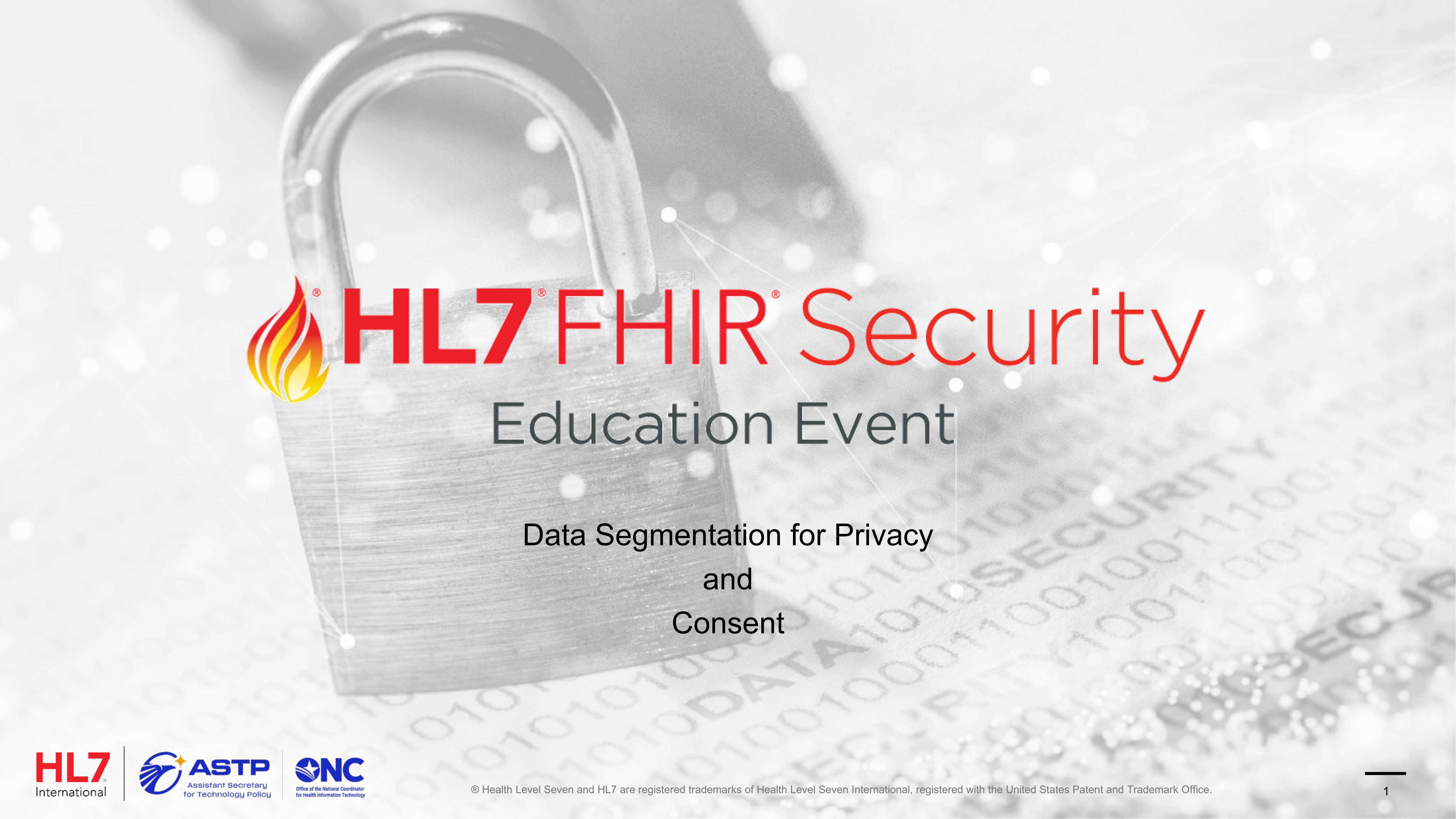


HL7® FHIR® Security

Education Event

Data Segmentation for Privacy

and

Consent

Data Segmentation for Privacy (DS4P)

Identifying certain data elements (*segments*),

that are subject to certain privacy or security controls

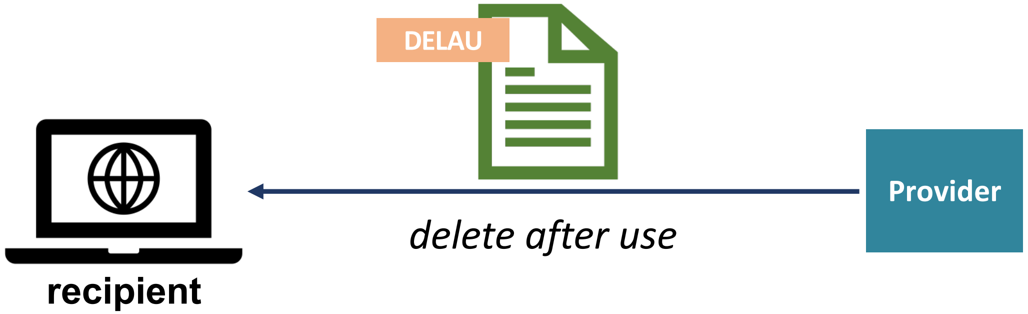
(based on policies),

to restrict access or apply specific access control mechanisms.

Marking these *segments* with metadata called *security labels*.

Security Labels

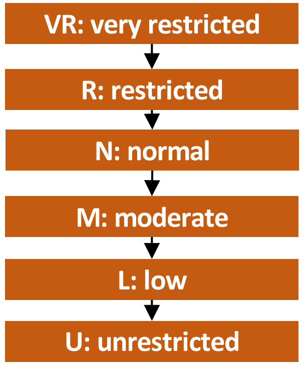
communicate terms of sharing and handling instructions to recipients and consumers



Security Labels

Enforce clearance-based mandatory access control (MAC)

- Implicit policies
- Example: “no reading up”: the user must have a clearance equal to, or higher than the confidentiality level of the resource



VR: very restricted

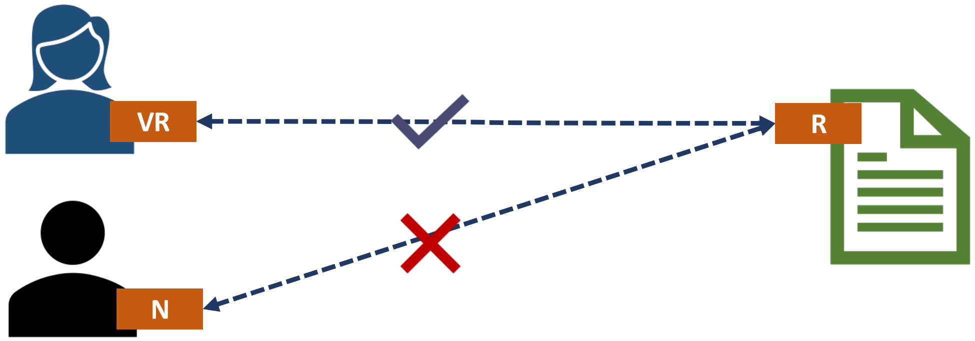
R: restricted

N: normal

M: moderate

L: low

U: unrestricted

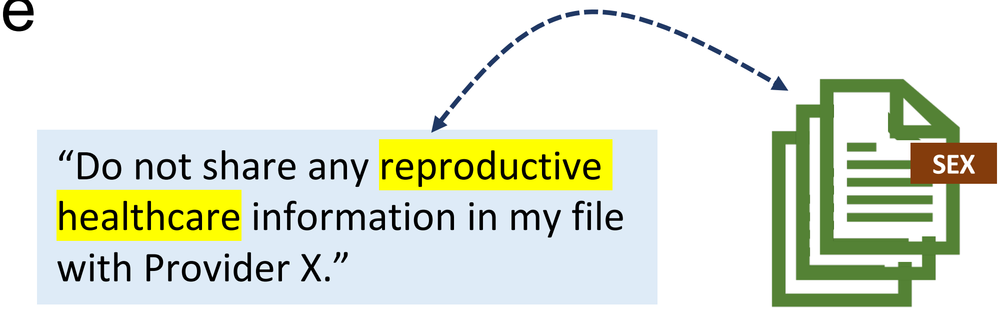


Security Labels

metadata reflecting sensitive content

- associated with a unit of data
  - bundle, resources, or portion of a resource

can be referenced in policies.

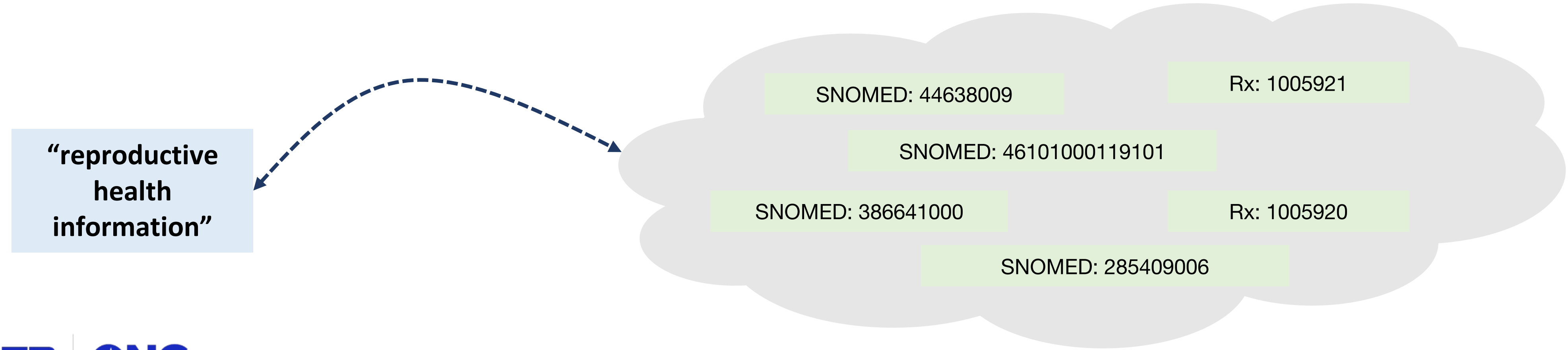


“Do not share any reproductive healthcare information in my file with Provider X.”

Sensitivity Labels

Semantic bridge between clinical vocabulary and known sensitive types of data

- Examples: *substance use treatment*, *reproductive health*, *psychotherapy notes*, *behavioral health*, etc.



“reproductive health information”

SNOMED: 44638009

Rx: 1005921

SNOMED: 46101000119101

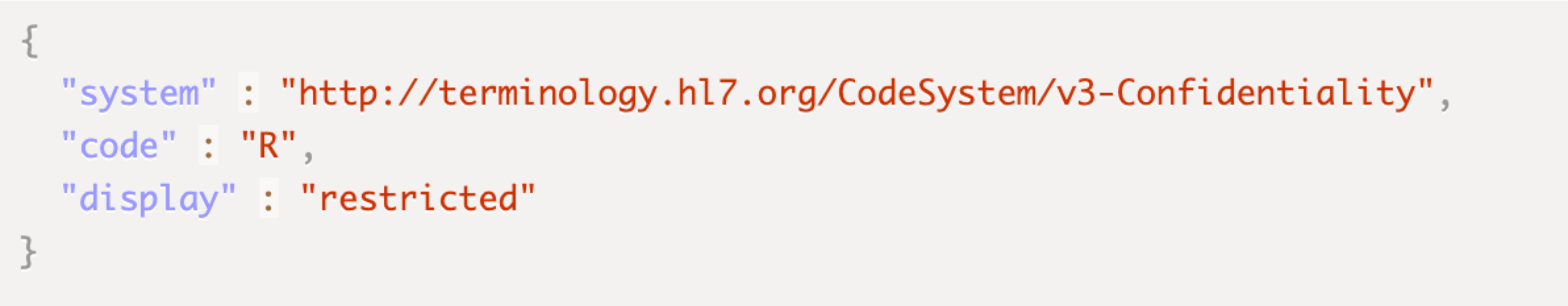
SNOMED: 386641000

Rx: 1005920

SNOMED: 285409006

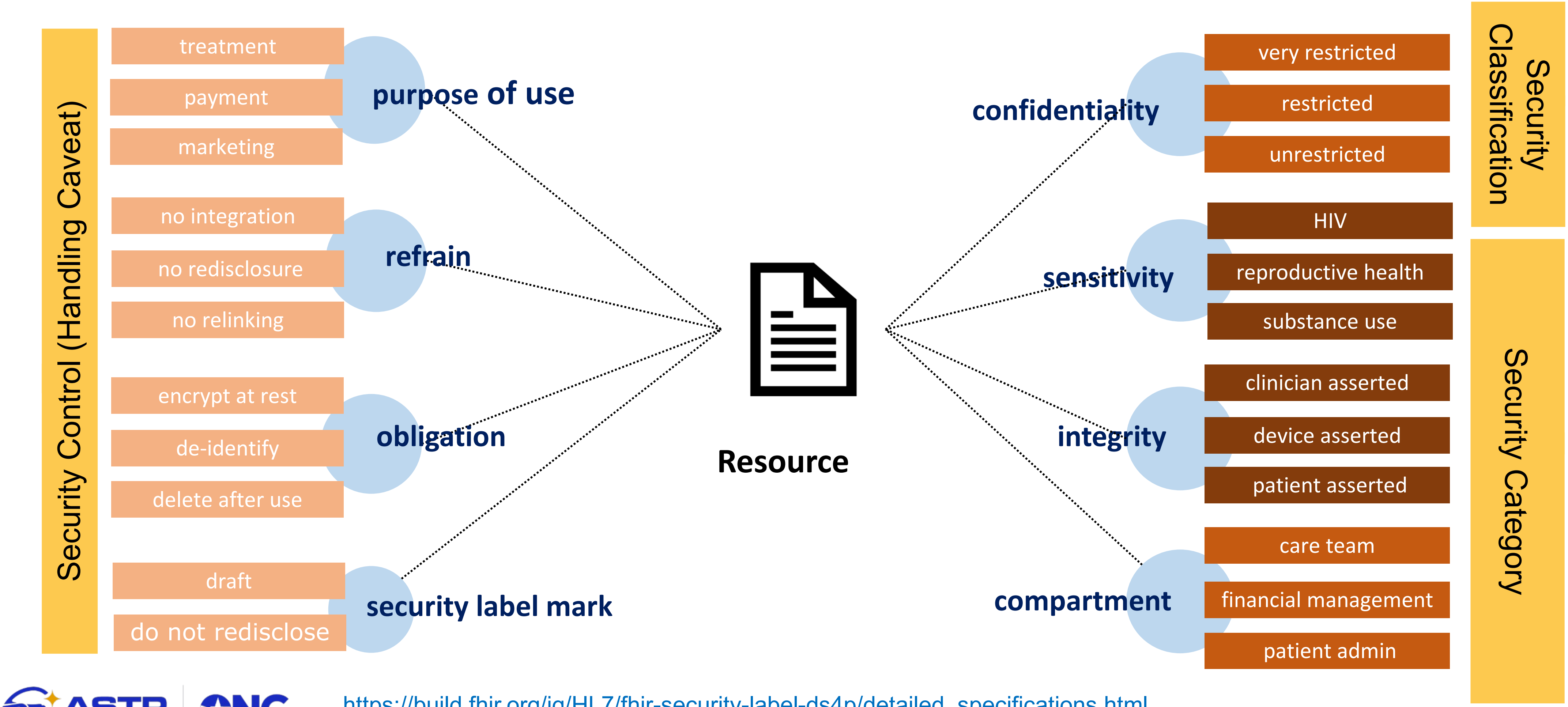
FHIR Security Label

- An instance of `Coding` data type:
  - System
  - Code
  - Optionally, system version and human-readable display value
- Recorded in `Resource.meta.security`



```json
{
  "system" : "http://terminology.hl7.org/CodeSystem/v3-Confidentiality",
  "code" : "R",
  "display" : "restricted"
}
```

Examples of Security Labels



Security Control (Handling Caveat)

purpose of use
- treatment
- payment
- marketing

refrain
- no integration
- no redisclosure
- no relinking

obligation
- encrypt at rest
- de-identify
- delete after use

security label mark
- draft
- do not redisclose

Resource

Security Classification

confidentiality
- very restricted
- restricted
- unrestricted

Security Category

sensitivity
- HIV
- reproductive health
- substance use

integrity
- clinician asserted
- device asserted
- patient asserted

compartment
- care team
- financial management
- patient admin

https://build.fhir.org/ig/HL7/fhir-security-label-ds4p/detailed_specifications.html

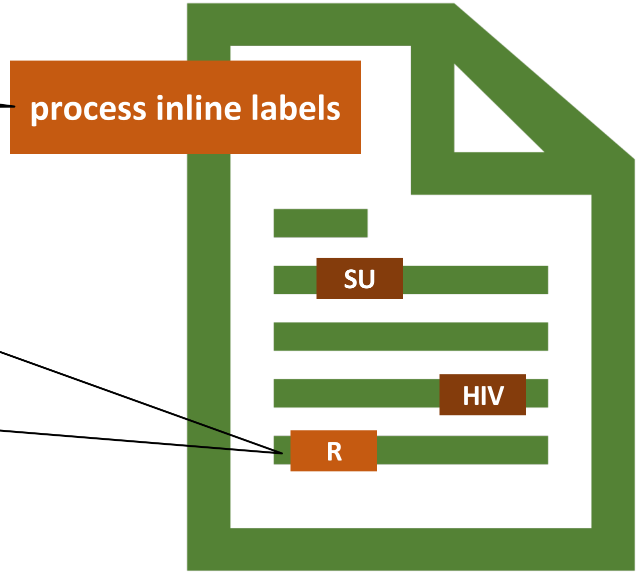
Inline (Sub-Resource) Labeling

- Applying a label to a portion of a FHIR resource
  - e.g., a patient addresses is *restricted*
  - e.g., a patient identifiers is *restricted*
  - e.g., link to the patient (identity) in an immunization resource is *patient reported*
- This extension that can appear anywhere in a resource
- Resource-level security label to direct to *process inline label*

Inline (Sub-Resource) Labeling

```
"meta": {
  "security": [
    {
      "system": "http://terminology.hl7.org/CodeSystem/v3-ActCode",
      "code": "PROCESSINLINELABEL"
    }
  ]
},
```

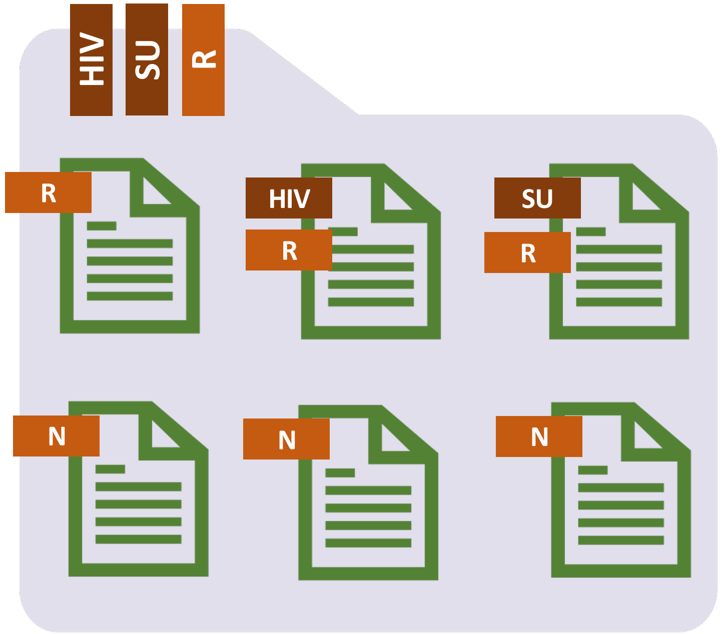
```
"extension": [
   {
     "url": "http://hl7.org/fhir/uv/security-label-ds4p/StructureDefinition/extension-inline-sec-label",
     "valueCoding": {
        "system": "http://terminology.hl7.org/CodeSystem/v3-Confidentiality",
        "code": "R",
        "display": "restricted"
     }
   }
]
```



High Watermark

- Security labels on a collection is determined based on the security labels of its contents
  - document composed of sections, a resource bundle containing multiple resources, resource with inline labels on some portions

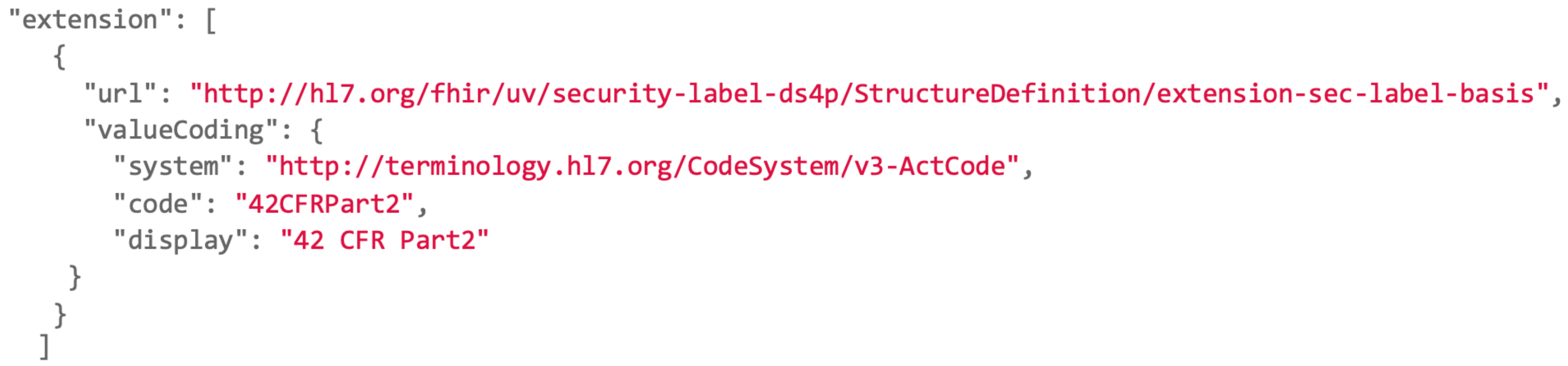
- Determines the safest treatment if granular processing is not possible
- For hierarchical/ordered labels, high watermark is the maximum (most restrictive) value
  - high watermark of a bundle with **Restricted** and **Normal** resources is **Restricted**
- For unordered labels, high watermark is the superset of all values
  - high watermark of a bundle with **HIV** and **Substance Use** resources is {**HIV**, **Substance Use**}



Label Metadata

Recording the *why*

- policy or law based on which a label has been assigned

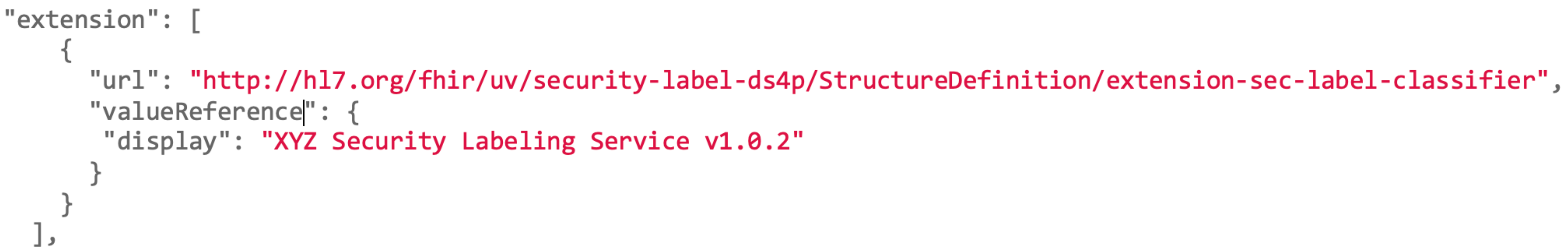


```
"extension": [
   {
     "url": "http://hl7.org/fhir/uv/security-label-ds4p/StructureDefinition/extension-sec-label-basis",
     "valueCoding": {
       "system": "http://terminology.hl7.org/CodeSystem/v3-ActCode",
       "code": "42CFRPart2",
       "display": "42 CFR Part2"
    }
   }
  ]
```

FHIR DS4P Extensions

Recording the *who*

- the entity that has assigned a label
  - e.g., security labeling software service, an individual, an organization, etc.
  - can be repeated to record more than one entity



```
"extension": [
    {
      "url": "http://hl7.org/fhir/uv/security-label-ds4p/StructureDefinition/extension-sec-label-classifier",
      "valueReference": {
       "display": "XYZ Security Labeling Service v1.0.2"
      }
    }
  ],
```

Key Components

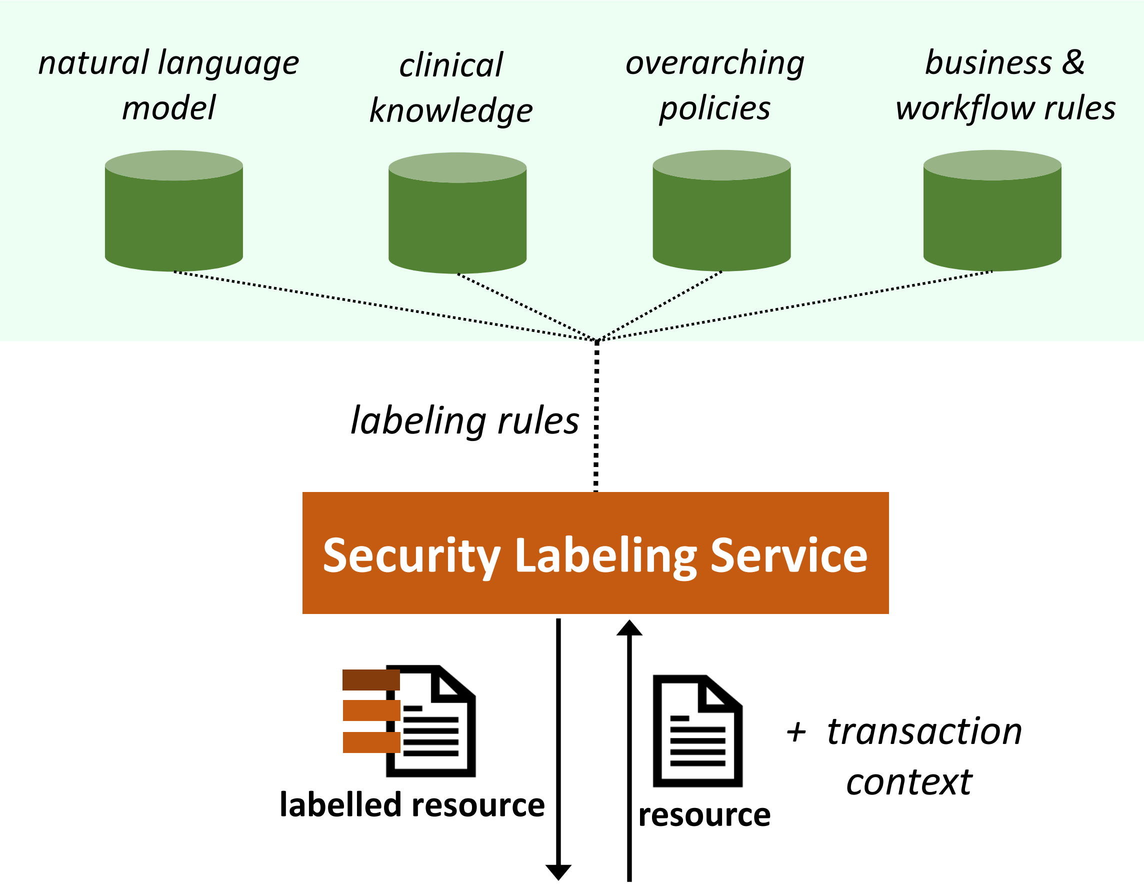
- Determine Labels
  - Determine the policy context
  - Analyze the content
  - Consider the workflow/transaction context
  - Determine whether the data element is subject to specific controls
  - Determine the security labels
  - Record the security labels

Key Components

- Record/Persist Labels
  - data structure to record the label
    - FHIR `meta.security` (as well as extensions)
    - CDA and v2
  - Standard codes for labels
    - HL7 Terminology
- Process Labels
  - Incorporate into authorization decision, e.g., consent enforcement
  - Incorporate into workflow, e.g., route sensitive information differently
  - Incorporate into UI/UX, e.g., render security labels or mask sensitive data.

Security Labeling Service

A rules engine backed by Labeling rules and policies



Input

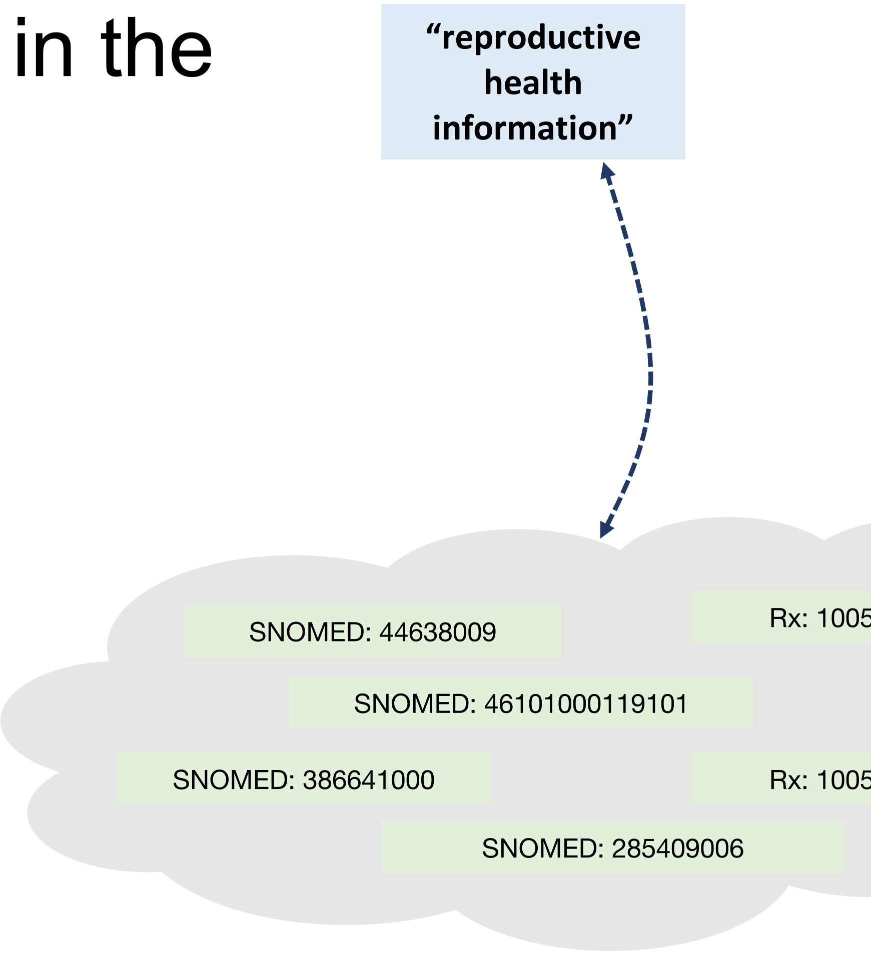
- Resource
  - FHIR resource, bundle, document
- Transaction context
  - recipient identity, purpose of use, etc.

Output:

- Labeled resource
- Labeling metadata
- Justification/reasoning for labeling

Naïve/Rudimentary Labeling

- Simple mapping between clinical codes in the resource and sensitivity labels
  - Intermediate, more fine-grained categories
- More sophisticated processing
  - Additional context
    - Related resources
    - Encounter context
    - Facility type
  - Unstructured text: NLP and LLM



“reproductive health information”

SNOMED: 44638009

Rx: 1005

SNOMED: 46101000119101

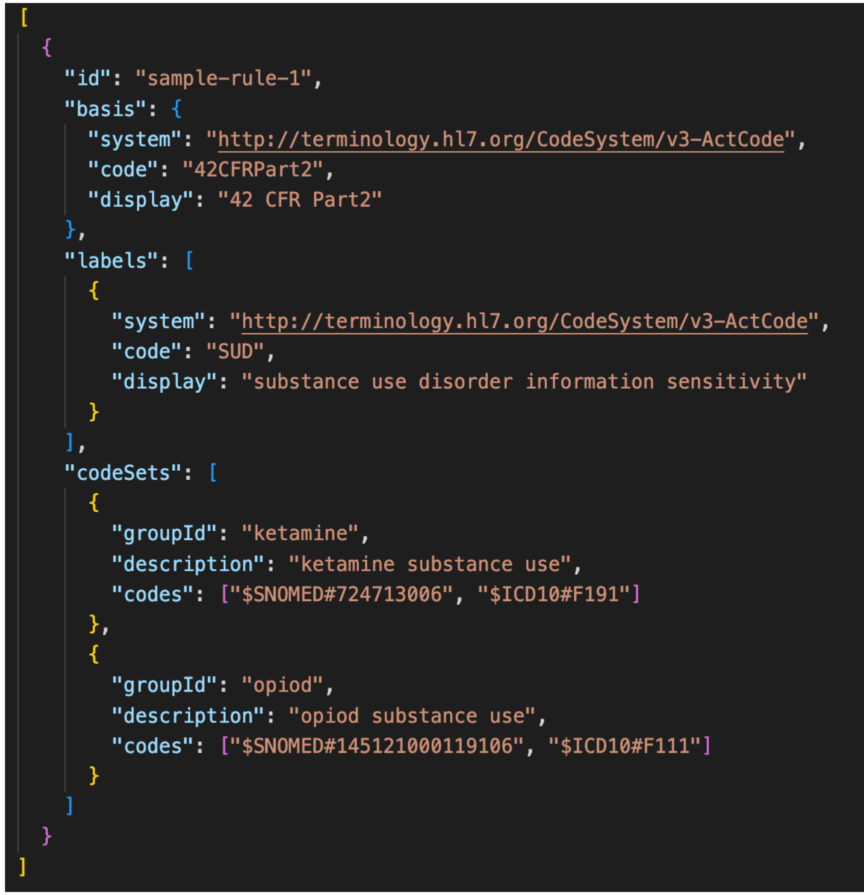
SNOMED: 386641000

Rx: 1005

SNOMED: 285409006

Example: LEAP Consent SLS

- Two-tier mapping of clinical codes to sensitivity codes
- Simple extraction of all coding elements
  - using `json-path`
  - `JSONPath({ path: "$..coding", json: resource })`
- Simple API:
  - Input: bundle
  - Output: labeled bundle
- Support for the `sec-label-basis` metadata



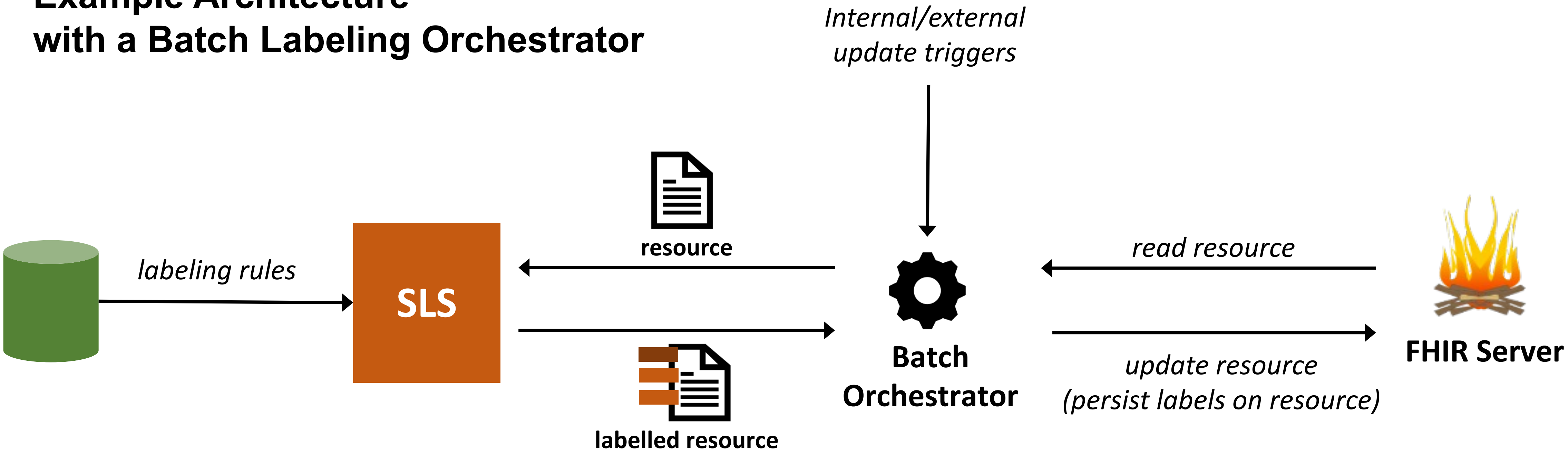
```
[
  {
    "id": "sample-rule-1",
    "basis": {
      "system": "http://terminology.hl7.org/CodeSystem/v3-ActCode",
      "code": "42CFRPart2",
      "display": "42 CFR Part2"
    },
    "labels": [
      {
        "system": "http://terminology.hl7.org/CodeSystem/v3-ActCode",
        "code": "SUD",
        "display": "substance use disorder information sensitivity"
      }
    ],
    "codeSets": [
      {
        "groupId": "ketamine",
        "description": "ketamine substance use",
        "codes": ["$SNOMED#724713006", "$ICD10#F191"]
      },
      {
        "groupId": "opiod",
        "description": "opiod substance use",
        "codes": ["$SNOMED#145121000119106", "$ICD10#F111"]
      }
    ]
  }
]
```

Technical Architecture Considerations

- Where does the labeling service reside?
  - EHR, HIE, third-party
- When does the labeling take place?
  - Batch (offline)
  - At the time of transaction (on the fly)

SLS Implementation Models: Batch

Example Architecture with a Batch Labeling Orchestrator



SLS Implementation Models: Batch Labeling

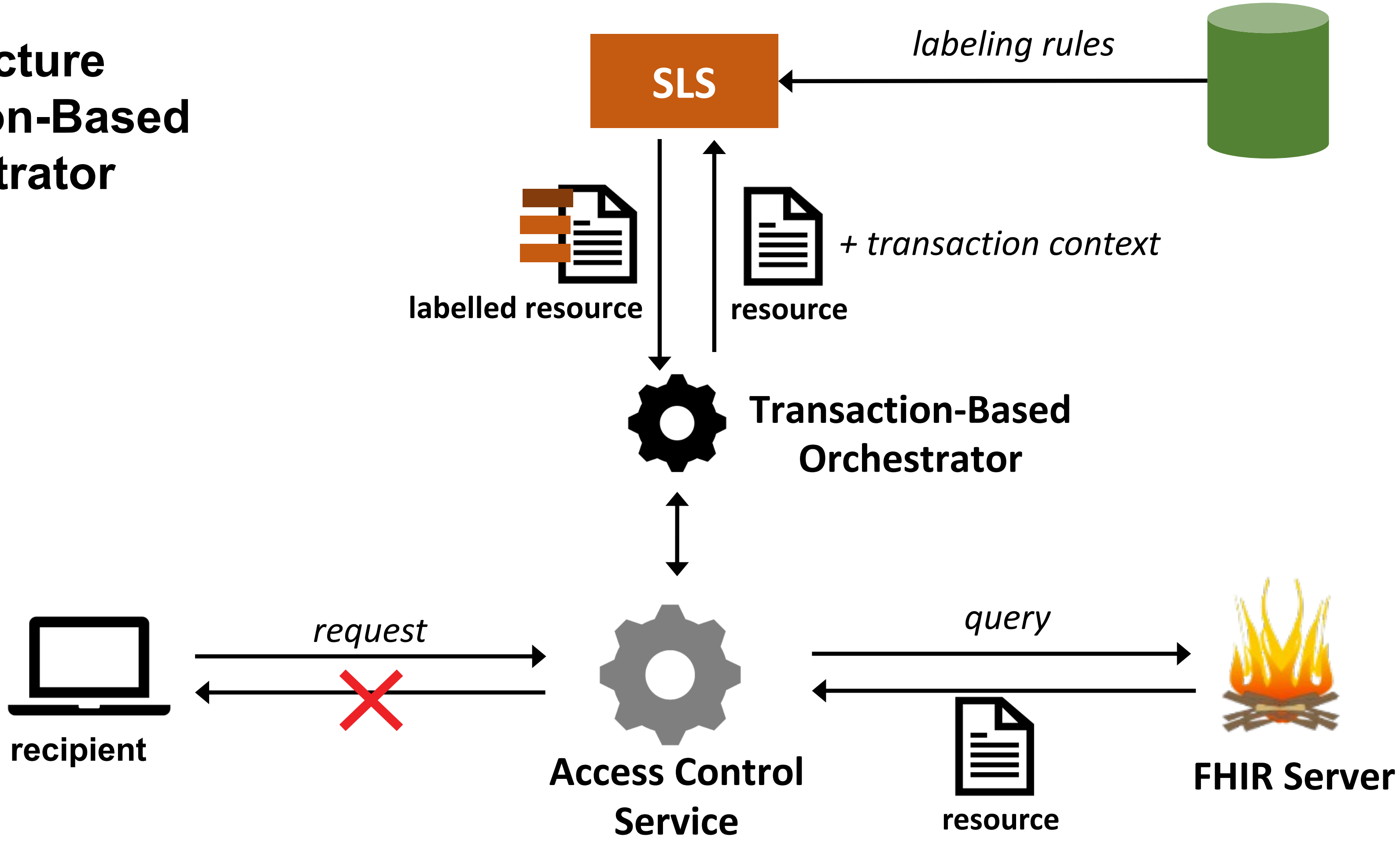
- based on internal or external triggers,
  - bulk import,
  - creating new resource,
  - change in resource content,
  - change in policies, etc.
- can tolerate longer response times
  - Accommodating of heavy computations such as natural language processing (NLP)

SLS Implementation Models: Batch Labeling

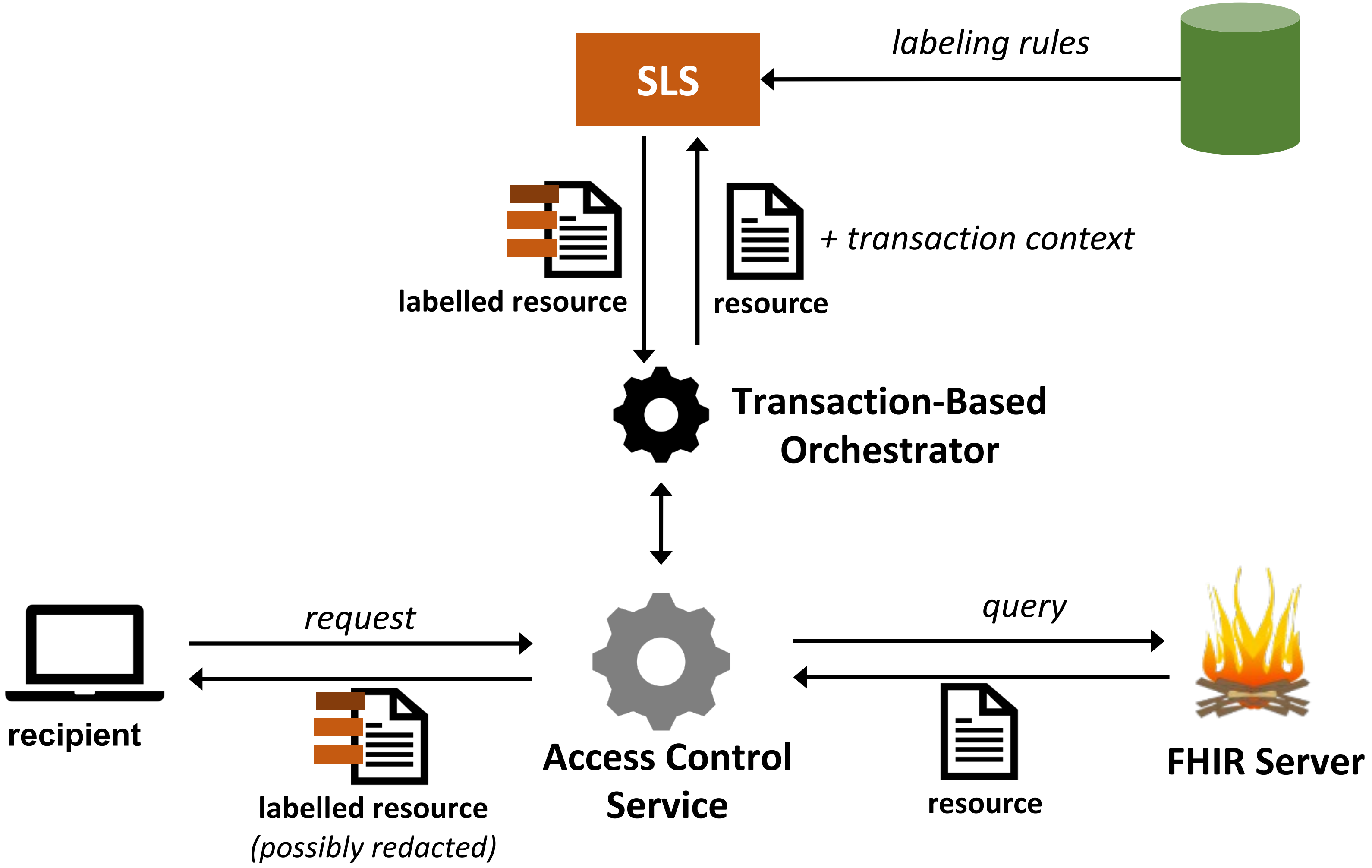
- No transaction context
  - Some transaction-specific labels cannot be determined
- must support the capability to persist labels
  - may not be the case for read-only FHIR adapters
- when resource content or policies change, data may need to be relabeled

SLS Implementation Models: Transaction-Based

Example Architecture with a Transaction-Based Labeling Orchestrator



SLS Implementation Models: Transaction-Based



SLS Implementation Models: Transaction-Based

- The orchestrator has access to *transaction context*,
  - context-dependent labels can be assigned
  - labels based on recipient's identity, purpose of use, etc.
- Labeling based on the most recent version of policies and resource content
- No need for persistence
- Synchronous labeling means some computationally expensive processing are not feasible
  - NLP and LLM

Policy Considerations

- Who is responsible for labeling the data?
- What sensitivity categories must be supported
  - A subset of sensitivity codes to be supported by all entities
- What labeling metadata to record?
- Redact vs. share with labels?
- What are the rules for processing labeling data for the recipient?

Challenges and Gaps

- HL7 specifications are available but need to be actively updated and maintained
- HL7 terminology for sensitive categories need to be overhauled
  - More granular codes
  - Deprecate old codes
  - Update definitions
- More implementation guidance on:
  - Standard HL7 codes to use for different classed of sensitive data identified in US regulations
  - Value sets (of clinical codes) tied to each sensitivity category

QUESTIONS AND DISCUSSION

CONSENT MANAGEMENT

Major Actors

Consenter

- patient, social services client, research participant, etc.

Administrator

- admin at healthcare provider, social services organization, etc.

Third-Party Systems

- Other consent management systems, providers, etc.

Consent Management Use Cases

- Initiate/Request Consent
- Review Consent
- Sign and File Consent
- Browse Consents
- Delegate
- Consent Event Notifications
- Revoke Consent
- View Consent Provenance

Request Consent

- A admin requests a consent from a consenter
  - The consent form may be selected explicitly or implied by the workflow
  - The admin must be able to identify the consenter
  - The consenter may be individually selected or as part of a group
  - The consenter should be notified about the request

- A `Questionnaire` resource captures the consent form to be reviewed by the consenter.
- A `Task` resource records the assignment of the request to the consenter and tracks its status.
- A `Subscription` resource is used to subscribe the requester to the events about this task/consent.
- A `Provenance` resource captures the event as part of the consent provenance

Review Consent

- A consenter navigates and reviews the consent form
  - It may be based on a request or self-initiated.
  - Some fields may need to pre-populated based on the context.

- A `QuestionnaireResponse` resource captures the partially- or fully completed form.
- A `Task` resource records the state of the process.
- A `Provenance` resource captures the event as part of the consent provenance.

Sign and File Consent

- A consenter signs and activates a completed consent form.

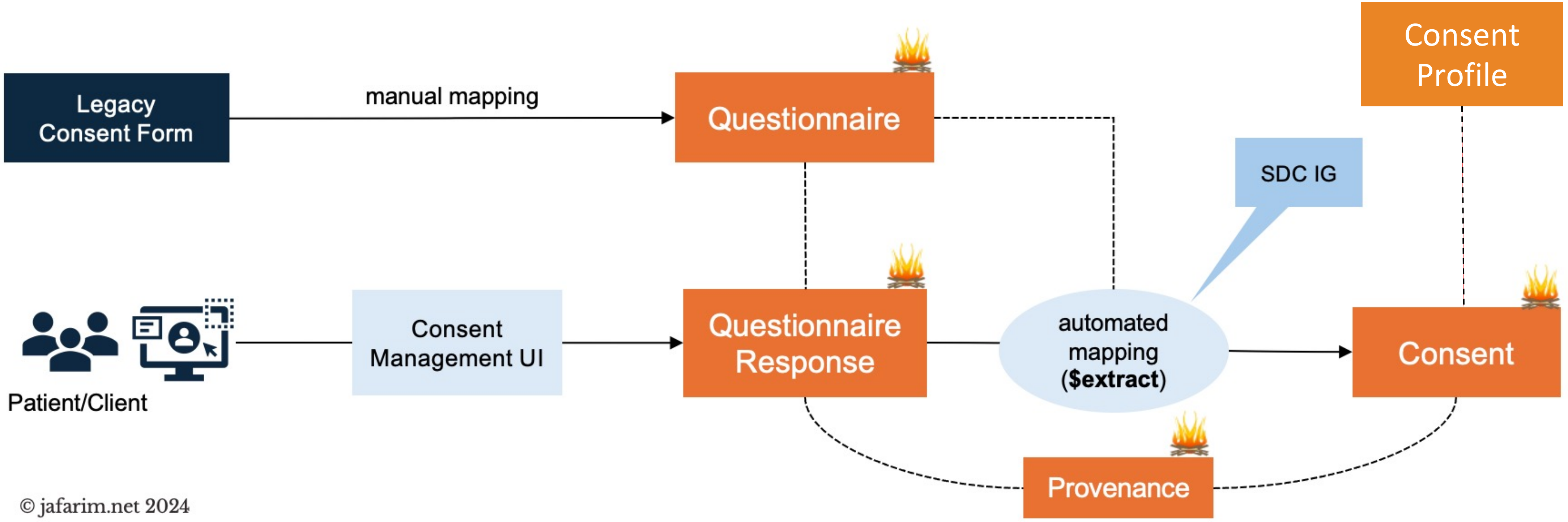
- A `QuestionnaireResponse` resource captures fully completed form.
- A `Task` resource records the completion of the process.
- A `Consent` resource records the final computable active consent.
  - `$extract` operation
  - A consent profile should usually be associated with the consent form.
- A `Provenance` resource captures the event.

Delegate

- A consenter or admin assigns a delegate to make consent decisions on behalf of a consenter.
  - The identity of the delegate should be discoverable.

- `Questionnaire` and `QuestionnaireResponse` resources can capture the delegation form.
- A `Consent` resource can capture the computable form of the delegation policy.

From Consent Form to Consent Resource



© jafarim.net 2024

Browse Consents

- A consenter or admin browses and reviews existing consents.
  - Requested consents (in progress)
  - Active Consents
  - Expired and Revoked Consents

- A `QuestionnaireResponse` captures the original response to the form
- A `Document` resource may capture a print-friendly version (e.g., PDF)
- A `Consent` resource records the computable form.

Review Consent Provenance

- A consenter or admin reviews the provenance of a consent including a record of lifecycle events
  - Requested by whom and when
  - Data and time of signing
  - Date and time of revocation
  - Record of sharing the consent with other parties.

- A `Provenance` resource captures the record of lifecycle events to show.
- An `AuditEvent` resource captures the events of sharing the consent with other parties.

Revoke Consent

- A consenter can revoke an existing active consent.
  - The consenter must be able to find and review their existing active consents
  - The consenter may have to sign a form to formalize the request to revoke

- A `Consent` resource records the change in the status of the consent.
- A `Provenance` resource to capture the revocation event.
- `Questionnaire` and `QuestionnaireResponse` resources may be used to captures the revocation record if required by policy.

Event Notifications

- Consent lifecycle events such as request, review, sign/file, and revocation is shared to authorized parties.

- A `Provenance` captures the record of lifecycle events.
- A `Subscription` resource captures how to share such events to other parties.

QUESTIONS AND DISCUSSION