CS455 – Cloud Computing
Project3

Objective

In this project you will use a "hybrid architecture" (similar to Project2) to create a solution to a problem that require the use of two Machine Learning algorithms. You will use the AWS Rekognition service along with other services.

Description:

Design a system that helps cities use cameras (installed at roads intersections) to catch and bill drivers that commit traffic violations at traffic lights. Such systems are already in use and are likely to be complex and involve challenging problems. However, we will make a few assumptions to make the problem tractable and suitable for a class project.

You can assume the following:

- The system will be installed in cities and towns in the state of California.
- A California license plate number is a combination of 7 characters and digits (A to Z and 0 to 9). And the state does not allow custom plate numbers. An example California license plate is shown in Figure 1.

Figure 1: Example California license plate: 7 characters long that consist of characters [A to Z] and digits [0 to 9].

- Cameras installed at intersections have Internet access and can communicate with the cloud.
- Cities and towns have budget shortfalls and want to save money. They decided not to use standard mail and stamps for mailing violation notices. All violations will be delivered to offenders via emails.
- All vehicles drivers have email accounts. And this information is in the Department of Motor Vehicle (DMV) database.
- The DMV database is not in the cloud. It is located in a local network at the state department of transportation.
- California has a lot of immigrants and international residents. Most speak excellent English, but a few don't. It was found that when people don't understand violation emails they tend to

ignore them. To increase the rate of collection, emails will be sent in the driver's preferred communication language (this is also available in the DMV database).

- The California DMV does not have access to DMV databases of other states. The license plates of non-California vehicles that are caught violating traffic rules must be placed in a dedicated S3 bucket to be manually processed.

- When a camera detects a traffic violation, it takes a picture of the front or rear of the offending vehicle (depends on the vehicle orientation with respect to the camera). The picture is first sent to an image processing software that extracts the plate from the picture. **As far as your system is concerned, you will always get "clean" license plates (Figure 2)**.
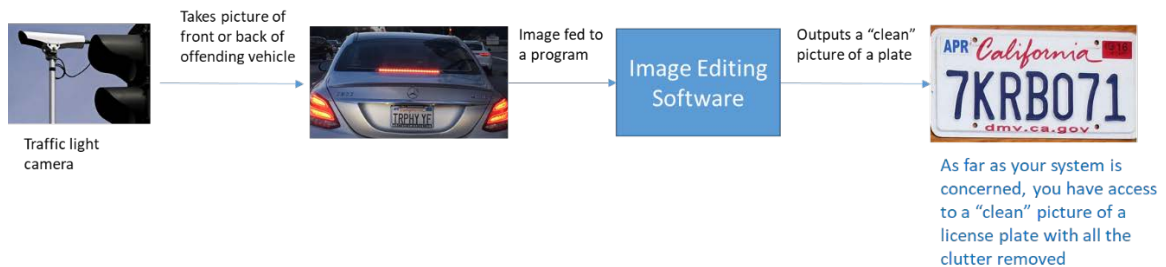

Figure 2: License plates are extracted out of pictures. This preprocessing happens prior to the plate arriving to your system. Your system only deals with "clean" plates.

- When the camera uploads the picture to the S3 bucket, it adds with the object 3 metadata (as S3 object tags):

Tag	Description and Value
Location	Intersection address (e.g., "Main St and 116th AVE intersection, Bellevue")
DateTime	The date and time the violation took place.
Type	The type of violation. Three types are supported: ➔ no_stop: vehicle didn't stop at the red light. ➔ no_full_stop_on_right: vehicle didn't come to a full stop before taking a right on red. ➔ no_right_on red: right turn on red not allowed.

- The type of violation determines the ticket amount the driver must pay:

  no_stop: $300.00
  no_full_stop_on_right: $75.00
  no_right_on_red: $125.00

- The email should have the following format:

  Your vehicle was involved in a traffic violation. Please pay the specified ticket amount by 30 days:

Note that only the blue portion needs to be translated to a language other than English. The green portion will always be in English.

For example:

Your vehicle was involved in a traffic violation. Please pay the specified ticket amount by 30 days:

Vehicle: Blue Ford Escort
License plate: BXT7765
Date: 5/26/2022 2:25:47 PM
Violation address: Main St and 116th AVE intersection, Bellevue
Vilation type: no_right_on_red
Ticket amount: $125

Build a solution such that when a license plate picture arrives to an S3 bucket, the corresponding vehicle owner is sent an email in his/her preferred language. If a license plate is not a California license plate (e.g., an out of state vehicle), the plate picture must be uploaded to a specific bucket (to be manually processed). Figure 3 shows the system workflow diagram (I also enclosed Figure 3 as a separate PDF in case you want to print it as a guide while working on the project).
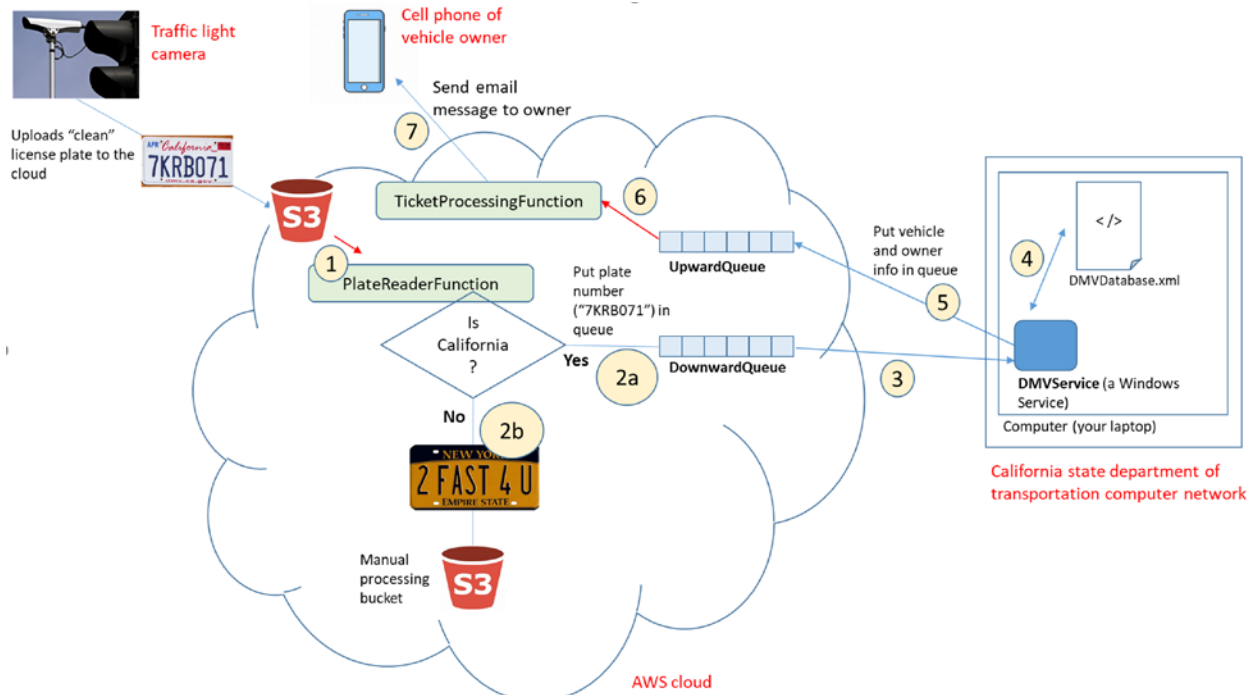
Figure 3: System workflow diagram.

Implementation Hints:

- Some parts of this project have strong resemblance to parts in Project2. Borrow code from your Project2 and fill in the blanks where needed.

- The California DMV database is the enclosed **DMVDatabase.xml** file.

  - Open the enclosed **DMVDatabase.xml** file.
  - Change the <contact>...</contact> elements to include your email and your partner email. This way you can test the messaging system.

- 8 license plates were given to you for testing (see folder **LicencePlates**). 5 are California plates and 3 are non-California plates.

- You can use any AWS service that you like and any programming language. For role, use the LabRole.

- You may find this function helpful:

```
/// <summary>
/// Gets an image from a stream object.
/// </summary>
/// <param name="s">The stream object</param>
/// <returns>The Image</returns>
1 reference
private Image GetImage(Stream s)
{
    Image image = new Image();

    MemoryStream ms = new MemoryStream();
    s.CopyTo(ms);

    image.Bytes = ms;
    return image;
}
```

To use the above function you need to add the following:

```
using System.IO;
using Amazon.Rekognition.Model;
```

- You may need to have a function that checks if a string is made up of capital letters and numbers (remember we are making the assumption that a California plate is made up of 7 alphanumerical characters - uppercase capital letters).

```
private bool IsCapitalLettersAndNumbers(string s)
{
    // Your code here
}
```

Grading rubric:

- (80 %) You can demo the solution and it works start-to-end (from dropping a license plate into S3 to getting a text message or email).
- (20 %) You can explain your code.

What to Submit:

Nothing to submit. You meet with me for 5 minutes and demo your work.