OpenLegacy
OPEN INNOVATION

THE HARTFORD

# Engagement Agreement

## Version History

| Date | Version | Description | Author |
|---|---|---|---|
| **09/03/2020** | 1.0 | Initial Use Case Definitions – POC | **Gus Delgado** |
| **09/29/2020** | 1.1 | Updated details on Use Case definitions | **Gus Delgado** |
| **10/1/2020** | 1.2 | Updated Additional Success Criteria | **Gus Delgado** |

## Contents

## Engagement Overview

The Hartford is currently researching integration platforms to expose existing services from their mainframe system, these systems run is transactions based on both IMS COBOL and CICS COBOL.

The Hartford needs to conduct a proof of concept for product evaluation purposes.

The evaluation will include the performance in request as well as development speed, in addition The Hartford also needs to evaluate the ease of use of the platform, the connections to their legacy instances and the integration to RESTful APIs.

**Current concerns and pain points:**

- Dependency in multiple MQ layers
- JSON to copybook translation
- Response consolidation in one single API service from multiple requests to mainframe, i.e. multiple COBOL programs
- Response of API needs to meet SLA post batch process
- Performance of APIs overall
- Repeatable automated process for online use cases and batch use cases (API Factor/CLI)
- Templating of NFRs such as regulations, security and such

## The OpenLegacy Approach and Impact

- **Simplification:**

  Deconstructing complex workflows into simple, granular digital microservices components will provide much greater level of agility, flexibility and speed in delivery and maintenance.

  The levels of granularity can vary from SDK orchestration to microservices choreography, depending on use case.  The Openlegacy architecture supports modern architectural principles in addition to supporting  CI/CD and DevOps methodologies.

- **Speed:**

  Creating services manually can be time consuming and prone to errors and vulnerabilities. Automating as much of that process as possible will provide tremendous speed benefits as well as consistency of quality, especially if combined with automated testing, deployment etc.

- **API Factory/CLI:**

  Automating the services process in a API factory approach delivers CLI functionality so entry-level developers can generate consistent APIs, supporting corporate standards without the need to edit code manually when making changes to the existing legacy environment.  This process provides accelerates to the delivery of innovative digital services in just a few  days.

## Engagement Objectives & Success Criteria

| Category | Description | Success Criteria |
|---|---|---|
| Objective 1: Post Online CICS Outbound GeoCoder | CICS client needs to post a RESTful service to an ERM Geocoder REST service provider and receive a response for Mainframe to continue process. | • For a CICS Transaction to trigger a 3rd party API through API Caller after a CICS online process is completed.<br>• Demonstrate API Specification to COBOL copybook translation<br>• API request/response to meet SLA requirements<br>• Demonstrate the ease of use of the tool to generate COBOL<br>• Successfully translate at runtime: COBOL to JSON and JSON to COBOL |
| Objective 2: Post batch IMS Outbound GeoCoder | IMS client needs to post a RESTful service to an ERM Geocoder REST service provider and receive a response for Mainframe to continue process. | • For an IMS Transaction to trigger a 3rd party API through API Caller after a batch process is completed.<br>• Demonstrate API Specification to COBOL copybook translation<br>• API request/response to meet SLA requirements<br>• Demonstrate the ease of use of the tool to generate COBOL |

| | | |
|---|---|---|
| | | • Successfully translate at runtime: COBOL to JSON and JSON to COBOL |
| Objective 3: Online CICS Inbound Policy Information | To implement a RESTful service that will call a Policy Information CICS COBOL program in the mainframe and receive a response back. | • Expose a CICS Transaction using OpenLegacy's IDE.<br>• Demonstrate the ease of use of the tool to generate SDKs and APIs<br>• Demonstrate the orchestration capabilities to present one single API contract from multiple COBOL programs<br>• Successfully expose the COBOL asset as an API in a Swagger document<br>• Successfully map COBOL data types to java data types within each field. |
| Objective 4: Online IMS Inbound Policy Information | To implement a RESTful service that will call a Policy Information IMS COBOL program in the mainframe and receive a response back. | • Expose a IMS Transaction using OpenLegacy's IDE.<br>• Demonstrate the ease of use of the tool to generate SDKs and APIs<br>• Demonstrate the orchestration capabilities to present one single API contract from multiple COBOL programs<br>• Successfully expose the COBOL asset as an API in a Swagger document |

| | | |
|---|---|---|
| | | • Successfully map COBOL data types to java data types within each field. |

## Additional Success Criteria

- Successfully install IDE within The Hartford development environment
- Successfully connect to IMS and CICS transactions
- Validate performance of the API calls to the mainframe (inbound/outbound) through a batch performance test (Load testing)
- Incorporate into The Hartford DevOps pipeline by:
    - Integration to the Jenkins CI/CD process
    - Generating Junit Test Cases
    - Containerize the solution within OpenShift
- Security Engineering team to scan generated code to identify vulnerabilities using Checkmarx, or any other scans that have already been used for OpenLegacy platform.

## Business Value Assessment – Operational Impact- Success Criteria

- Digital Driven Integration – Business Request to Production
- How many hours (on AVG) per API
- How many months (on AVG) per API

- Labor Hours Cost / API

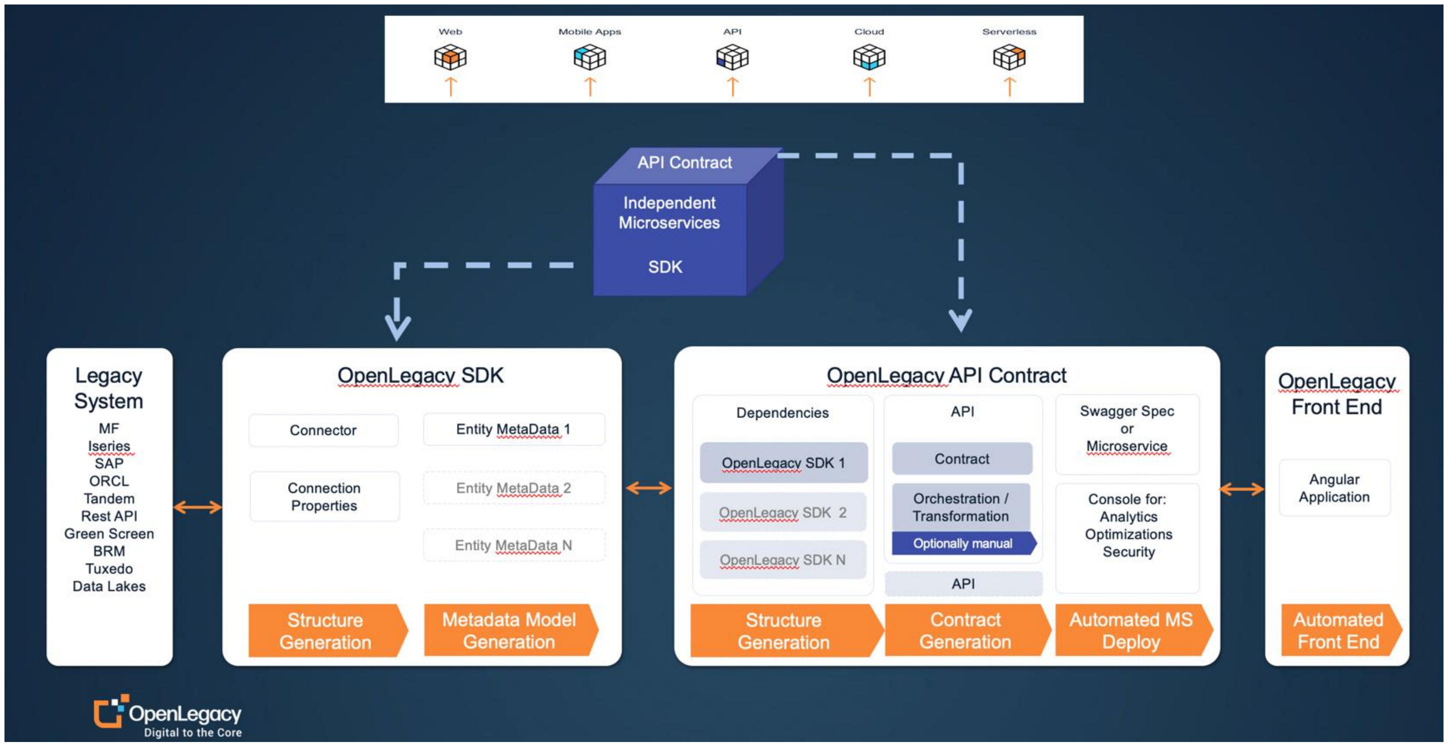
## OpenLegacy Architecture

OpenLegacy provides a collection of connectors that allow communicating and working with the backend using a simple Java SDK.

This Java layer is comprised of a combination of connectors and Java models (also called entities), representing backend resources relevant for the operation. For example, an entity can describe the input and output parameters of a legacy asset, so that using a connector to this legacy system such as mainframe COBOL, we can invoke the program, knowing what to send and what we expect to receive, in a single line of Java Object.

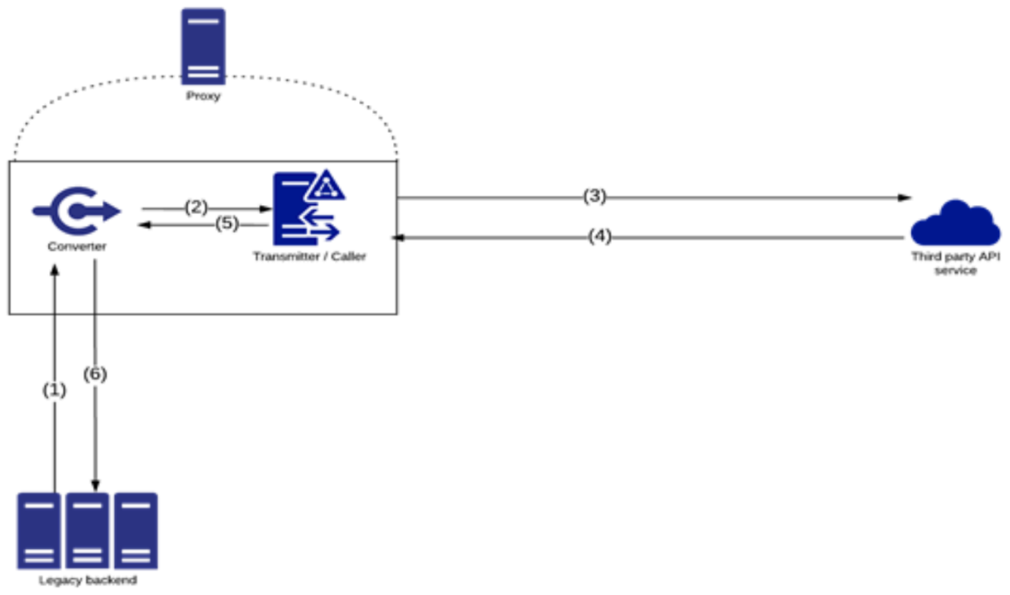
## Legacy for Inbound APIs

A typical use case is enabled with the architecture described below. OpenLegacy, at design-time provides automatic tools to analyze and generate entities automatically from the legacy resource, i.e. COPYBOOK files whether it is legacy source code, a trail file describing terminal screens, or even a direct connection to the database. At run-time OpenLegacy delivers a production ready microservice that provides endless deployment capabilities.



## Legacy for Outbound APIs

The API Caller is a component of the OpenLegacy platform that enables calling RESTful APIs from the legacy application. At design time the API Caller consumes the API specification of the service that needs to be called and generates a legacy asset representation of the same, i.e. Copybook. The API Caller also generates an end-point that will be used by the legacy system to make calls to the API it consumed. The legacy asset generated can then be downloaded. At run time the legacy program can call the API using the generated program and use the API Caller as a proxy for RESTful requests/responses. The API Caller converts the legacy buffer sent to a REST request and calls the API end-point,  the response is received and is converted to a legacy representation and sends it back to the legacy program.



## Participants

### Customer Participants

| Resource | Role | Phone | Email |
|---|---|---|---|
| **Steven Ruscik** | | | steven.ruscik@thehartford.com |
| **Chandra Bhat** | | | chandra.bhat@thehartford.com |
| **Akila Rajasekaran** | | | akila.rajasekaran@thehartford.com |

*Mandatory Role

### OpenLegacy Participants

| Resource | Role | Email |
|---|---|---|
| **Shane Kent** | General Manager – NA | shane.k@openlegacy.com |
| **Gus Delgado** | Director of Solutions Engineering | gus.d@openlegacy.com |

## Prerequisites

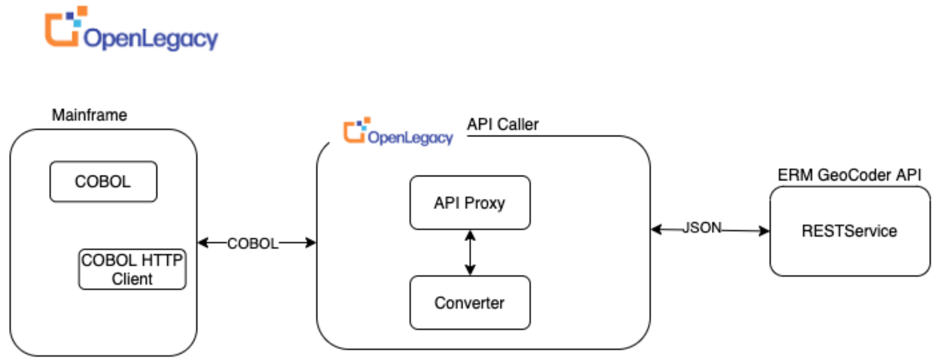
| Prerequisite | Description |
|---|---|
| **Engagement Document** | Engagement document to be reviewed by The Hartford group and Openlegacy to agree on Use Case and Success Criteria |
| **OpenLegacy IDE** | OpenLegacy to deliver the OpenLegacy IDE for installation within a The Hartford developer desktop |
| **CICS Adapter Installation** | OpenLegacy to provide the CICS Adapter code to install in the lower environment mainframe |
| **API Caller Installation** | OpenLegacy to provide the API Caller module to install within a The Hartford developer desktop or server |
| **License** | OpenLegacy to provide a 30-day license to the platform for the purpose of the POC |
| **Connectivity Testing Session** | The Hartford and OpenLegacy to have a working session where we can validate installation and connectivity |
| **Resources/Assets** | The Hartford to provide a Point-Of-Contact to support OpenLegacy during the POC and the assets needed for each of the use cases, i.e. API specifications, COBOL copybooks |
| **Environment Access** | The Hartford to provide OpenLegacy with access to a resource that has the necessary access to the development environment. |

## Use Case Definition(s)

### Use Case 1

| | | | |
|---|---|---|---|
| **Use Case ID:** | **UC1** | | |
| **Use Case Name:** | **Use Case 1:** Post batch CICS Outbound GeoCoder | | |
| **Created By:** | **Gus Delgado** | **Last Updated By:** | |
| **Date Created:** | **09/03/2020** | **Date Last Updated:** | |
| **Description:** | CICS client needs to post a RESTful service to an ERM Geocoder REST service provider and receive a response for Mainframe to continue process. | | |

**Suggested State:**



**OpenLegacy API Caller Generates at Design Time**

-COBOL HTTP Client
-COBOL copybook representation of REST API specification

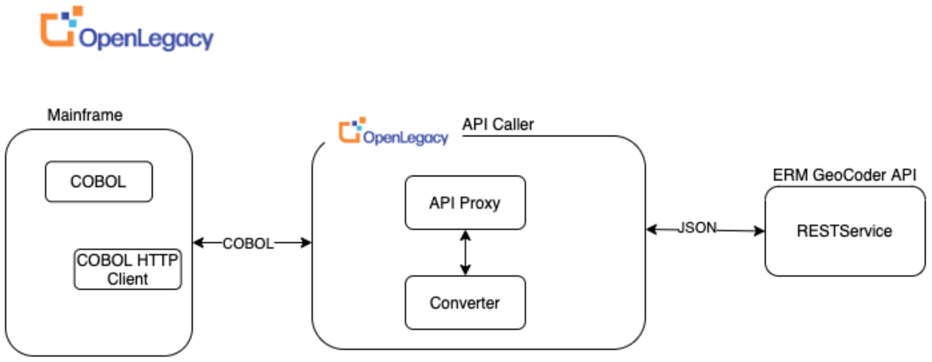
**OpenLegacy API Caller at Run Time**

-Captures HTTP Client Call
-Converts COBOL to JSON
-Calls 3rd Party API
-Converts JSON response to COBOL
-Sends Response back to Mainframe

## Use Case 2

| | | | |
|---|---|---|---|
| **Use Case ID:** | **UC2** | | |
| **Use Case Name:** | **Use Case 2:** Post batch IMS Outbound GeoCoder | | |
| **Created By:** | **Gus Delgado** | **Last Updated By:** | |
| **Date Created:** | **09/03/2020** | **Date Last Updated:** | |
| **Description:** | IMS client needs to post a RESTful service to an ERM Geocoder REST service provider and receive a response for Mainframe to continue process. | | |

**Suggested State:**



**OpenLegacy API Caller Generates at Design Time**

-COBOL HTTP Client
-COBOL copybook representation of REST API specification

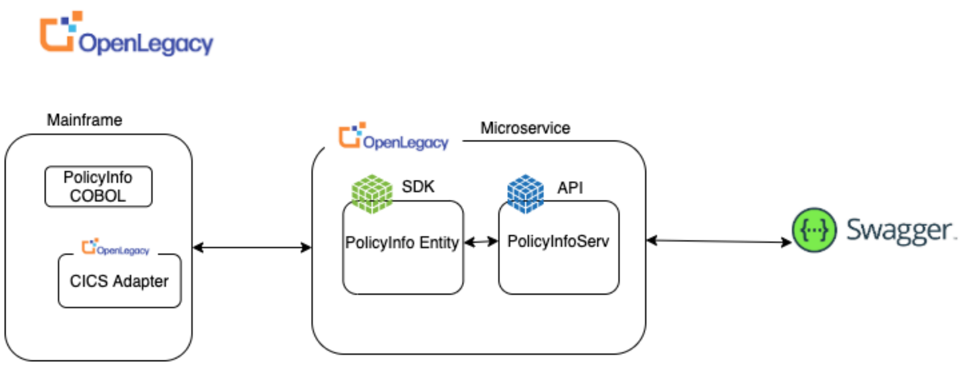
**OpenLegacy API Caller at Run Time**

-Captures HTTP Client Call
-Converts COBOL to JSON
-Calls 3rd Party API
-Converts JSON response to COBOL
-Sends Response back to Mainframe

## Use Case 3

| | | | |
|---|---|---|---|
| **Use Case ID:** | **UC3** | | |
| **Use Case Name:** | **Use Case 3:** Online CICS Inbound Policy Information | | |
| **Created By:** | **Gus Delgado** | **Last Updated By:** | |
| **Date Created:** | **09/03/2020** | **Date Last Updated:** | |
| **Description:** | To implement a RESTful service that will call a Policy Information CICS COBOL program in the mainframe and receive a response back. | | |

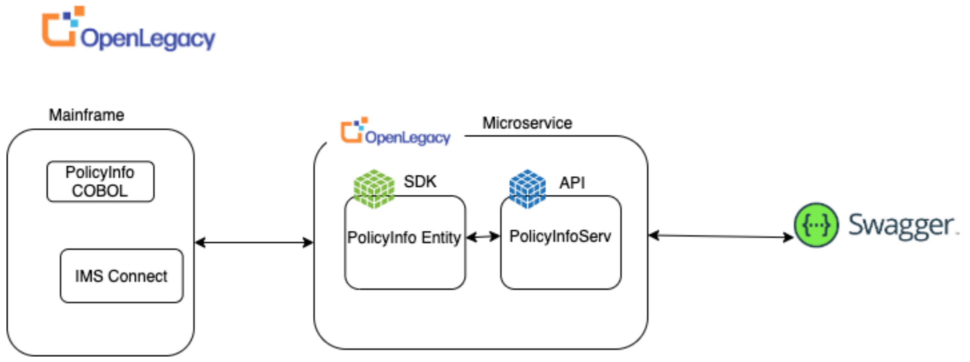
**Suggested State:**



## Use Case 4

| | | | |
|---|---|---|---|
| **Use Case ID:** | **UC4** | | |
| **Use Case Name:** | **Use Case 4:** Online IMS Inbound Policy Information | | |
| **Created By:** | **Gus Delgado** | **Last Updated By:** | |
| **Date Created:** | **09/03/2020** | **Date Last Updated:** | |
| **Description:** | To implement a RESTful service that will call a Policy Information IMS COBOL program in the mainframe and receive a response back. | | |

**Suggested State:**



## Engagement Acceptance

*We look forward to working with The Hartford and supporting your initiative. We are confident that we can face future challenges and be prepared to provide an effective solution.*

*Engagement Period of Performance*

*Start Date: __ /__/____ End Date: __ /__ / ____*

*Engagement Assumptions*

*During the engagement process, all assigned participants from both organizations will participate in a "two times per week" – Standup Meeting Huddle – 45 minute meeting. The objective of the Standup Meeting is to make sure that all stakeholders are aware of the status, aware of issues and aligned on next steps throughout the entire engagement cycle.*

*This engagement comes with a 30-day license. Following the successful completion of the engagement (period of performance), the deployed and installed environment remains available for the customer to continue to evaluate the solution in a non-Production manner. Upon the license expiration date, the deployed and installed solution will become non-functional.*

*Customer is fully aware that OpenLegacy is providing the services and associated trial license usage rights to the customer for the term of the above reference period of performance, for a $0.00 ( Zero ) dollar cost. It is mutually understood that the customer is committing to have a Digital Transformation budget or API budget allocated within 6 months of the above referenced period of performance start date at the range of $XXX per month/year.*

## *POC Sandbox Test Environment options:*

*The Proof of Concept is usually conducted at the client site using The Hartford sandbox to resemble production use as closely as possible. This provides a realistic environment to prove that a design concept has the potential to be acceptable.*

*As an alternative, OpenLegacy can provide a POC Sandbox as the testing environment to support the Proof of Concept. This should be decided and agreed upon prior to the start of the POC.*

*In the case OpenLegacy has successfully delivered on the above defined Engagement and met the above mutually agreed Success Criteria, the customer commits to enter a commercial license negotiation with OpenLegacy.*

*Engagement Acceptance*

*Customer:*

*Executive Sponsor:* _______________________________________ *Date:* _____________

*Budget Owner:* __________________________________________ *Date:* _____________

*OpenLegacy:*

*Engagement Manager:* ____________________________________ *Date:* ______________

*Customer Success Officer:* __________________________________ *Date:* ______________

*Our representative Shane Kent [General Manager NA], can communicate through the telephone number (215) 776-0277 and through his email shane.k@openlegacy.com.*

*Soon we will contact you to organize a closing meeting of this proposal.*

*Thanks for your consideration*

## Post POC Activities

| Activity | Date |
|---|---|
| **POC Executive Presentation** | |
| **POC Read Out Results** | |

## Appendix A