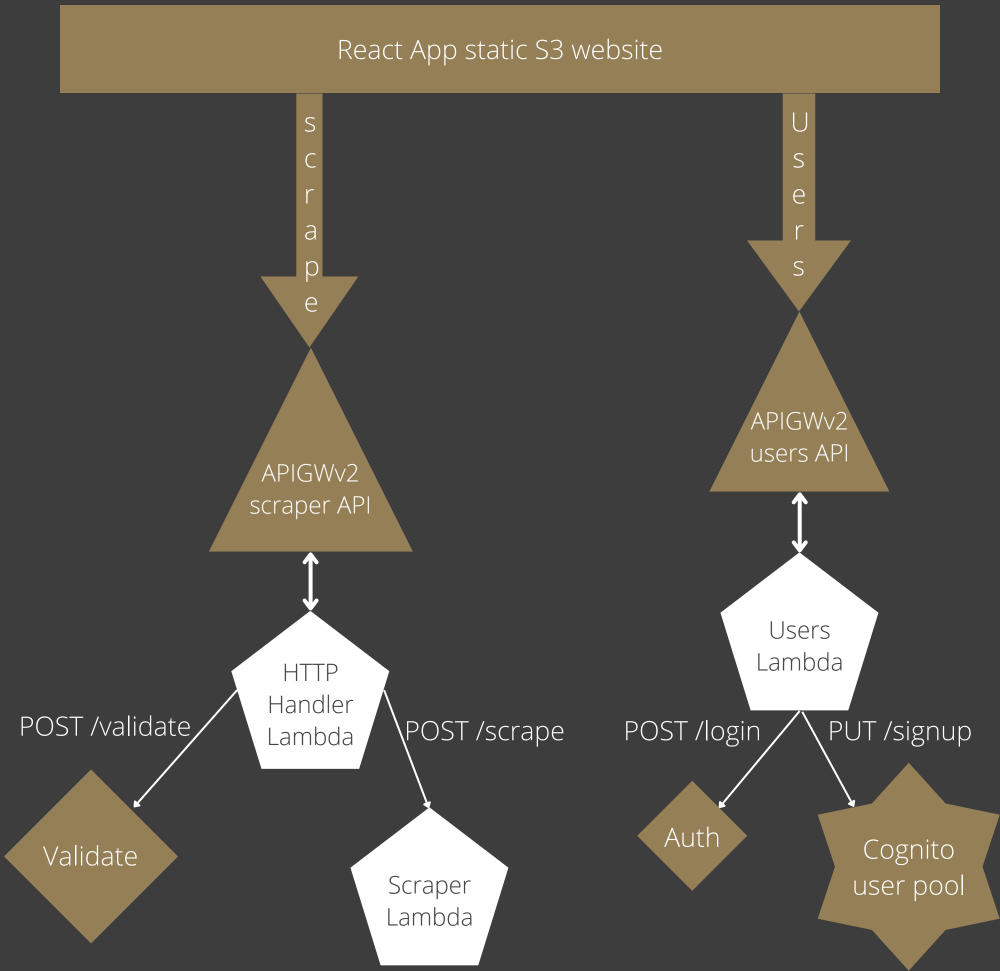
React App static S3 website
scrape
Users
APIGWv2 scraper API
APIGWv2 users API
HTTP Handler Lambda
Users Lambda
POST /validate
POST /scrape
POST /login
PUT /signup
Validate
Scraper Lambda
Auth
Cognito user pool

SCRAPER FUNCTIONALITY

The scraper should get a few arguments for a successful run:

- "scrapepath" header with a correct /<PATH> to scrape, example: "/scrape"
- $request.body with a list of puppeteer selectors to scrape from aforementioned /<PATH>,
  example: [".class1","htmlTag1","htmlTag2.class2"]
- "scrapedestination" header with a URL to a form or destination to fill up the scraped data with

Scraper Lambda

USERS FUNCTIONALITY

This lambda will deal with sign-in and sign-up

It will be called when a user signs up, logs in or otherwise has to be authenticated or handled.
If a user is not signed up, the lambda will create a new user in the app's storage bucket, it will assign that user a new JWT and return that to the app to be stored in the user's local storage / cookies.

It will also be called when a user wants to sign in and authenticate, then it will trigger an authentication process that is TBD, after which is the user is valid user, they will be granted access.

DAL FUNCTIONALITY

This lambda will deal with Data Access to mongoDB

It will be called when the Scraper needs a Data Set of selectors and other data attached to each one of them from mongoDB, updating the correct document with metadata when done.

It will also be called when a user wants to view the Statistics Page in the frontend-app.

DAL Lambda

SCRAPING PAGE

This is the page where the scraped data comes from

There are two routes to scrape:

1. Invoices - contains a simple HTML invoice
2. Tables - contains a single table with an option to use a checkbox table when filling up, instead of a regular Invoice form.

DAL
Lambda