload capacity, file size, records limit per file

Archive or delete DB entries once processed after keeping them for number of days or months as per backup policy.

QUERY

There must be APIs to read the information at file level and at record level based on input parameters such as file ID, record ID, file upload date and processed dates if required, file status, etc.