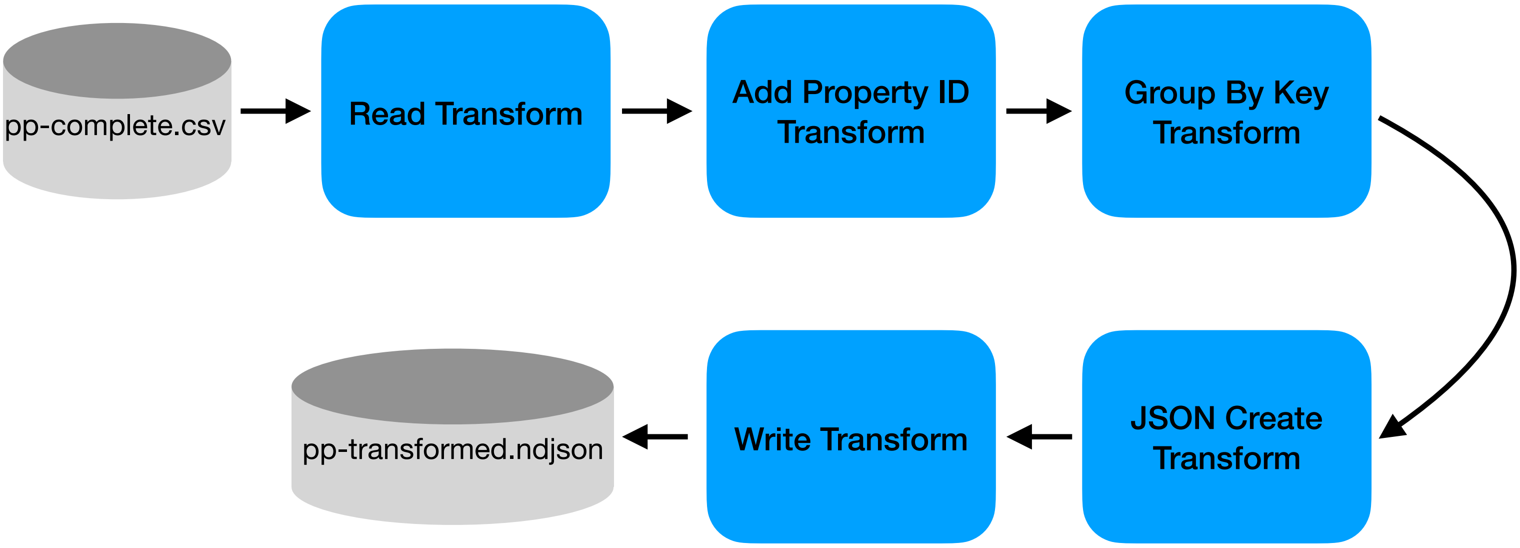
Street Group Price Paid Data Pipeline Task

The task at had was to create a pipeline that takes the freely available government data on price paid in property transactions; group it together by property and output it in the ndjson format. This required multiple transformations in the data pipeline, please refer to the diagram below for a full overview.



All transforms apart from “Add Property ID” and “JSON Create” are standard callable transforms included in Apache Beam by default.

Add Property ID is a custom transform that takes the data_element as an input, splits it into an iterable list, creates a unique property ID using a sha1 function. The sha1 function takes the address fields as input, assuming that the address fields per specific property do not change with years, the same property will get the same hash_property_id output from the sha1 function. The iterable list is then combined with a constant dict_keys into a dictionary to make it easier to convert into a JSON format at a later stage. Lastly the function yields the hash_property_id and the data_element as a tuple, for it to be later Grouped By Key (hash_property_id).

JSON Create is also a custom transform that takes in the already Grouped By Key data_element. First it separates the hash_property_id from the transaction data stored in the data_element, in a dictionary format. It then recombines the hash_property_id with the data_element in a nested dictionary, for ease of json conversion. Lastly it returns a json converted nested dictionary to the pipeline for it to be written to file.

The pipeline described above is able to transform the whole pp-complete.csv file into one .ndjson format file. To process the complete data set this pipeline takes 1h54m45s of CPU time, at the end of the processing it creates a 12+ GB .ndjson file with the transactions in the newline delimited format grouped by property.