Cloud Privacy Labs

Selective Disclosure using the LSA

Burak Serdar
Cloud Privacy Labs
bserdar@cloudprivacylabs.com

https://layeredschemas.org
https://github.com/cloudprivacylabs/lsa

LSA ≠ OCA

LSA	OCA
Harmonize disparate data	Standardize data capture
Pipelines (capture, transform)	Schemas + overlays (capture)
Self-describing data: Ingested data includes schema annotations	Schemas + cryptographically linked function-specific overlays
Graph (hierarchical and cyclic structures)	Flat (spreadsheet)

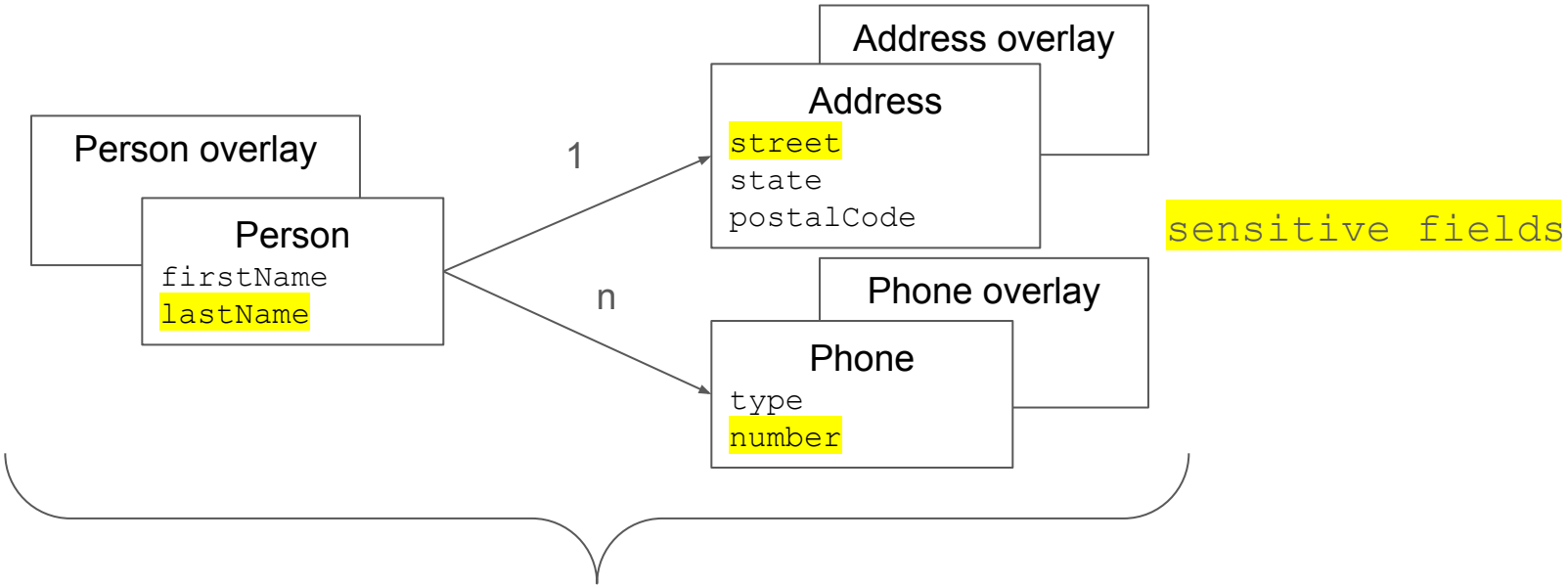
Selective Disclosure

Idea: Use Schema Annotations to Classify Fields

This idea is not new. Some schema languages support annotations

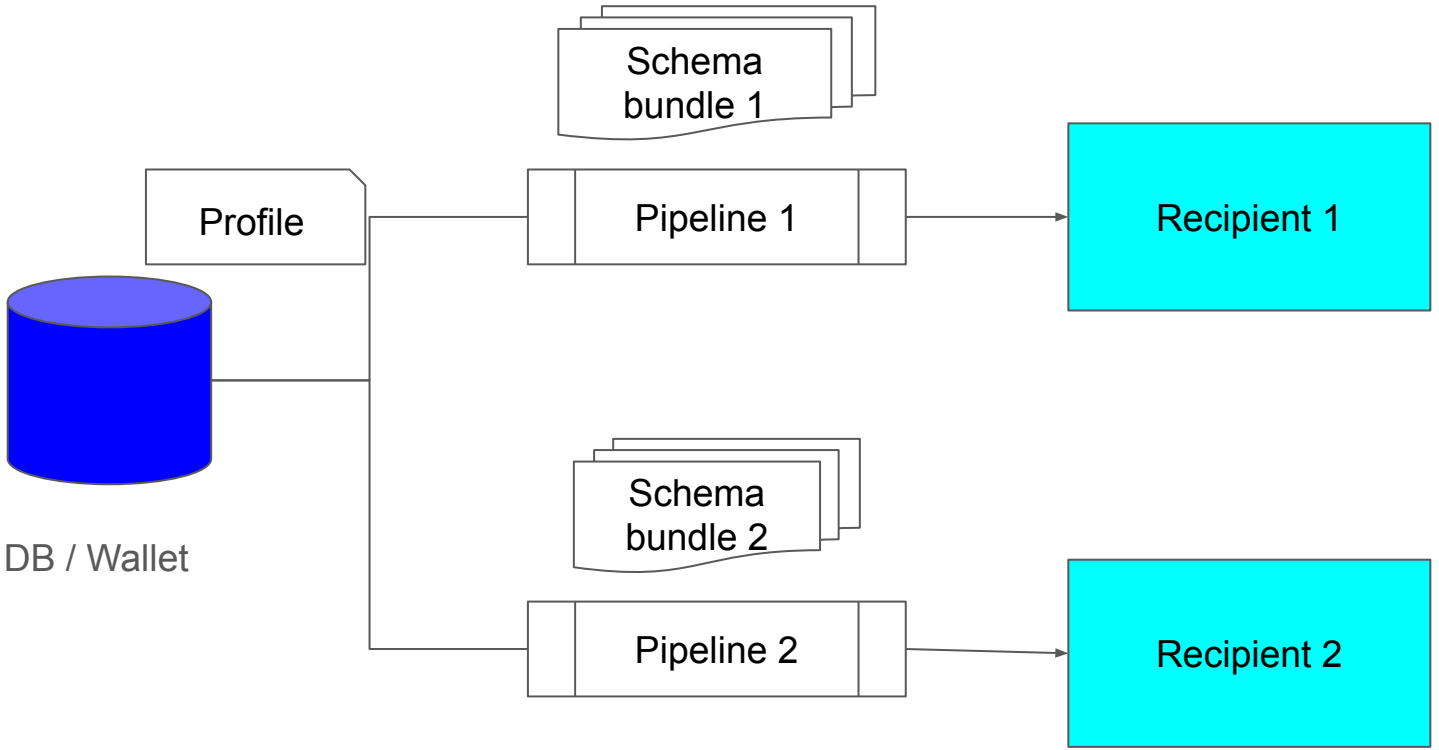
- XML schema supports annotations
- JSON schema supports application specific extensions
- Cap'n Proto (Cloudflare): Data serialization system has built-in schema annotations

Selective Disclosure Example: Profile Data Model



Use a schema bundle to create variants and link them

Separate Schema Bundle for Each Use Case



Example: User Profile Schema and Data

profile.schema.json

```
{
  "definitions": {
    "Profile": {
      "type": "object",
      "properties": {
        "firstName": {"type": "string"},
        "lastName": {"type": "string"},
        "address": {"$ref": "#/definitions/Address"},
        "phone": {
          "type": "array",
          "items": {"$ref": "#/definitions/Phone"}
        }
      }
    },
    "Address": {
      "type": "object",
      "properties": {
        "street": {"type": "string"},
        ...
      }
    },
    "Phone": {
      "type": "object",
      "properties": {
        "number": {"type": "string"},
        "type": {"type": "string"}
      }
    }
  }
}
```

profile.json

```
{
    "firstName": "john",
    "lastName": "doe",
    "address": {
        "street": "123 Main St.",
        "city": "Anycity",
        "state": "CO",
        "postalCode": "80000",
        "country": "US"
    },
    "phone": [
        {
            "type": "cell",
            "number": "123-123 1234"
        }
    ]
}
```

Sensitive Fields

profile.schema.json

```
{
  "definitions": {
    "Profile": {
      "type": "object",
      "properties": {
        "firstName": {"type": "string"},
        "lastName": {"type": "string"},
        "address": {"$ref": "#/definitions/Address"},
        "phone": {
          "type": "array",
          "items": {"$ref": "#/definitions/Phone"}
        }
      }
    },
    "Address": {
      "type": "object",
      "properties": {
        "street": {"type": "string"},
        ...
      }
    },
    "Phone": {
      "type": "object",
      "properties": {
        "number": {"type": "string"},
        "type": {"type": "string"}
      }
    }
  }
}
```

sensitive.ovl.json

```
{
    "definitions": {
        "Profile": {
            "properties": {
                "lastName": {
                    "x-ls":{"privacyLevel": "sensitive"}
                }
            }
        },
        "Address": {
            "properties": {
                "street": {
                    "x-ls": {"privacyLevel": "sensitive"}
                }
            }
        },
        "Phone": {
            "properties": {
                "number": {
                    "x-ls": {"privacyLevel": "sensitive"}
                }
            }
        }
    }
}
```

Cloud Privacy Labs

Sensitive Profile Bundle

profile.schema.json

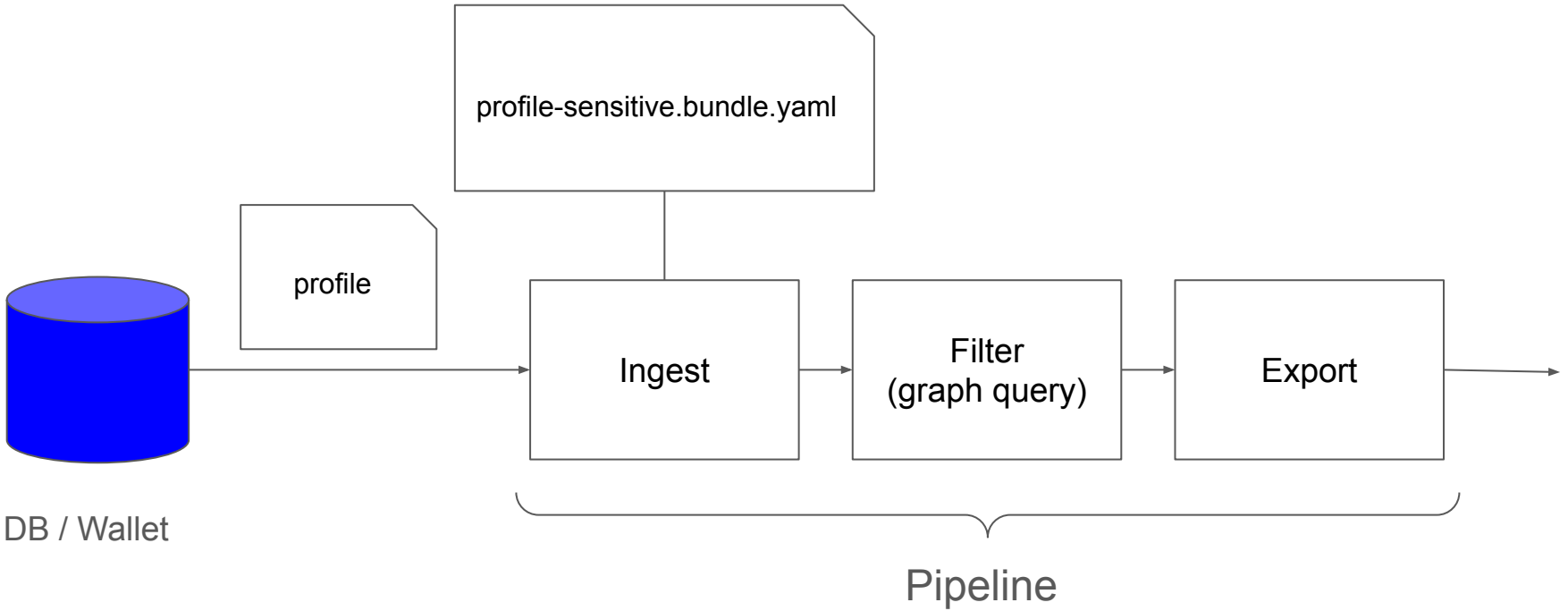
sensitive.ovl.json

profile-sensitive.bundle.yaml

Schema variant adjusted with privacy levels

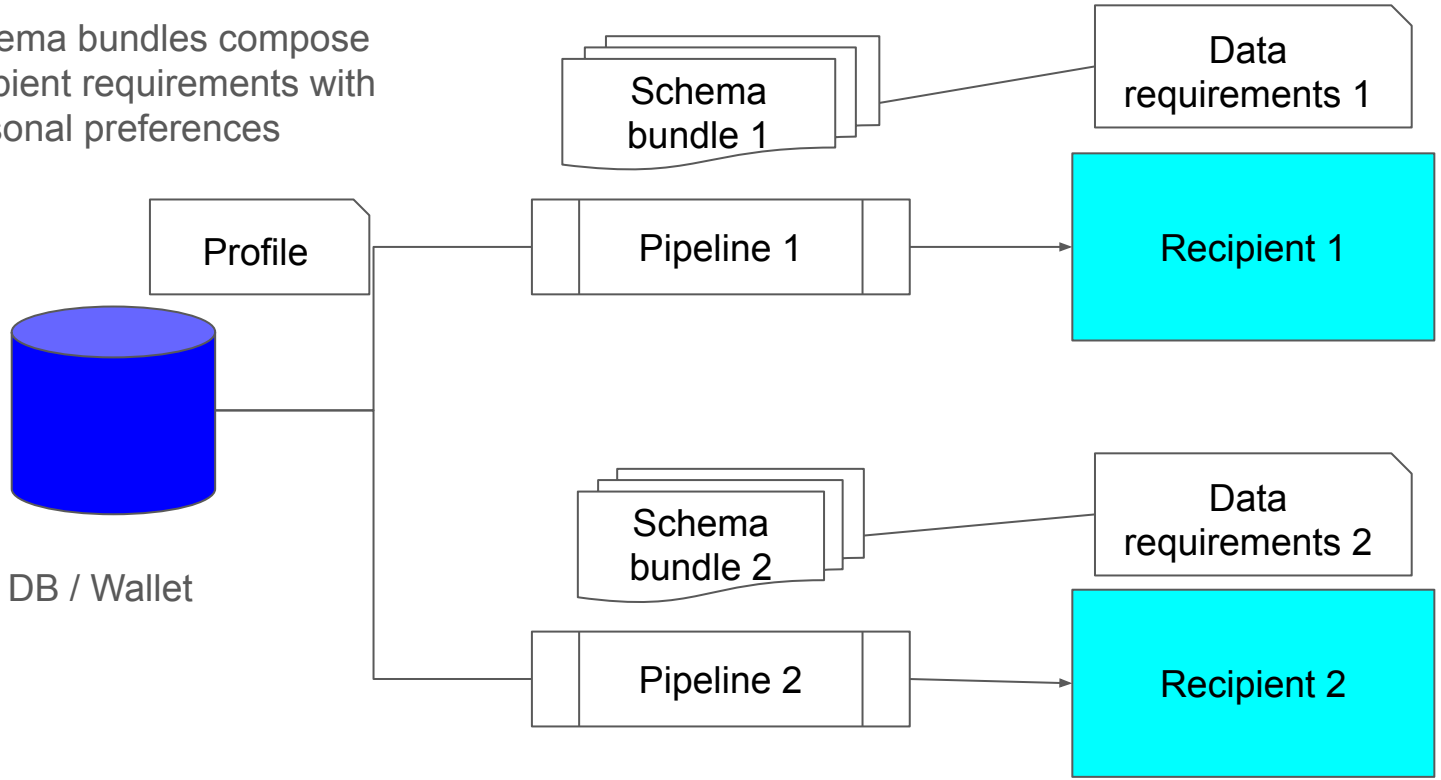
```
{
  "definitions": {
    "Profile": {
        "lastName": {
            "type": "string",
            "x-ls":{"privacyLevel": "sensitive"}
        }
         …
   },
   "Address": {
     "type": "object",
     "properties": {
       "street": {
           "type": "string",
           "x-ls": {"privacyLevel": "sensitive"}
       }
        ...
     }
   },
   "Phone": {
     "type": "object",
     "properties": {
       "number": {
          "type": "string",
           "x-ls": {"privacyLevel": "sensitive"}
       }
     }
   }
 }
}
```

Pipeline


profile-sensitive.bundle.yaml
profile
Ingest
Filter
(graph query)
Export
DB / Wallet
Pipeline

Published Data Requirements

Schema bundles compose recipient requirements with personal preferences



Recipients publish their data requirements as schemas

Use Cases

- API interoperability
- Data capture, display, exchange
- Different schema variant (bundle) for each use case
- Difference schema variants can
  - Add new fields,
  - Add new annotations or modify existing ones
  - Enforce constraints

Limitations:

- Labeling is field-based, not content-based
  - Content of non-sensitive field may be sensitive in different contexts (especially relevant to health-related data)
  - Content classification is part of commercial LSA

Questions?

bserdar@cloudprivacylabs.com