**Immunizations Flow**

**HTTPInput node:**

IIB will receive the input from APIC on (/phr/v1/medications/details). By default, HTTPInput node will parse BLOB message. (Irrespective of XML or JSON)

We have enabled event monitoring for Transaction Start, Failure terminal and Catch terminal.

**UAF Authorization Sub flow:**

To check the user authorization based on the expected format. (IdSource, IdValue etc.,)

**Compute node (PrepareCallForUAF):**

Compute mode: Message

If the user is authorized, the main flow continues. Otherwise it will throw back an error. If we get any error in the sub flow compute node (PrepareaCallForUAF), the fail terminal will trigger to FailError compute node which is in the main flow.

Then the FailError compute node will throw an error based on the request.

This compute node is for:

1. To find the name space of the root element of input message.
2. To call User-defined configurable service to retrieve UAF webservice endpoint.
3. For calling the Stored Procedure from the common library to prepare the service auth.object
4. To call UDCS only if shared variable is null.
5. This will throw user exception values if no UAF Service Endpoint found.
6. To Set webservice endpoint

**HTTPRequest node (EDBUserAuthorizationService):**

The webservice URL for this node is: <http://localhost:7800/technical/v2/identitymanagement/userauthorization>

We have enabled event monitoring for event source In terminal, Out terminal, Failure terminal and Error terminal.

If the request is success, it will reach the RestoreOriginalMessage node (Compute node).

**Compute node (RestoreOriginalMessage):**

This node will:

1. Find namespace of root element of UAF Service Response.
2. Check the response from UAF service and process accordingly.
3. Call the function to Restore the original message.

**Failure node (In Sub flow):**

**Main flow:**

**Compute node (PreTransform):**

Compute mode: Local Environment and Message

This node will continue after the success of the sub flow.

This node is to filter out whether the content type is xml or JSON and validate the data against schema based on the content type

Functions it will perform:

1. validateXMLData

2. validateJSONData

This will pass to RouteToLabel (RouteToOperation) based on the routing logic which will be defined in this compute node.

Routing mode: RoutetoLast

**Java Compute node:**

We have JDBC to the database to call the procedure.

**Compute node (FailError):**

Compute mode: All

Purpose:

FailError node detects whether the message is coming from catch/fault path or not. The received error/fault will be handled here and response will be prepared and return back to the service consumer.

The ErrorType has been set based on the case as show below:

|  |  |
| --- | --- |
| **Error Number (Between)** | **Set To** |
| 8000 And 8999 | 'COMMANDS ERROR' |
| 7000 AND 7999 | 'PUBLISH/SUBSCRIBE’ |
| 6000 AND 6999 | 'WEBSPHERE MQ |
| 5000 AND 5999 | 'MESSAGE PARSING |
| 3000 AND 4999 | 'MESSAGE FLOW |
| 2949 AND 2999 | 'USER GENERATED EXCEPTIONS' |
| 2701 AND 2948 | 'BROKER ERROR' |
| 2610 AND 2700 | 'MQSERIES ERROR' |
| 2599 AND 2609 | 'BROKER ERROR' |
| 2401 AND 2598 | 'ESQL ERROR' |
| 2325 AND 2400 | 'BROKER ERROR' |
| 2321 AND 2324 | 'DATABASE ERROR' |
| 2000 AND 2320 | 'BROKER ERROR' |
| 1000 AND 1999 | ‘CONFIGURATION ERROR' |
| ELSE | 'UNKNOWN /EXTERNAL SYSTEM ERROR' |