Terminology Overview

The following are definitions that are potentially helpful when listening to our presentation. We have provided simple definitions as well as links to more in-depth explanations. The definitions are provided

Machine Learning

SSIO - SightScale Input/Output, our module that runs on the trucks and sends images to the cloud

Machine Learning Model - A piece of code that has been trained to recognize certain patterns in data.

Classifier - A specific type of machine learning model that is specialized at sorting data into a fixed number of named groups, such as apples, oranges, etc.

Backend - Code running on the server

The user does not interact with it directly.

API - Application Programming Interface. In our case, having a web API, it is a separate module of code that connects to the database and answers requests made by our Front-End.

NestJS - An opinionated framework designed to help you build scalable APIs.

RBAC - Role-Based Access Control. An authorization system that gives users access to an API based on their assigned roles within the organization.

Authentication and Authorization - Authentication is when a user signs into a website, and the website verifies their login information to make sure they are who they say they are. Authorization occurs after the website knows who the user is. It is concerned with whether the user should have the permission to reach a specified page/endpoint.

JWT - JSON Web Token. A small *signed* document that is given to our front-end by Amazon's authentication service. The fact that it is signed guarantees that it is not tampered with. It specifies information about the user, such as their username, which is used in our RBAC system to check their permissions.

Frontend - The User Interface of the application.

The user interacts with this code on their browser, mobile device, or other type of app.

React - A Front-end Library that provides a method of writing HTML-like syntax within normal JavaScript or TypeScript code. Allows created components to be reused

JavaScript - A dynamically typed language written originally to be executed by web browsers to make dynamic web pages.

TypeScript - A statically typed superset of JavaScript that gives JavaScript the advantages of a strongly-typed compiled language.

HTML - HyperText Markup Language. The most basic building block of the Web. It defines the meaning and structure of web content.

AWS - Amazon Web Services.

Amazon's collection of cloud services offered to developers and other technical users.

Lambda - A method for invoking functions such as APIs. Involves AWS spinning up a running instance only when it is needed, and shutting off / repurposing the resources when the code is not needed.

RDS - Relational Database Service. Amazon's implementation of a scalable database service.

S3 - Simple Storage Service. Allows for storage and retrieval of files. Our front-end uses S3 to store its files and serve them to browsers.

SQS - Simple Queue Service. Allows various types of messages to be stored in a queue and dealt with by other services. The results of our contaminant classifier are placed in an SQS queue to be processed by another lambda and put in the database.

VPC - Virtual Private Cloud. Allows Amazon cloud resources to be placed

EC2 - Elastic Compute Cloud. Used for virtual computing. And used for our machine learning classification and training.

CloudFront - Distributes web content to users

Route53 - A scalable domain name system

DNS - Domain Name System. Links website names like google.ca to the content they are providing.

Database - Organized collection of information stored in rows similar in ways to Excel but used by many users and has set columns that are used.

SQL / MySQL - Structured Query Language. It is a programming language to easily access data from a database.

Sequelize - an object-relational mapping tool (ORM) that allows you to query the database by creating the SQL queries for you.

CI/CD - Continuous Integration and Continuous Deployment

Continuous Integration is a way to check your code with linting, testing and other checks every time updates are made to the main versions that developers are working off of. It provides fast feedback to developers when the changes they have made do not meet the current standards so that the issues can be fixed to ensure better code quality.

Continuous deployment (sometimes called delivery) happens after code is integrated and has passed all of the quality checks. Continuous deployment deploys changes to AWS so that the application used by users will be updated.

Linting - Checks code to make sure it matches set styles. This can be naming conventions like: variableName vs variable_name. This helps to ensure code is more maintainable, readable and consistent.

Testing - Testing is used to check that code is running properly and providing expected responses. It adds reliability to the system and catches errors that might not have been easily found.

Test coverage - Test coverage is a metric of how much code your tests cover. It lets you know if there are conditions that you are missing in your tests and how many times something is tested.

Test Suite - A test suite is a set of related tests that run together at once.

CloudFormation -

Pair Programming - Pair Programming is programming together to talk over ideas, catch errors and use the experience of two people on a problem.

Code Reviews - Code reviews have other developers look over code that someone has written. This allows code to be improved by having someone not directly involved in the code find missed errors or potential improvements to add after it has been committed.

Response Times - How long it takes data to be received once it has been requested.

Batch Processing - Processing reports and data in groups to reduce response time and the size of data being transferred.

Indexing - Making the database more efficient by changing the order of rows so when data is retrieved it doesn't have to check each row.