

Connect the SAP Ariba Network to Third-Party invoice status

TABLE OF CONTENTS

1.	FUNCTIONAL REQUIREMENT	3
2.	TECHNICAL DESIGN	3
3.	SYSTEM INFORMATION.....	4
4.	TEST AND OPERATIONAL ASPECTS.....	14

1. FUNCTIONAL REQUIREMENT

Standard SAP Ariba Network adapter does not pull Invoice attachments linkages from SAP Ariba Network. Due to which there a need to develop an interface in order to get Invoice cXML and its Attachments from the SAP Ariba Network.

Below are functions inputs to be considered:

1. Invoice Attachments need to be converted in Base64 string and embedded inline of Invoice cXML
2. Invoice cXML to be modified as per the Third-Party system accepted format.
3. Only single Invoice data will be sent to the Third-Party system at a time.
4. cXML to be embedded in a Third-Party system provided WSDL format.
5. Invoice cXML should be formatted in "UTF-8" text format.

2. TECHNICAL DESIGN

2.1 Technical Flow

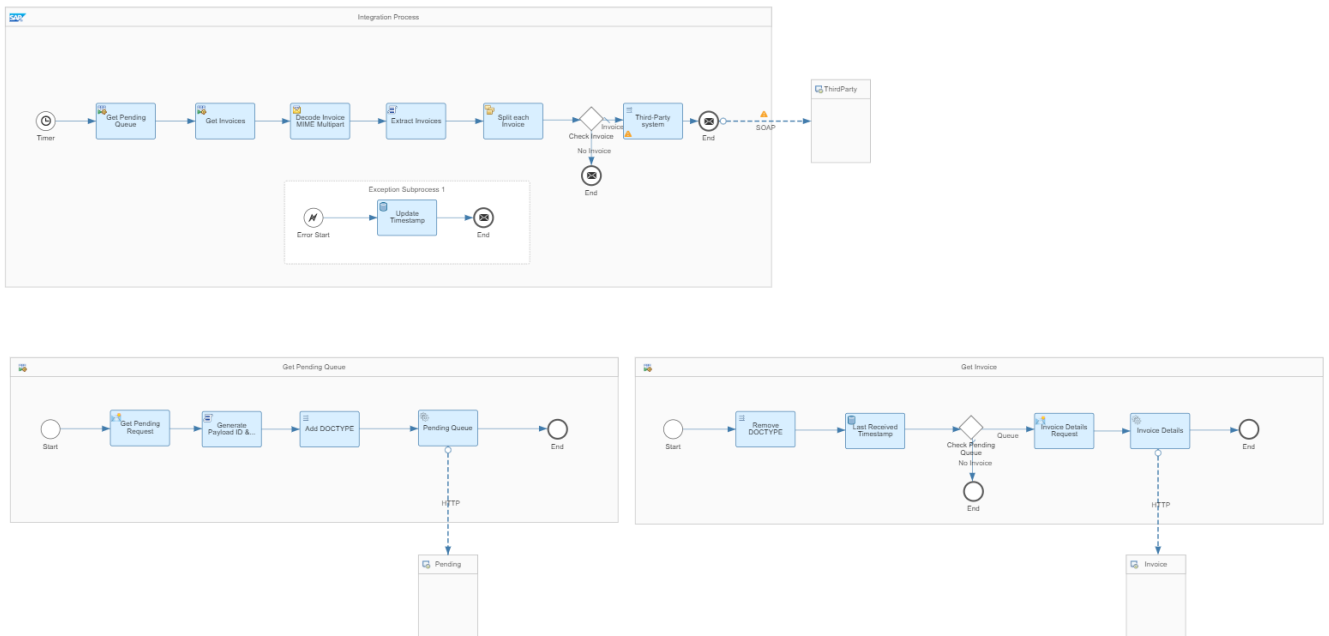
Asynchronous interface for fetching Pending Invoice data and its attachment from SAP Ariba Network and Send to the Third-Party system

2.1.1 Description

2 REST service calls will be required to receive information from the SAP Ariba Network. Below are the details:

1. Get Pending Invoice queue
2. Get Invoice data & attachment

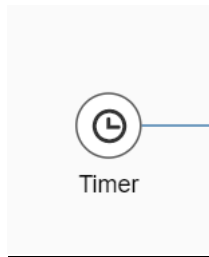
2.1.2 Graphical Overview



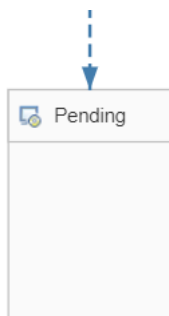
3. SYSTEM INFORMATION

3.1 System Name

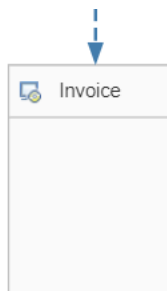
Sender System – Timer



Receiver System 1- SAP Ariba Network(Pending)



Receiver System 2- SAP Ariba Network(Invoice)



Receiver System 3- Third-Party System



3.2 Channel Details

Channel Details	Receiver Channel to get Invoice Pending queue from the SAP Ariba Network
Channel Name	Get_Pending
Protocol	HTTP/HTTPS
Authentication	None

Channel Details	Receiver Channel to get Invoice data & attachments from The SAP Ariba Network
Channel Name	Invoice_Details
Protocol	HTTP/HTTPS
Authentication	None
Channel Details	Receiver Channel to send Invoice to the Third-Party System
Channel Name	SOAP
Protocol	HTTPS
Authentication	Basic

3.3 HTTP Adapter (Pending) Receiver Tab - Connection

Section	Parameter	Value
Connection Details	Address	https://service-2.ariba.com/service/transaction/cxml.asp
	Proxy Type	Internet
	Method	POST
	Authentication	None
	Timeout (in ms)	60000

3.4 HTTP Adapter (Invoice) Receiver Tab - Connection

Section	Parameter	Value
Connection Details	Address	https://service.ariba.com/VendData.aw
	Proxy Type	Internet
	Method	POST
	Authentication	None
	Timeout (in ms)	60000


3.5 SOAP Adapter (Third-Party system) Receiver Tab - Connection


Section	Parameter	Value
Connection Details	Address	http://thirdpartysystem.com/InvoiceReceiver.asmx
	Proxy Type	On-Premise
	Location ID	
	URL to WSDL	/wsdl/InvoiceReceiver.WSDL
	Service	p1:InvoiceReceiver
	EndPoint	p1:InvoiceReceiverSoap
	Operation Name	p1:InvoiceReceiverAriba
	Authentication	Basic
	Credential Name	SAP
	Timeout (in ms)	60000
	Compress Message	False
	Allow Chunking	True
	Clean-up Request Headers	True



3.6 Externalized Parameters

Name	Value
Address	https://service-2.ariba.com/service/transaction/cxml.asp
Credentials	AN_CREDENTIALS
cXML Version	1.2.031
maxMessages	2
MessageType	InvoiceDetailRequest
Mode	test
SystemID	IQ4CLNT100
UserAgent	Ariba SN

3.7 Mapping * Refer to the documentation section for example resources.

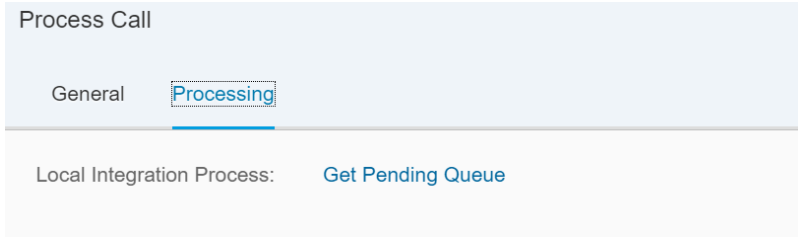
Mapping Name	Add DOCTYPE
Mapping Type	XSLT Mapping
Source XSD/WSDL	NA
Target XSD/WSDL	NA
Source Operation (Optional)	NA
Target Operation (Optional)	NA
Operation Mapping(optional)	NA
Mapping Document (Attachment)	 Add DOCTYPE.xsl

Mapping Name	Remove DOCTYPE
Mapping Type	XSLT Mapping
Source XSD/WSDL	NA
Target XSD/WSDL	NA
Source Operation (Optional)	NA
Target Operation (Optional)	NA
Operation Mapping(optional)	NA
Mapping Document (Attachment)	 Remove DOCTYPE.xsl

Mapping Name	ThirdParty
Mapping Type	XSLT Mapping
Source XSD/WSDL	NA
Target XSD/WSDL	 InvoiceReceiver.WSDL
Source Operation (Optional)	NA
Target Operation (Optional)	p1:InvoiceReceiverAriba
Operation Mapping(optional)	NA
Mapping Document (Attachment)	 ThirdParty.xsl

3.8 Integration Flow

Step 1: Local SubProcess



Step 2: Content Modifier

This content modifier is used to create various properties which will be used in SAP Ariba Network Request Payloads which helps in authentication, cXML version etc.

Name: Get Pending Request

Exchange Property:

Properties:						
Action	Name	Type	Data Type	Value	Default	
Create	Credentials	Constant		AN_CREDENTIALS		
Create	lastReceivedTimestamp	Local Variable		lastReceivedTimestamp		
Create	timestamp	Expression		\$(date.now:yyyy-MM-dd'...		
Create	cXML Version	Constant		1.2.031		
Create	SystemID	Constant		IQ4CLNT100		
Create	UserAgent	Constant		Ariba SN		
Create	maxMessages	Constant		2		
Create	MessageType	Constant		InvoiceDetailRequest		
Create	Connection Mode	Constant		test		

Message Body:

Message body in this content modifier is used to generate an Ariba Network Request payload, to get Invoice PendingnQueue. **“property.timestamp”** is the important detail in this payload which defines the timespan from what time data need to be pulled from SAP Ariba. Timestamp in SAP Ariba is used to identify if an Invoice need to be removed from Pending queue or not.

Type: Expression

Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE cXML SYSTEM "http://xml.cxml.org/schemas/cXML/${property.cXML
Version}/cXML.dtd">
<cXML payloadID="${property.payloadID}" timestamp="${property.timestamp}" xml:lang="en-US"
version="${property.cXML Version}">
  <Header>
    <From>
      <Credential domain="NetworkID">
        <Identity></Identity>
      </Credential>
      <Credential domain="SystemID">
        <Identity>${property.SystemID}</Identity>
      </Credential>
    </From>
    <To>
      <Credential domain="NetworkID">
        <Identity>AN0100000001</Identity>
      </Credential>
    </To>
    <Sender>
      <Credential domain="NetworkID">
        <Identity></Identity>
        <SharedSecret></SharedSecret>
      </Credential>
      <UserAgent>${property.UserAgent}</UserAgent>
    </Sender>
  </Header>
  <Request deploymentMode="${property.Connection Mode}">
    <GetPendingRequest lastReceivedTimestamp="${property.lastReceivedTimestamp}"
maxMessages="${property.maxMessages}">
      <MessageType>${property.MessageType}</MessageType>
    </GetPendingRequest>
  </Request>
</cXML>
```

Step 3: **Groovy Script**

This groovy script is used to perform below activities:

- Generate a Unique Payload ID to be used in every SAP Ariba Network Request.
- Get SAP Ariba Network credentials from CPI Security Material and use it in SAP Ariba Network Request payloads.

See section 4.6 for script details

Groovy Script

General Processing

Name:

Step 4: XSLT Mapping

Add "DOCTYPE cXML SYSTEM" to payload. As, this is required for the SAP Ariba Network to process Request payload.

XSLT Mapping

General **Processing**

Source:

Resource: [/mapping/Add DOCTYPE.xml](#)

Output Format:

Step 5: Request Reply

This helps in connecting to the external system. Here Request-Reply helps to connect to the SAP Ariba Network through HTTP adapter.

Request Reply

General

Name:

Step 6: Local SubProcess

Process Call

General **Processing**

Local Integration Process: [Get Invoice](#)

Step 7: XSLT Mapping

Remove "DOCTYPE cXML SYSTEM" from the SAP Ariba Network payload. As, this might cause issues when using XPath expressions.

XSLT Mapping

General **Processing**

Source:

Resource: [/mapping/Remove DOCTY...](#)

Output Format:

Step 8: Write Variables

Creates a Local Variable named “lastReceivedTimestamp” to the SAP Ariba Network response timestamp which will be used for next Pending Queue call.

Write Variables ?

General Processing

Variables:

Name	Type	Data Type	Value	Global Scope
lastReceivedTimestamp	Xpath	java.lang.String	/cXML@timestamp	<input type="checkbox"/>

Step 9: Router

A router is used to identify if there are invoices waiting in Pending Queue or not. In case there are no Invoices pending invoices in Queue, complete the lflow.

Router Externalize ?

General Processing

Order	Route Name	Condition Expression	Default Route
1	Queue	/cXML/Response/GetPendingResponse/...	<input type="checkbox"/>
2	No Invoice		<input checked="" type="checkbox"/>

Step 10: Content Modifier

This content modifier is used to create Pending Message property which will be used in the SAP Ariba Network. Request payload to request Invoice data

Name: Invoice Details Request

Exchange Property:

Content Modifier Externalize ?

General Message Header Exchange Property Message Body

Properties: Add Delete

<input type="checkbox"/>	Action	Name	Type	Data Type	Value	Default
<input type="checkbox"/>	Create	Pending Message	XPath	java.lang.String	/cXML/Respo...	Select

Message Body:

Message body in this content modifier is used to generate a SAP Ariba Network Request payload to get Invoice data Using Queue number. "Pending Message" property is used to extract payloads from particular queue number from the SAP Ariba Network.

Type: Expression

Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE cXML SYSTEM "http://xml.cxml.org/schemas/cXML/${property.cXML
Version}/cXML.dtd">
<cXML payloadID="${property.payloadID}" timestamp="${property.timestamp}"
version="${property.cXML Version}" xml:lang="en-US">
  <Header>
    <From>
      <Credential domain="NetworkID">
        <Identity>${property.username}</Identity>
      </Credential>
    </From>
    <To>
      <Credential domain="NetworkID">
        <Identity>AN01000000001</Identity>
      </Credential>
    </To>
    <Sender>
      <Credential domain="NetworkID">
        <Identity>${property.username}</Identity>
        <SharedSecret>${property.password}</SharedSecret>
      </Credential>
      <UserAgent>${property.UserAgent}</UserAgent>
    </Sender>
  </Header>
  <Request deploymentMode="${property.Connection Mode}">
    <DataRequest>
      <InternalID domain="PendingMessages">${property.Pending Message}</InternalID>
    </DataRequest>
  </Request>
</cXML>
```

Step 11: Request Reply

This helps in connecting to the external system. Here Request-Reply helps to connect to the SAP Ariba Network through HTTP adapter

Request Reply	
General	
Name:	<input type="text" value="Invoice Details"/>

Step 12: MIME Multipart Decoder

This helps to decode Multipart data received from the SAP Ariba Network containing Invoices & attachments

The screenshot shows the configuration for the 'MIME Multipart Decoder' component. The 'General' tab is selected, and the 'Processing' sub-tab is active. The 'Name' field is set to 'Decode Invoice MIME Multipart'.

Step 13: Groovy Script

This groovy script is used to perform below activities:

- Extract Invoice cXML from data payload.
- Convert Attachments into Based64 String and embed them in Invoice cXML.
- Modify cXML payload as required by the Third-Party system.
- Ensure "UTF-8" encoding is followed.

See section 4.6 for script details

The screenshot shows the configuration for the 'Groovy Script' component. The 'General' tab is selected, and the 'Processing' sub-tab is active. The 'Name' field is set to 'Extract Invoices'.

Step 14: General Splitter

General Splitter is used to split each invoice before sending it to the Third-Party system.

The screenshot shows the configuration for the 'Iterating Splitter' component. The 'Processing' tab is selected. The 'Expression Type' is set to 'XPath'. The 'XPath Expression' is set to '/MultiCxmlMessage/cXML'. The 'Grouping' field is empty. The 'Streaming' checkbox is checked. The 'Parallel Processing' checkbox is unchecked. The 'Stop on Exception' checkbox is checked.

Step 15: Router

A router is used to identify if there is an actual invoice cXML in the payload or not. In case there are no Invoice in payload, complete the Iflow.

The screenshot shows the 'Router' configuration window with the 'Processing' tab selected. Under 'ERROR HANDLING', the 'Throw Exception' checkbox is unchecked. Under 'ROUTING CONDITION', there is a table with the following data:

Order	Route Name	Condition Expression	Default Route
1	Invoice		<input checked="" type="checkbox"/>
2	No Invoice	/cXML/Response/Status@code = "200"	<input type="checkbox"/>


Step 16: XSLT Mapping


This mapping is used to prepare final payload which will be sent to the Third-Party system. It transforms the cXML Into the Third-Party system and formatted as per WSDL provided.

The screenshot shows the 'XSLT Mapping' configuration window with the 'Processing' tab selected. The configuration fields are as follows:

- Source: Integration Flow (dropdown)
- Resource: /mapping/ThirdParty.xml (with a 'Select' button)
- Output Format: String (dropdown)

4.6 Scripts *Refer to the documentation section for example resources.


Script Name	Generate Payload ID & Get Credentials
Script Type	groovy
Short Description	Generate Payload ID & Get Credentials
Code (Attach the code)	 Credentials.groovy

Script Name	Extract Invoices
Script Type	groovy
Short Description	Extract Invoices
Code (Attach the code)	 Extract Invoices.groovy

4.7 Library

Name of the External Jar	NA
	NA

4.7 WSDL / XSD

Name	Type	Attachment
InvoiceReceiver	WSDL	 InvoiceReceiver.WSDL

4.8 Additional Information

Name	Value / Attachment
NA	NA

5 TEST AND OPERATIONAL ASPECTS

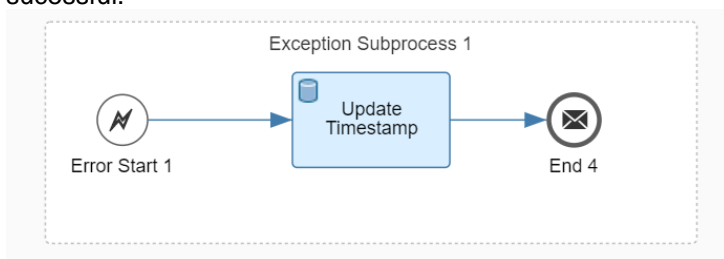
5.1 Monitoring

Monitoring is via the standard SAP CPI monitoring via the WebU.

5.2 Error Handling

Error case	Error Handling procedure
N/A	Standard CPI error handling

An exception subprocess is created which will preserve last time stamp used to retrieve data from the SAP Ariba Network in case there are failures in Inflow. This will help to keep retrying failed Invoices until successful.



5.3 Ticket Component

Component	Comment
LOD-HCI-PI-CNT-IDG	If any issues occur please log a ticket to the component described