Frameworks and tools:

OpenCV (Image Processing): OpenCV is a library of programming functions tool for image processing and performing computer vision tasks. The library is cross-platform and free for use under the open-source Apache 2 License.

Flask (API Implementation): Flask is used for developing web applications using python, there is a built-in development server and a fast debugger provided.

GRPC (API): gRPC is a modern open source high performance Remote Procedure Call (RPC) framework that can run in any environment.

Swagger (API Documentation): It is a web-based API documentation framework. It is used to create interactive documents for APIs which are built to serve a specific purpose.

Design Pattern:

As we are building an image processing pipeline consisting of several image transformations and output of first transformation is given to subsequent transformation, I will be implementing the Chain-of-responsibility Design Pattern for creating a pipeline of image processing phases.

Assumptions:

1. File Format supported: *.jpeg, *.png, *.tiff, *.bmp
2. Max File Size: 15MB
3. Gray scale operation cannot be performed multiple times.

API Details:

1. **Name: Image Transforming service**

   HTTP Method: POST

   Description: This API is responsible for performing image processing steps on input image and resultant image will be sent as API response. The order of transformation is decided by user and image processing is done in the same sequence.

   URL: protocol://domainOrIPAddress:port/processimage

   Request Body: File + JSON

   Response: Output Image

   Example:

   Datas = [{"name": "ROTATE", "direction": "RIGHT"}, {"name": "ROTATE", "direction": "LEFT"}, {"name": "ROTATE_BY_ANGLE", "degree": 190}, {"name": "RESIZE", "height": 400, "width": 250}, {"name": "GRAYSCALE"}, {"name": "FLIP"}, {"name": "THUMBNAIL"}]

   file= 'user_picture.jpg'

url = http://localhost:5000/transformimage

http -> post(url, files=file, data=json)

System Context Diagram: