Project Requirements

Functional Requirements

Backend

- Receive polygon data from user frontend

- Three types of acceptable polygon data: point, box and polygon

- Deal with point and box first, expand to polygon type later

- Create resource and task

- Resource made up of dataset and geospatial extent

- Send resource to Eratos primary node and aggregate task to gateway node as Post requests

- Acceptable formats: Well known text, Well known binary and GeoJSON

- Check task state

- The task will be completed by the Eratos gateway in its own time

- Send Get requests periodically to enquire about task completion

- If the task was unsuccessful, an error notification to be sent to the admin node

- Process completed task

- If the task was successful, a success notification sent to the user node

- Resource associated with successfully completed task shared with primary node (user frontend to fetch the completed resource from primary node separately)

- User authorization

- Apart from processing tasks, the backend is also required to perform authorization of the user requesting the task with the Eratos primary node

- Authorization based on Auth0 API

- Authorization decides whether the user has permission to perform the selected task (such as if the user hasn't paid or account expired, etc.)

- Result caching and other optimizations (optional / future)

- Several processing optimizations can be performed at the backend to reduce the number of Post and Get requests sent to the Eratos servers

- These optimizations can cache frequently queried results or user privileges for faster processing

User Frontend

- Map Interface

  - Full Screen.

  - Can use TerriaJS or Mapbox.

  - Switch view styles: Street, Terrain & Satellite.

- Lookup and select a module (model).

  - Look up and view information on the module and its outcomes

  - Extensibility.

- Lookup and select location

  - Either search a location and/or draw a polygon for a selected area

  - Searchable locations based on administrative boundaries, places, or address lookup

  (polygon is automatically drawn based on boundary selected)

- Component to identify processing is occurring.

- If a payment is needed to process, the user is flagged and asked to put in payment details and confirm the purchase.

- Component notifying the processing is complete
- View and Visualise results.

  - The ability to visualise the results: table, line graph, etc...

  - Ability to filter the results

  - Ability to download results

- Account login and settings

  - Ability to log in or create an account

  - Ability to change payment details

  - Ability to view the history of purchases and downloads.

Admin Frontend

- Module Control Space (Admin operations over the modules on the system)

- View all the modules available on the system

- View detailed information (metadata) of a selected module

- Add a new module to the system

- Edit a selected module

- Enable or disable a selected module by a toggle switch

- User information

- View all the users on the system

- Search for a particular user by username or full name

- View the searched user's information

- Purchase information

- View all the purchase transactions on the system

- Look up particular purchases by purchase ID, date or price range

- View the selected purchase details, such as the price paid, the cost to the process and the time of the process taken

- Third-party integrations (Nice to have)

- User analytics (View the analytic report about a user)

- User support

Non-functional Requirements