Flat vs. Structured

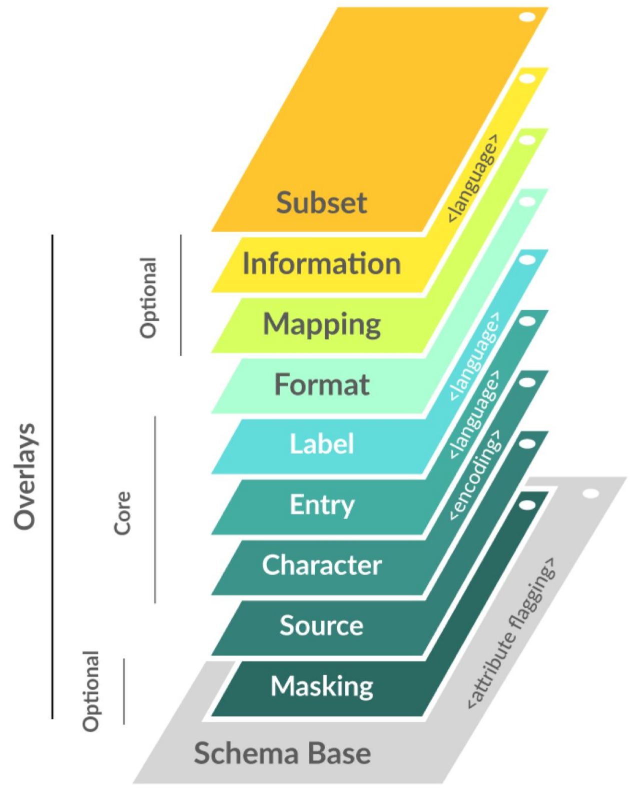
Can be flattened

```
{
    "entry": [
        {
            "resource": <Patient>
        },
        {
            "resource": <Immunization>
        },
        {
            "resource": <Provider>
        }
    ]
}
```

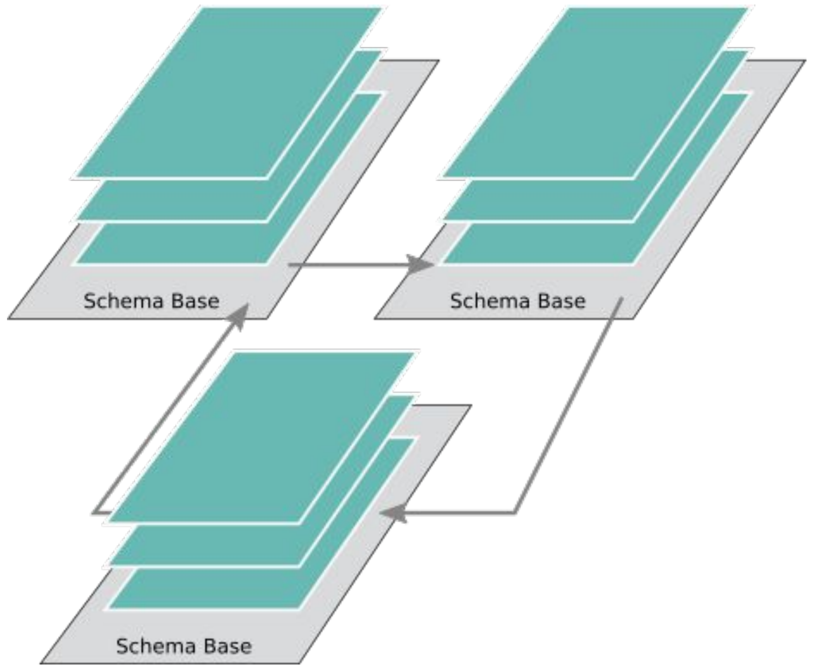
Structured

```
{
    "entry": [
        {
            "resource": <Patient>
        },
        {
            "resource": <Immunization>
        },
        {
            "resource": <Immunization>
        },
        {
            "resource": <Immunization>
        },
        {
            "resource": <Provider>
        }
    ]
}
```

OCA

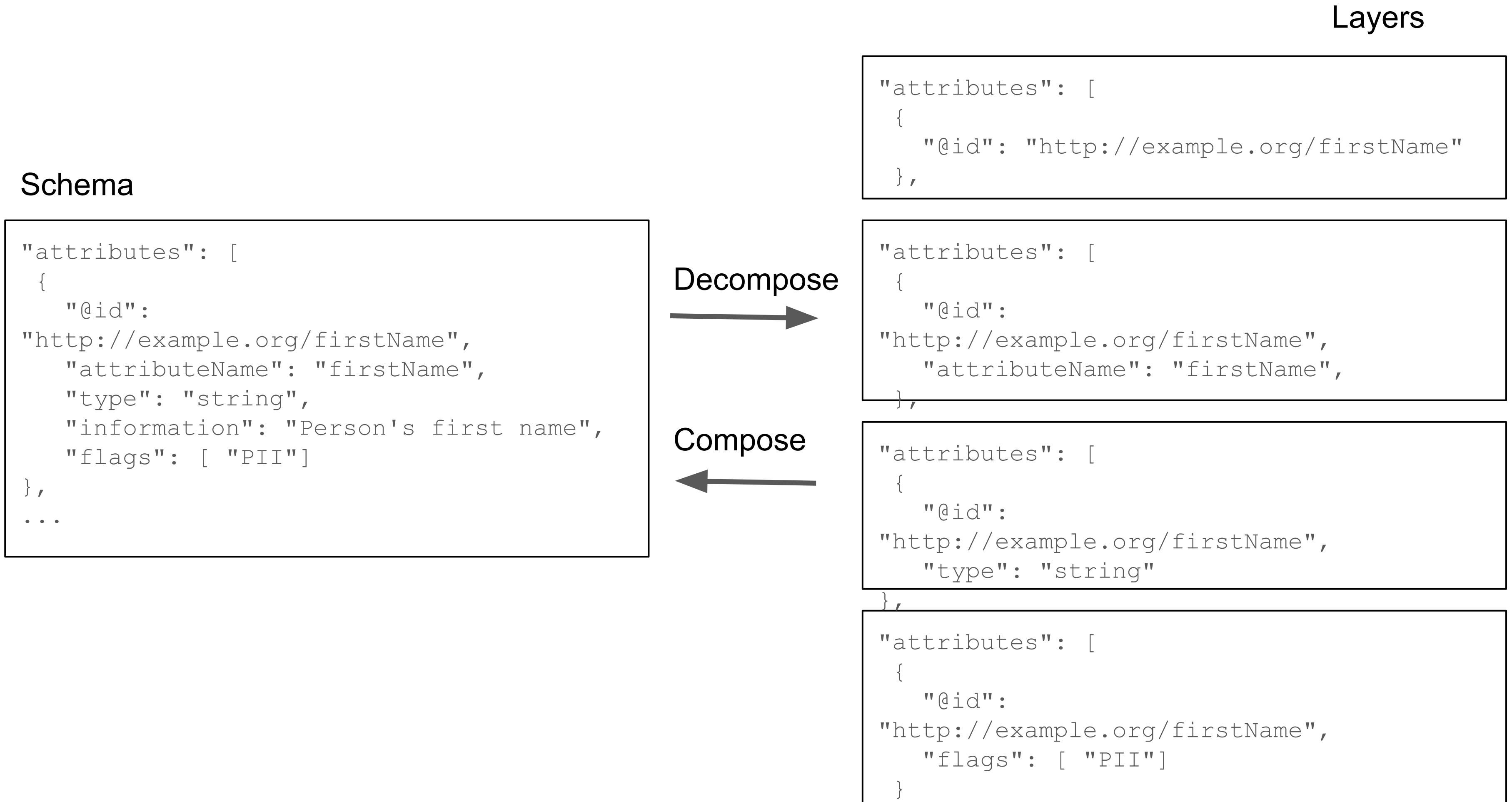


Layered Schemas



OCA	**Layered Schemas**
Schema base + overlays	Layers (schema base is not structurally different from an overlay)
Overlay type determines function (encoding overlay, label overlay, …)	Ontology term determines function (“encoding”, “label”,...)
Flat	Nested objects, arrays, references, polymorphism, composition
Schema base is the addressable object (Person schema base + overlays)	Defined entity is the addressable object (Person + context)

Schema Decomposition/Composition



```
"attributes": [
 {
   "@id":
"http://example.org/firstName",
   "attributeName": "firstName",
   "type": "string",
   "information": "Person's first name",
   "flags": [ "PII"]
},
...
```

```
"attributes": [
 {
   "@id": "http://example.org/firstName"
 },
```

```
"attributes": [
 {
   "@id":
"http://example.org/firstName",
   "attributeName": "firstName",
 },
```

```
"attributes": [
 {
   "@id":
"http://example.org/firstName",
   "type": "string"
},
```

```
"attributes": [
 {
   "@id":
"http://example.org/firstName",
   "flags": [ "PII"]
 }
```

Schema Base

```
"@type": "SchemaBase",
"objectType": "http://example.org/Person",
"attributes": {
  "http://example.org/name": {
   "attributeName": "name"
  },
  "http://example.org/work": {
   "attributes”: {
      "http://example.org/jobTitle": {},
      "http://example.org/department": {}
  },
  "http://example.org/accountId": {
   "reference”: "http://example.org/Account"
},
  "http://example.org/contact": {
   "attributeName": "contact",
   "arrayItems": {
      "reference":
"http://example.org/Contact"
   }
  },
 ...
}
```

Object defined by this schema layer

Attribute ID

Attribute Name (can be in an overlay)

Nested object

Reference to another object

Array attribute

Composition

```
"@type": "SchemaBase",
"objectType": "http://example.org/Person",
"attributes": {
  "http://example.org/USAddress": {
      "allOf": [
         {
           "reference": "http://example.org/basicAddress"
         },
         {
           "attributes": {
             "state": {}
           }
         },
          ...
      ]
   }
  },
 ...
}
```

Polymorphism

```
"@type": "SchemaBase",
"objectType": "http://hl7.org/fhir/Bundle",
"attributes": {
  "http://hl7.org/fhir/Bundle.entries": {
   "arrayItems": {
      "oneOf": [
         {
           "reference": "http://hl7.org/fhir/Patient"
         },
         {
           "reference":
"http://hl7.org/fhir/Encounter"
         },
          ...
      ]
   }
  },
 ...
}
```

Overlay

```
"@type": "Overlay",
"objectType": "http://example.org/Person",
"attributes": {
  "http://example.org/name": {
   "type": "string"
   "privacyClassification":
"http://example.org#BIT"
  },
 "http://example.org/jobTitle": {
   "type": "string"
  },
 "http://example.org/department": {
   "type": "string"
 },
 "http://example.org/newAttribute": {
    "type": "string"
 },
 "http://example.org/accountId": {
    "required": true
 }
 ...
}
```

The object annotated by this layer

Refer to attributes using @id

Metadata for the attribute

Overlay can define new attributes

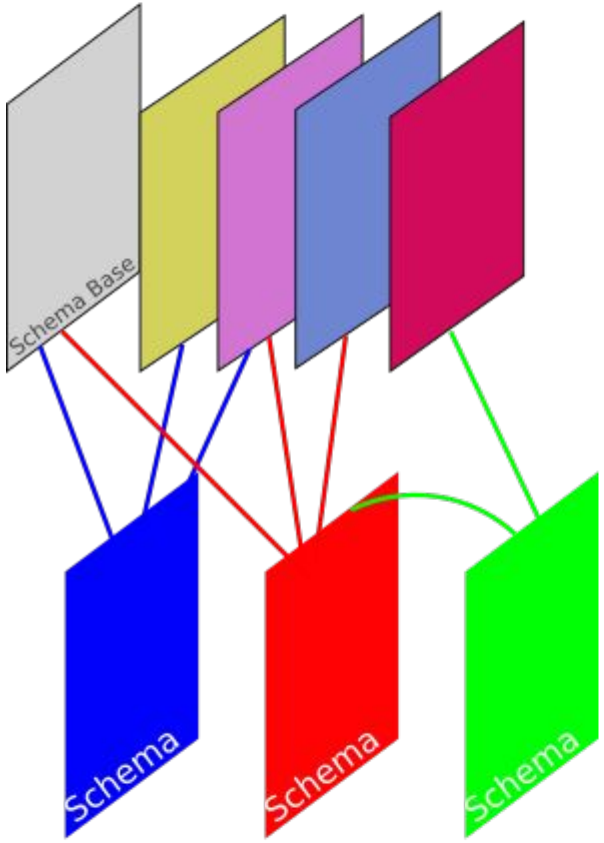
Overlay can add/remove constraints

Schema

The schema links a schema base and overlays to create a schema that is localized, adopted to a particular context/jurisdiction, and versioned. An object may have many variations for different contexts.

```
{
  "@context": "http://schemas.cloudprivacylabs.com/schema.jsonld",
  "@type": "Schema",
  "@id": "schema Id",
  "issuedBy": "...",
  "issuerRole": "...",
  "issuedAt": "...",
  "purpose": "...",
  "classification": "...",
  "objectType": "http://example.org/Person",
  "objectVersion": "...",
  "schemaBase": "http://example.org/Person/schemaBase",
  "overlays": [
     "http://example.org/Person/ovl/info" ,
     "http://example.org/Person/ovl/BIT" ,
     ...
   ]
}
```

Schemas



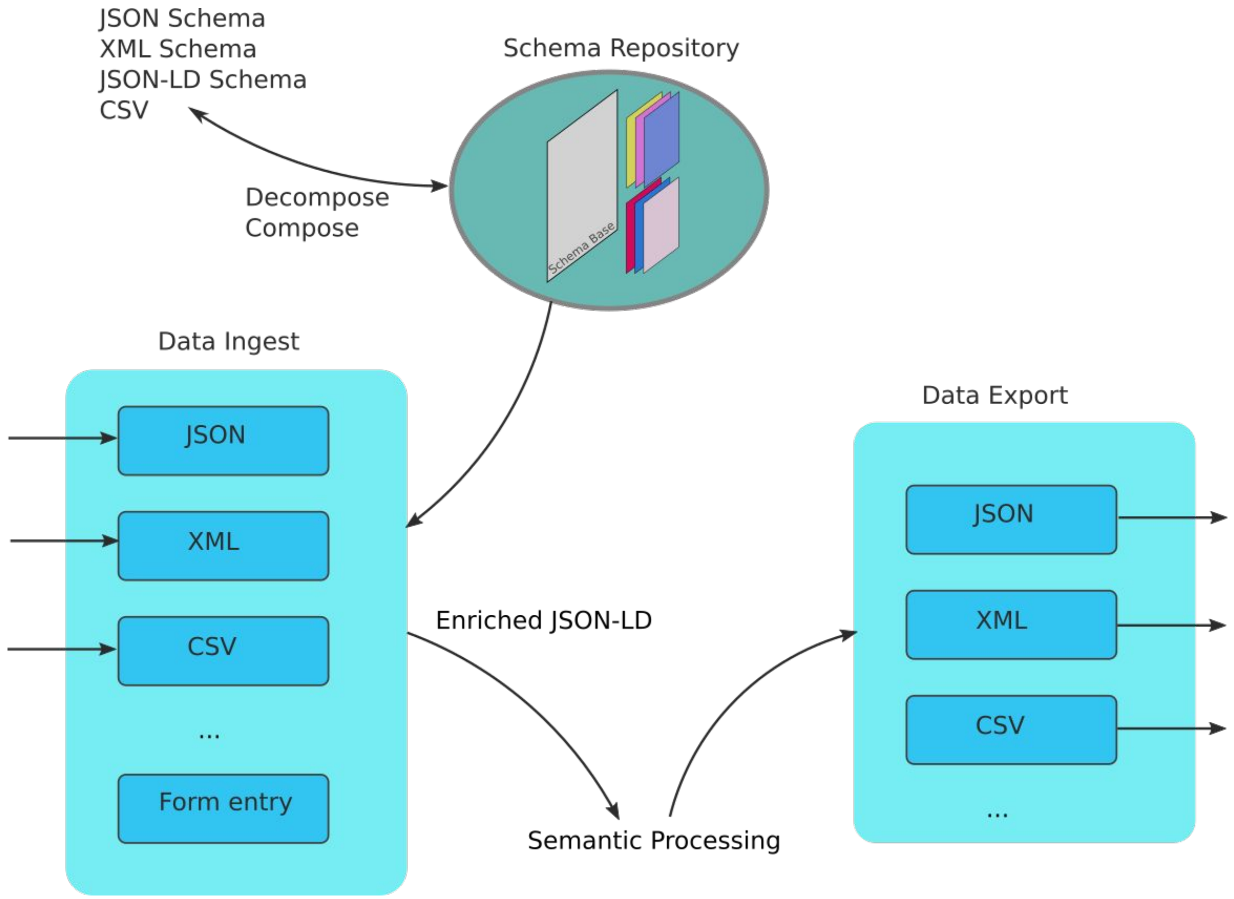
Schema = Schema base + layers

Schema = schema + layers
(use another schema as schema base)

An object can have multiple variations as different schemas

Schema variant selection problem:
Person -> Contact
Person (variant a) -> Contact (which variant?)

Layered Schema Architecture


JSON Schema
XML Schema
JSON-LD Schema
CSV
Schema Repository
Schema Base
Decompose
Compose
Data Ingest
JSON
XML
CSV
...
Form entry
Enriched JSON-LD
Semantic Processing
Data Export
JSON
XML
CSV
...