PROJECT SILVAFLUX

Design Review

Big Data Computing and Interface for Tropical Forest Regeneration

By Team Clean Carbon

NAU

Team Introduction

Meet the team:

Curtis McHone - Team Lead
Justin Stouffer
Richard McCue
Jonathan Bloom
Shayne Sellner

Team Sponsor:

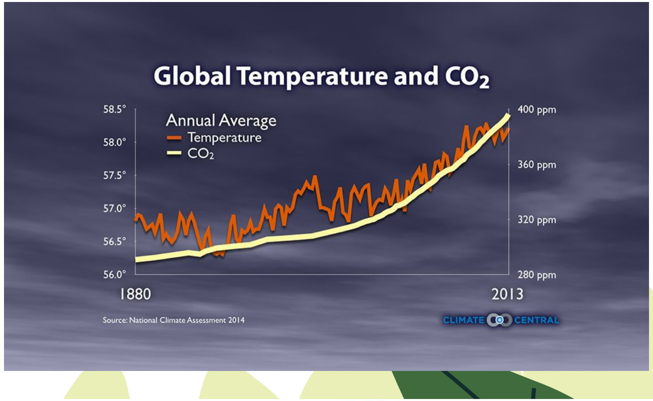
Allie (Alexander) Shenkin

Team Mentor:

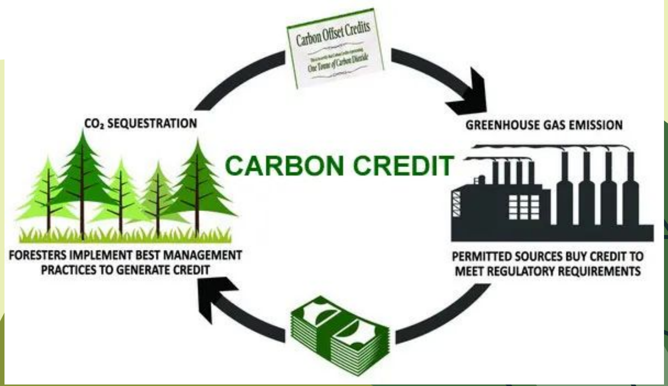
Vahid Nikoonejad Fard

Project Introduction

- Researchers are seeing higher levels of carbon dioxide and other greenhouse gases in our atmosphere
- When greater amounts of these gases get trapped in our atmosphere, we start to see temperature levels rise globally
- Many companies have began investing in carbon credit based projects
- Carbon credits are a purchasable “credit” developers can sell based off how much carbon dioxide they remove from the atmosphere
- Our sponsor Allie (Alexander) Shenkin and his team have discovered a new climate cooling technology that allows for 30% more carbon credits per designated plot of land


Global Temperature and CO_2

Source: National Climate Assessment 2014


CARBON CREDIT

CO_2 SEQUESTRATION

GREENHOUSE GAS EMISSION

FORESTERS IMPLEMENT BEST MANAGEMENT PRACTICES TO GENERATE CREDIT

PERMITTED SOURCES BUY CREDIT TO MEET REGULATORY REQUIREMENTS

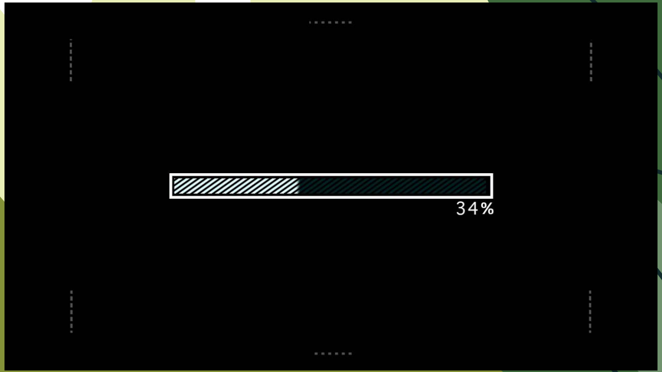
Problem

- This discovery makes investing in carbon credit based projects much more profitable, helping not only the investor but the planet

- However, the current software that is used to calculate the amount of carbon credits per plot of land is simply too slow

- Our team has been asked to develop an application that will run much faster than the current design, as well as providing users with a fluid, user-friendly interface

- To fix some of the bottlenecks the current prototype has, we envision transferring the backend to either Monsoon, or AWS



Solution Overview

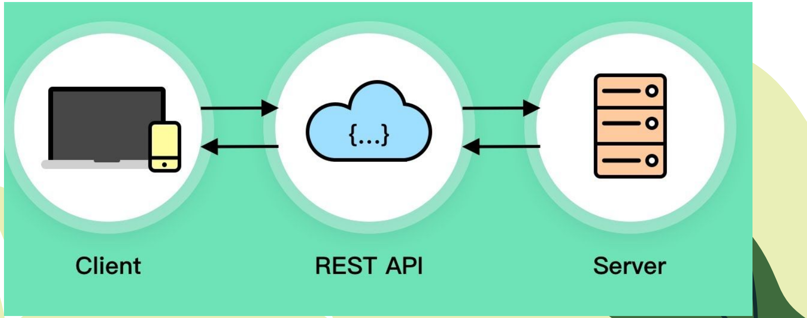
Front End

- Web interface
- Ability to upload shapefiles or draw polygons
- User friendly and responsive

Solution Overview

API

- Link web interface and back end
- Send shapefile/polygon to the back end
- Send results to the front end

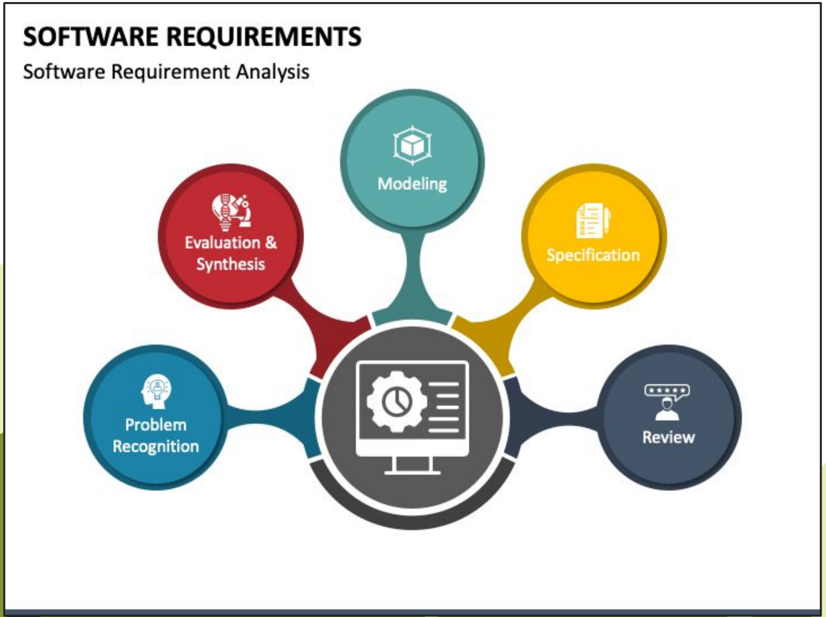


Solution Overview

Back End and Database

- High performance linux based server
- Python based prediction system
- Raster layer storage
- User access control

Requirements Acquisition



SOFTWARE REQUIREMENTS

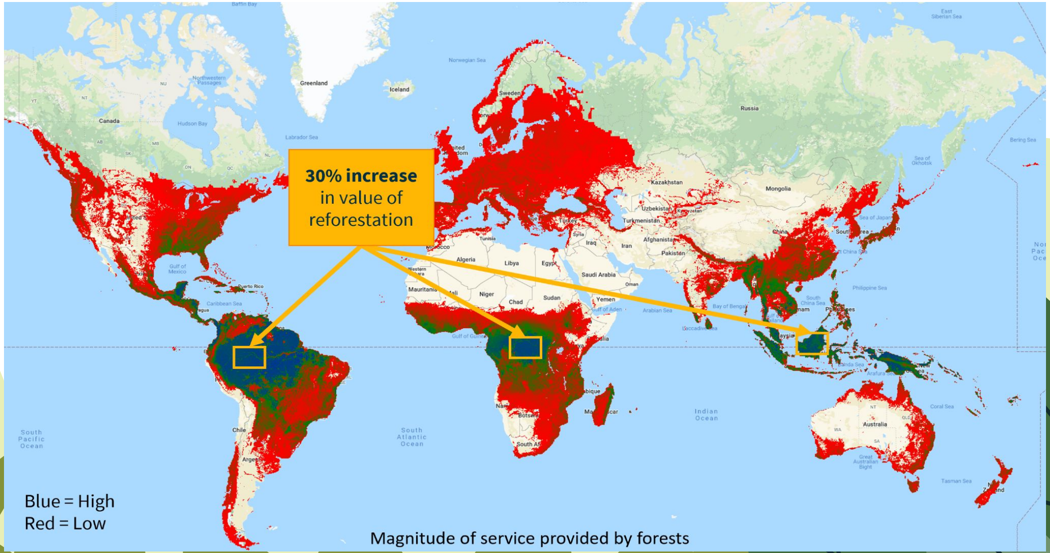
Software Requirement Analysis

- Weekly client meetings
- External research & sponsor recommendations
- Initial project description and overview

Key Requirements

User Requirements

- Ease of use
- Quick response time
- Up to date data for the most accurate carbon credit prediction
- Visualization of reforestation area on interactive map



30% increase in value of reforestation

Blue = High
Red = Low

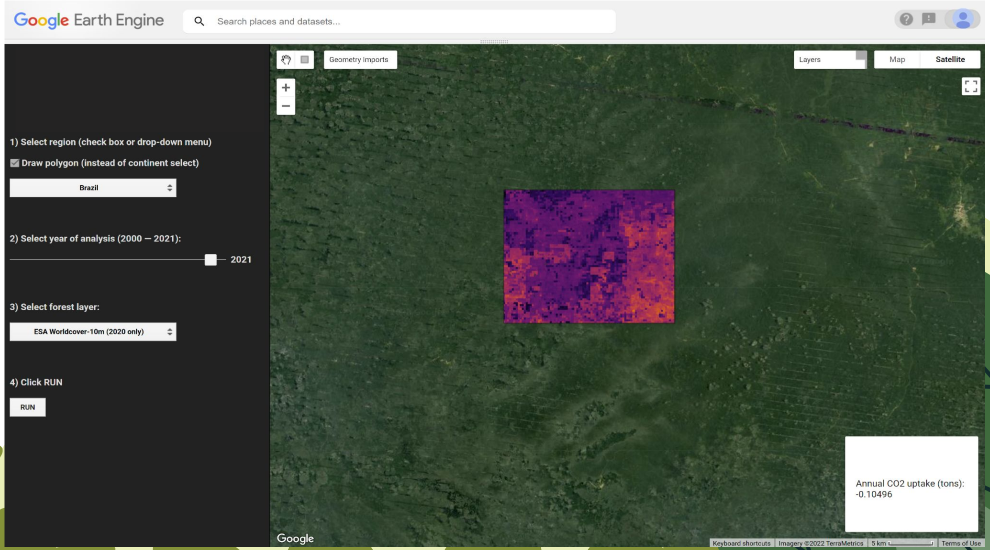
Magnitude of service provided by forests

Functional Requirements

Requirement 1: Simple web interface with an interactable map

- Zoomable and responsive map
- Ability to draw out a polygon on the map or upload a shapefile
- Carbon credit prediction display

Current Prototype:



1) Select region (check box or drop-down menu)

☑ Draw polygon (instead of continent select)

Brazil

2) Select year of analysis (2000 — 2021):

2021

3) Select forest layer:

ESA Worldcover-10m (2020 only)

4) Click RUN

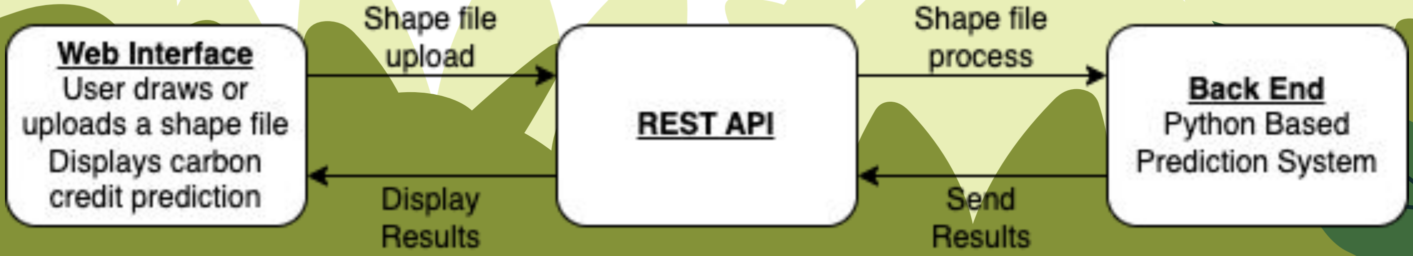
RUN

Annual CO2 uptake (tons):
-0.10496

Functional Requirements

Requirement 2: REST API facilitating communication between the backend and the user / web interface

- Ability to transfer a shapefile from the web application to the backend
- Ability to return the necessary data to the user and visualize their query



Web Interface
User draws or uploads a shape file
Displays carbon credit prediction

Shape file upload

REST API

Shape file process

Back End
Python Based Prediction System

Send Results

Display Results

Functional Requirements

Requirement 3: Computationally efficient backend that will run the prediction on a specified plot of land

- Accept a shapefile for a proposed reforestation plot and run a carbon credit prediction on the land
- Use the correct raster data set for year and time of prediction
- Ex: For a 1000 hectare plot of land, the prediction should be computed within 10 seconds to be returned back to the user

Functional Requirements

Requirement 4: Database with encryption to store user account information

- Store user account information for user access privileges

Stretch goal

- Ability to store history of users spatial queries for usage based billing

PostgreSQL

Key Requirements

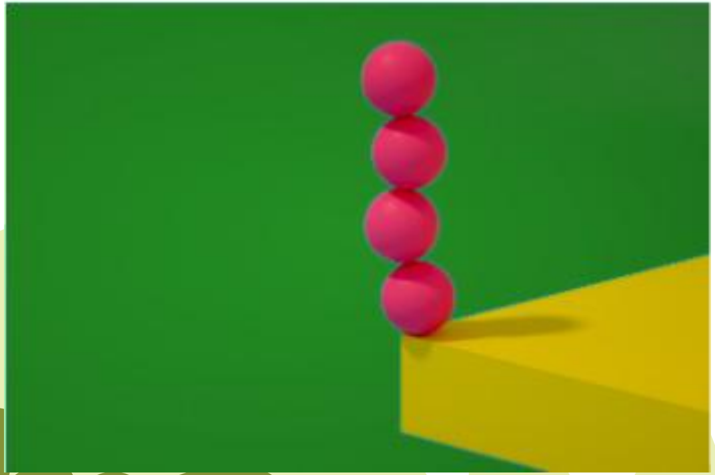
Environmental

- Prediction computation located on the backend and written in python
- High performance backend computer cluster

Stretch Goals

- ArcGIS plugin to connect to ESRI's ArcGIS tool
- Performance upgrades such as tiling rasters

Risks



The main risks that we are concerned about are:

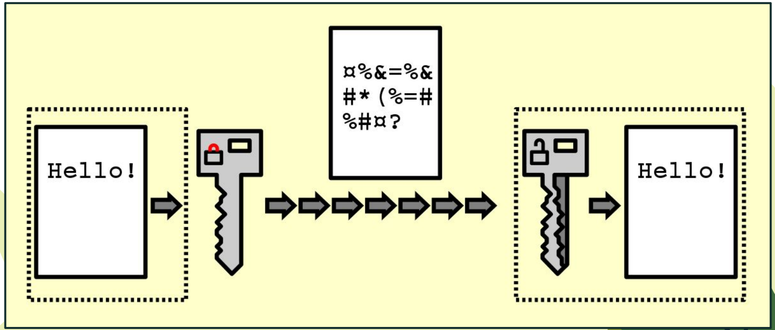
- Account Security / User Privacy - High Risk
  - Must ensure that user's accounts and information are secure
- Accurate Computation Results - High Risk
  - Must ensure that our prediction system produces correct results
- Accurate Data Representation - Low Risk
  - Must ensure that the results we compute are represented accurately

Risk Mitigation/Feasibility

How we are planning to mitigate these risks, feasibility of these solutions:

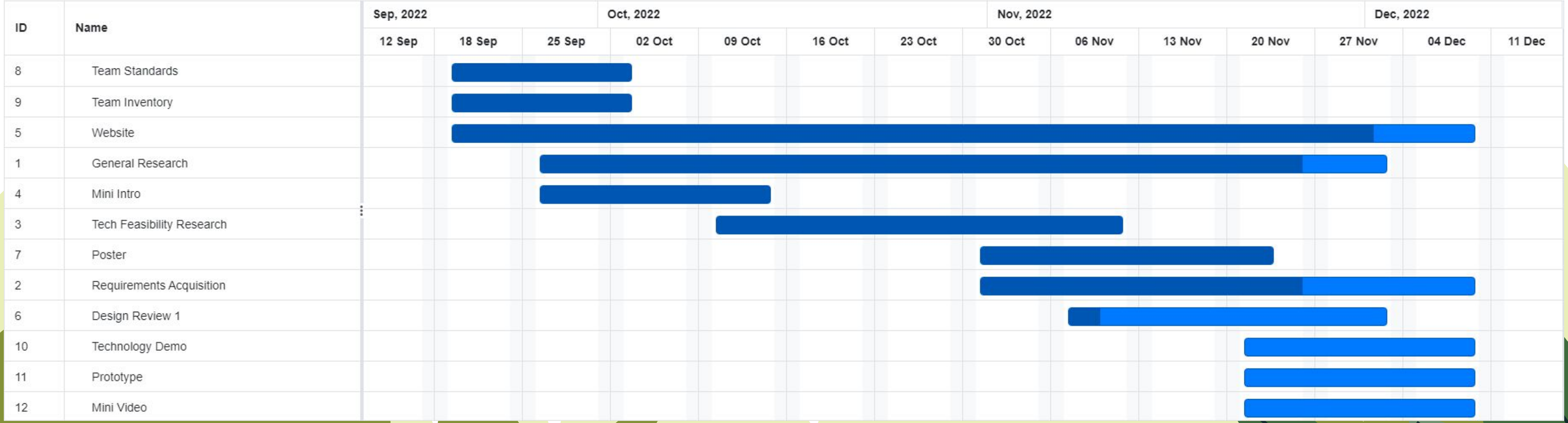
- Account Security / User Privacy
  - Encryption will be used to protect sensitive data
- Accurate Computation Results
  - Testing with our sponsor to ensure accurate results are being calculated
- Accurate Data Representation
  - Collaboration with our sponsor to validate data display

Encryption Example:



Schedule

Gantt Chart:



ID	Name
8	Team Standards
9	Team Inventory
5	Website
1	General Research
4	Mini Intro
3	Tech Feasibility Research
7	Poster
2	Requirements Acquisition
6	Design Review 1
10	Technology Demo
11	Prototype
12	Mini Video

Conclusion

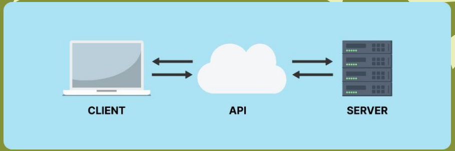
The future of our planet is directly affected by the amount of carbon dioxide that continues to be released into the atmosphere. This is the problem that our sponsor, Allie Shenkin, is trying to help solve with his new discovery. His discovery allows for investors to receive 30% more carbon credits than before; however, the current implementation to estimate the amount of carbon credits per plot of land is very underdeveloped. This is where our team comes in. Our envisioned solution includes:

- A responsive and convenient web interface front-end
- REST API that securely and swiftly transmits information between our web interface and backend (computation)
- Secure backend that holds all of the user access information, with potential to include storage of previous queries

Functional Requirements:

Simple web interface with a interactable map

REST API facilitating communication between the backend and the user / web interface



Computationally efficient backend that will run the prediction on a specified plot of land

Database with encryption to store user account information

Conclusion

Performance Requirements:

- 10 seconds maximum response time per every 1000 hectares plot of land
- Responsive back end and web front end

OpenLayers

Risks/Feasibility:

- Account Security / User Privacy
- Accurate Computation Results
- Accurate Data Representation

PostgreSQL

django

Environmental Requirements:

- Prediction computation located on the backend and written in python
- High performance backend computer cluster

Bing Maps