AEC Download

Situation/Pain – Customer Area based plans are being sold and our tracking system is not adequate. The solution leaves gaps in what we report on, reporting itself is not accurate, leaves only a single coverage estimate with now proof of work for the client to see, and is not scalable putting a lot of work on the data science team.

Ideal State – Customer is able to use geo-based pricing models. API is being developed but this is probably a year away.

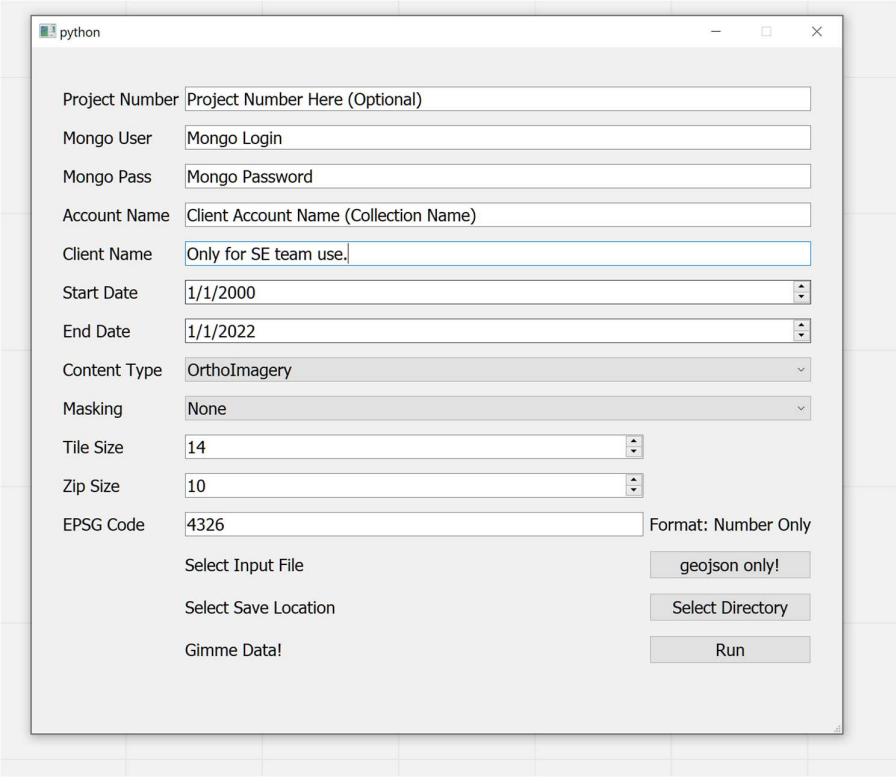
Solution – Clients who are pulling data via an api end point can simply use a database tracking system where they have the ability to write data and we have admin access to view. We then can have the database track exact coordinates of the data requests to charge area-based pricing.

Also, out of the 90 clients on this system around half are well over the allowed amounts. 500 percent over is not uncommon.
https://www.dropbox.com/sh/i10l2trb3effe6l/AAB7lY_kRAc0T80PolmVVv6aa?dl=0

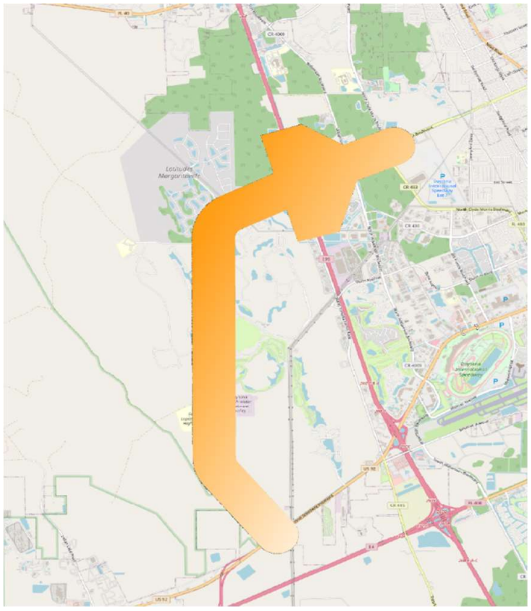
MongoDB is a database that is hosted on the cloud and uses a json file structure to write to. This is accessible from any coding language and has built in security such as user authentication and IP restrictions. Nearmap would set up a database for each client having these tracking issues to write to and we could track and print reports from that.

Example Workflow

Customer Runs Python Library with attached Front End tied to tracking database Via Mongo.



Front end takes in customer request data and saves high res clipped Ortho Locally.



Customer AOI (geojson)

Customer Imagery Return, Merged/Reprojected

Tracking:

All tracking is done through a no SQL database mongodb. Account is set up with only write access to the client account prior to the tool being distributed.

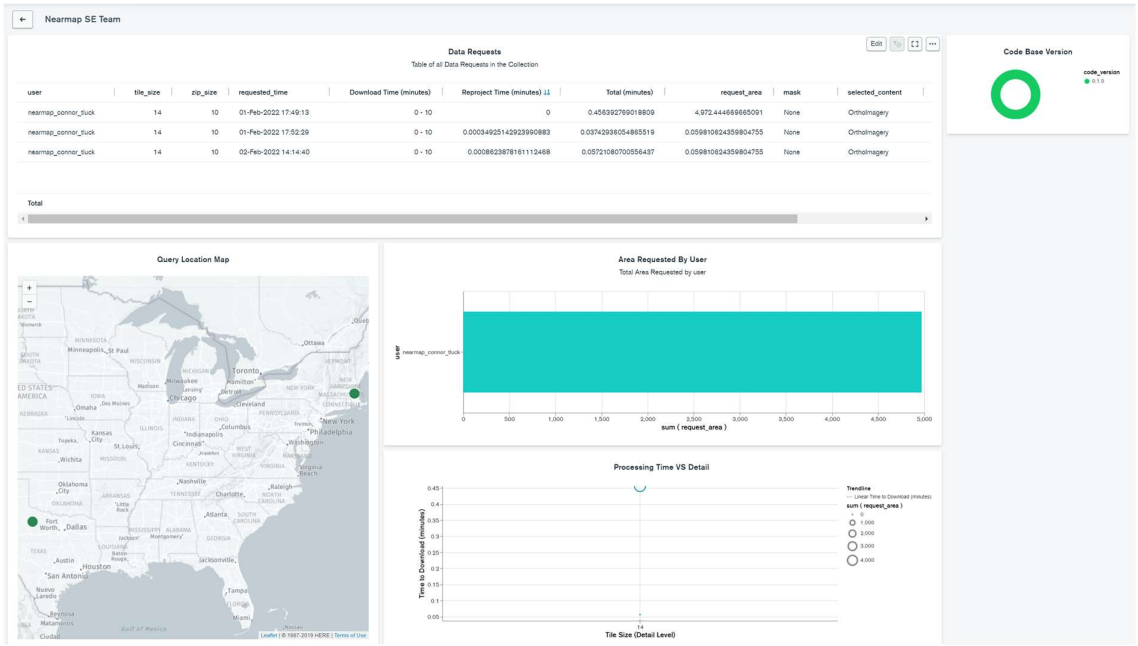
Schema:


```
_id: ObjectId("61f9b9116cf7920db4ac6138")
error_code: "Coverage Gaps Found, No Data Returned."
coverage_binary: 0
code_version: "0.1.0"
project_number: "Project Number Here (Optional)"
user: "nearmap_connor_tluck"
collection_name: "nearmap"
client_name: "nearmap"
input_file: "D:/AEC_Downloader/map (1).geojson"
output_file: "D:/AEC_Downloader/throw_away_folder/t1"
requested_time: 2022-02-01T17:49:13.631+00:00
start_date: 2000-01-01T00:00:00.000+00:00
end_date: 2022-01-01T00:00:00.000+00:00
projection: 4326
selected_content: "OrthoImagery"
tile_size: 14
zip_size: 10
masking_method: "None"
time_to_deliver_data: 0.456392769018809
time_to_reproject_data: 0
total_processing_time: 0.456392769018809
request_area: 4972.444669665091
> request_centroid: Array
> geojson_data: Object
```

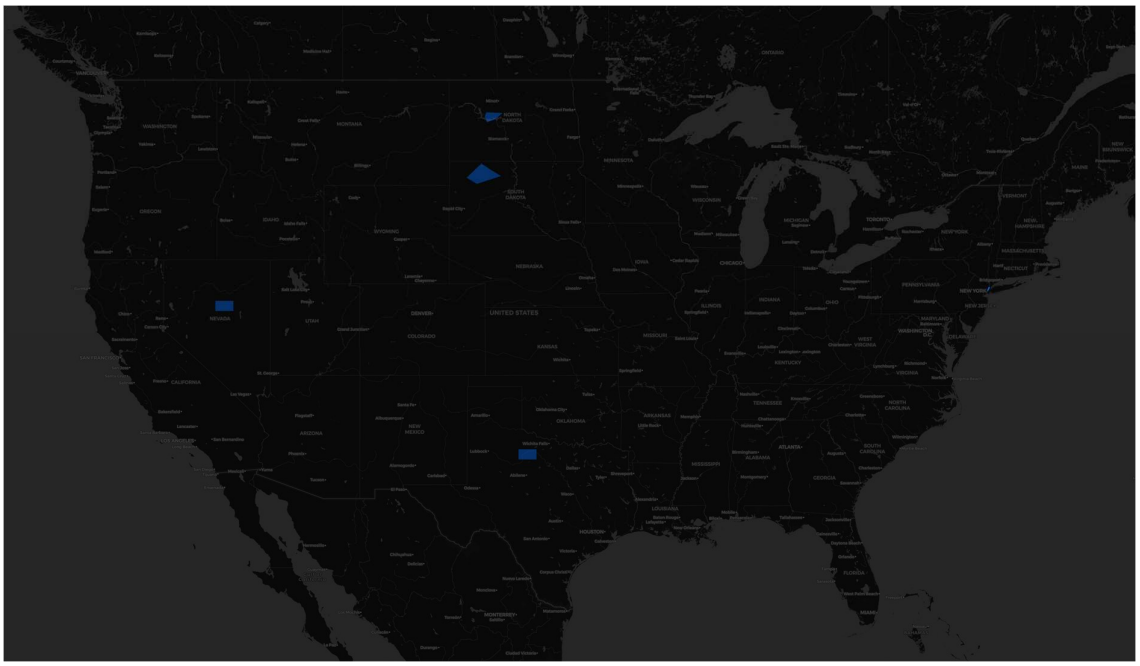

Color Indicates storage Format

Tracking:

Tracking can be done via queries of the database itself or through inhouse Mongo Tracking Dashboards. Note Geojsons ARE stored but as of now mongo does not support direct integration of this into dashboard so we use centroids. Below are examples of Dashboard and Folium map making direct Query on another Dataset.



Mongo Dashboard.



Folium Query with Geojson being updated Live.