Technical Challenge

Back End Challenge

You are a senior member of a team that has been tasked with developing a programmatic image storage and processing service called ProgImage

Unlike other image storage services that have a web front-end and target end-users, ProgImage is designed as a specialised image storage and processing engine to be used by other applications, and will (only) provide high-performance programmatic access via its API.

Apart from bulk image storage and retrieval, ProgImage provides a number of image processing and transformation capabilities such as compression, rotation, a variety of filters, thumbnail creation, and masking.

These capabilities are all delivered as a set of high-performance web-services that can operate on images provided as data in a request, operate on a remote image via a URL, or on images that are already in the repository. All of the processing features should be able to operate in bulk, and at significant scale.

Required

1. Build a simple microservice that can receive an uploaded image and return a unique identifier for the uploaded image that can be used subsequently to retrieve the image.

2. Extend the microservice so that different image formats can be returned by using a different image file type as an extension on the image request URL.

3. Write a series of automated tests that test the image upload, download and file format conversion capabilities.

Stretch

1. Write a series of microservices for each type of image transformation. Coordinate the various services using a runtime virtualisation or containerisation technology of your choice.

2. Design a language-specific API shim (in the language of your choice) for ProgImage as a reusable library (eg Ruby Gem, Node Package, etc). The library should provide a clean and simple programmatic interface that would allow a client back-end application to talk to the ProgImage service. The library should be idiomatic for the target language.