# Project Overview

# In order to combat climate change and meet New York City's 2030 energy storage goals, the Sustainable CUNY Smart DG Hub has teamed with the NYC Mayor's Office of Sustainability (MOS), NYC Fire Department (FDNY), NYC Department of Buildings (DOB) and Con Edison to provide real-time progress tracking of all energy storage projects in the Five Boroughs.

A comprehensive NYC energy storage system (ESS) database will be developed to store key ESS project details and metrics, and will be accessible internally by NYC agencies in the form of an interactive dashboard for data visualization and exploration.

# Deliverables

* Comprehensive SQL-based database of energy storage projects
* Interactive dashboard for data exploration, accessible via user login landing page

# Components

[Component diagram here]

## Database

* .db or .sqlite database from CSV
* Populated from three (3) publicly available sources
* (Data pipeline diagram)

## Web framework

* Flask-powered RESTful API
* (Directory tree diagram)

## HTML/CSS landing page and dashboard

* Bootstrap-based dashboard layout
* User landing page with login

## JavaScript

* Interactive charts using Highcharts.js library
* Leaflet map using project location data and map overlays

Maps overlays:

* NYC Open Data - Borough boundaries: <https://data.cityofnewyork.us/City-Government/Borough-Boundaries/tqmj-j8zm>
* BetaNYC zip codes: <https://beta.nyc/products/boundaries-map/>

# Data Sources

## Con Edison SIR Inventory

* Description:
* Type: web scrape / CSV
* Source: <http://www3.dps.ny.gov/W/PSCWeb.nsf/96f0fec0b45a3c6485257688006a701a/286d2c179e9a5a8385257fbf003f1f7e/$FILE/Con%20Ed%20Public%20January%202020.xlsx>
* Filename: Con Ed Public January 2020.xlsx

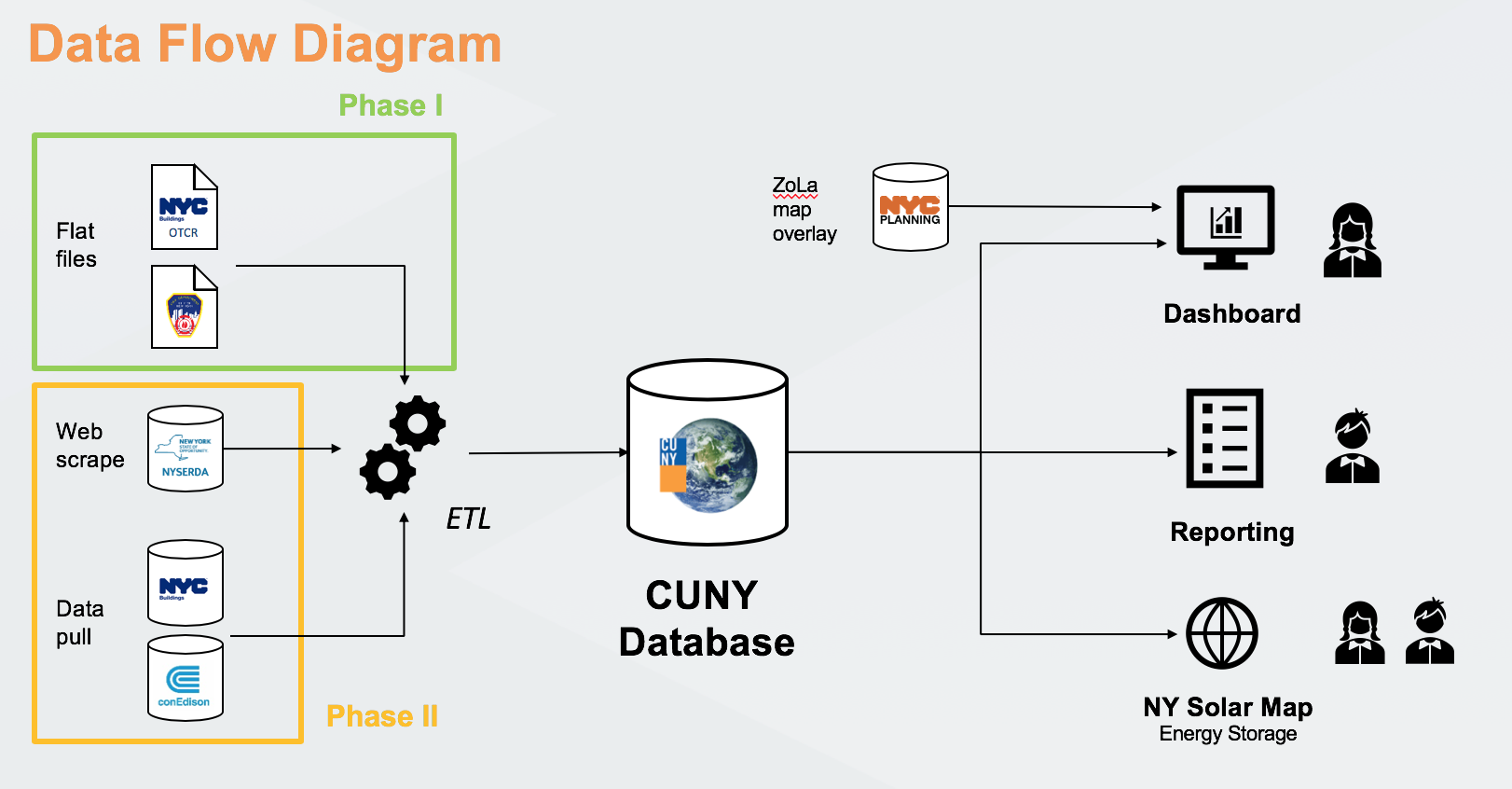
## NYISO Interconnection Queue

* Description:
* Type: web scrape / CSV
* Source: <https://www.nyiso.com/documents/20142/1407078/NYISO-Interconnection-Queue.xlsx/3bc8b922-769a-7bfe-8aea-ffa1e3d5669a?t=1579017010942>
* Filename: NYISO-Interconnection-Queue.xlsx

## NYSERDA DER Characteristic Data (web scrape / CSV)

* Type: web scrape / CSV
* Source: <https://der.nyserda.ny.gov/download/facility-metric/>
* Filename: NYSERDA\_DER\_Metric\_Data.xlsx

# Data Pipeline



# Process

## Step 1. Identify and acquire data

Our data is taken from three publicly available sources providing metrics on energy storage systems located in New York City:

1. [Con Edison SIR Interconnection Inventory](http://www3.dps.ny.gov/W/PSCWeb.nsf/96f0fec0b45a3c6485257688006a701a/286d2c179e9a5a8385257fbf003f1f7e/$FILE/Con%20Ed%20Public%20January%202020.xlsx), which lists all completed and upcoming interconnection applications for various types of distributed energy resources serving the Con Ed grid (ESS systems < 5 MW),
2. [NYISO Interconnection Queue](https://www.nyiso.com/documents/20142/1407078/NYISO-Interconnection-Queue.xlsx/3bc8b922-769a-7bfe-8aea-ffa1e3d5669a?t=1579017010942) – a similar list of projects, but containing ESS systems > 5 MW or greater, and serving the greater NYISO area.
3. [NYSERDA DER Characteristic Data](https://der.nyserda.ny.gov/download/facility-metric/) containing New York State-based ESS systems which have applied for funding under NYSERDA grants.

Because all three of these sources are only available online as CSV files – the names of which are updated every month - it is easiest, for our purposes, to simply download the files and access locally, as opposed to other, more advanced web-scraping techniques.

## Step 2. Clean data

The data is then cleaned using the **‘clean\_data.ipynb’** script. This file reads the flat, unprocessed CSV files, creates dataframes for each, then performs several “cleaning” operations as needed. These cleaned dataframes are then exported to new, more useable CSV files from which query-able databases can then be created.

## Step 4. Create database and APIs

Each CSV file is then imported in DB Browser and exported to a database (.db) file which can then be queried for detailed analytics and visualization down the road.

## Step 5. Create Flask-based web framework for serving files

## Step 6. Create HTML skeleton for dashboard elements

## HTML Bootstrap theme with top navbar and interactive elements

## Custom styling for HTML theme

## Step 7. Interactive JavaScript elements using Highcharts.js library

## Step 8. Leaflet map using project data and map overlays

## (Create JSON and geoJSON files for use with Highcharts and Leaflet)

## Step 9. User login landing page

## Step 10. Documentation, Packaging, and Sharing

## 