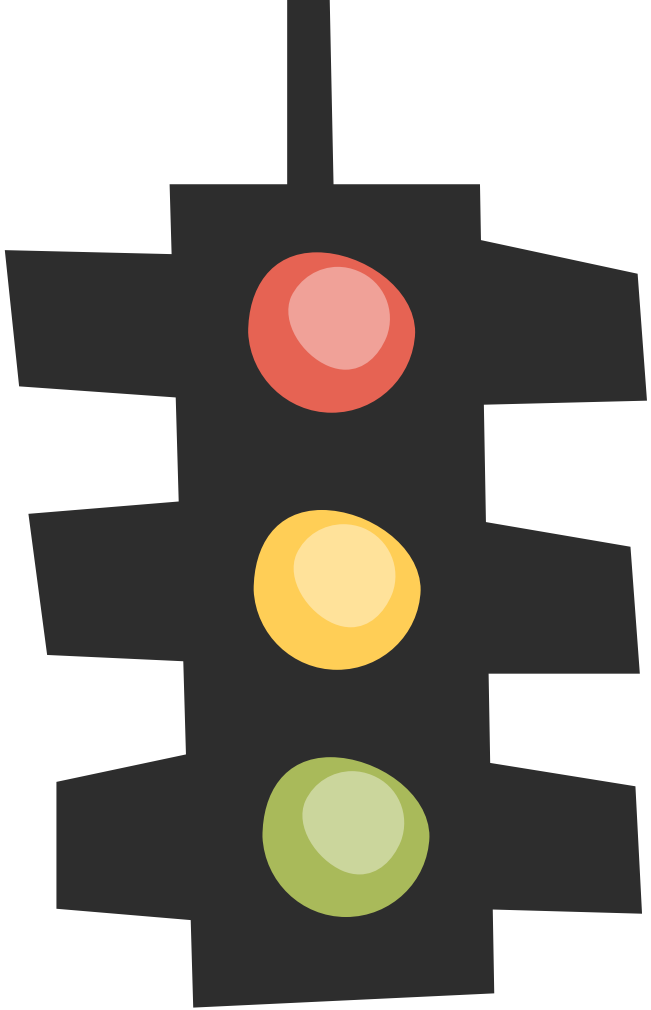


aws

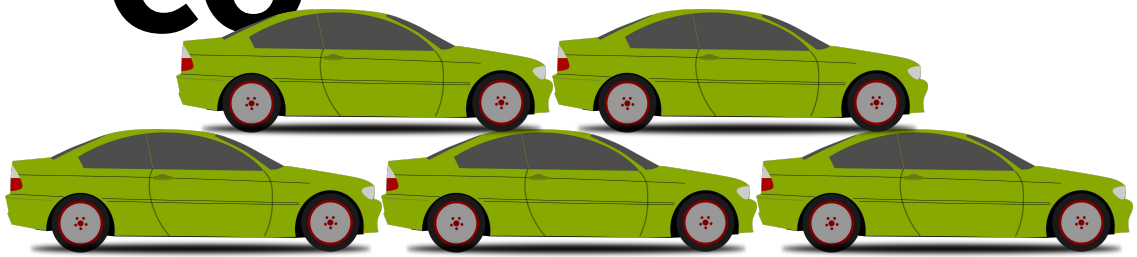
Gr  n Light

Neha Balamurugan, Brooke Bolsinger, Aron Chu, Junayd Lateef, Jayashree Adivarahan

Green Light API provides the access to data needed to integrate traffic lights, which minimizes idling, abrupt accelerations, and congestion thus reducing carbon emissions

• simple solution to a huge problem •

30 million tons of CO



5 million cars worth of CO_2 per year

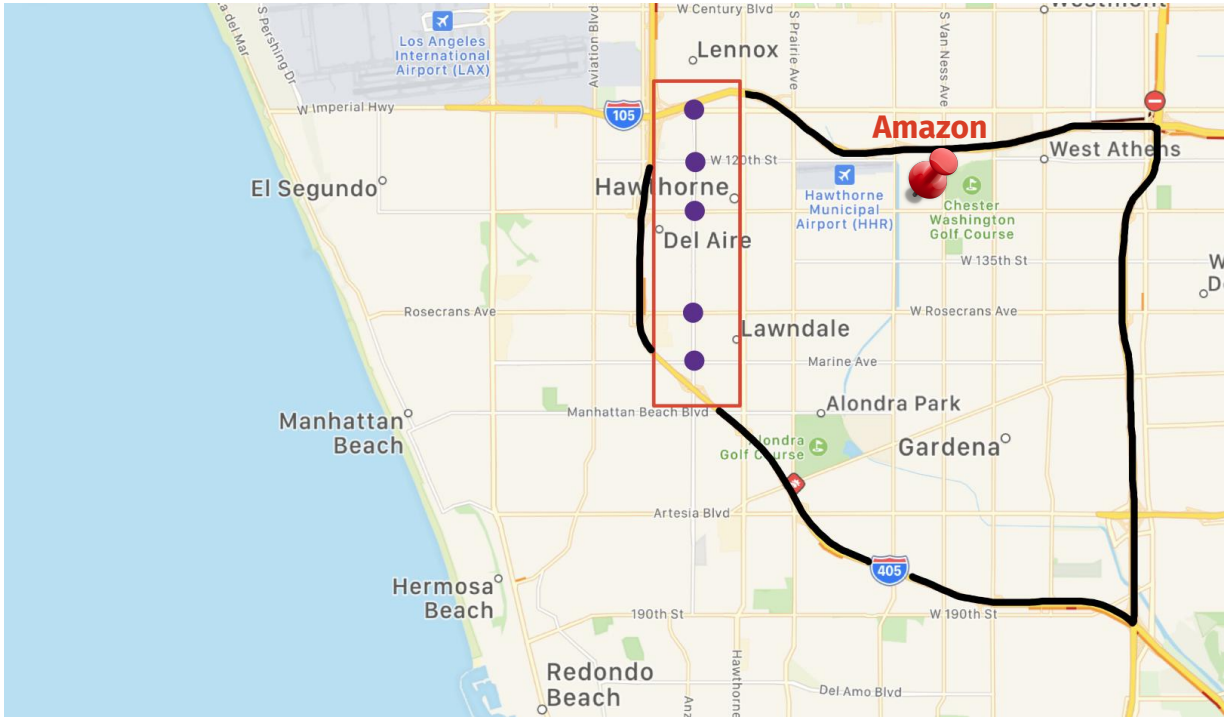
- High Idling Time
- Road Conditions
- Weather Conditions
- Abrupt Accelerations

Latitude, Longitude Coordinate of Intersection

Number, Type of Vehicle

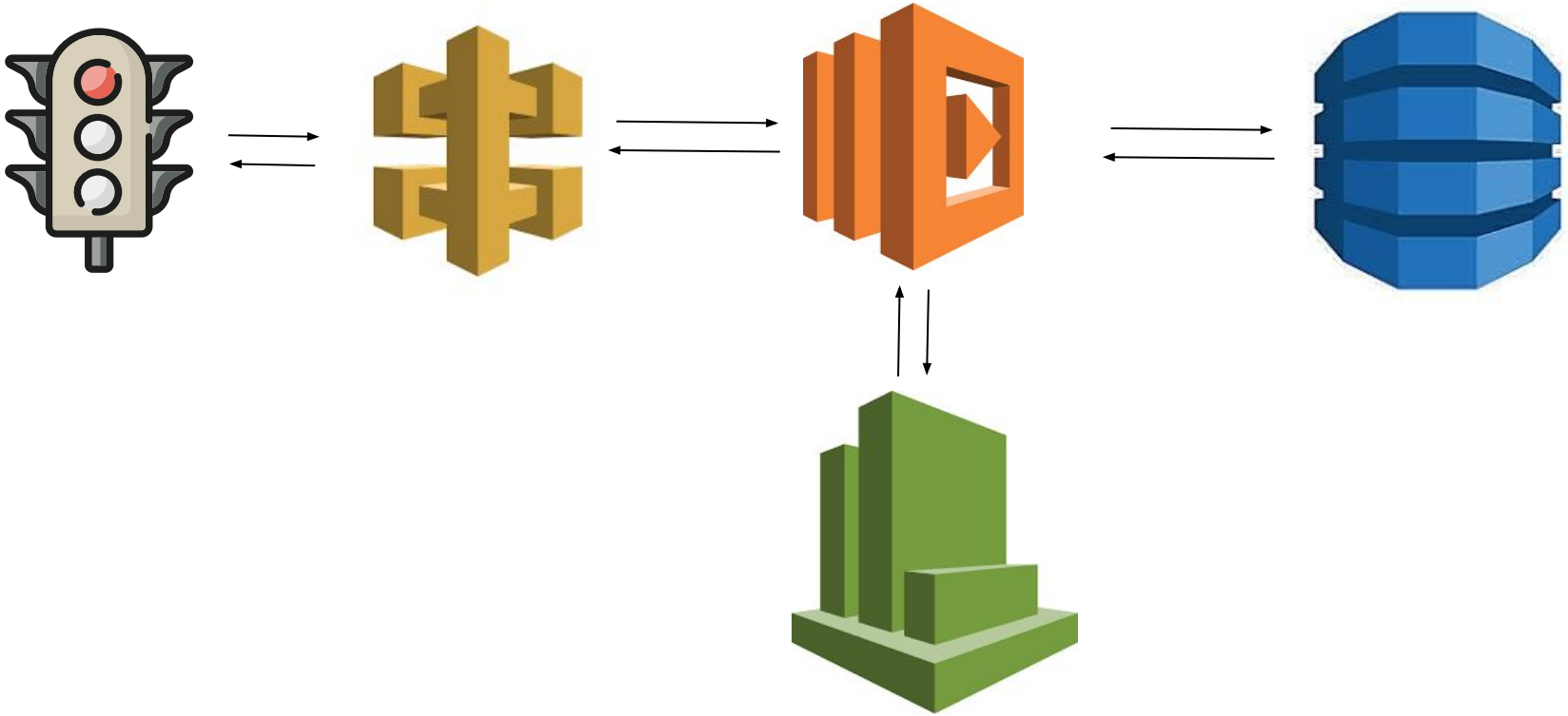
Road: Pavement, Friction

Wait time of vehicles



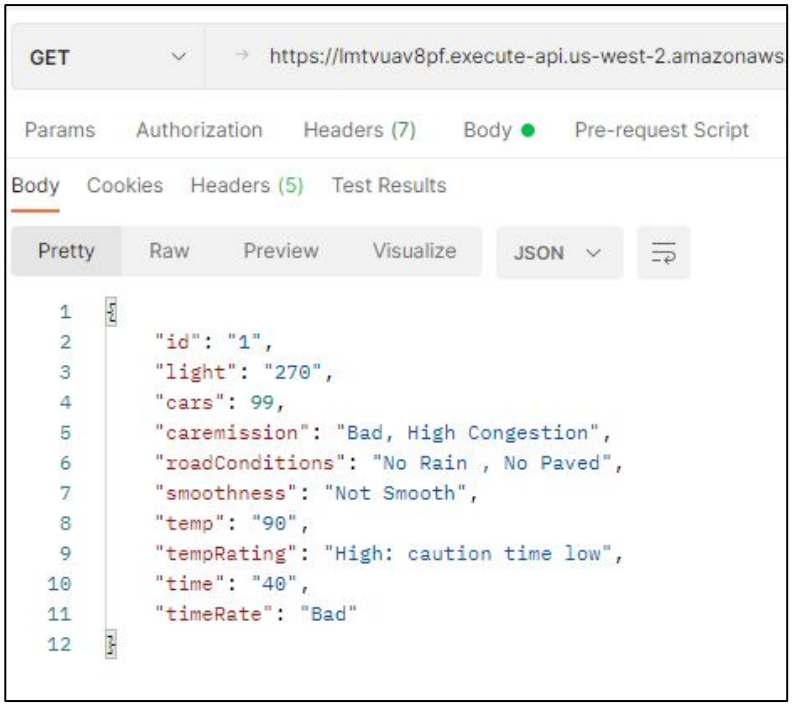
Green Light

Architecture Diagrams



Green Light

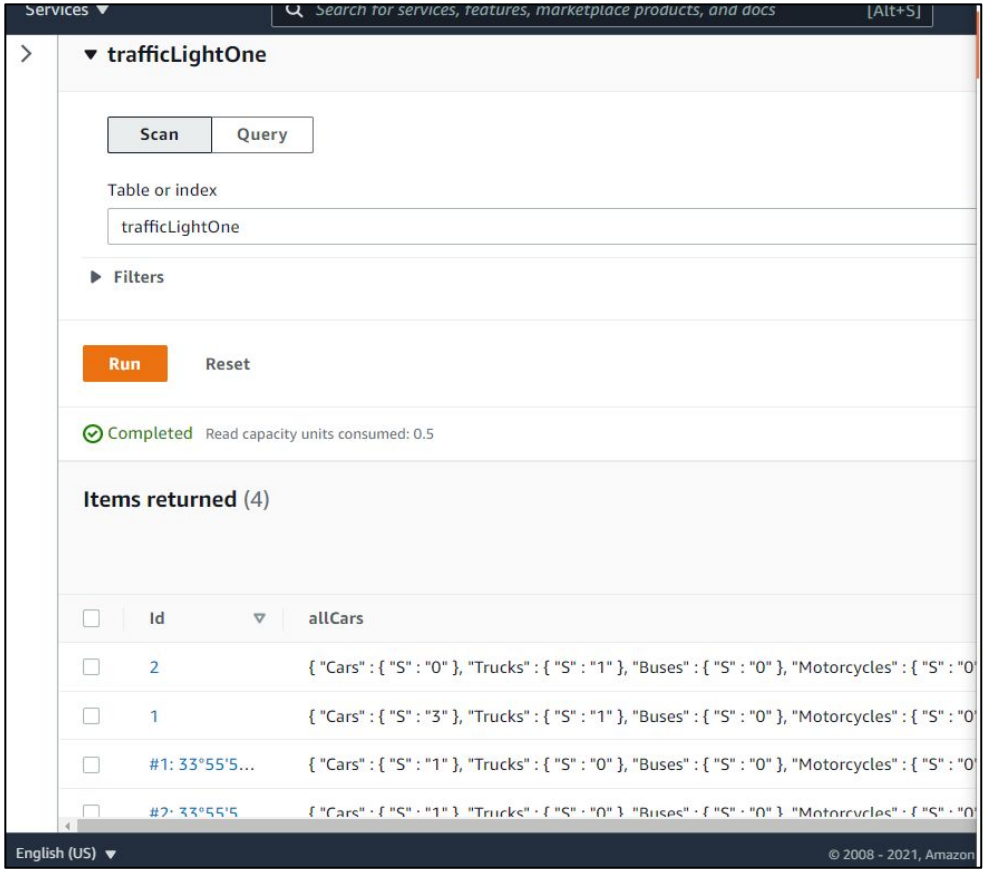
Single Get

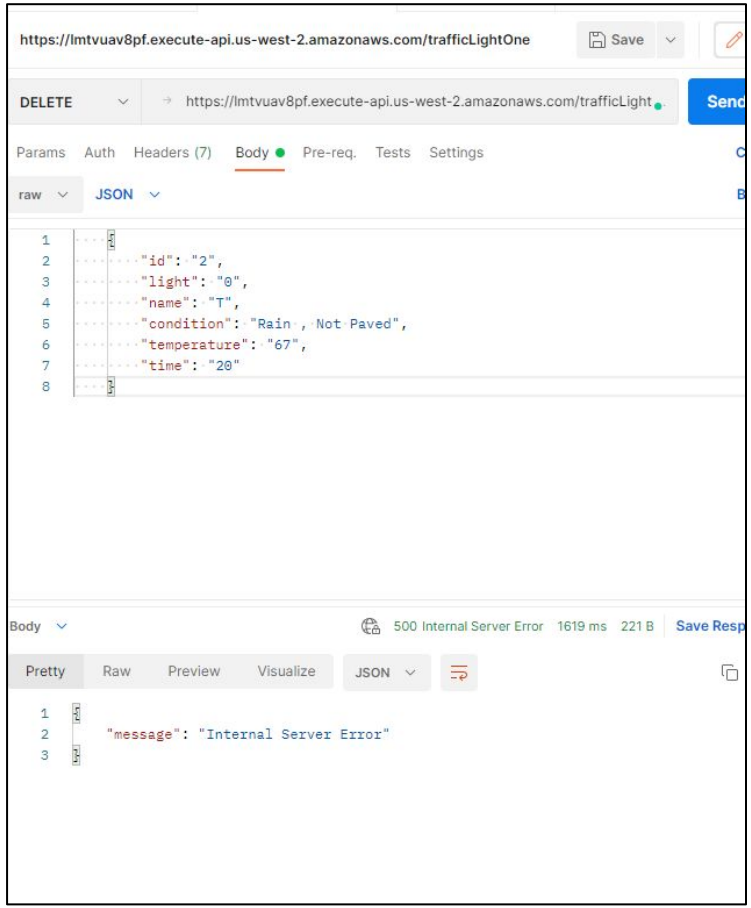


Multiple Gets

Green Light

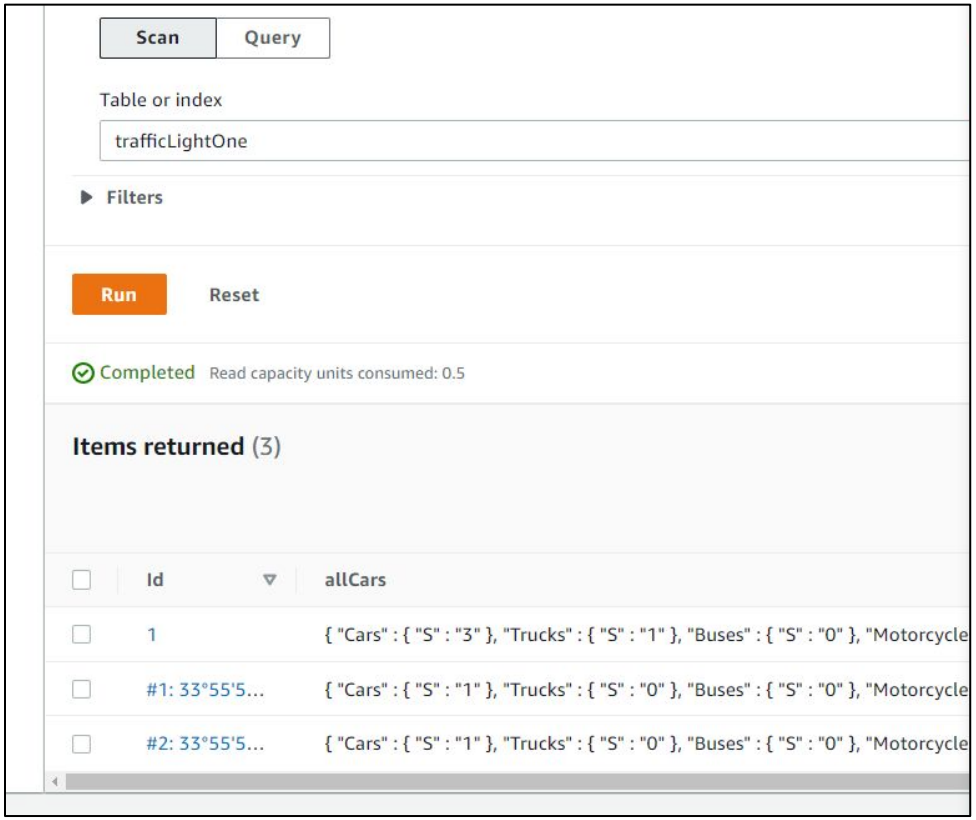
Before Delete

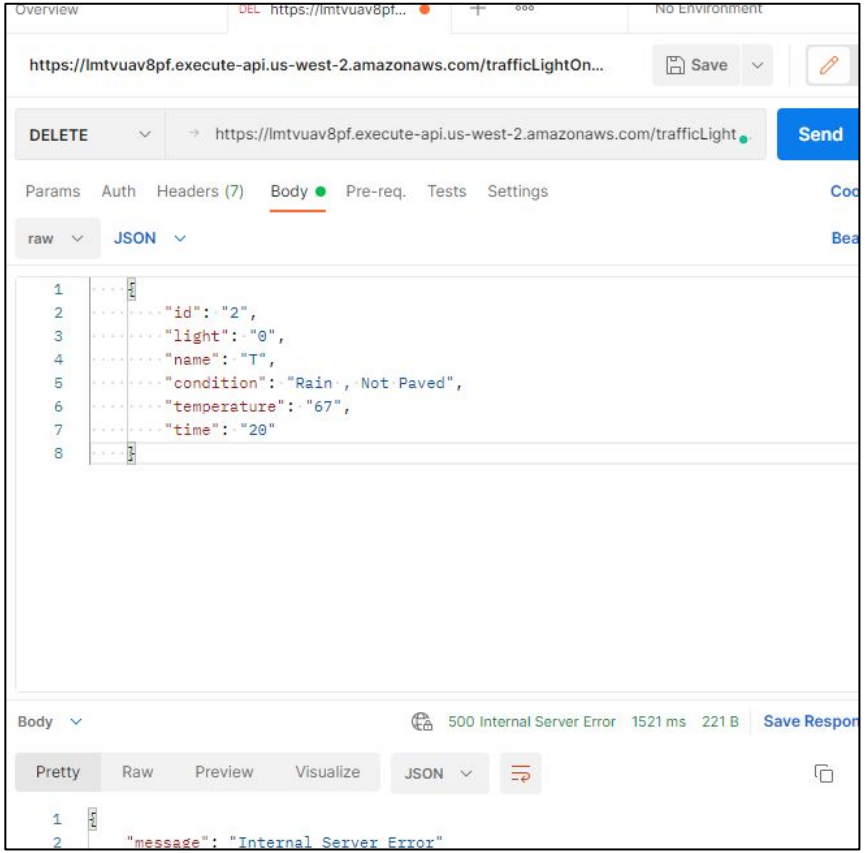




Green Light

After Delete





Green Light

AWS Lambda Conditionals Code

```
    case 'POST':
        let requestBody = JSON.parse(event.body);
        let id = requestBody.id;
        let name = requestBody.name;
        let temperature = requestBody.temperature;
        let condition = requestBody.condition;
        let time = requestBody.time;
        let light = requestBody.light;
        let smoothness;
        let tempRating;
        let timeRating;
        let emissionnumber;
        let emissionstatus;
        counter++;

        emissionnumber = (c*1+m*2+t*3+b*4);
        if (emissionnumber >= 45) {
            emissionstatus = "Bad, High Congestion";
        }
        if (emissionnumber>20 && emissionnumber<45) {
            emissionstatus = "Ok, Medium Congestion";
        }
        if (emissionnumber<=20) {
            emissionstatus = "Good, Low Congestion";
        }

        if (temp != id) {
            c = 0;
            t = 0;
            b = 0;
            m = 0;
        }
```

AWS Lambda Conditionals Code

```
temp = id;
if (time <= 12) {
    timeRating = "Good";
}
if (time > 12 && time < 20) {
    timeRating = "Ok";
}
if (time >= 20) {
    timeRating = "Bad";
}
if (temperature >= 80 ){
    tempRating = "High: caution time low";
}
if (temperature >= 50 && temperature <= 80 ){
    tempRating = "Medium: caution time medium";
}
if (temperature < 50 ){
    tempRating = "Low: caution time high";
}

if (condition.substring(0, 4) == "Rain" && condition.substring(7) == "Paved") {
    smoothness = "Very smooth";
}
if (condition.substring(0, 4) == "Rain" && condition.substring(7) != "Paved") {
    smoothness = "Smooth";
}
if (condition.substring(0, 4) != "Rain" && condition.substring(10) == "Paved") {
    smoothness = "Smooth";
}
if (condition.substring(0, 4) != "Rain" && condition.substring(7) != "Paved") {
    smoothness = "Not Smooth";
}
if (name == "C") {
    c++;
}
if (name == "T") {
    t++;
}
if (name == "B") {
    b++;
}
if (name == "M") {
    m++;
}
```

JSON Input File

```
{
    "id": "#1: 33°55'51.4'N 118°21'40.8'W",
    "light": "0°",
    "name": "C",
    "condition": "Rain , Not Paved",
    "temperature": "67",
    "time": "20"
}
```

DynamoDB Table

Id	allCars	caremiss...	LightAngle	roadCon...	smoothn...	tempera...	tempRat...	timeRating	timeWai...
#2 33°55'2...	{ "Cars" : { "S" : "12" }, "Trucks" : { "S" : "0" }, "B...	Ok, Mediu...	90	No Rain , Pa...	Not Smooth	15	Low: cautio...	Bad	45
2	{ "Cars" : { "S" : "0" }, "Trucks" : { "S" : "1" }, "Bus...	Good, Low ...	270	No Rain , N...	Not Smooth	90	High: cauti...	Bad	40
#4 33°55'2...	{ "Cars" : { "S" : "17" }, "Trucks" : { "S" : "4" }, "B...	Bad, High C...	270	No Rain , N...	Not Smooth	90	High: cauti...	Bad	40
#2 233°55'...	{ "Cars" : { "S" : "0" }, "Trucks" : { "S" : "5" }, "Bus...	Good, Low ...	180°	Rain , Not P...	Smooth	35	Low: cautio...	Bad	21
1	{ "Cars" : { "S" : "0" }, "Trucks" : { "S" : "1" }, "Bus...	Bad, High C...	270	No Rain , N...	Not Smooth	90	High: cauti...	Bad	40
#1 33°55'2...	{ "Cars" : { "S" : "18" }, "Trucks" : { "S" : "7" }, "B...	Bad, High C...	0	Rain , Paved	Very smooth	90	High: cauti...	Bad	45
#1 233°55'...	{ "Cars" : { "S" : "1" }, "Trucks" : { "S" : "3" }, "Bus...	Good, Low ...	0°	No Rain , Pa...	Not Smooth	35	Low: cautio...	Ok	15
#3 233°55'...	{ "Cars" : { "S" : "0" }, "Trucks" : { "S" : "2" }, "Bus...	Good, Low ...	90	No Rain , Pa...	Not Smooth	35	Low: cautio...	Good	10
#3 33°55'2...	{ "Cars" : { "S" : "5" }, "Trucks" : { "S" : "0" }, "Bus...	Good, Low ...	180	No Rain , N...	Not Smooth	50	Medium: ca...	Good	10
#4 233°55'...	{ "Cars" : { "S" : "5" }, "Trucks" : { "S" : "2" }, "Bus...	Ok, Mediu...	270	Rain , Paved	Very smooth	70	Medium: ca...	Good	10

Calculate most efficient traffic flow

Real-time sensor readings
(1) Timing sensor
(2) Caution light
(3) Vehicle type identification

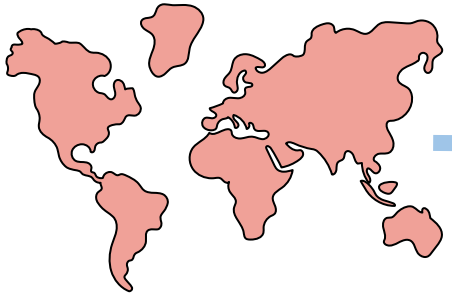
Local weather API

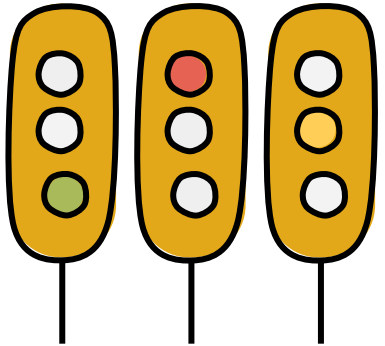
More data points
(1) Speed limit before light
(2) Toward/Away from pollution hotspot

API

Large, specific datasets

AI/ML Algorithm





Green Light

Pillars of Green Light

Enforce

System change creates big impact, easy to enforce

Scalable

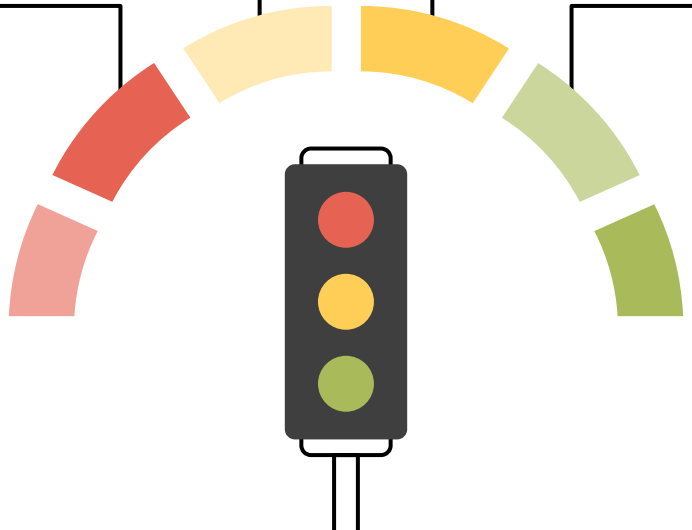
Can be scalable to multiple different regions

Cheap

Integrable into current tech with minimal cost

Comfortable

User benefits: high mileage, less spending fuel, lower time



Thank You