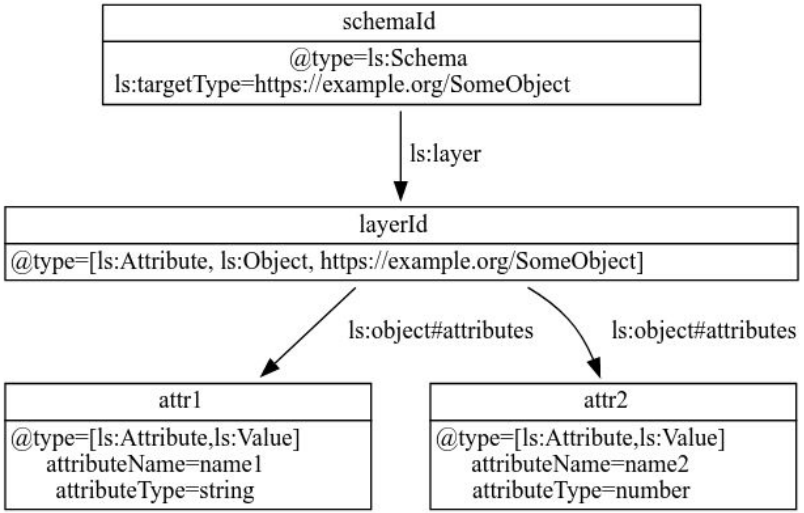
Layered Schemas

Schema



Possible Instances

```
{
   "name1": "value1",
   "name2": 1
}
```

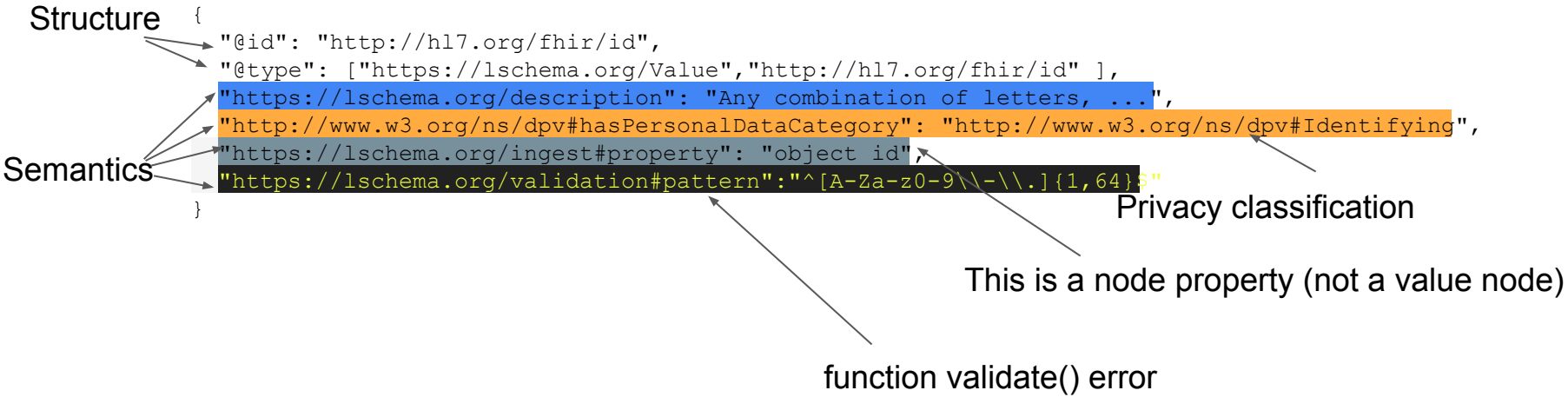
```
<name1>value1</name1>
<name2>1</name2>
```

```
name1, name2
"value1", 1
```

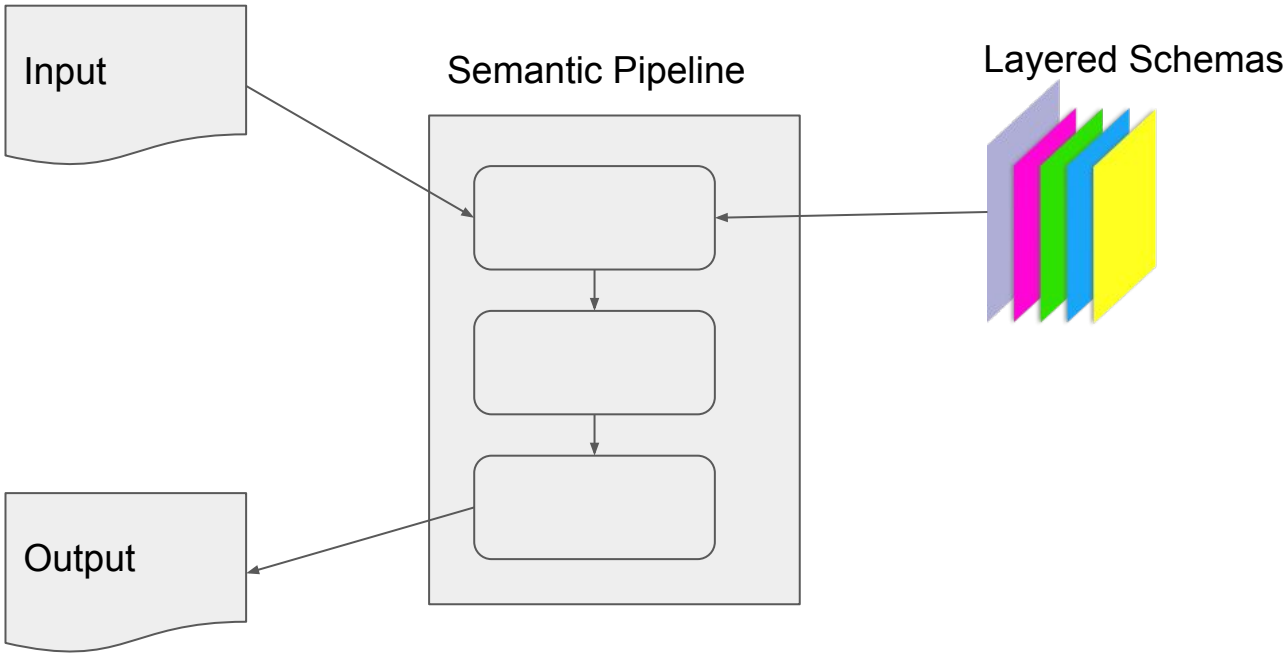
- Labeled property graph
- Defines data structure and semantics using tags
- Different layers change the shape and annotations of a schema
- Layers capture variations in data representation and semantics
- Tags are open-ended (e.g. vocabularies, labels, codes,...)

Layered Schemas: Structure + Semantics

- Metadata: Terms, codes, identifiers
- Metadata defines meaning **and** functionality
- Change meaning and functionality using overlays
- Different layers → different metadata/semantics → different graph

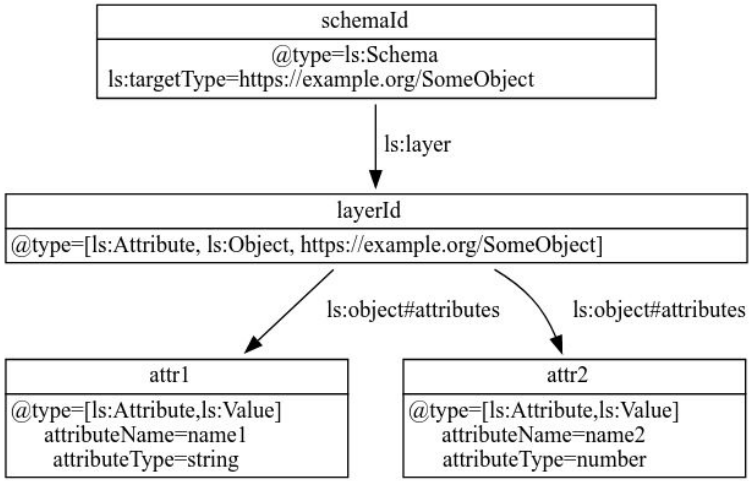


Semantic Pipelines



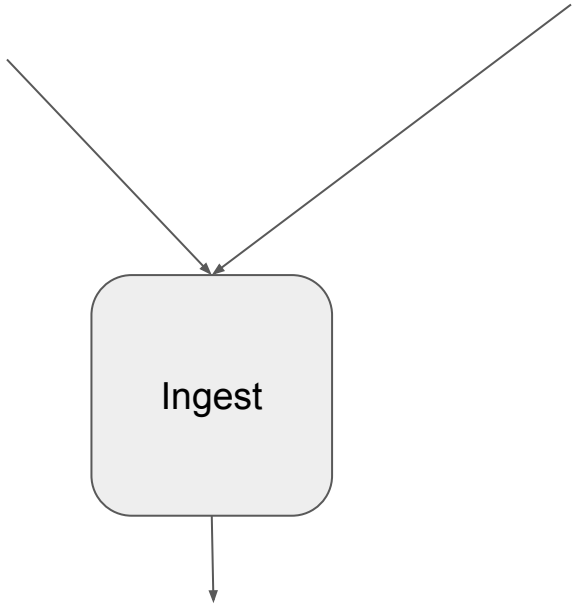
Stages: Ingest (data + schema → graph)

Layered Schema

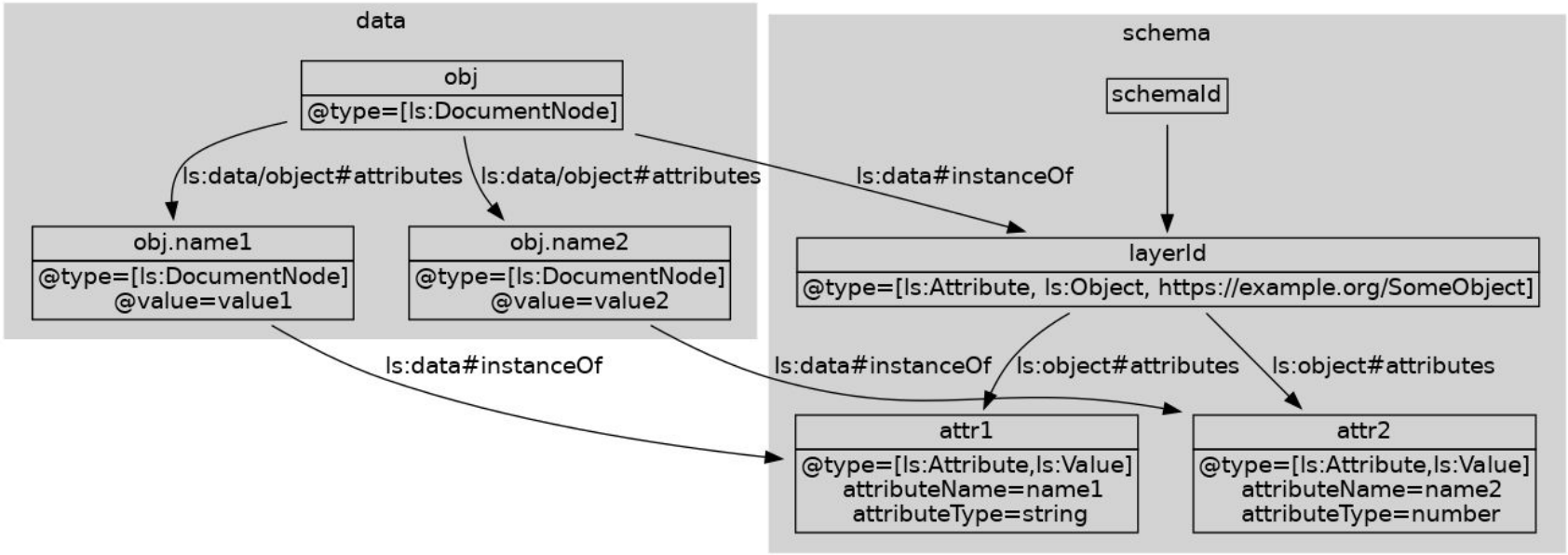


Data

```
{
   "name1": "value1",
   "name2": "value2"
}
```



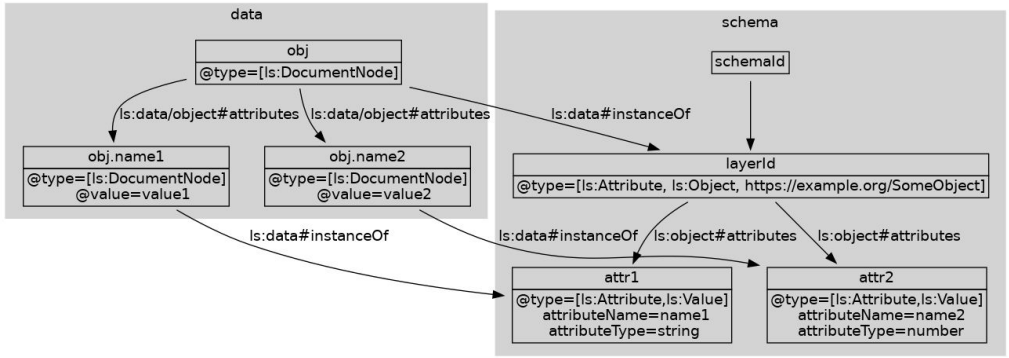
Stages: Ingest



Labeled property graph with embedded schema

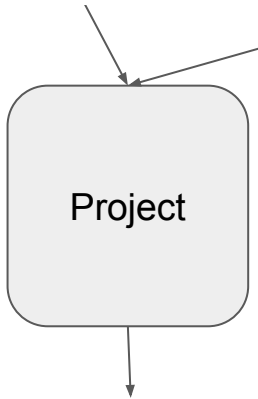
Stages: Projection (subgraph)

Graph

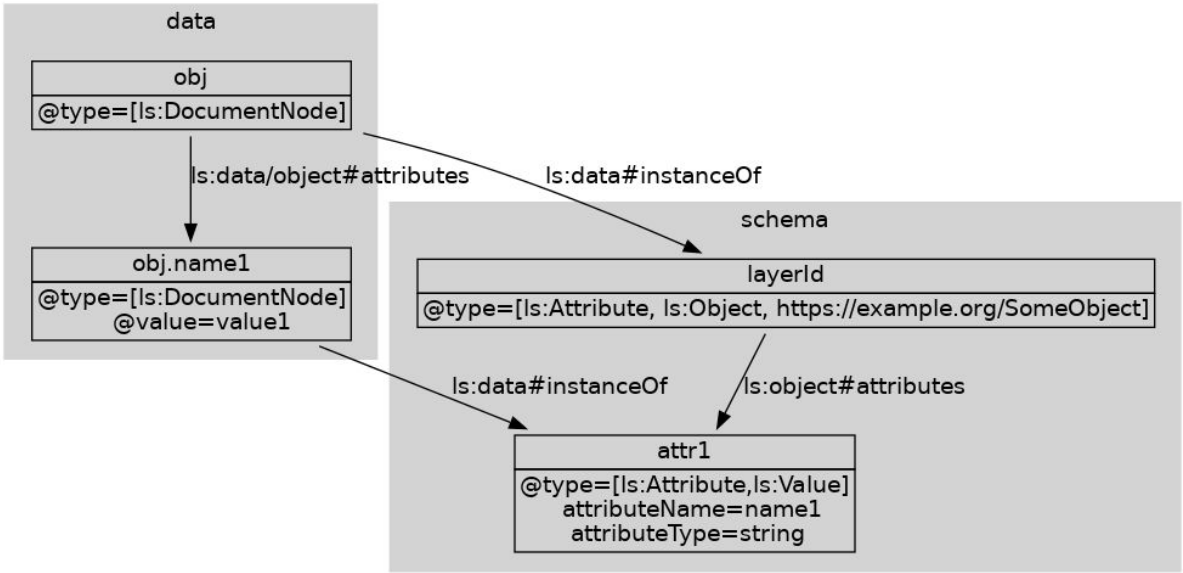


Projection specification using graph language

```
nodes:
  - include: |
      node->(node.type.has('ls:DocumentNode')) ||
      node->(node.instanceOf('attr1')
edges:
  - include: |
      edge->(edge.label=='ls:data#instanceOf')
```



Stages: Projection

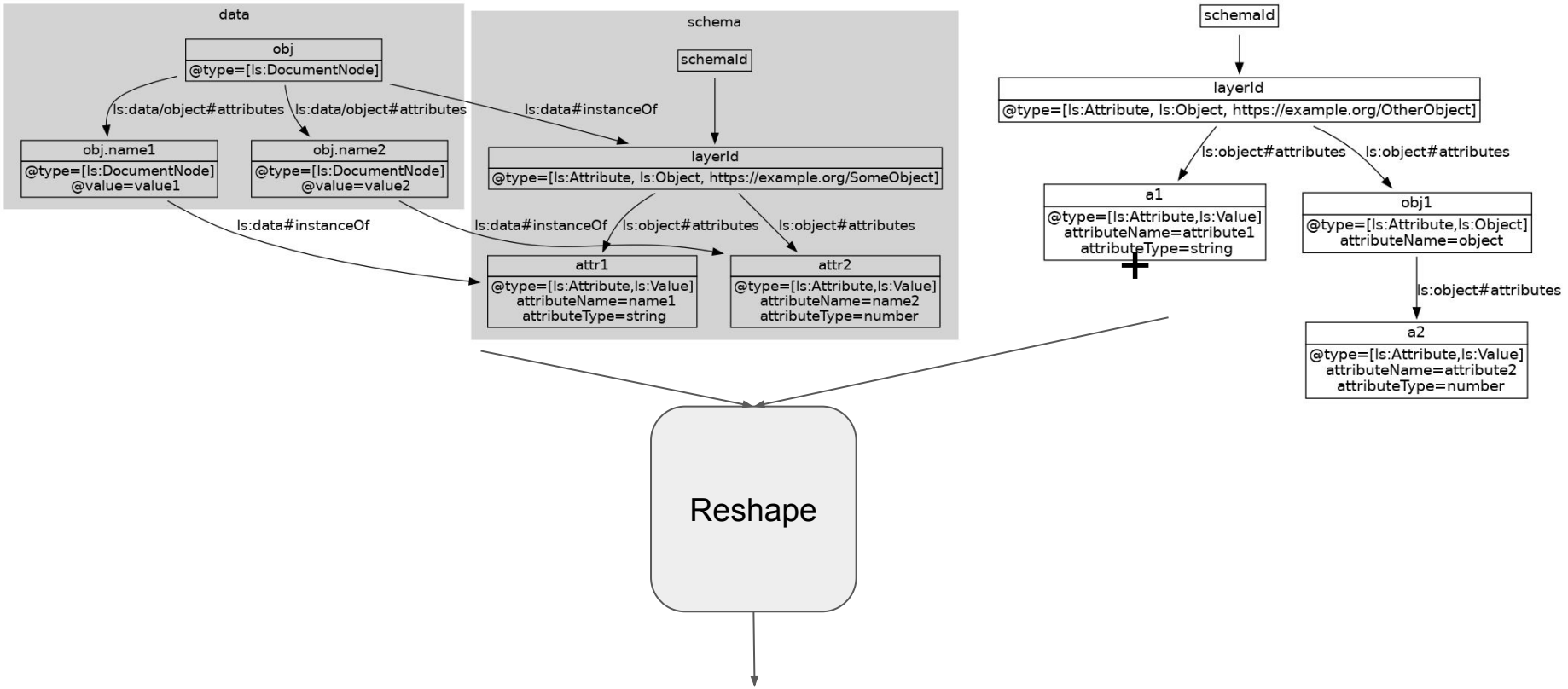


Subgraph with selected nodes and edges

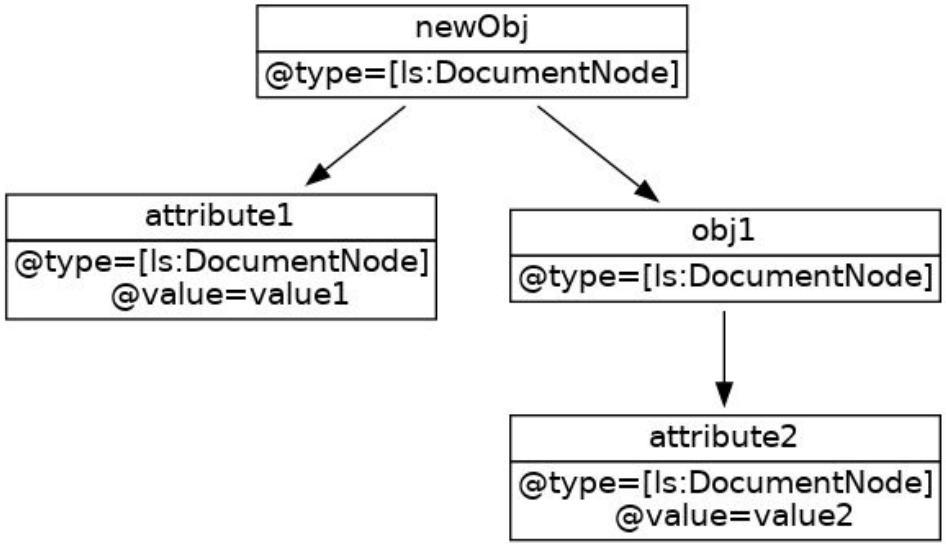
Stages: Reshape (graph + schema → new graph)

Graph

Target schema (with rule layers)



Stages: Reshape (graph + schema → new graph)



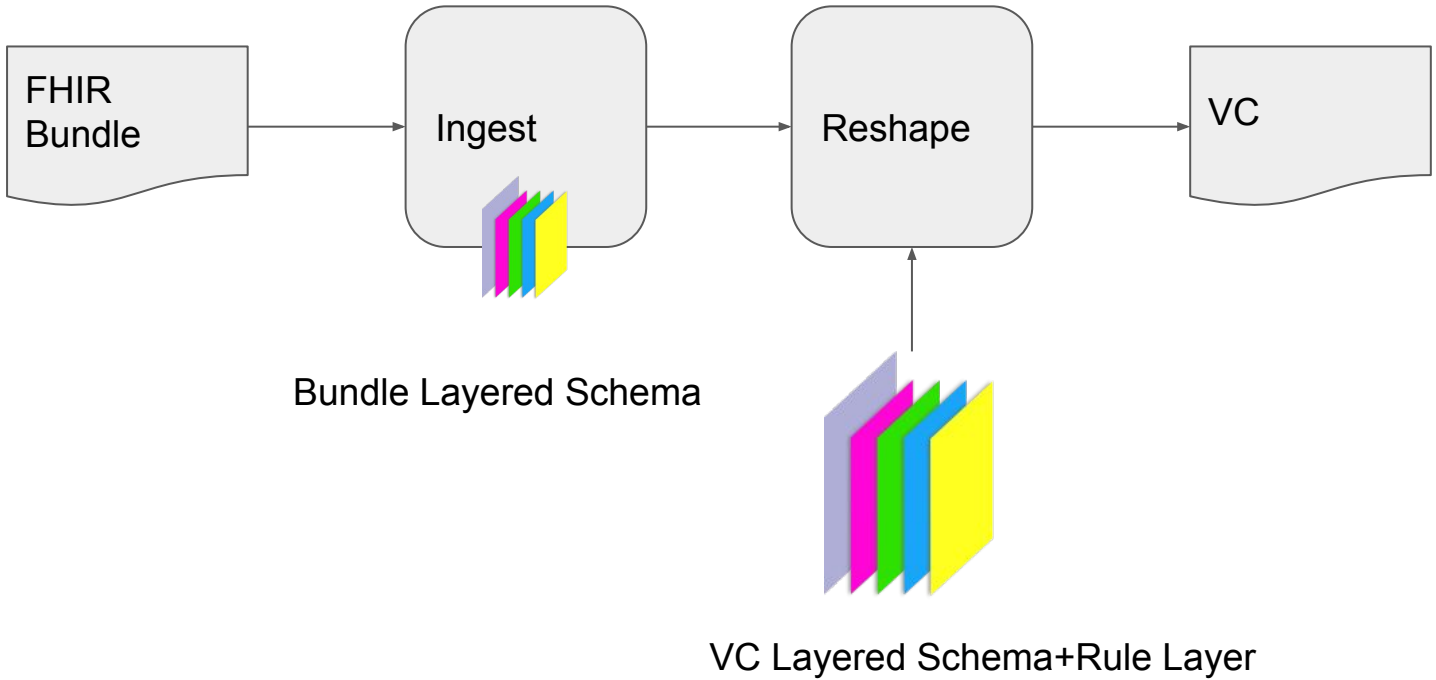

```
{
    "name1": "value1",
    "name2": "value2"
}
```

Reshape

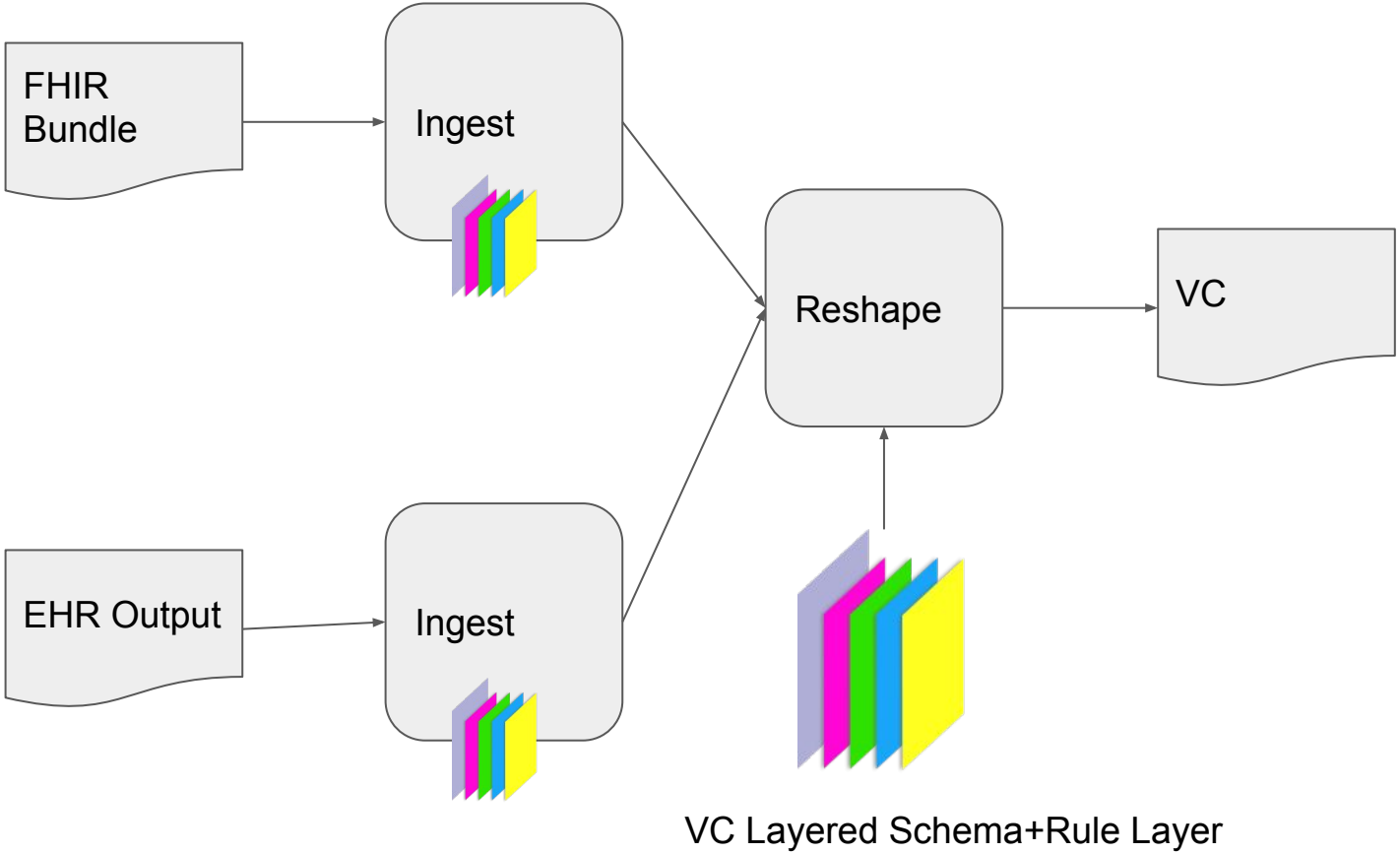
```
{
    "attribute1": "value1",
    "obj1": {
        "attribute2": "value2"
    }
}
```

Data reshaped to conform to another schema

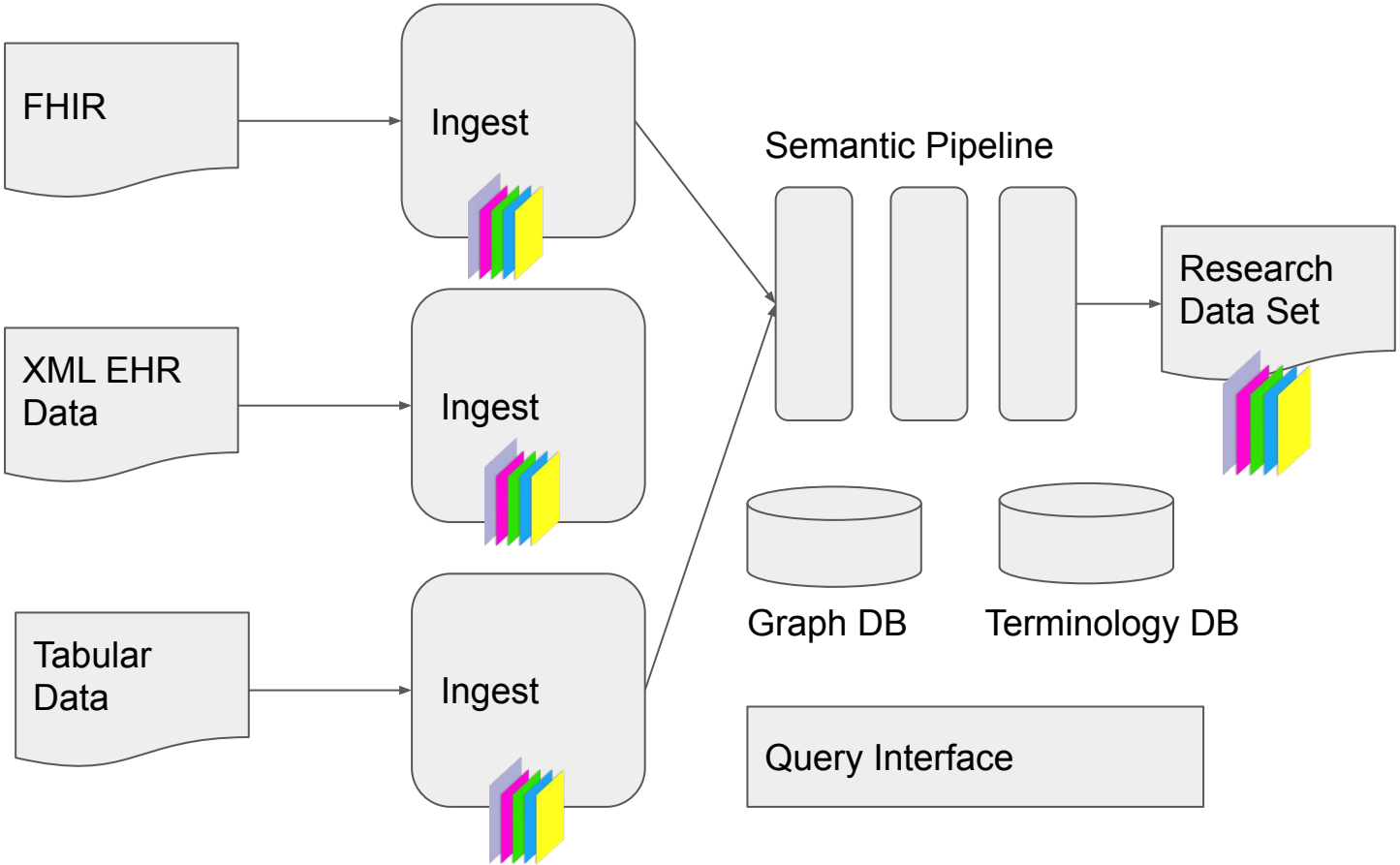
Example: FHIR → Unsigned VC



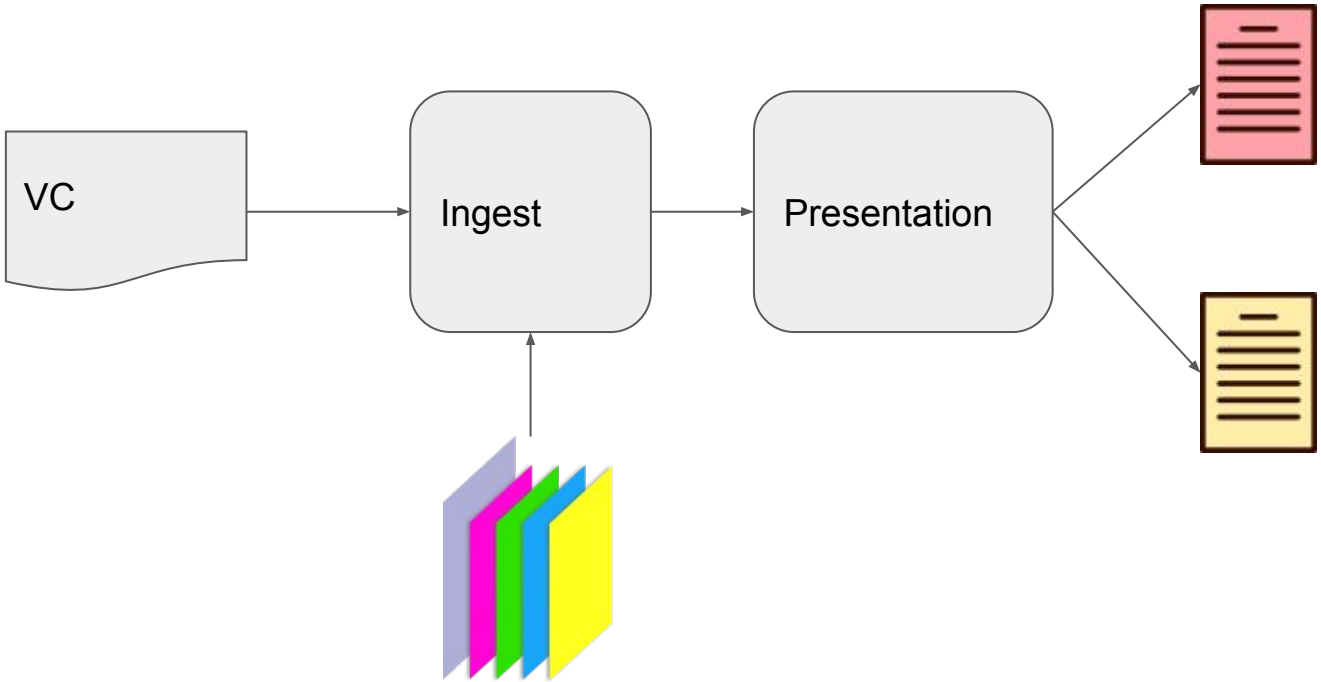
Example: Health data → Unsigned VC



Example: Semantic Data Warehouse for Health Data



Example: VC → Presentation



VC Layered Schema+Presentation Layer

Semantic Pipelines as Service