Technical Design Document

1. Tech Stack

Programming language : Python 3.7
Libraries : OpenCV 3.4.5, pytesseract 0.2.6
Database : Amazon dynamoDB
Framework : Flask for python
Infrastructure : AWS API gateway, AWS lambda

2. Accounts and Infrastructure

2.1 Development

For primary development we are going to use our own local machines to develop base tool (classifying images and extracting text from OBD II images). Once the basic functionality is achieved we are going to develop web service on top of it. For training and developing model we are going to use AWS Service. To use AWS services we need and account and this has to be discussed with ACV.

2.2 Production

We are going to host REST web service using AWS API gateway and AWS lambda. REST api service will be a part of API gateway and model/code goes into AWS lambda. To use this services we need AWS account.

Data Sources, Models, Timing

Whenever vehicle inspector inspects car he/she takes pictures of OBD II reader and those pictures are stored on ACV's persistent data store (database). This images are associated with auction IDs. We feed this images data to image classification module and the text extraction module to train the models. Our exposed REST endpoint secured with basic auth will take an image as input (POST request) and image is sent to the server.

1.1 Data Sources

Data will be generated by Vehicle inspectors while inspecting vehicle. After clicking pictures images are sent to backend server and database. An existing dataset of the images are used for training and testing the model. Training data would be the initial set of images which would be available through Amazon Simple Storage Service.

1.2 Data Models and Structure

The structure of the OBD II reader images is dynamic depending upon the screen of display. The structure will be determined based on the modes available in the OBD-II reader. OBD II extracted codes will be displayed in the following JSON format.

Eg. P0171 - System too lean, EVAP - evaporative emission

```
{
    auction_id  : “xsdg2324bhddf”,
    mil_status : “ON”,
    readiness_supported : 1
    readiness_completed: 3
    readiness_not_supported: 4
    datastream_supported : 20
    Ignition_type: “spark”
    protocol_type: “CAN”

}
```

1.3 Timing

Data lives forever in ACV’s data store. Currently ACV has OBD II reader images data from last couple of years. Information obtained from OBD II reader will enable them to predict actual cash value for cars.

System architecture diagram :