Perfect — let’s break down **Phase 1** thoroughly. This is the foundation of your SUMO Swiss railway network pipeline, and it’s critical to understand every aspect before moving forward.

**✅ PHASE 1: extract\_nodes\_and\_edges.py**

**Goal:**  
Extract and project the Swiss railway nodes and edges from the SwissTNE GeoPackage and save them into clean CSV files that will later be transformed into SUMO XML inputs.

**📥 Input File**

**Path:**  
data/Swiss/raw/swissTNE\_Base\_20240507.gpkg  
**Format:**  
GeoPackage (.gpkg) — a spatial database format.

**Layers used:**

* bn\_node – contains railway network nodes (stations, junctions, endpoints).
* bn\_edge – contains railway network edges (rail tracks between nodes).

**🧠 Logic / Operations**

1. **Load each layer using GeoPandas.**
2. **Project to EPSG:2056 (LV95 / CH1903+)**, the official Swiss coordinate system.
3. **For nodes (bn\_node):**
   * Add a new ID: node\_id = "n\_" + object\_id
   * Extract projected coordinates: x, y
   * Keep geometry for possible GIS usage
4. **For edges (bn\_edge):**
   * Add a new ID: edge\_id = "e\_" + object\_id
   * Generate from\_node and to\_node using their object IDs and prefix n\_
   * Store the track length in length
   * Retain the geometry (a LineString) that traces the shape of the rail

**📤 Output Files**

**Folder:** data/Swiss/processed/

1. rail\_nodes.csv

| **Column** | **Description** |
| --- | --- |
| node\_id | Unique ID used in SUMO (e.g. n\_12345) |
| object\_id | Raw SwissTNE object ID |
| x | Projected x-coordinate (meters) |
| y | Projected y-coordinate (meters) |
| geometry | Point geometry in EPSG:2056 |

1. rail\_edges.csv

| **Column** | **Description** |
| --- | --- |
| edge\_id | Unique ID used in SUMO (e.g. e\_12345) |
| object\_id | Raw SwissTNE object ID |
| from\_node | node\_id of starting point |
| to\_node | node\_id of ending point |
| length | Track length in meters |
| geometry | LineString representing the rail track |

**📌 Design Assumptions**

* The object\_id field in both layers is unique and consistent across the dataset.
* Node IDs are prefixed with n\_, edge IDs with e\_.
* All geometries are projected to a **common metric CRS** to ensure geometric and distance operations (e.g. Euclidean distance for nearest neighbor mapping) work correctly in later steps.
* Output CSVs will later be parsed to write SUMO <node> and <edge> XML definitions.

**🔍 Example**

If bn\_node contains a station with:

{ object\_id: 76234, geometry: POINT (2623851.2 1200735.6) }

It becomes:

node\_id,object\_id,x,y

n\_76234,76234,2623851.2,1200735.6

If bn\_edge connects from\_node\_object\_id=76234 to to\_node\_object\_id=83012:

edge\_id,object\_id,from\_node,to\_node,length

e\_55501,55501,n\_76234,n\_83012,247.5

Would you like me to move on to **Phase 2** (writing SUMO .nod.xml) or do you want to discuss or inspect anything else in Phase 1 first?