ÆGLÉ
INFORMATIQUE MÉDICALE

Rapport LabBook Connect

Phase 1

08 décembre 2023

Contact pour ce projet :

M. Philippe BRUN
phb@aegle.fr

FONDATION MÉRIEUX
M. Nicolas STEENKESTE
17, rue Bourgelat
69002 LYON

LabBook

Rapport LabBook Connect Phase 1

1. Objectifs Phase 1

La phase 1 du projet LabBook Connect comprenait plusieurs éléments :

- l'étude d'une solution basée sur des composants standardisés OpenHIE avec un accès FHIR,
- la prise en charge dans LabBook des éléments nécessaires aux échanges avec les analyseurs,
- une première implémentation de LabBook Connect incluant un simulateur d'analyseur.

2. Étude

L'étude réalisée s'est focalisée dans un premier temps sur le projet Instant OpenHIE et plus particulièrement son Core Package qui fournit une mise en œuvre de HAPI FHIR connectée à OpenHIM dans une plateforme de conteneurs d'applications Kubernetes.

L'espoir en utilisant cette plateforme standardisée était de disposer d'une base sur laquelle raccorder des compsants déjà mis en œuvre dans le cadre d'autres projets. Il était clair dès le début qu'il n'existait rien de standardisé concernant la connexion d'analyseurs de laboratoire, mais il était intéressant d'explorer la bibliothèque des éléments disponibles à la recherche de composants matures utilisables dans le cadre du projet LabBook.

Si de nombreux prototypes sont disponibles, nous n'avons pas trouvé de composants suffisamment matures pour être utilisés dans un contexte de production.

En revanche, cette prise de contact avec la communauté OpenHIE nous a permis de participer aux premiers pas d'une initiative de mutualisation des développements de connecteurs d'analyseurs de laboratoire Decoupled Analyzer Interface System (DAIS platform) de la part de la communauté Laboratoire de OpenHIE OpenHIE Lab Information Systems Community. Cette initiative, si elle est encore dans une phase préliminaire, peut représenter une opportunité de mutualisation et de visibilité pour le projet LabBook Connect. Nous allons naturellement suivre son évolution, y participer dans la mesure du possible, et bien sûr nous conformer au mieux à ses préconisations.

Par ailleurs, grâce aux contacts de Mr Steenkeste avec BioMérieux, nous avons pu échanger avec diverses personnes impliquées à la fois dans le développement et la mise en œuvre de middleware de connexion d'analyseurs ainsi que dans les instances de standardisation internationales comme IHE https://www.ihe.net/ et HL7 http://www.hl7.org/index.cfm ou des consortiums de fournisseurs d'analyseurs comme IVD Industry Connectivity Consortium (IICC).

A l'issue de ces différentes expérimentations et contacts, nous avons statué sur la première réalisation concrète de LabBook Connect qui s'articule autour de standards et reste au maximum ouverte en prévision des évolutions de OpenHIE. Cette position est détaillée dans un document “LabBook Connect first steps” que vous trouverez en annexe. En résumé il s'agit d'un middleware qui implémente les transactions standardisées par IHE-LAW au format HL7v2 à travers des échanges HTTP qui sont la norme au sein des middlewares santé.

3. Réalisation

Les premiers développements ont permis d'ajouter à LabBook des éléments fonctionnels qui lui manquaient pour gérer les échanges avec les analyseurs :

- un écran de visualisation des analyseurs connectés.
- une adaptation de la saisie de prescription d'analyses qui permette de lier les analyses demandées avec les produits pathologiques des prélèvements réalisés,
- la gestion du dialogue avec LabBook Connect avec principalement l'envoi des demandes d'analyses et la réception des résultats.

Ces développements de LabBook seront intégrés dans LabBook version 3.4.

Côté LabBook Connect, une fois le cadre fixé, les choix principaux de réalisation ont été assez rapides :

- Langage de développement ; language java, utilisé de manière très majoritaire dans le domaine de l'interopérabilité de santé, ainsi que pour la plateforme DAIS évoquée plus haut.
- Architecture technique : conteneurisée pour être compatible avec OpenHIE et s'installer facilement sur une machine LabBook qui utilise déjà cette technologie.
- Architecture logicielle à base d'extensions qui permettent d'isoler le code spécifique à un analyseur dans un composant logiciel autonome pouvant être développé en dehors du projet LabBook Connect

Ainsi LabBook Connect est composé en pratique :

- d'un composant central qui gère les échanges avec LabBook,
- de plusieurs extensions logicielles (plugins, 1 par type d'analyseur) correspondant aux analyseurs présents dans le laboratoire. A l'issue de cette phase 1 le seul plugin disponible simule un analyseur générique.

Pour ajouter un nouvel analyseur dans une installation LabBook + LabBook Connect, il suffit de télécharger quelques fichiers :

- la description du type d'analyseur,
- la table de correspondance entre le dictionnaire de LabBook et celui de l'analyseur,
- l'extension logicielle (plugin) spécifique au type d'analyseur

4. Conclusion

Cette phase 1 du projet LabBook Connect a permis de mettre en place une architecture sur laquelle nous allons pouvoir capitaliser au cours de la phase 2 avec le développement d'extensions pour les divers analyseurs prévus.

Si cette première phase n'a pas encore permis de mettre en œuvre les middleware open source envisagés, qui sont avérés trop peu aboutis à ce stade, l'architecture choisie qui respecte au maximum les standards est celle qui pourra le plus facilement s'adapter aux évolutions apportées par les diverses initiatives recensées.

ANNEXE 1

LabBook Connect first steps

1 - Context

The LabBook Connect project's first goal is to connect LabBook to analyzers (In Vitro Diagnostic devices), but its architecture must also allow future data exchanges with other types of healthcare informatics systems like Electronic Medical Records or Health Management Information Systems.

In this context, the OpenHIE Framework seems a good starting point with a modular standards- based architecture. Following these guidelines, the LabBook Connect project is planned to leverage the following components :

- a FHIR server to store the information exchanged between systems,
- a middleware component for routing, orchestrating and translating requests between systems

2 - Connecting to IVD devices

Connections between IVD devices and healthcare informatics systems in clinical laboratories have been in place for years. Standards are being developped but some of the interactions remain specific.

Connecting various IVD devices to a LIS like LabBook means dealing with at least 2 specific elements:

- managing the physical connection to the analyzer and the messages and workflows it implements,
- mapping the various codes used inside the LIS to those of the analyzer.

In this context some unifying efforts seem relevant:

- The IVD Industry Connectivity Consortium (IICC) is a global, nonprofit organization dedicated to creating and encouraging adoption of a unified connectivity standard to reduce the cost and variability of data exchange between IVD devices and healthcare informatics in clinical laboratories. It promotes the use of the IHE-LAW profile to define the physical connection, message definitions, and workflow definitions between instruments, middleware, and LIS systems in the laboratory. It defines the LIVD digital format for publishing mappings between LOINC and vendor defined tests.
- Decoupled Analyzer Interface System (DAIS platform) is a recent initiative of the OpenHIE Lab Information Systems Community which aims to create a community-built and supported software that will connect to clinical analyzers in the lab via ASTM, HL7 v2, or text file outputs, and convert them into FHIR objects which any FHIR-ready system can consume, providing lab results to any system that needs them (LIS, EMR or surveillance system). It also mentions the OCL tool to maintain the concept sets and link to other reference terminologies as needed.
- One of the founding blocks of any communication middleware today is HTTP so it is worth mentionning the HL7 over HTTP initiative by the HAPI project.

The ideas behind the DAIS platform are fully aligned with the planned architecture for LabBook Connect, but there is a lot of work to be done to agree upon and to develop a common FHIR-to-analyzer system. One useful first step could be to develop a minimal HTTP bridge to analyzers, that would implement the IHE-LAW transactions over HTTP, without any code conversion.

Such a component could be useful as a building block in an orchestrating middleware or as a standardized access point from a LIS. The necessary mappings between IVD vendor codes and standard or LIS specific codes will be performed by upstream components in a middleware or by the LIS itself if it connects directly to the bridge.

The reasons why IHE-LAW is probably the right standard to agree upon for this mininal HTTP bridge to analyzers are expressed on IICC home page, from which we can cite:

- [LAW] Addresses all the shortcomings of outdated laboratory connectivity standards such as CLSI LIS1-A (ASTM 1391) and CLSI LIS2 (ASTM E1394).
- LAW will be a global standard (CLSI AUTO16).
- [the LAW specification is] available for download and do not require any licensing or fees for implementation.

To the best of our knowledge there is no standard expression of IHE-LAW based on FHIR for the time being, so we propose to stick to HL7v2 messages. The HL7v2 to FHIR conversion will be performed by upstream components in the middleware. The extra work of keeping the HL7v2 layer is justified by the consensus upon IHE-LAW and the time saving and stability it provides.

For analyzers that already implement IHE-LAW the bridge will be very similar to hapi-hl7overhttp. For the others it will have to manage the conversion to and from the LAW messages and workflow according to the IVD vendor specifications.

In the following sections we propose sequence diagrams for the HTTP interactions based on the IHE-LAW profile described in the IHE PaLM TF Vol1 document (pdf from IHE) and transactions described in the IHE PaLM TF Vol2b document (pdf from IHE or IICC).

3 - IHE-LAW transactions

All the diagrams present the same actors:

- Analyzer represents the IVD device
- Bridge is the HTTP component acting as server and/or client
- Analyzer Manager is the term used in LAW and stands in our case for the upstream middleware or LIS system

AWOS is for “Analytical Work Order Steps”, see §5 of IHE PaLM TF Vol1.

3.1 - Query for AWOS [LAB-27]

See full description in §3.27 of IHE PaLM TF Vol2b.

In this transaction the bridge transmits the query from the analyzer, it needs an HTTP endpoint to contact, it will act as the client.

This transaction is used with an Analyzer working in “Query Mode”, it conveys the query and is combined with AWOS Broadcast [LAB-28], which carries the actual response to the query, as zero or more AWOS.

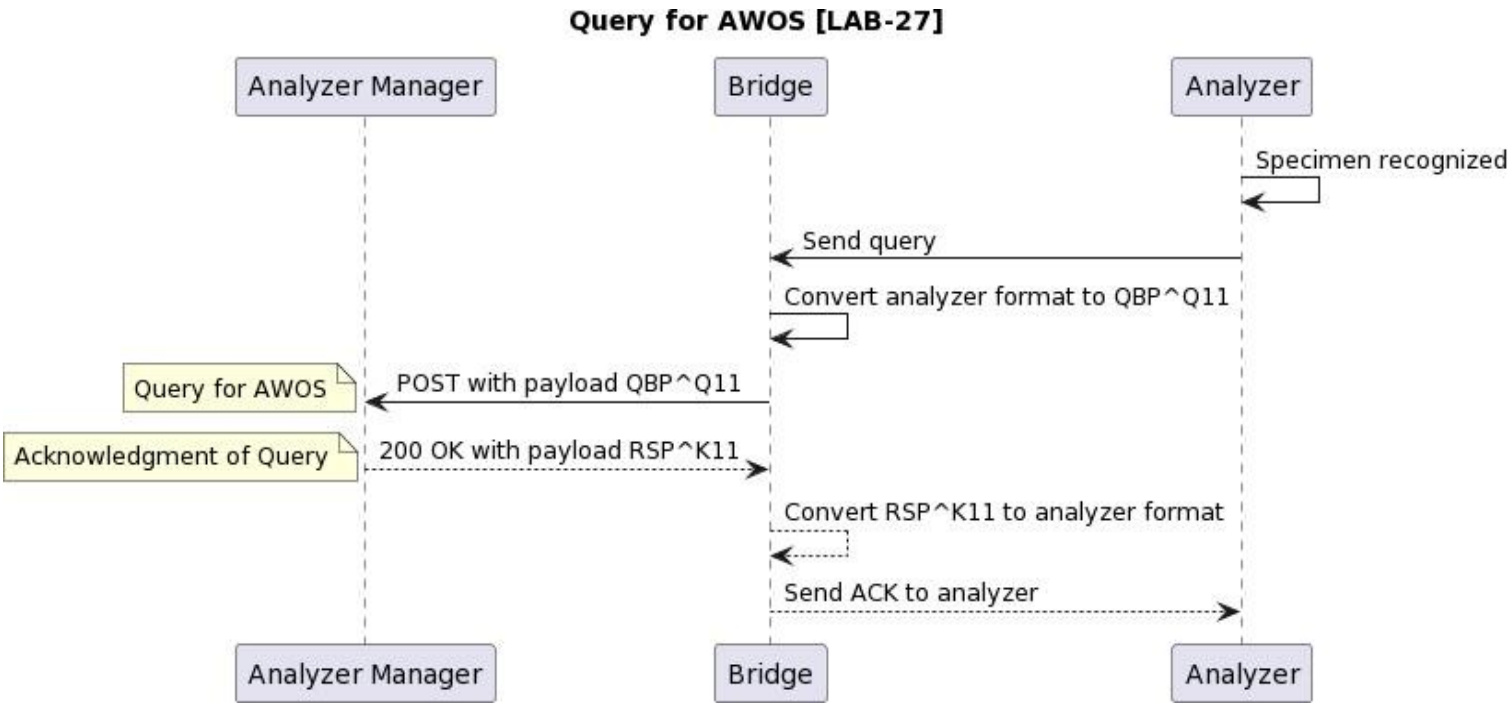

Illustration 1: Query for AWOS [LAB-27]

3.2 - AWOS Broadcast [LAB-28]

See full description in §3.28 of IHE PaLM TF Vol2b.

In this transaction the bridge receives new order and cancel order resquests from upstream, it needs to offer an HTTP endpoint, it will act as the server.

This transaction can be unsolicited if the Analyzer operates in “Broadcast Mode” or in response to a query initiated via transaction [LAB-27] by the Analyzer operating in "Query Mode".

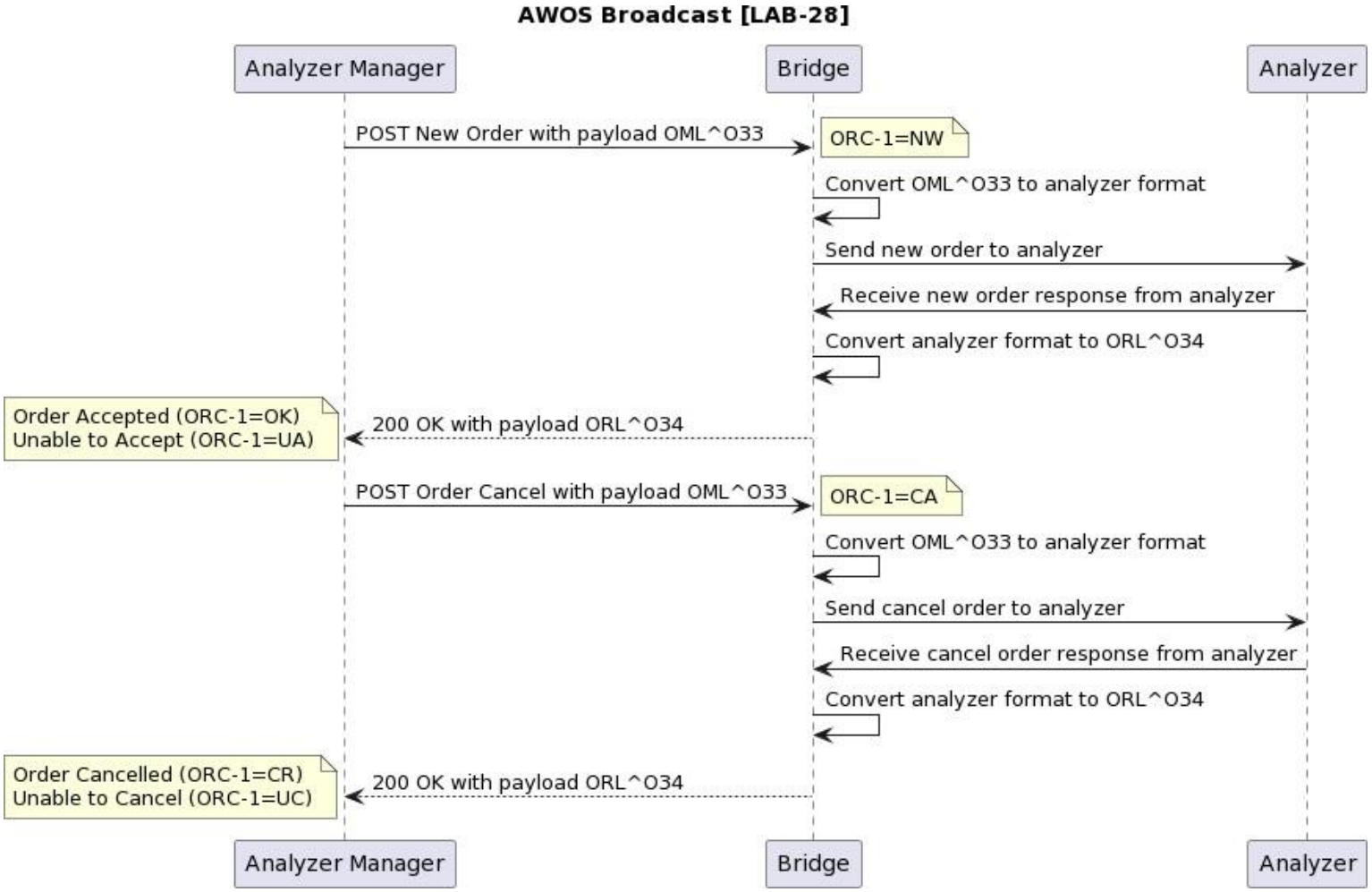

Illustration 2: AWOS Broadcast [LAB-28]

3.3 - AWOS Status Change [LAB-29]

See full description in §3.29 of IHE PaLM TF Vol2b.

In this transaction the bridge transmits the test results and AWOS status changes from the analyzer, it needs an HTTP endpoint to contact, it will act as the client.

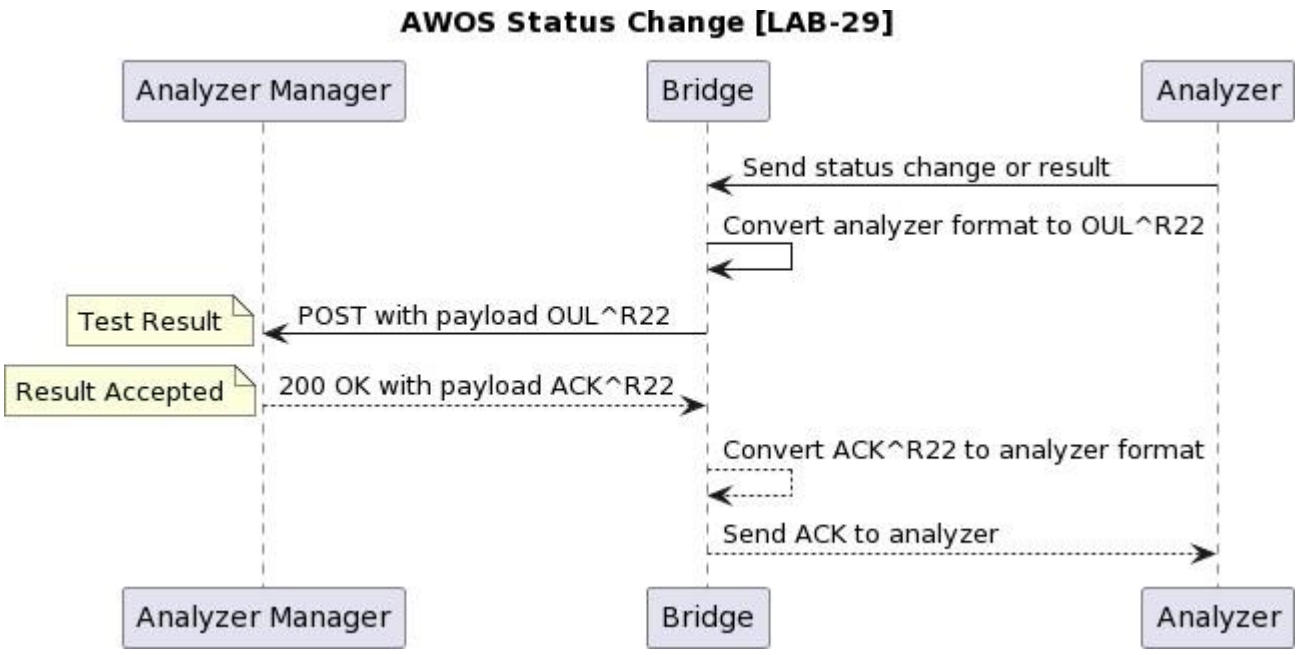

Illustration 3: AWOS Status Change [LAB-29]

3.4 - Summary

The HTTP Bridge converts specific analyzer messages to and from IHE-LAW HL7 messages, and transports them over HTTP implementing the HL7 over HTTP specification.

It can be implemented as one or several instances of a program acting as an HTTP server with a lab28 endpoint accepting POST requests and as an HTTP client placing POST requests to lab27 and lab29 upstream endpoints.

A bridge connected to an analyzer operating in “Query Mode” will use the 3 endpoints.

A bridge connected to an analyzer operating in “Broadcast Mode” will use the lab28 and lab29 endpoints.

4 – Questions

4.1 Does the bridge need to expose raw messages to and from analyzer for auditing/debug ?

If so simple payloads may be converted to MIME structures to carry the HL7 and the raw messages. An alternative is to log raw messages, and to implement IHE ATNA Profile transactions to centralize the logs in the future.

4.2 Is it possible to adapt any specific IVD device workflow to IHE-LAW ?

The HTTP Bridge is based on the assumption that it is possible to adapt any specific IVD device to the synchronous workflow of IHE-LAW. In particular, when a response is needed from the device, it has to come within a reasonnable delay.

4.3 The bridge should be able to expose an https endpoint

Either via auto-signed certificates or an imported certificate chain.

4.4 Support additional information in the payload ?

IHE-LAW messages may be insufficient in corner cases. Should we support to carry additional information to the IHE-LAW messages ? Using MIME structures instead of bare HL7v2 messages ?