PMPA Conversion Structure

Description of the project

- PMPA is the name of the project that includes a SCADA conversion.
- The customer is coming from a Survalent system and moving into Monarch.
- The conversion includes the following databases.
  - SCADA
  - STATES
  - FEP
  - ICCP
  - OPENCALC

Similarities to CLAY

- The code can be written in python and jupyter.
- A mapping document will be provided and will be updated around two times a month.
- Most of the inputs in .csv and outputs to .dat or .csv.
- The output of the script will be in the same format.
  - Commented lines
  - Starting the file with the name of the database
  - 0 to separate between objects.
  - 0 to end the file
  - Etc.

Folder structure

- To keep the files in order the following folders will be used.
  - Dat_files
  - Inputs
  - Populateables
  - Scripts
  - Xrefs
  - Mappings

Scripts folder

- This folder will contain all the .py files. The goal is to be able to run the whole conversion by running one main file only, lets call it PMPA_Conversion.py.
- PMPA_Conversion.py will call other .py files like:
  - Scada_code.py
  - Fep_code.py
  - Iccp_code.py
  - Merge_scada.py
  - Merge_fep.py
  - Xml_to_csv
  - Etc.
- Having a main script that can run all the other smaller ones will help when converting all databases instead of just a SCADA, or just an object from SCADA.
- On the other hand having separated small scripts will help with troubleshooting whenever we want to work on a specific object instead of running the whole conversion.

Dat_files folder

- All files with extension .dat will be saved in this location.
- The object creator scripts (scada_code, fep_code, etc) will write the outputs here.
- The merging scripts will read the files from here.
- There will be some .dat files that will be static. These will be manually created and will reside in this folder too. I.e FEP_GSD_v1.dat, SCADA_AOR_v2.dat.

Populateables folder

- Will contain the complete databases in .dat format.
- This folder has the output of the conversion and will be used to grab the files from here and populate them in the live system.
- Files should end in two digits for year, two digits for month and two digits for day.
  - Example for FEP database created in July 29th 2024→ FEP_240729.dat
- SCAN_DATA.dat is an object of FEP database, this is the only exception of a singular object that will be in the populateables folder.

Xrefs folder

- The folder will contain the extracts of the input .csv files and can be utilized to verify what is being written to the .dat files. See workflow diagram in the next slides.
- As most of the columns of the input files are not needed, the xref file will contain the usable information of the input .csv files. And will also include the information that is going to be written into the .dat files.
- The xref files will be .csv and the name of the files will end in “*_xref.csv”.
- Most of the objects of each database will have its own xref file. The objects that are created manually won’t have an xref file.

Inputs folder

- This will contain all of the .csv that will be part of the input files. The files that are extracted from the customer's system will go in:
  - /inputs/survalent
- In the case that there is another type of file that will act as an input file then it will go into a different folder in /inputs/ . This could be xmls, or RTAC_points:
  - /inputs/xmls
  - /inputs/RTAC_points

Mappings folder

- The mappings folder will contain the latest mapping document.

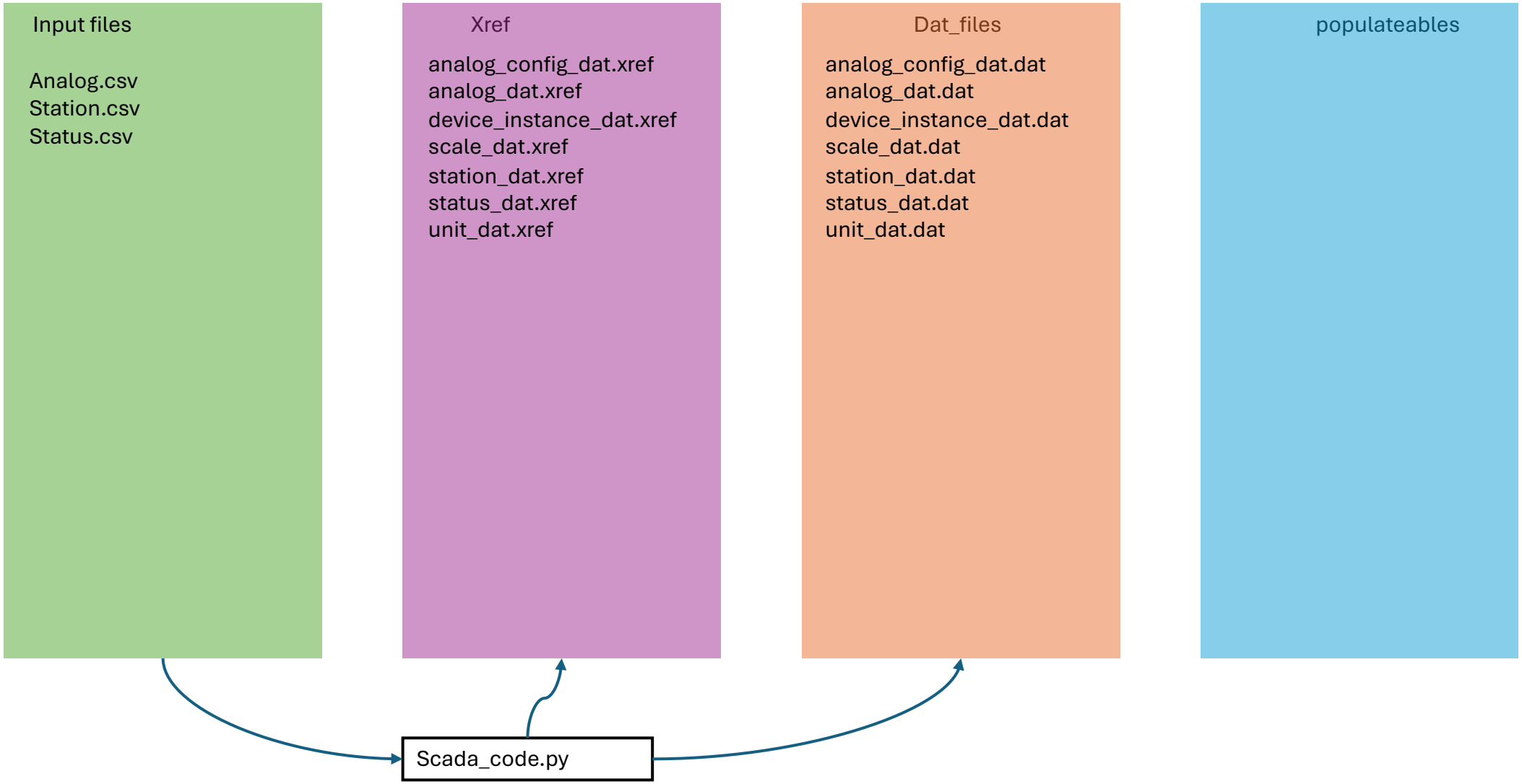
Git

- Branches
- Merge and Pull Requests to keep history and timeline of the project.
  - This could be achieved by syncing local environment with git. Open to other solutions.

Requirements

- Coding in English
  - Variables
  - Function names
  - Comments
  - Printed text during execution.
- Time counter
  - The main script must print “Starting 00:00” and start a counter when it starts running.
  - Print elapsed time and a reference of what was processed after every section of the code, this could be after every object, merging files, functions that are n^2 or take longer than other functions, script, etc.

Files and folders per script



Files and folders per script

Input files

Analog.csv
Station.csv
Status.csv

Xref

analog_config_dat.xref
analog_dat.xref
device_instance_dat.xref
scale_dat.xref
station_dat.xref
status_dat.xref
unit_dat.xref

Dat_files

analog_config_dat.dat
analog_dat.dat
device_instance_dat.dat
scale_dat.dat
station_dat.dat
status_dat.dat
unit_dat.dat

populateables

SCADA_yymmdd.dat

Main script calls