Security Working Group

Security Labeling Service: Notes on Implementation

Mohammad Jafari,
Kathleen Connor, John M. Davis, Christopher Shawn

Version 1.3

March 17, 2021

Table of Contents

1 INTRODUCTION

This paper presents and discusses some notes on the implementation of Security Labeling Services (SLS) in health information systems, such as an Electronic Health Record system (EHR).

2 COMPONENTS AND FLOWS

While the Security Labeling Service (SLS) provides the functionality of assigning security labels to given data objects, it has to be connected and integrated with the rest of system workflows in order to ensure data objects are labeled as needed. The components and flows are discussed in this section.

2.1 Security Labeling Service

As shown in Figure 1, the SLS determines and assigs security labels to data objects based on different types of applicable labeling rules, such as security and privacy policies, patient consent, and business and workflow requirements. In general, the SLS should be able to label any healthcare data object (e.g., Health Level Seven v2 (HL7v2) messages, Composite Clinical Document Architectures (CDAs), or Fast Healthcare Interoperability Resources (FHIR) resources), but an implementation may choose to support only a subset of all possible such data types. For simplicity, this paper focuses on FHIR resources, although the concepts are straightforwardly applicable to other types of clinical data objects.

The SLS provides an Application Programming Interface (API) for assigning labels to data objects. An SLS client is any internal service invokes the SLS to get the applicable security labels to a resource. To invoke the SLS, the client should provide:

- a data object (e.g., a FHIR resource),
- other relevant contextual information (e.g., the transaction metadata including the client's identity or roles and its purpose of use), and
- requested types of labeling.

In response, the SLS provides a labeled data object to the client. This may be an individual data object or a collection data object containing other data objects (e.g., a FHIR bundle).

By specifying the type of security labels in its request, the client can control the level of processing involved in the labeling. For example, the client may specify that it is only interested in assigning *handling caveats* since the resource has already been assigned sensitivity and confidentiality labels beforehand. Or, the client may request labeling for unstructured text portions of the data object, which requires invoking the Natural Language Processing (NLP) to process the unstructured text and extract clinical concepts in order to determine the applicable sensitivity labels.

This parameter gives the client the flexibility to minimize the SLS processing by choosing the details of the types of labels that are of interest, for example, if the data object has already been partially labeled with sensitivity labels, the client can request that the SLS only assigns the confidentiality labels. Furthermore, the client can use this parameter to choose the depth of inspection based on the context of the labeling request to avoid computationally expensive operations such as the NLP, when it is not needed or does not suit the synchronousness requirements of the workflow context. For example, a client labeling a set of resources imported via a bulk transfer may choose to activate the NLP since there is no real-time requirements for the labeling of the resources, but a client working in the context of real-time transactions may choose to opt out of the computationally expensive NLP processing to avoid the potentially prohibitive delays.

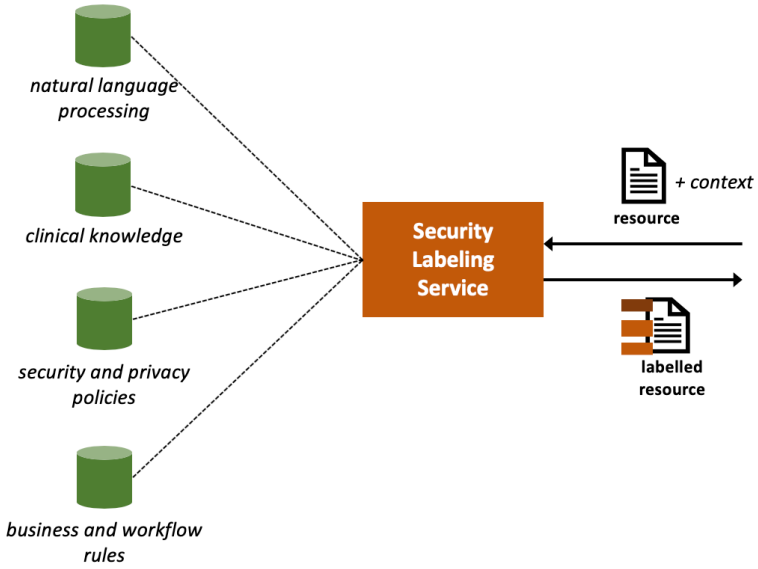

Figure 1: Security Labeling Service

Depending on the level of effort and types of labeling requested, the SLS interface may be synchronous or asynchronous. If the computational effort for labeling a resource is high, the SLS may not be able to guarantee completion within a reasonable time that meets the timeout requirements of a synchronous protocol. This can be the case when the data objects are very large (e.g., large CDA documents) or when a computationally-heavy processing such as NLP is involved.

As shown in Figure 2, in asynchronous cases, the SLS accepts the labeling request but instead of returning a response, returns a message indicating that the request has been accepted and will be processed in due course. This message also includes a reference which the client can use in subsequent calls to check whether the results are ready and optionally receive a job progress value (e.g., *the processing is 60% done*) to assist the client in determining when to schedule the next call-back, and also to provide feedback and responsiveness when the client is facing a human user. In this polling model, the client must use discretion as to when and how frequently to check again with the SLS to see if the labeling job has been completed to fetch the results.

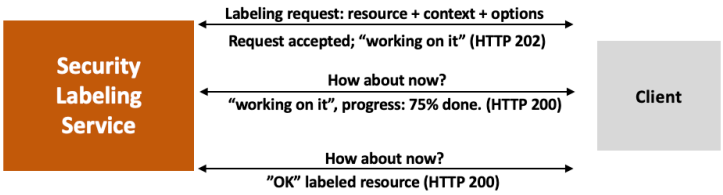

Figure 2: Asynchronous calls to SLS

Alternatively, the SLS can support a callback model in which the client registers an endpoint with the SLS to call and post the results when the labeling job is completed. This enables the client to avoid polling which could be wasteful of computational resources and network bandwidth.

2.2 Labeling Orchestrator

The SLS provides the core capability of assigning security labels to data objects; but in order to be used in an EHR, it must be integrated with the rest of the system workflows in order to ensure correct labeling exists when resources are accessed locally or shared with another system. Labeling orchestrators are the software components providing this integration logic. Labeling orchestrators are tightly coupled with the EHR workflows; the main functions of the Labeling orchestrator are:

- Determining when and at what point in the workflow the SLS must be invoked.
- Determining the data objects that must be submitted to be labeled.
- Modifying data objects and replacing them with their labeled version in the workflow.

Generally, an efficient implementation of security labeling hinges on a hybrid approach where the two types of labeling orchestrators are used in tandem.

2.2.1 Batch Orchestrator

A batch orchestrator submits resources for labeling on an offline basis, outside of active transactions. Because this takes place in an offline workflow, it can handle longer processing times and asynchronous responses, and therefore, more costly processes such as NLP. On the other hand, since the labeling is outside the context of a transaction, transaction-specific labels (such as *handling caveats* that depend on the transaction-specific details) cannot be determined by this type of orchestrator.

A system may have multiple batch orchestrators that monitor different events and queue the data objects that need to be labeled (or re-labeled). Once the response from the SLS is received and incorporated, the security labels are persisted on the data object.

Often, a priority strategy is necessary to determine which data objects should be labeled first. For example, data objects that are more likely to be sensitive (e.g., diagnosis notes imported from a mental health provider), or data objects that are more likely to be accessed (e.g., newer data objects) may be prioritized for labeling.

Some examples of batch orchestrators are discussed in the rest of this section.

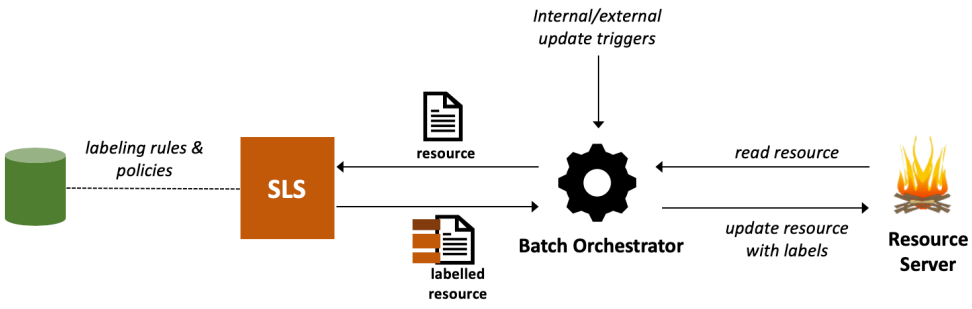

Figure 3: Batch Orchestrator

2.2.1.1 Legacy Data Orchestrator

When the SLS is deployed or activated at an EHR for the first time, a batch orchestrator must initiate the labeling by creating an index of all pre-existing data objects in the system and form a queue to submit them to the SLS based on a suitable priority strategy as discussed above.

2.2.1.2 Bulk Data Orchestrator

When new data objects are imported to the EHR in bulk (by using the standard bulk data transfer or by other mechanisms such as a direct database import), the orchestrator must queue the new data objects for labeling. In some cases, such as a jurisdictional change, data objects may be subject to new labeling rules and policies, so even if the data is labeled, a re-labeling may be necessary.

2.2.1.3 Event-Based Orchestrator

An event-based orchestrator monitors important events and queues the affected data objects for labeling or re-labeling. For example, after creating a new data objects or after updating an existing data object, and depending on the type of the date object created or the type of update, labeling or re-labeling may be necessary. As a more specific example, in a FHIR service where automatic subscriptions can lead to new FHIR resources or updates to existing resources, an event-based orchestrator should monitor the endpoint at which the subscription notifications are received and queue resources implied in a create or update.

The orchestrator may implement sophisticated strategies in determining whether an update to the content of an existing data object is substantive enough to warrant a re-labeling. For example, an update to a prescription that adds a new medication item, or removes an existing one, potentially changes the sensitivity labels of the prescription data object; but an update to the state of a prescription (e.g., from *active* to *stopped*) is unlikely to lead to a change in the sensitivity labels and does not warrant a re-labeling.

Another example for an event that could call for re-labeling is a change in policies, for example, a change in the patient consent, that might require re-labeling some data objects accordingly.

2.2.2 Transaction-Based Orchestrator

Transaction-Based Orchestrators submit data objects to the SLS on-the-fly and in the course of a transaction. Since this orchestrator is aware of the transaction context (e.g., the identity of the recipient and the purpose of use), it can submit the transaction context metadata to the SLS to enable assigning transaction-dependent labels such as *handling caveats*.

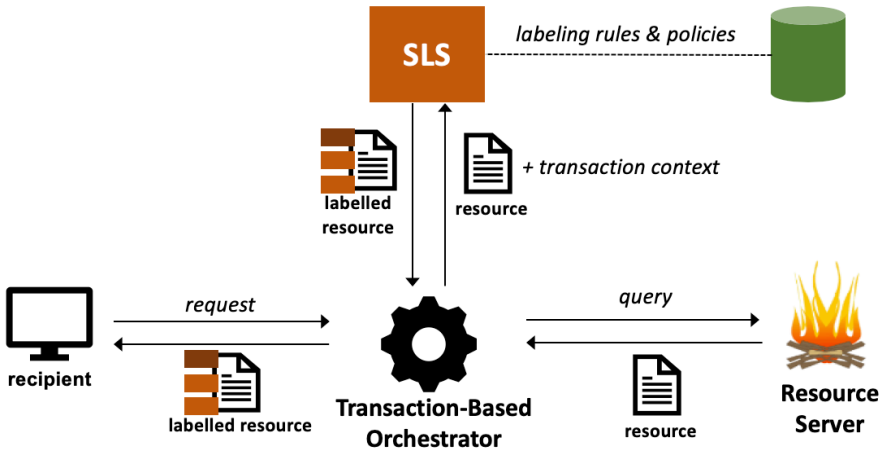

Figure 4: Transaction-Based Orchestrator

3 ACRONYMS

API	Application Programming Interface
CDA	Clinical Document Architecture
EHR	Electronic Health Records (System)
FHIR	Fast Healthcare Interoperability Resources
HL7	Health Level Seven
NLP	Natural Language Processing
SLS	Security Labeling Service