*InvolvedPartySearch Service Design Document*

Service Design Specification

Prepared by: Praveen Kumar

Facilitating PMO: SNET

Version: 1.2

Date: August 02, 2016

Purpose: The Service Design Specification Document is to define the requirements and scope in technical terms that will drive the development of the service.

**T** **ABLE OF CONTENTS**

[I Approvals 4](#_Toc431947041)

[II Document Revision History 5](#_Toc431947042)

[III General Description 6](#_Toc431947043)

[Service Definition 6](#_Toc431947044)

[Overview 6](#_Toc431947045)

[Service Type 6](#_Toc431947046)

[Operations 6](#_Toc431947047)

[Design Approach 7](#_Toc431947048)

[Assumptions 7](#_Toc431947049)

[Scope 7](#_Toc431947050)

[IV Service Analysis 8](#_Toc431947051)

[Service Perspective 8](#_Toc431947052)

[Business Requirements and Service Analysis Specification 8](#_Toc431947053)

[Use Cases 9](#_Toc431947054)

[Class Diagrams 10](#_Toc431947055)

[Alternative Approaches Considered 10](#_Toc431947056)

[V Service Design 11](#_Toc431947057)

[Service Overview (Contract) 11](#_Toc431947058)

[Sequence Diagram – searchInvolvedParties 12](#_Toc431947059)

[Sequence Diagram – searchIndividuals 14](#_Toc431947060)

[Sequence Diagram – searchOrganizations 17](#_Toc431947061)

[Sequence Diagram – searchInvolvedPartiesBySafeBox 18](#_Toc431947062)

[Sequence Diagram – searchInvolvedPartiesByANI 20](#_Toc431947063)

[Service Composition 21](#_Toc431947064)

[Service System Context Diagram 21](#_Toc431947065)

[Service Model 21](#_Toc431947066)

[Service Component Design 21](#_Toc431947067)

[There are four WMB project artifacts involved in this framework. 22](#_Toc431947068)

[WMB Implementation Details 23](#_Toc431947069)

[B. searchIndividuals\_Request\_Response.subflow: 25](#_Toc431947070)

[B. searchInvolvedParties\_Request\_Response.subflow: 26](#_Toc431947071)

[B. searchOrganizations\_Request\_Response.subflow: 28](#_Toc431947072)

[B. searchInvolvedPartiesBySafeBox\_Request\_Response.subflow: 29](#_Toc431947073)

[B. searchInvolvedPartiesByANI\_Request\_Response.subflow: 32](#_Toc431947074)

[Service Specification Design 33](#_Toc431947075)

[WSDL Definition 34](#_Toc431947076)

[Service Message Design 35](#_Toc431947077)

[Service Data Design 38](#_Toc431947078)

[Layout 39](#_Toc431947079)

[WSDL 40](#_Toc431947080)

[Service Binding or Protocol 40](#_Toc431947081)

[Service Data Mapping 40](#_Toc431947082)

[Request and Response Mapping 40](#_Toc431947083)

[Legacy System Data Definition 40](#_Toc431947084)

[VI Unit Testing Strategy 41](#_Toc431947085)

[Definitions of Processing Cycles 41](#_Toc431947086)

[Interface Dependencies 41](#_Toc431947087)

[Testing Tool(s) 41](#_Toc431947088)

[SOAP UI 41](#_Toc431947089)

[Environmental Considerations 41](#_Toc431947090)

[Test Considerations 41](#_Toc431947091)

[VII Appendix 42](#_Toc431947092)

[SNET Error Conditions Summary 42](#_Toc431947093)

[Source System Error Conditions Summary 42](#_Toc431947094)

[Error Info Model 43](#_Toc431947095)

[Supporting Documentation 43](#_Toc431947096)

# 

# I Approvals

|  |  |
| --- | --- |
| **Service Name:** | **InvolvedPartySearch** |
| I have reviewed the information contained in the Service Specification Document version 1.0, and agree to the baseline commitments specified herein. If approval is via e-mail, type Name and Title of person, Signature on File above signature and date of e-mailed approval. Place the e-mail approval in the project repository. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| Enterprise Architect |  | Signature |  | Date |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| Integration Architect |  | Signature |  | Date |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| Integration Architect |  | Signature |  | Date |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Additional Stakeholders** (when necessary as per Governance policy) | | | | |
|  |  |  |  |  |
| Program Architect |  | Signature |  | Date |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| Name, Title |  | Signature |  | Date |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| Name, Title |  | Signature |  | Date |

# II Document Revision History

|  |  |  |
| --- | --- | --- |
| **No.** | **Date** | Reason |
| 1.0 | 05/06/2010 | Draft Version |
| 1.1 | 09/30/2015 | As part of 16.1 AML Consumer Lending project, made the following changes:   1. Added mapping for new element (**MaxReturn**) in the request for searchIndividuals and searchOrganizations operations.   2. Added mapping for CIN (**ClientIDNumber** and **ClientIDType**) in the response for searchIndividuals and eearchOrganizations operations.  These are optional elements only so there is no WSDL change; only mapping changes.  A new searchIndividuals search option was added: Last Name + SSN + DOB + MaxReturn.  A new searchOrganizations search option was added: Name + TIN + MaxReturn  These searches are called “exact match” searches because they will return only one party. If more than one party matches the search criteria, CIS will return a message stating there were multiple matching parties but will NOT return the multiple parties.  MaxReturn is used by CIS to control the maximum parties in the response. For the new search options, it will be set to “1”.  These search options are available in both the CIS online api and the CIS batch api. |
| 1.2 | 08/02/2016 | As part of DNC project 16.3 release, enhanced searchIndividuals to include search by phone number, updated the mapping sheet accordingly. |

# III General Description

## Service Definition

The purpose of the InvolvedPartySearch function is to enable the process of searching for both personal and business customers including prospects and viewing client search results.  This search function is client-focused meaning that all search types return client records as a result.  The search, results, and accessing the client profile process support both personal and business consumers and includes prospects as well as existing customers

## Overview

**InvolvedPartySearch**

InvolvedPartySearch is a service, which offers atomic operations to retrieve the personal and business client profile

The atomic operation provided by this service is

* searchIndividuals
* searchInvolvedParties
* searchOrganizations
* searchInvolvedPartiesBySafeBox
* searchInvolvedPartiesByANI

## Service Type

Select the type of service that this specification represents

Composite Service (composite, course grained, business services)

Atomic Service (Fine grained, structural, enablement of backend systems)

Integration Service Realization Approach (the design approach for implementing service realization

BEPL Process (WPS)

Mediations (Transactional, typically WESB implementations)

Utility Service (security, monitoring, logging, etc)

Data Service

## Operations

|  |  |
| --- | --- |
|  | **Operation** |
| 1 | searchIndividuals |
| 2 | searchInvolvedParties |
| 3 | searchOrganizations |
| 4 | searchInvolvedPartiesBySafeBox |
| 5 | searchInvolvedPartiesByANI |

## Design Approach

The InvolvedPartySearch service is a service with atomic operations and the service realization is implemented as a mediation service that leverages the WESB to perform the desired business functionality.

## Assumptions

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Assumption** | **Risk Factor** | **Comments** |
| 1 | Wildcard characters will be replaced with a % prior to calling this service | Medium |  |

## Scope

* Provides the ability to retrieve the individual information based on Name/NameAddress/Address/NameTin/ Last Name+TIN+DOB+MaxReturn by calling the EC service CIS Search Party.
* Provides the ability to retrieve the Party Information based on Account by calling EC service CIS Search Contract.
* Provides the ability to retrieve organization information based on Name/NameAddress/Address/NameTin by calling the EC service CIS Search Org.
* Provides the ability to retrieve the party information for individual/organization based on TIN/CIN/Telephone by calling the EC service CIS Search Party. Name can be a legal name or the alias name for both Organization and Individual
* Provides the ability to search secondary signers from Safebox.
* Provides ability to search party information based on ANI (Automated Number Identification / phone number from eIVR).

# IV Service Analysis

## Service Perspective

New Service (No service that closely matches the requirements of this service)

Existing Service(s)

From the above requirements it was identified that there is no service that closely matches the needs for this Service, so a new service Utility was added.

## Business Requirements and Service Analysis Specification

Business Requirements which are met by the Following use cases are implemented by this service.

**Brief Description of Functionality**

The InvolvedPartySearch service can be used to retrieve the personal and business client profile. It has got the following atomic operations:

1. searchIndividuals – This is an atomic operation that calls EC’s service CISSearchParty which is used to retrieve the involved party based on the following search criteria
   * Name (Legal or Alias)
   * NameAddress
   * Address
   * NameTIN
   * Last Name (full or wildcard), TIN, DOB, MaxReturn
2. searchInvolvedParties– This is an atomic operation that calls EC’s service CISSearchParty which is used to retrieve the involved party for the individual and business client profile search based on the following search criteria
   * TIN
   * CIN
   * Telephone
   * IPNumber

This atomic operation also is used to retrieve the parties involved in arrangement based on the following input criteria and it calls EC’s CISSearchContract operation

* + ContractNumber(AccountNumber)

1. searchOrganizations – This is an atomic operation that calls EC’s service CISSearchParty which is used to retrieve the business client profile based on the following search criteria
   * Name (Legal or Alias)
   * NameAddress
   * Address
   * Name, TIN, MaxReturn
2. searchInvolvedPartiesBySafeBox – This is a composite operation that calls EC’s service CISSearchContract which is used to retrieve the involved party for the individual and business client profile search based on the account number which is concatenation of BoxNumber and Center Number provided in the input. It also retrieves the Secondary Parties involved in Arrangement from RetrieveSecondarySigner operation from SafeBoxInquiryIntg Service.
3. searchInvolvedPartiesByANI – This is an atomic operation, calls CIS interface CISANILookup to perform search/lookup based on ANI (phone number).

## Use Cases

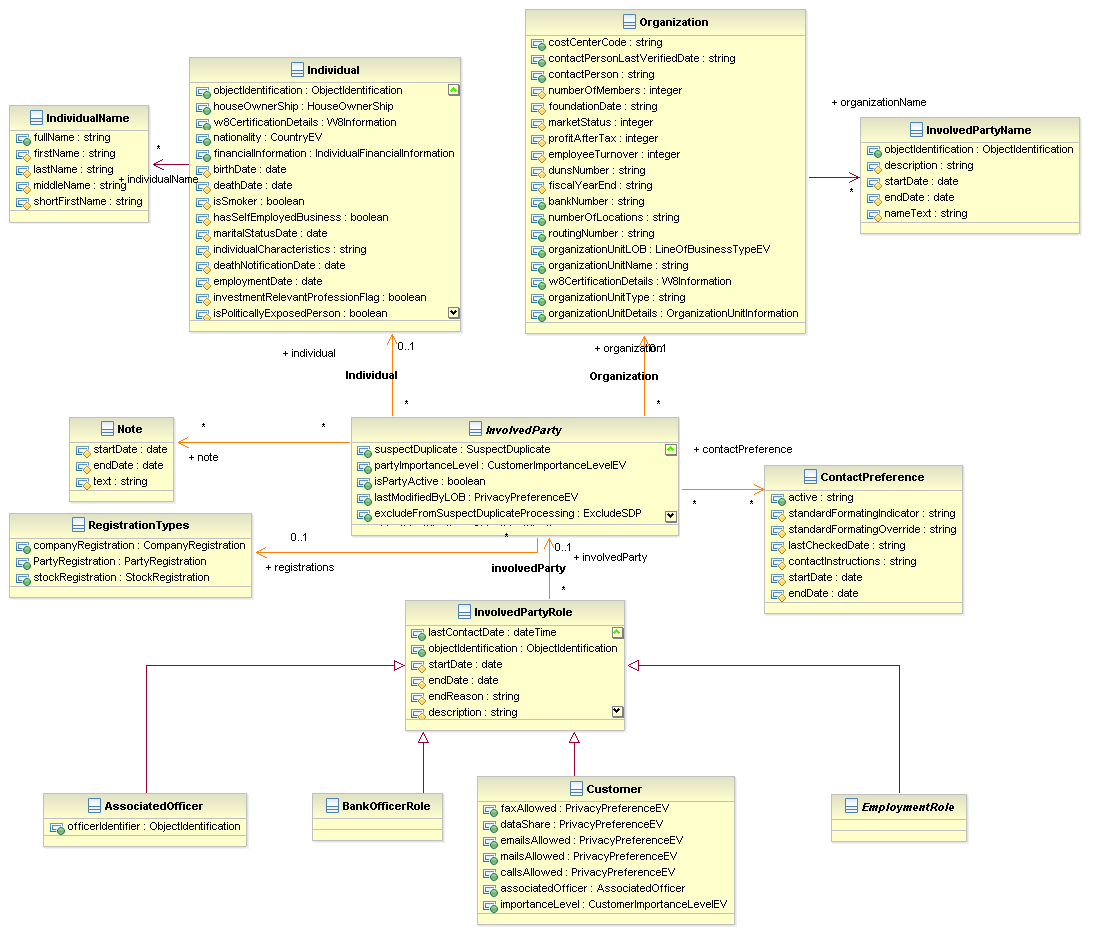
The various scenarios are summarized in the table below. Individual Use Case Details follow.

|  |  |  |  |
| --- | --- | --- | --- |
| **UC #** | **UC Name** | **Requirements Reference** | **Screenshot Figure #** |
|  | **Use Case Alternate Flows** |  |  |
| A1 | Search by Account |  |  |
| A2 | Search by Account And Safebox Number |  |  |
| A3 | Search by TIN |  |  |
| A4 | Search by Address |  |  |
| A5 | Search by Name |  |  |
| A6 | Search by Name and TIN |  |  |
| A7 | Search by Name and Address |  |  |
| A8 | Search by CIN |  |  |
| A9 | Search by Telephone Number |  |  |
| A10 | Search by Last Name, TIN and DOB |  |  |
| A11 | Search by ANI |  |  |

## Class Diagrams

**Qualified Name: InvolvedParty: Main**

This view shows us the InvolvedParty main class diagram



## 

## Alternative Approaches Considered

NA

# V Service Design

## Service Overview (Contract)

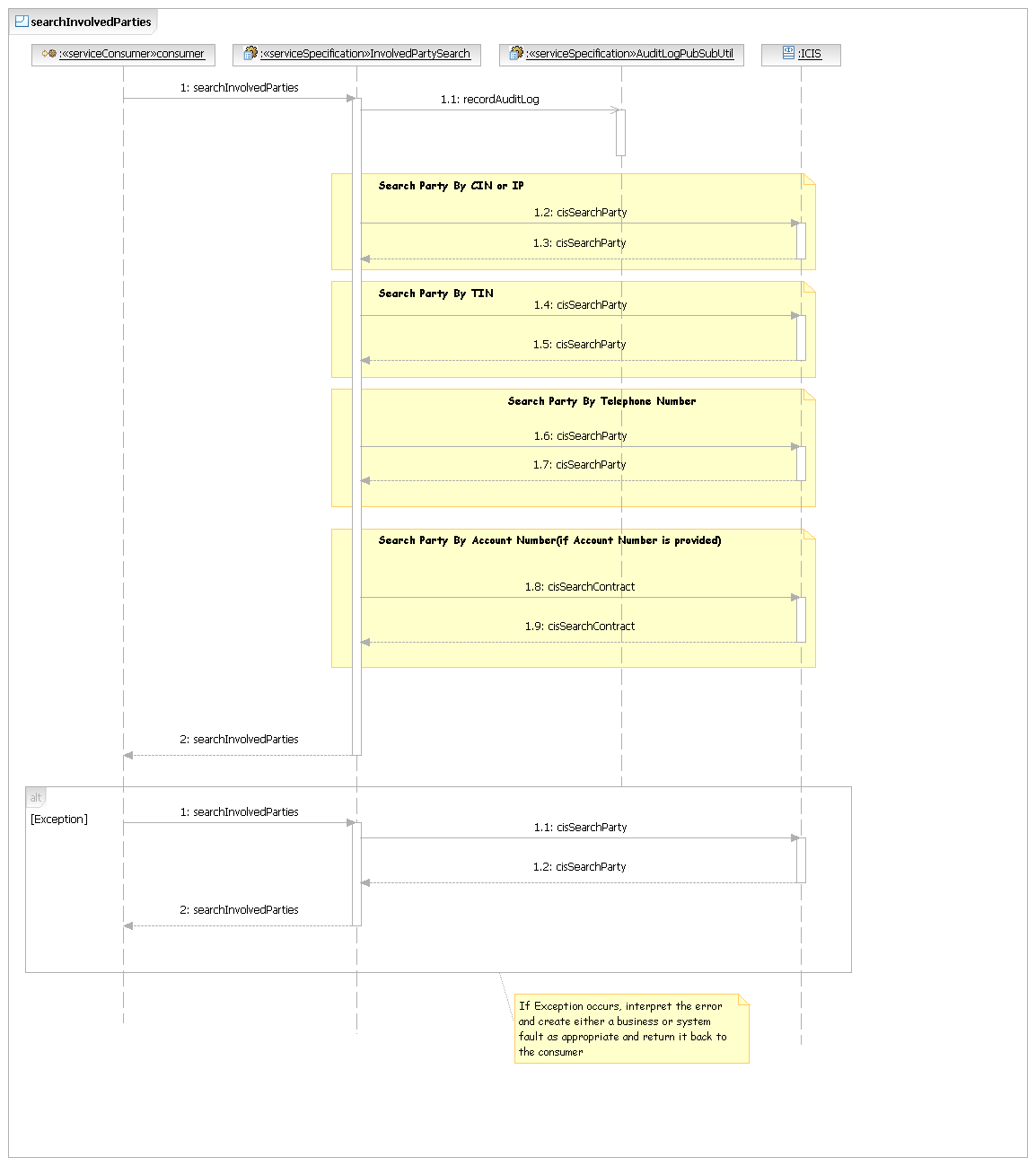
New Service (No service that closely matches the requirements of this service)

Existing Service

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** |  | | | |
| **Description** | This service encompasses the inquiry functions for an <Involved Party>.  For the search function <Involved Party’s> will be performed using the relevant populated items on the Involved Party Search Criteria Instance as search criteria. | | | |
| **Component** | Involved Party Management | | | |
| **Interface** | InvolvedPartySearch | | | |
| **Operations** | 1. searchInvolvedParties | | | |
| **Parameters** | **Name** | **Dir** | **Type** | **Default** |
| searchInvolvedPartiesRequest | In | Request Message with:  InvolvedPartySearchCriteria  ProductArrangementSearchCriteria  MessageHeader,  AuditHeader |  |
| Success:  searchInvolvedPartiesResponse  Failure: Faultinfo | Out | Response Object will include:  InvolvedParty(Individual or Organization) |  |
| **Notes** | The client application will not pass in the Tax ID Type. We will need to make the call to the EC System thrice to ensure we can capture the possibility that either a TIN (Tax ID Number) or SSN or ITIN or IPNumber was entered.  Client can search with any one of the following search criteria using the SNet operation searchInvolvedParties.   * TIN * CIN * Telephone Number * IPNumber * Account Number   The CustomerLifeCycleStatus is mapped to the PartyFilter at the EC end and this is an optional field.  <PartyFilter>Active</PartyFilter> - Fetches only Active records from operational table  <PartyFilter>Inactive</PartyFilter> - Fetches only Inactive records from operational table  <PartyFilter>All</PartyFilter> - Fetches both Active and Inactive records from operational table  <PartyFilter> </PartyFilter> (blank) - Fetches only Active records from operation table | | | |

## Sequence Diagram - searchInvolvedParties

The following sequence diagram shows the interaction between service consumer service and the backend system for the operations in InvolvedPartySearch service.



**Steps for searchInvolvedParties Operation**

1. Client calls the searchInvolvedParties operation by passing CIN/TIN/Telephone Number

1.1 The searchInvolvedParties service performs mapping from input request based on the CIN or IP to EC object model and invokes the “searchParty” EC transaction.

1.2 Get the response back from the EC

1.3 The searchInvolvedParties service performs mapping from input request based on the TIN to EC object model and invokes the “searchParty” EC transaction.

1.4 Get the response back from EC

1.5 The searchInvolvedParties service performs mapping from input request based on the Telephone Number to EC object model and invokes the “searchParty” EC transaction.

1.6 Get the response back from EC

1.6 The searchInvolvedParties service performs mapping from input request based on the Account Number to EC object model and invokes the “searchContract” EC transaction.

1.7 Get the response back from EC

1.8 If error encountered during the EC call perform the error mapping and send the error response back to the service consumer

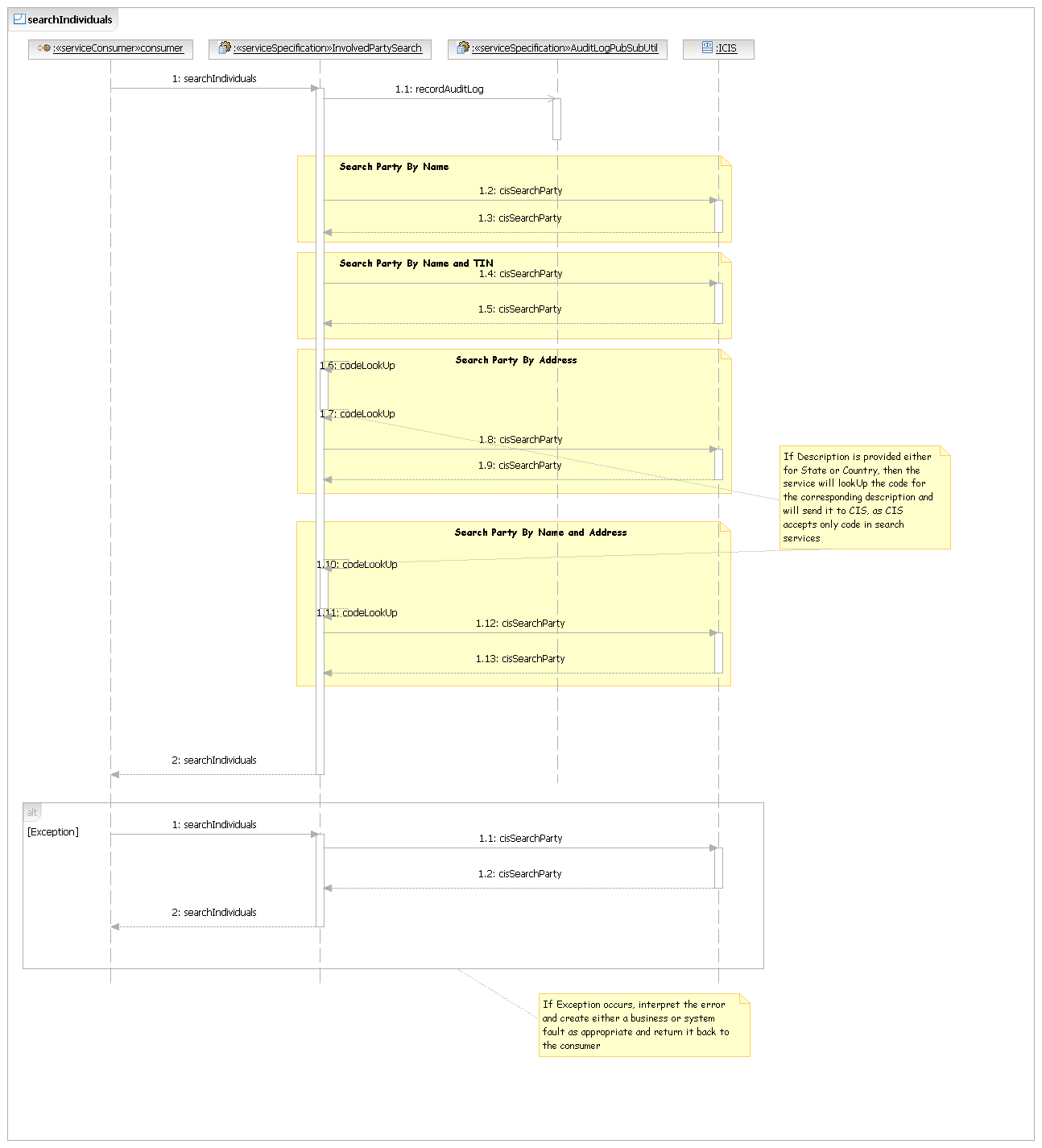
1.9 All the inquiry details will be audit logged.

2. If the EC call is success map the response message against the consumer model and send the success response to the service consumer.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Operations** | 2. searchIndividuals | | | | | |
| **Parameters** | **Name** | | **Dir** | **Type** | **Default** | |
| searchIndividualsRequest | | In | Request Message with:  IndividualSearchCriteria,  MessageHeader,  AuditHeader  MaxReturn |  | |
| Success:  searchIndividualsResponse  Failure: ErrorInfo | | Out | Response Object will include:  InvolvedParty(Individual)  ClientIDNumber  ClientIDType |  | |
| **Notes** | | SNet will maintain a static value ‘P’ for the individuals.  Client can search with any one of the following search criteria using the searchIndividuals operation.   * Name (Legal or Alias) * Address * Name and Address * Name and TIN * Last Name, TIN, DOB, MaxReturn   SNet will receive the NameUsageType value “6” from the consumer for the alias name, if no value present in the request xml for the NameUsageType then that should be considered as a legal name search. Here SNET will send the NameUsageType value “1” to CIS for the Legal name search.  The CustomerLifecycleStatus is mapped to the PartyFilter at the EC end and this is an optional field.  <PartyFilter>Active</PartyFilter> - Fetches only Active records from operational table  <PartyFilter>Inactive</PartyFilter> - Fetches only Inactive records from operational table  <PartyFilter>All</PartyFilter> - Fetches both Active and Inactive records from operational table  <PartyFilter> </PartyFilter> (blank) - Fetches only Active records from operation table | | | |

## Sequence Diagram - searchIndividuals

The following sequence diagram shows the interaction between service consumer service and the backend system for the operations in InvolvedPartySearch service.

****

**Steps for searchIndividuals Operation**

1. Client calls the searchIndividuals operation by passing search criteria in the request.

1.1 The searchIndividuals service performs mapping from input request based on the Name to EC object model and invokes the “searchParty” EC transaction.

1.2 Get the response back from the EC

1.3 Convert all the literals into code for example state.

1.5 The searchIndividuals service performs mapping from input request based on the Address to EC object model and invokes the “searchParty” EC transaction.

1.6 Get the response back from EC

1.7 Convert all the literals into code for example state.

1.8 The searchIndividuals service performs mapping from input request based on the Name and Address to EC object model and invokes the “searchParty” EC transaction.

1.10 Get the response back from EC

1.11 The searchIndividuals service performs mapping from input request based on the Name and TIN to EC object model and invokes the “searchParty” EC transaction.

1.12 Get the response back from the EC

1.13 If error encountered during the EC call perform the error mapping and send the error response back to the service consumer

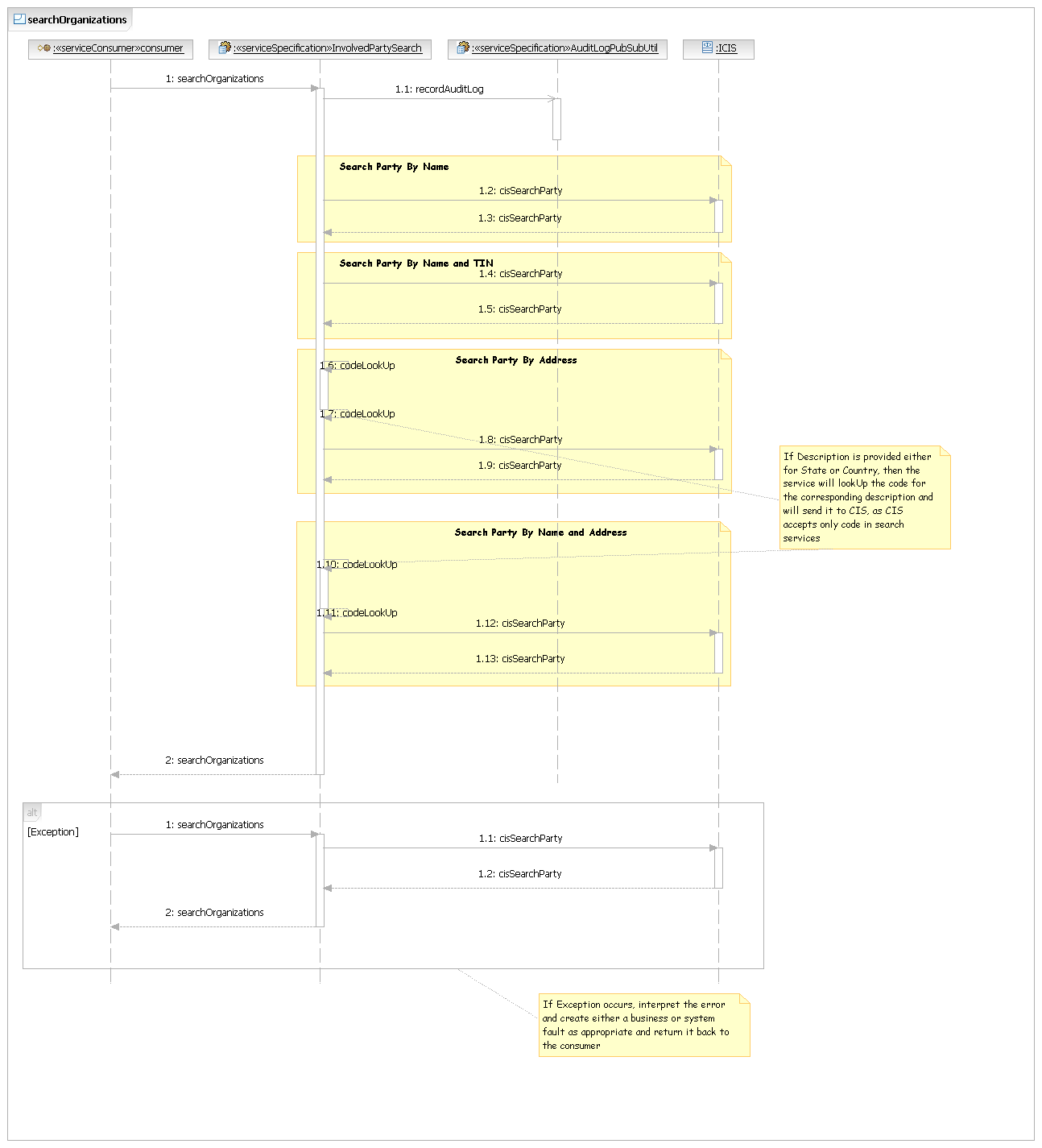
1.17 All the inquiry details will be audit logged.

2. If the EC call is success map the response message against the consumer model and send the success response to the service consumer.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Operations** | 3. searchOrganizations | | | |
| **Parameters** | **Name** | **Direction** | **Type** | **Default** |
| searchOrganizationsRequest | In | Request Message with:  organizationSearchCriteria  MessageHeader  AuditHeader  MaxReturn |  |
| searchOrganizationsResponse | Out | Response Object will include:  InvolvedParty(Organization)  ClientIDNumber  ClientIDType |  |
| **Notes** | SNet will maintain a static value ‘O’ for the organizations.  Client can search with any one of the following search criteria using the retrieveIndividuals.   * Name (Legal or Alias) * Address * Name and Address * Name, TIN, MaxReturn   SNet will receive the NameUsageType value “100001” from the consumer for the alias name, if no value present in the request xml for the NameUsageType then that should be considered as a legal name search. As per CIS no NameUsageType is required for the organization legal name search, so SNet will not send any code for organization legal name search.  The CustomerLifecycleStatus is mapped to the PartyFilter at the EC end and this is an optional field.  <PartyFilter>Active</PartyFilter> - Fetches only Active records from operational table  <PartyFilter>Inactive</PartyFilter> - Fetches only Inactive records from operational table  <PartyFilter>All</PartyFilter> - Fetches both Active and Inactive records from operational table  <PartyFilter> </PartyFilter> (blank) - Fetches only Active records from operation table | | | |

## Sequence Diagram - searchOrganizations

The following sequence diagram shows the interaction between service consumer service and the backend system for the operations in InvolvedPartySearch service.

****

**Steps for searchOrganizations Operation**

1. Client calls the searchOrganizations operation by passing Name/Name and TIN/Address /Name and Address

1.1 The searchOrganizations service performs mapping from input request based on the Name to EC object model and invokes the “searchParty” EC transaction.

1.2 Get the response back from the EC

1.3 Covert all the literals into code for example state.

1.5 The searchOrganizations service performs mapping from input request based on the Address to EC object model and invokes the “searchParty” EC transaction.

1.6 Get the response back from EC

1.7 Covert all the literals into code for example state.

1.8 The searchOrganizations service performs mapping from input request based on the Name and Address to EC object model and invokes the “searchParty” EC transaction.

1.10 Get the response back from EC

1.11 The searchOrganizations service performs mapping from input request based on the Name and TIN to EC object model and invokes the “searchParty” EC transaction.

1.12 Get the response back from the EC

1.13 If error encountered during the EC call perform the error mapping and send the error response back to the service consumer

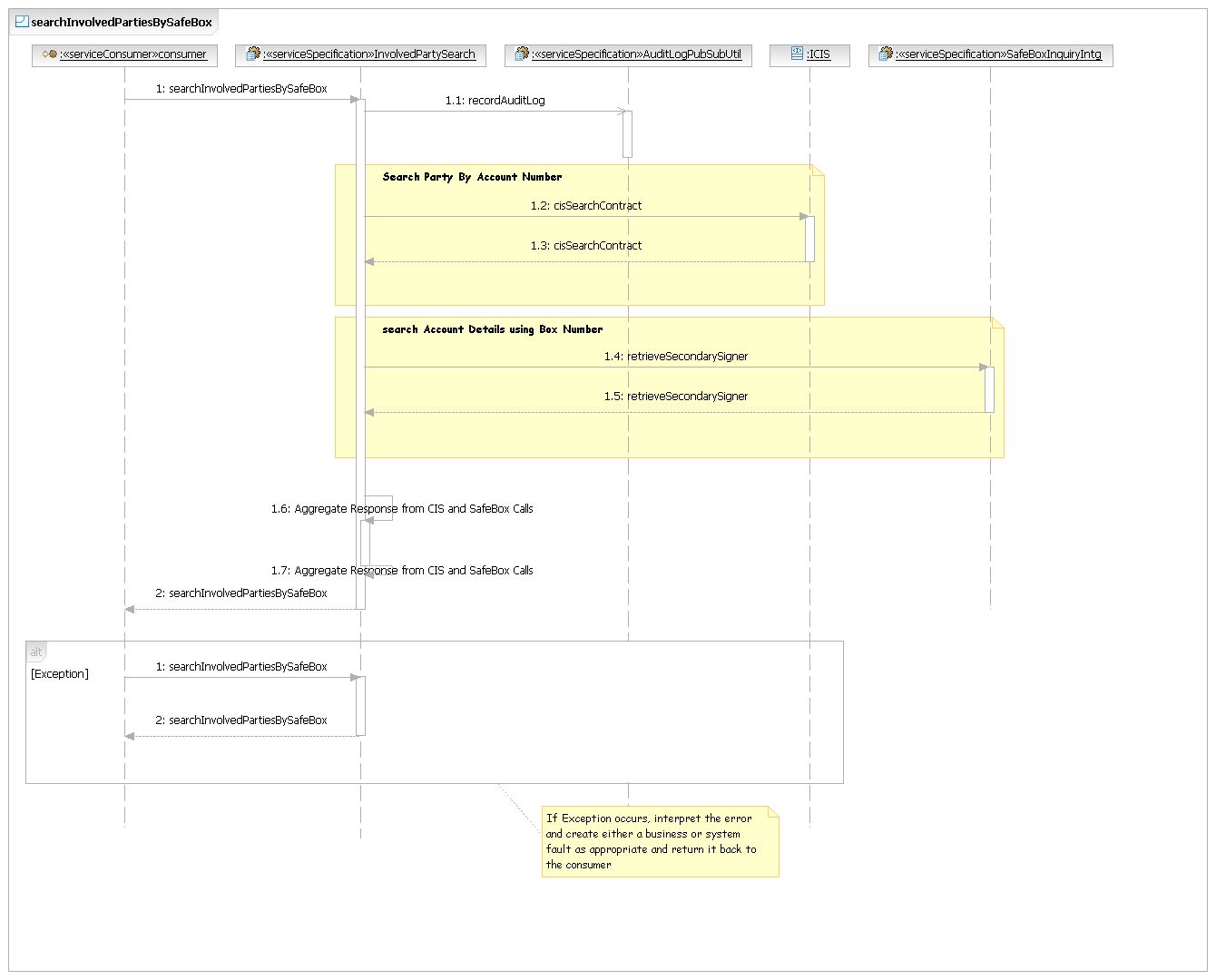
1.17 All the inquiry details will be audit logged.

2. If the EC call is success map the response message against the consumer model and send the success response to the service consumer.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Operations** | 4. searchInvolvedPartiesBySafeBox | | | |
| **Parameters** | **Name** | **Direction** | **Type** | **Default** |
| searchInvolvedPartiesBySafeBox Request | In | Request Message with:  Header  InvolvedPartyType  ProductArrangementSearchCriteria |  |
| searchInvolvedPartiesBySafeBox Response | Out | Response Object will include:  messageHeader  InvolvedpartyRole(0-\*) |  |
| **Notes** |  | | | |

## Sequence Diagram - searchInvolvedPartiesBySafeBox

The following sequence diagram shows the interaction between service consumer service and the backend system for the operations in InvolvedPartySearch service.



**Steps for searchInvolvedPartiesbySafeBox Operation**

1. Client calls the searchInvolvedpartiesBySafeBox operation by passing Box Number, Center Number, Bank Number and Account Type

1.1 The searchInvolvedpartiesBySafeBox service performs mapping from input request to EC object model and invokes the “searchContract” EC transaction.

1.2 Get the response back from the EC

1.3 The searchInvolvedpartiesBySafeBox service performs mapping from input request based on the Box Number and Center (Branch) Number to SNET’s Service Data Model Object model and invokes the “retrieveSecondarySigner” Operation in SafeBoxInquiryIntg Service.

1.4 Get the response back from retrieveSecondarySigner Operation

1.5 The searchInvolvedpartiesBySafeBox aggregates data from EC and SafeBox Calls and sends it back to the consumer who invoked it.

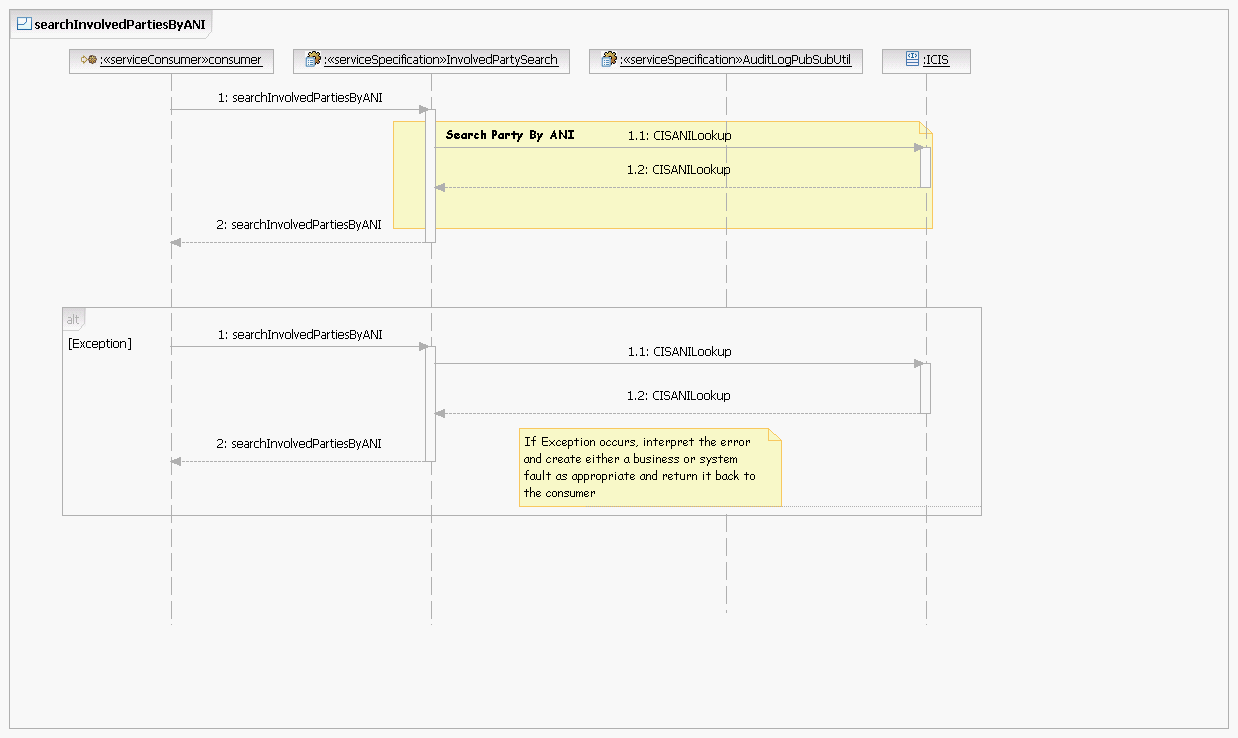
1.6 If error encountered, perform the error mapping and send the error response back to the service consumer

1.7 All the inquiry details will be audit logged.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Operations** | 5. searchInvolvedPartiesByANI | | | |
| **Parameters** | **Name** | **Dir** | **Type** | **Default** |
| searchInvolvedPartiesByANIRequest | In | Request Message with:  InvolvedPartySearchCriteria  MessageHeader |  |
| Success:  searchInvolvedPartiesByANIResponse  Failure: Faultinfo | Out | Response Object will include:  InvolvedParty(Individual or Organization) |  |
| **Notes** | Consumer can lookup party details by search criteria using the SNet service searchInvolvedPartiesByANI.   * ANI or phone number | | | |

## Sequence Diagram - searchInvolvedPartiesByANI

The following sequence diagram shows the interaction between service consumer service and the backend system for the operations in InvolvedPartySearch service.



**Steps for searchInvolvedPartiesByANI Operation**

1.1 Client calls the searchInvolvedPartiesByANI operation by passing phone number/ANI as is from consumer. No transformation or validation is performed on input value.

1.2 SNET operation creates request from input and invokes CISANILookup.

1.3 Receives response back from CIS interface.

1.4 Service responds back to consumer with party details.

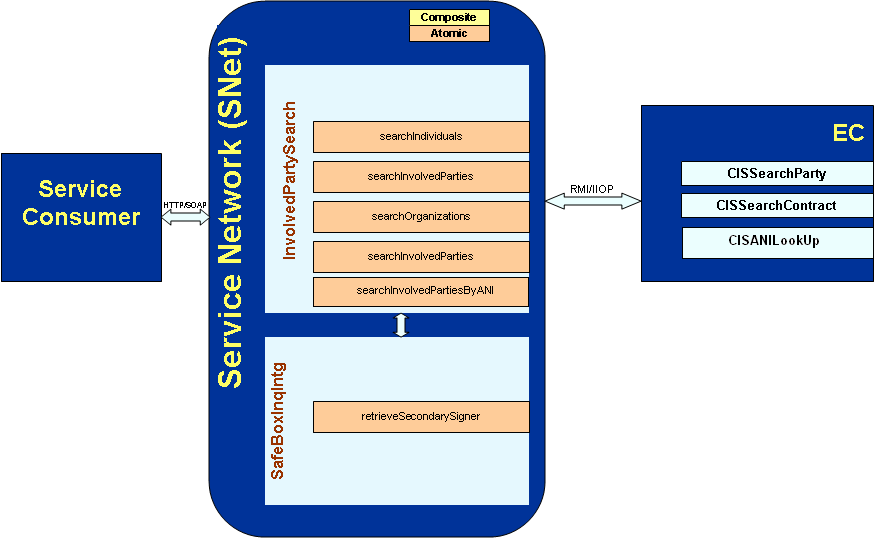
1.5 If error encountered during call to CIS, service responds error/fault to service consumer.

## Service Composition

N/A

## Service System Context Diagram

System Context Diagram represents the external entities that interact with the system.



**Figure 1: InvolvedPartySearch Service context**

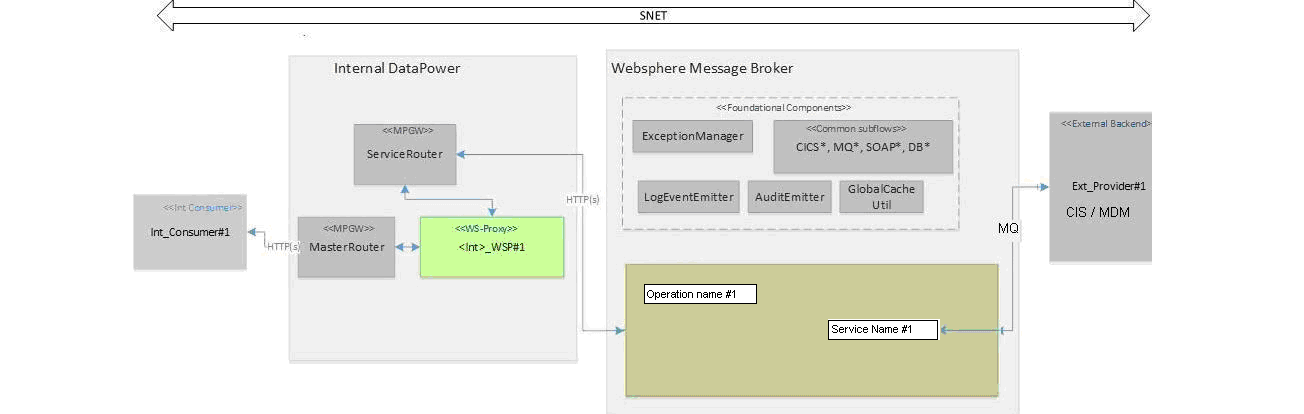
## Service Model

Please refer the following sections.

### *Service Component Design*

This section shows how the service provider implements the services interface. Notice that the diagram shows the message signatures. For more details on the parameters and return types, see the sections below.

**Data Power:**

****

**CIS Visio Diagram**

The following sections explain the request/reply message flows.

**WMB:**

### There are four WMB project artifacts involved in this framework.

|  |  |  |
| --- | --- | --- |
| Container Type |  | Description |
| InvolvedPartySearchService | Service | This is the application with a defined interface that acts as a container for a Web Services solution. It contains flows, sub flows, implement the specified Service operations. This service has dependencies to SNET\_Framework library. |
| SNET\_Framework | Library | SNET2.0 reusable framework library having sub flows for logging and error handling. |
| MSGSET\_Provider\_MDMRequest | MessageSet | MessageSet for Reference  Implementation and Validation |
| MSGSET\_Provider\_MDMResponse | MessageSet | MessageSet for Reference  Implementation and Validation |

The following sections explain the request/reply message flows.

## 

## WMB Implementation Details

**Execution Group :**

|  |  |
| --- | --- |
| Execution Group | EG\_SNE\_BRKA\_051,EG\_SNE\_BRKA\_052 |

**Queues :**

|  |  |
| --- | --- |
| **Queue Type** | **Queue Name** |
| Input Queue | SBK.IPSEARCHSERVICE.SBK.REQ.I.I.S.L |
| Error Queue | SBK.IPSEARCHSERVICE.SBK.REQ.I.I.S.E |
| Audit Queue | SBK.ALS\_RECORDAUDITLOG.SBK.REQ.I.I.S.L |
| Logger Queue | SBK.LOG\_EG\_SNE\_BRKA\_05.SBK.REQ.I.I.U.L |

**A. InvolvedPartySearchService.msgflow:**

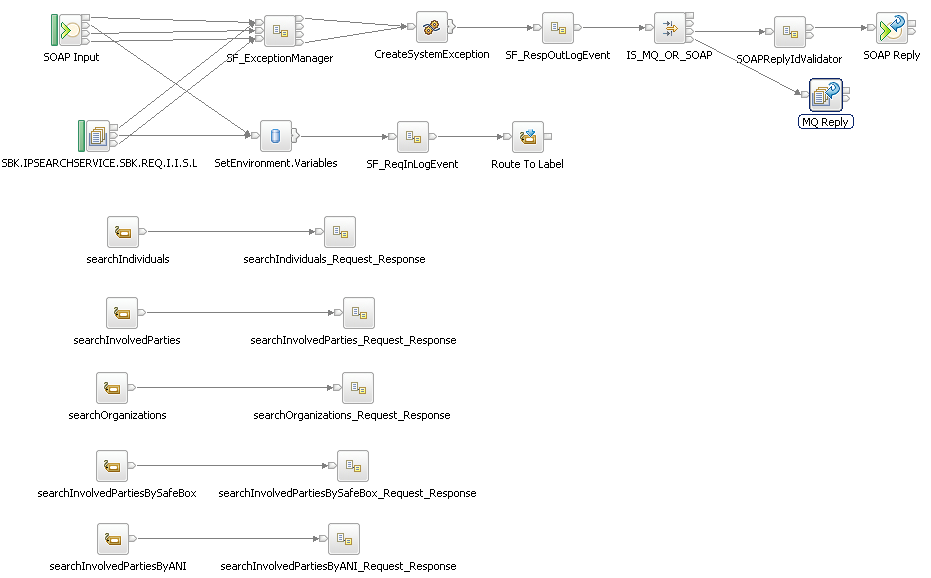
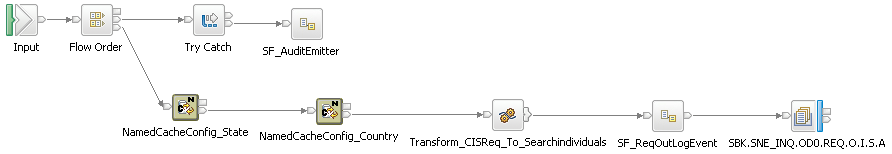


Figure 1: Depicts the service implementation main flow.

|  |  |  |  |
| --- | --- | --- | --- |
| **Node Name** | **WMB Node Type** | **Description** | **Non-default property names – values** |
| **SOAP Input** | SOAPInput | This node accepts and processes SOAP messages. The node is configured using deployable WSDL. The SOAP Input node routes each message that it retrieves successfully to the Out terminal. Both the Failure and Catch terminals are connected to the SF\_ExceptionManager sub flow’s “Failure In” and “Catch In” terminals respectively. HTTP timeout terminal is connected to “Time Out In” terminal of SF\_ExceptionManager. HTTP Timeout terminal is fired if there is no message received within the configured “Maximum client wait time”. In case of timeout a SOAP fault is generated and sent back to client indicating that the service is timed out.  Since “Set destination list” property is enabled on the SOAP Input node, method binding names are added to the route-to-label destination list. So that we can use a RouteToLabel node in the message flow after the SOAPInput node as shown in this flow. Based on the operation name the message comes to corresponding Label node. | Maximum client wait Time (sec) =20  Transaction Mode = no  Validation = none |
| **SF\_ReqInLogEvent** | Subflow | This subflow is used to prepare and emit the response in [00] log message. |  |
| **CreateSystemException** | Compute | In this compute node, the generic fault which is generated from the Exception Manager is wrapped in to service specific system exception. |  |
| **SF\_RespOutLogEvent** | Subflow | Subflow used to prepare and emit the log messages for transaction type “response out” [11] and it send back the response to the caller via MQOutput. |  |
| **IS\_MQ\_OR\_SOAP** | Filter | Filter node has the logic to check whether incoming message is MQ or SOAP message. Based on the response message route the control to either success or fault terminals. |  |
| **SetEnvironment.Variables** | Database | This is used to set environment variables and routing the message according to operation name and storing required fields in cache |  |
| **SOAPReplyIdValidator** | Subflow | This Subflow validates the Soap ReplyID. |  |
| **MQ Reply** | MQ Reply | Node to send MQ reply to consumer in case of exceptions. |  |
| **SOAP Reply** | SOAP Reply | Node to send SOAP reply to consumer in case of exceptions. |  |

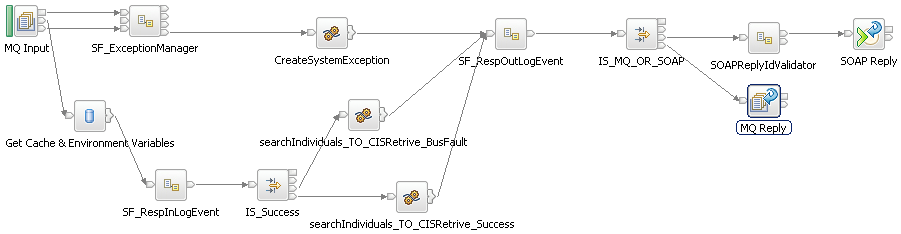
## 

## B. searchIndividuals\_Request\_Response.subflow:



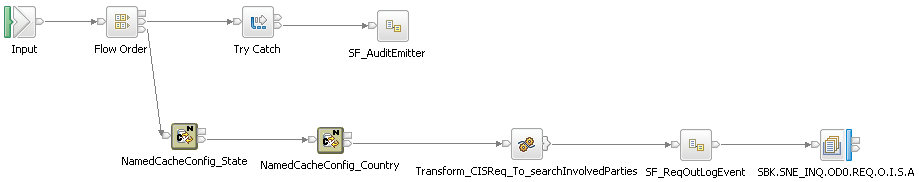
|  |  |  |  |
| --- | --- | --- | --- |
| **Node Name** | **WMB Node Type** | **Description** | **Non-default property names – values** |
| **NamedcacheConfig\_state** | NamedcacheConfig | This NamedcacheConfig node configures the state properties. |  |
| **NamedcacheConfig\_Country** | NamedcacheConfig | This NamedcacheConfig node configures the country properties. |  |
| **Transform\_CISReq\_To\_searchIndividuals** | compute | This compute node maps the incoming request xml and prepares Individuals request message |  |
| **SF\_ReqOutLogEvent** | SubFlow | This subflow is used to prepare and emit the MQ request in [01] log message. |  |
| **MQ Output** | MQ Output | This node places the request messages into CIS Queue. |  |

**C. searchIndividuals\_Response.msgflow:**



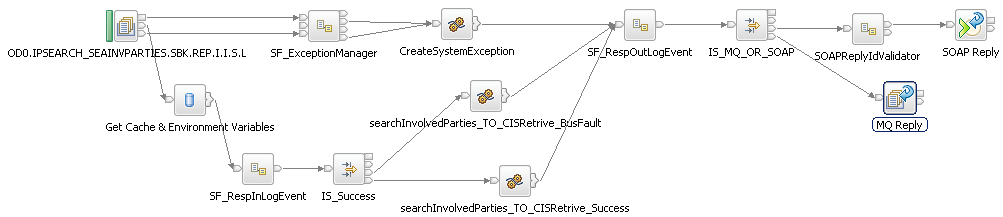
|  |  |  |  |
| --- | --- | --- | --- |
| **Node Name** | **WMB Node Type** | **Description** | **Non-default property names – values** |
| **MQInput** | MQInput | This node gets the response from CIS and processes it for logging. |  |
| **Get Cache & Environment.Variables** | Database | This node retrieves the data and Environment Variables from cache |  |
| **SF\_RespInLogEvent** | Subflow | This subflow is used to prepare and emit the MQ request in [10] log message. |  |
| **IS\_Success** | Filter | Filter node has the logic to check the fault in the provider response message and based on the result route the control to either true or false terminals |  |
| **searchIndividuals\_TO**  **\_CISRetrive\_BusFault** | Compute | This Compute node Prepares service Business Exception message in case if provider response message has Success <> ‘true’. |  |
| **searchIndividuals\_TO**  **\_CISRetrive\_Success** | Compute | This Compute node Prepares service response message in case if provider response message has Success = ‘true’ |  |
| **SF\_RespOutLogEvent** | Subflow | This subflow is used to prepare and emit the MQ request in [11] log message. |  |
| **IS\_MQ\_OR\_SOAP** | Filter | Filter node has the logic to check whether incoming message is MQ or SOAP message. Based on the response message route the control to either success or fault terminals. |  |
| **SOAPReplyIdValidator** | subflow | This Subflow validates the Soap ReplyID. |  |
| **SOAP Reply** | SOAP Reply | Node to send SOAP reply to consumer in case of exceptions. |  |
| **MQ Reply** | MQ Reply | Node to send MQ reply to consumer in case of exceptions. |  |

## B. searchInvolvedParties\_Request\_Response.subflow:



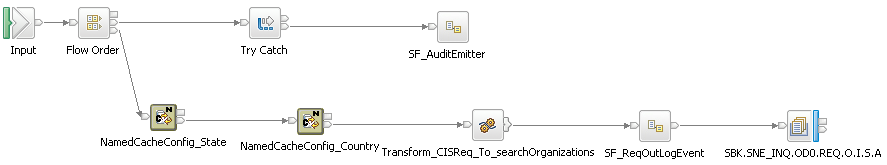
|  |  |  |  |
| --- | --- | --- | --- |
| **Node Name** | **WMB Node Type** | **Description** | **Non-default property names – values** |
| **NamedcacheConfig\_state** | NamedcacheConfig | This NamedcacheConfig node configures the state properties. |  |
| **NamedcacheConfig\_Country** | NamedcacheConfig | This NamedcacheConfig node configures the country properties. |  |
| **Transform\_CISReq\_To\_searchInvolvedParties** | compute | This compute node maps the incoming request xml and prepares involvedParty request message |  |
| **SF\_ReqOutLogEvent** | SubFlow | This subflow is used to prepare and emit the MQ request in [01] log message. |  |
| **MQ Output** | MQ Output | This node places the request messages into CIS Queue. |  |

**C. searchInvolvedParties\_Response.msgflow:**



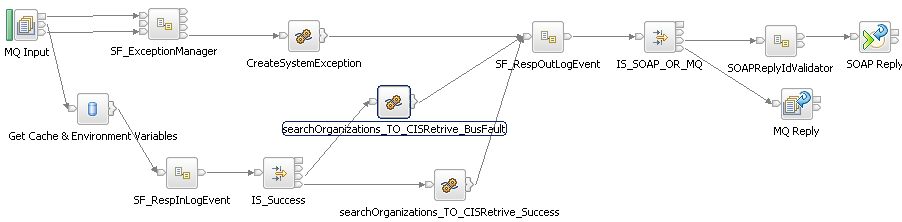
|  |  |  |  |
| --- | --- | --- | --- |
| **Node Name** | **WMB Node Type** | **Description** | **Non-default property names – values** |
| **MQInput** | MQInput | This node gets the response from CIS and processes it for logging. |  |
| **Get Cache & Environment.Variables** | Database | This node retrieves the data and Environment Variables from cache |  |
| **SF\_RespInLogEvent** | Subflow | This subflow is used to prepare and emit the MQ request in [10] log message. |  |
| **IS\_Success** | Filter | Filter node has the logic to check the fault in the provider response message and based on the result route the control to either true or false terminals |  |
| **searchInvolvedParties\_TO\_CISRetrive\_BusFault** | Compute | This Compute node Prepares service Business Exception message in case if provider response message has Success <> ‘true’. |  |
| **searchInvolvedParties\_TO\_CISRetrive\_Success** | Compute | This Compute node Prepares service response message in case if provider response message has Success = ‘true’ |  |
| **SF\_RespOutLogEvent** | Subflow | This subflow is used to prepare and emit the MQ request in [11] log message. |  |
| **IS\_MQ\_OR\_SOAP** | Filter | Filter node has the logic to check whether incoming message is MQ or SOAP message. Based on the response message route the control to either success or fault terminals. |  |
| **SOAPReplyIdValidator** | Subflow | This Subflow validates the Soap ReplyID. |  |
| **SOAP Reply** | SOAP Reply | Node to send SOAP reply to consumer in case of exceptions. |  |
| **MQ Reply** | MQ Reply | Node to send MQ reply to consumer in case of exceptions. |  |

## B. searchOrganizations\_Request\_Response.subflow:



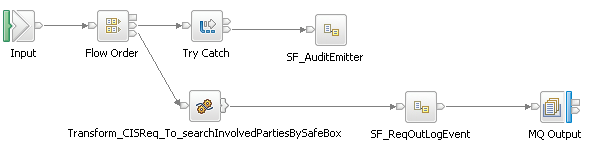
|  |  |  |  |
| --- | --- | --- | --- |
| **Node Name** | **WMB Node Type** | **Description** | **Non-default property names – values** |
| **NamedcacheConfig\_state** | NamedcacheConfig | This NamedcacheConfig node configures the state properties. |  |
| **NamedcacheConfig\_Country** | NamedcacheConfig | This NamedcacheConfig node configures the country properties. |  |
| **Transform\_CISReq\_To\_searchOrganizations** | compute | This compute node maps the incoming request xml and prepares Organization request message |  |
| **SF\_ReqOutLogEvent** | SubFlow | This subflow is used to prepare and emit the MQ request in [01] log message. |  |
| **MQ Output** | MQ Output | This node places the request messages into CIS Queue. |  |

**C. searchOrganizations\_Response.msgflow:**



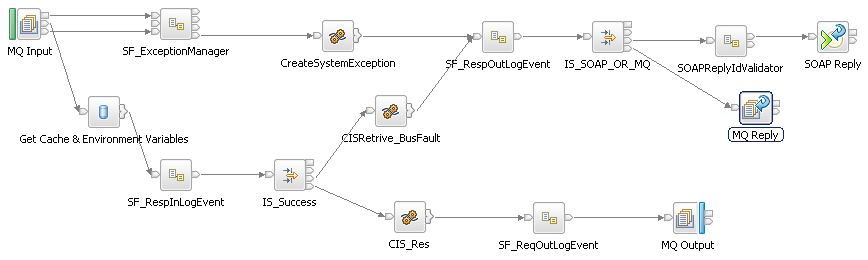
|  |  |  |  |
| --- | --- | --- | --- |
| **Node Name** | **WMB Node Type** | **Description** | **Non-default property names – values** |
| **MQInput** | MQInput | This node gets the response from CIS and processes it for logging. |  |
| **Get Cache & Environment.Variables** | Database | This node retrieves the data and Environment Variables from cache |  |
| **SF\_RespInLogEvent** | Subflow | This subflow is used to prepare and emit the MQ request in [10] log message. |  |
| **IS\_Success** | Filter | Filter node has the logic to check the fault in the provider response message and based on the result route the control to either true or false terminals |  |
| **searchOrganizations\_TO**  **\_CISRetrive\_BusFault** | Compute | This Compute node Prepares service Business Exception message in case if provider response message has Success <> ‘true’. |  |
| **searchOrganizations\_TO**  **\_CISRetrive\_Success** | Compute | This Compute node Prepares service response message in case if provider response message has Success = ‘true’ |  |
| **SF\_RespOutLogEvent** | Subflow | This subflow is used to prepare and emit the MQ request in [11] log message. |  |
| **IS\_SOAP \_OR\_ MQ** | Filter | Filter node has the logic to check whether incoming message is MQ or SOAP message. Based on the response message route the control to either success or fault terminals. |  |
| **SOAPReplyIdValidator** | subflow | This Subflow validates the Soap ReplyID. |  |
| **SOAP Reply** | SOAP Reply | Node to send SOAP reply to consumer in case of exceptions. |  |
| **MQ Reply** | MQ Reply | Node to send MQ reply to consumer in case of exceptions. |  |

## B. searchInvolvedPartiesBySafeBox\_Request\_Response.subflow:



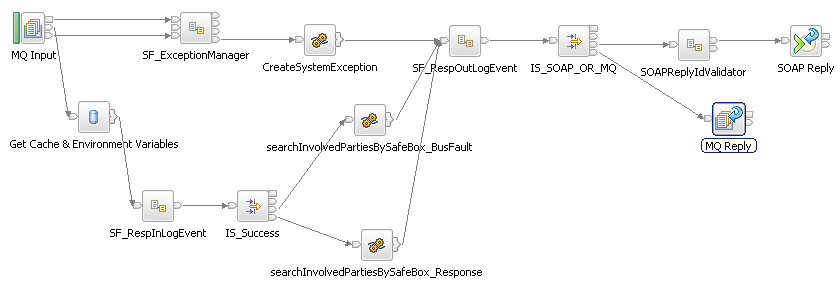
|  |  |  |  |
| --- | --- | --- | --- |
| **Node Name** | **WMB Node Type** | **Description** | **Non-default property names – values** |
| **Transform\_CISReq\_To\_searchInvolvedPartiesBySafeBox** | compute | This compute node maps the incoming request xml and prepares involvedParty request message |  |
| **SF\_ReqOutLogEvent** | SubFlow | This subflow is used to prepare and emit the MQ request in [01] log message. |  |
| **MQ Output** | MQ Output | This node places the request messages into CIS Queue. |  |

**C. searchInvolvedPartiesBySafeBox\_CIS\_Response.msgflow:**



|  |  |  |  |
| --- | --- | --- | --- |
| **Node Name** | **WMB Node Type** | **Description** | **Non-default property names – values** |
| **MQInput** | MQInput | This node gets the response from CIS and processes it for logging. |  |
| **Get Cache & Environment.Variables** | Database | This node retrieves the data and Environment Variables from cache |  |
| **SF\_RespInLogEvent** | Subflow | This subflow is used to prepare and emit the MQ request in [10] log message. |  |
| **IS\_Success** | Filter | Filter node has the logic to check the fault in the provider response message and based on the result route the control to either true or false terminals |  |
| **CISRetrive\_BusFault** | Compute | This Compute node Prepares service Business Exception message in case if provider response message has Success <> ‘true’. |  |
| **CIS\_Res** | Compute | This Compute node Prepares service response message in case if provider response message has Success = ‘true’ |  |
| **SF\_RespOutLogEvent** | Subflow | This subflow is used to prepare and emit the MQ request in [11] log message. |  |
| **IS\_SOAP \_OR\_ MQ** | Filter | Filter node has the logic to check whether incoming message is MQ or SOAP message. Based on the response message route the control to either success or fault terminals. |  |
| **SOAPReplyIdValidator** | subflow | This Subflow validates the Soap ReplyID. |  |
| **SOAP Reply** | SOAP Reply | Node to send SOAP reply to consumer in case of exceptions. |  |
| **MQ Reply** | MQ Reply | Node to send MQ reply to consumer in case of exceptions. |  |

**D. searchInvolvedPartiesBySafeBox\_Response.msgflow:**



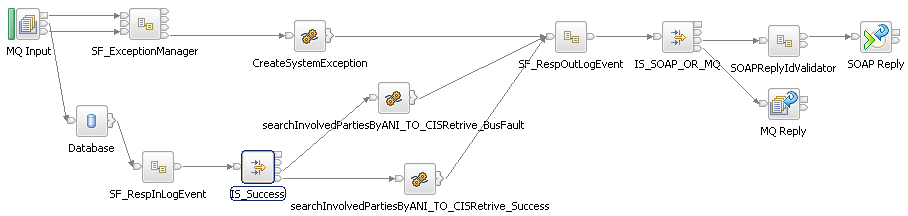
|  |  |  |  |
| --- | --- | --- | --- |
| **Node Name** | **WMB Node Type** | **Description** | **Non-default property names – values** |
| **MQInput** | MQInput | This node gets the response from CIS and processes it for logging. |  |
| **Get Cache & Environment.Variables** | Database | This node retrieves the data and Environment Variables from cache |  |
| **SF\_RespInLogEvent** | Subflow | This subflow is used to prepare and emit the MQ request in [10] log message. |  |
| **IS\_Success** | Filter | Filter node has the logic to check the fault in the provider response message and based on the result route the control to either true or false terminals |  |
| **searchInvolvedPartiesBySafeBox\_TO\_CISRetrive\_BusFault** | Compute | This Compute node Prepares service Business Exception message in case if provider response message has Success <> ‘true’. |  |
| **searchInvolvedPartiesBySafeBox\_TO\_CISRetrive\_Success** | Compute | This Compute node Prepares service response message in case if provider response message has Success = ‘true’ |  |
| **SF\_RespOutLogEvent** | Subflow | This subflow is used to prepare and emit the MQ request in [11] log message. |  |
| **IS\_SOAP \_OR\_ MQ** | Filter | Filter node has the logic to check whether incoming message is MQ or SOAP message. Based on the response message route the control to either success or fault terminals. |  |
| **SOAPReplyIdValidator** | subflow | This Subflow validates the Soap ReplyID. |  |
| **SOAP Reply** | SOAP Reply | Node to send SOAP reply to consumer in case of exceptions. |  |
| **MQ Reply** | MQ Reply | Node to send MQ reply to consumer in case of exceptions. |  |

## B. searchInvolvedPartiesByANI\_Request\_Response.subflow:



|  |  |  |  |
| --- | --- | --- | --- |
| **Node Name** | **WMB Node Type** | **Description** | **Non-default property names – values** |
| **Transform\_CISReq\_To\_searchInvolvedPartiesByANI** | compute | This compute node maps the incoming request xml and prepares involvedParty request message |  |
| **SF\_ReqOutLogEvent** | SubFlow | This subflow is used to prepare and emit the MQ request in [01] log message. |  |
| **MQ Output** | MQ Output | This node places the request messages into CIS Queue. |  |

**C. searchInvolvedPartiesByANI\_Response.msgflow:**

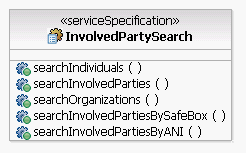


|  |  |  |  |
| --- | --- | --- | --- |
| **Node Name** | **WMB Node Type** | **Description** | **Non-default property names – values** |
| **MQInput** | MQInput | This node gets the response from CIS and processes it for logging. |  |
| **Get Cache & Environment.Variables** | Database | This node retrieves the data and Environment Variables from cache |  |
| **SF\_RespInLogEvent** | Subflow | This subflow is used to prepare and emit the MQ request in [10] log message. |  |
| **IS\_Success** | Filter | Filter node has the logic to check the fault in the provider response message and based on the result route the control to either true or false terminals |  |
| **searchInvolvedPartiesByANI\_TO**  **\_CISRetrive\_BusFault** | Compute | This Compute node Prepares service Business Exception message in case if provider response message has Success <> ‘true’. |  |
| **searchInvolvedPartiesByANI\_TO**  **\_CISRetrive\_Success** | Compute | This Compute node Prepares service response message in case if provider response message has Success = ‘true’ |  |
| **SF\_RespOutLogEvent** | Subflow | This subflow is used to prepare and emit the MQ request in [11] log message. |  |
| **IS\_SOAP \_OR\_ MQ** | Filter | Filter node has the logic to check whether incoming message is MQ or SOAP message. Based on the response message route the control to either success or fault terminals. |  |
| **SOAPReplyIdValidator** | subflow | This Subflow validates the Soap ReplyID. |  |
| **SOAP Reply** | SOAP Reply | Node to send SOAP reply to consumer in case of exceptions. |  |
| **MQ Reply** | MQ Reply | Node to send MQ reply to consumer in case of exceptions. |  |

### Service Specification Design

The InvolvedPartySearch is a simple UML service interface defined in the service specification package. It provides the atomic operation and it has the parameters defined and can raise exceptions.

***Involved Interface diagram :***



### *WSDL Definition*

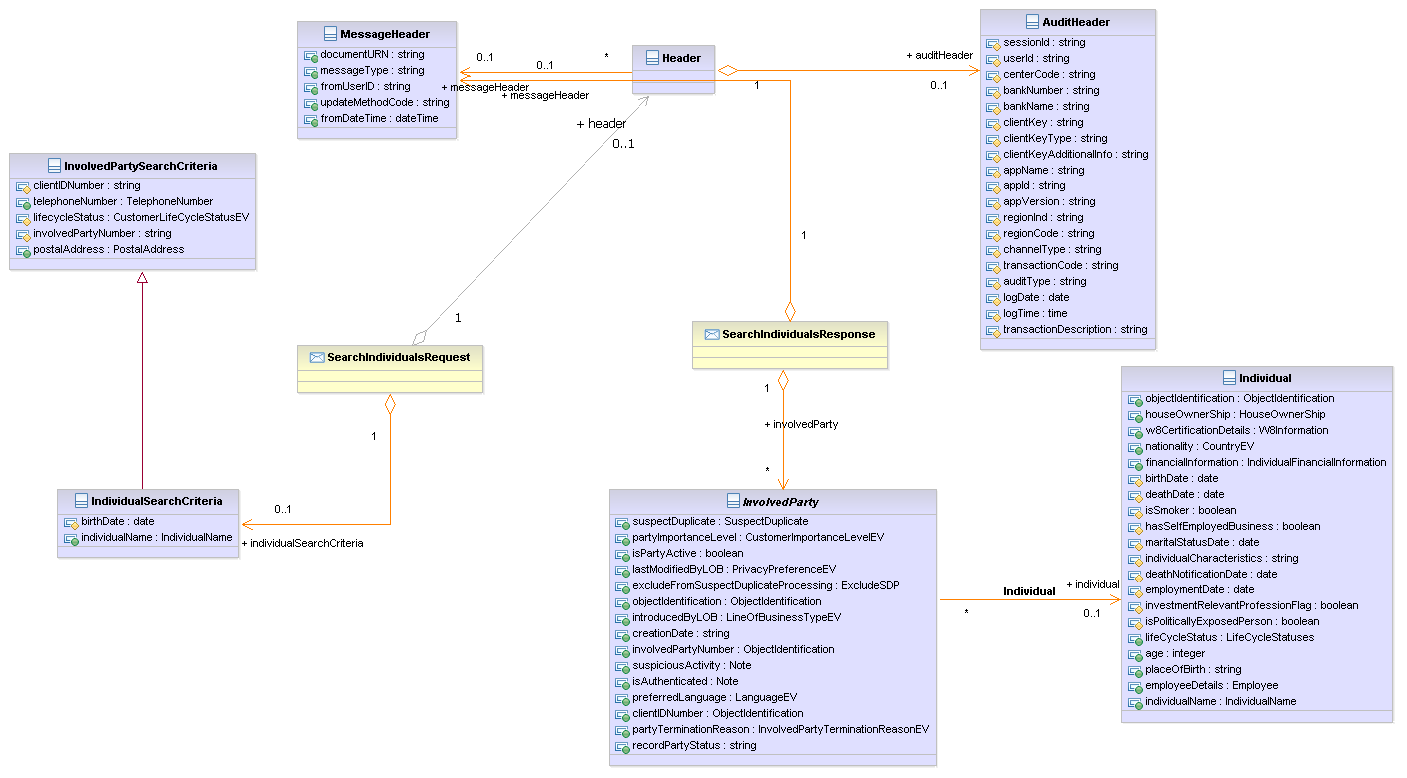
### *Service Message Design*

The following diagrams represent the Request and Response pairs for each service

operation. For more details on the complex types, see the Service Data section.

1. **searchIndividuals**

**Request/Response**

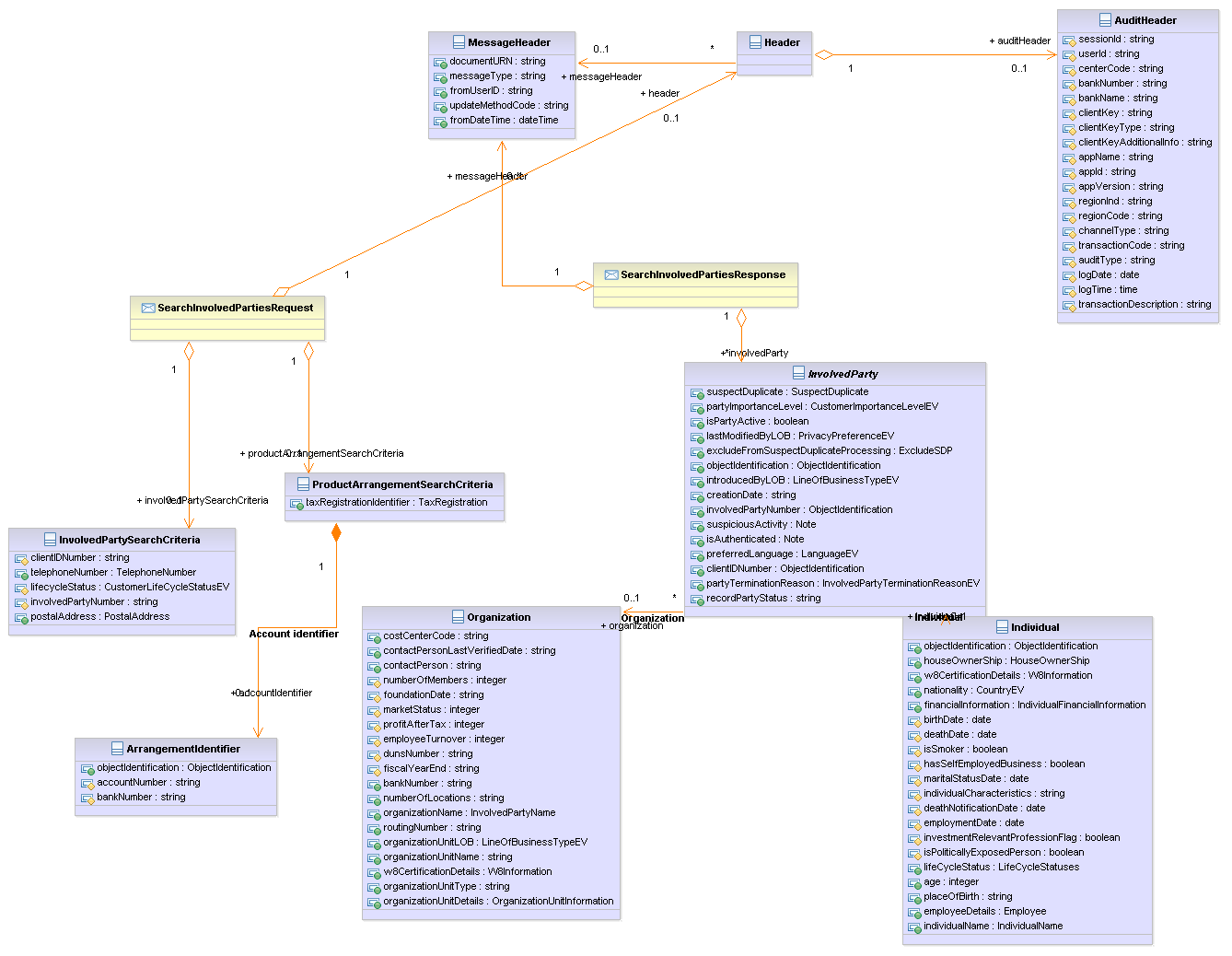


**Static fields setting:**

* TCRMTxType=searchParty
* For Person
  + TCRMTxObject=TCRMPersonSearchBObj

1. **searchInvolvedParties**

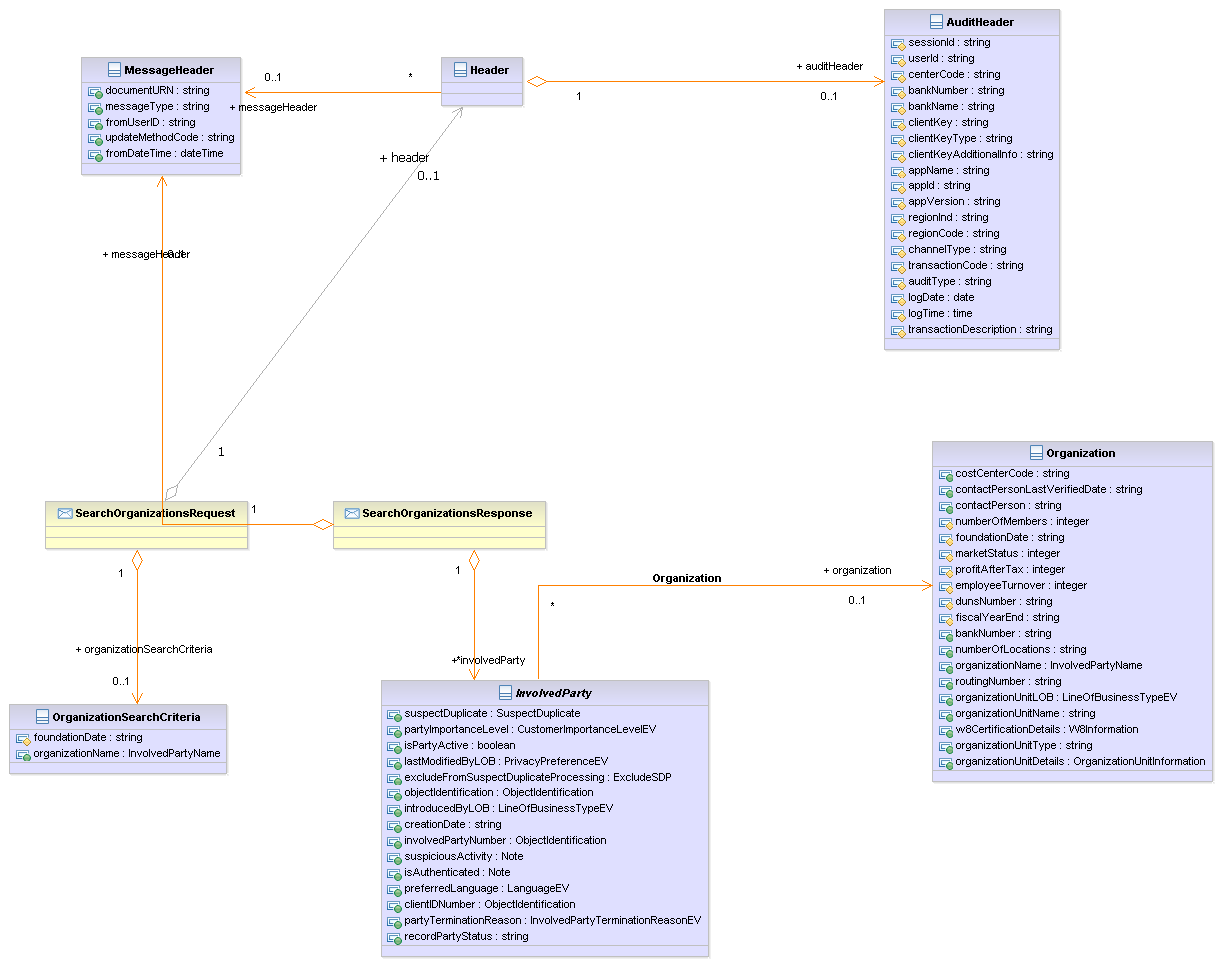
**Request/Response**



* TCRMTxType=searchParty
  + For Person and Organization
    - TCRMTxObject=TCRMPersonSearchBObj
    - TCRMTxObject=TCRMOrganizationSearchBObj
* TCRMTxType=searchContract – (if Account Number is present in the input)
  + For Contract
    - TCRMTxObject=TCRMContractSearchBObj

1. **searchOrganizations**

**Request/Response:**

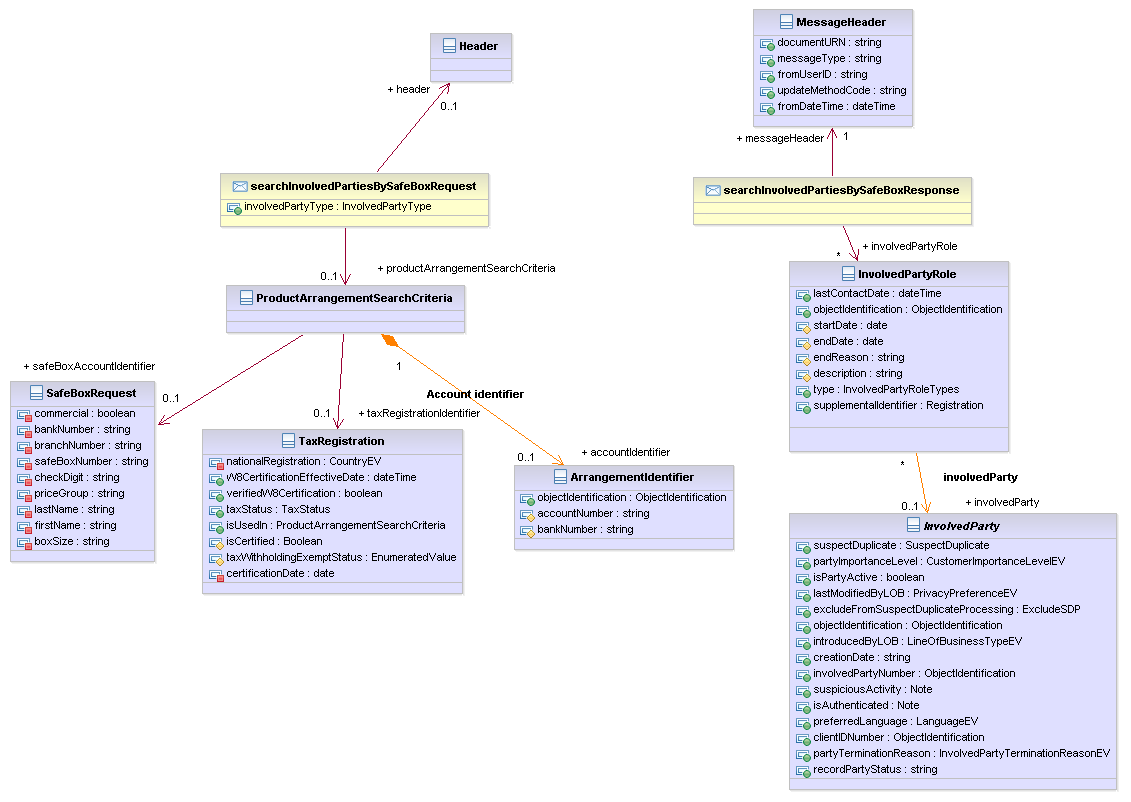


**Static fields setting:**

* + - TCRMTxType=searchParty
    - TCRMTxObject=TCRMOrganizationSearchBObj

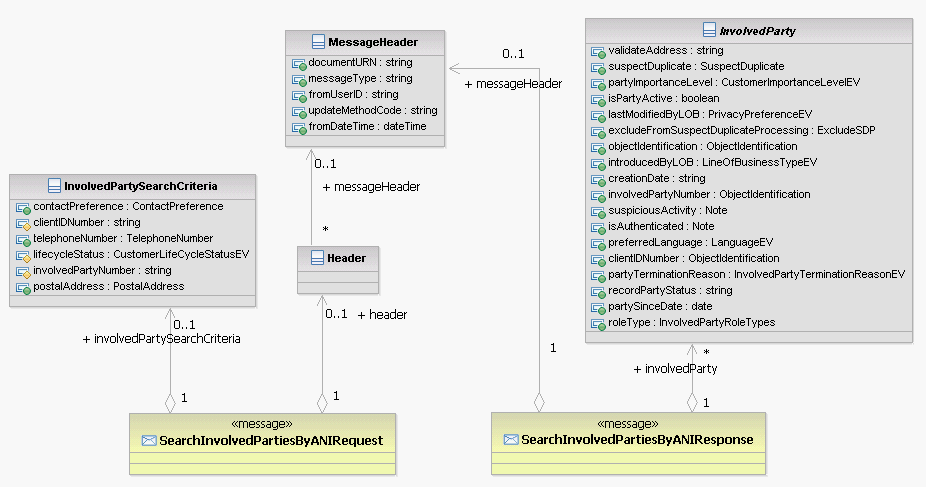
1. **searchInvolvedPartiesBySafeBox**

**Request/Response:**

****

1. **searchInvolvedPartiesByANI**

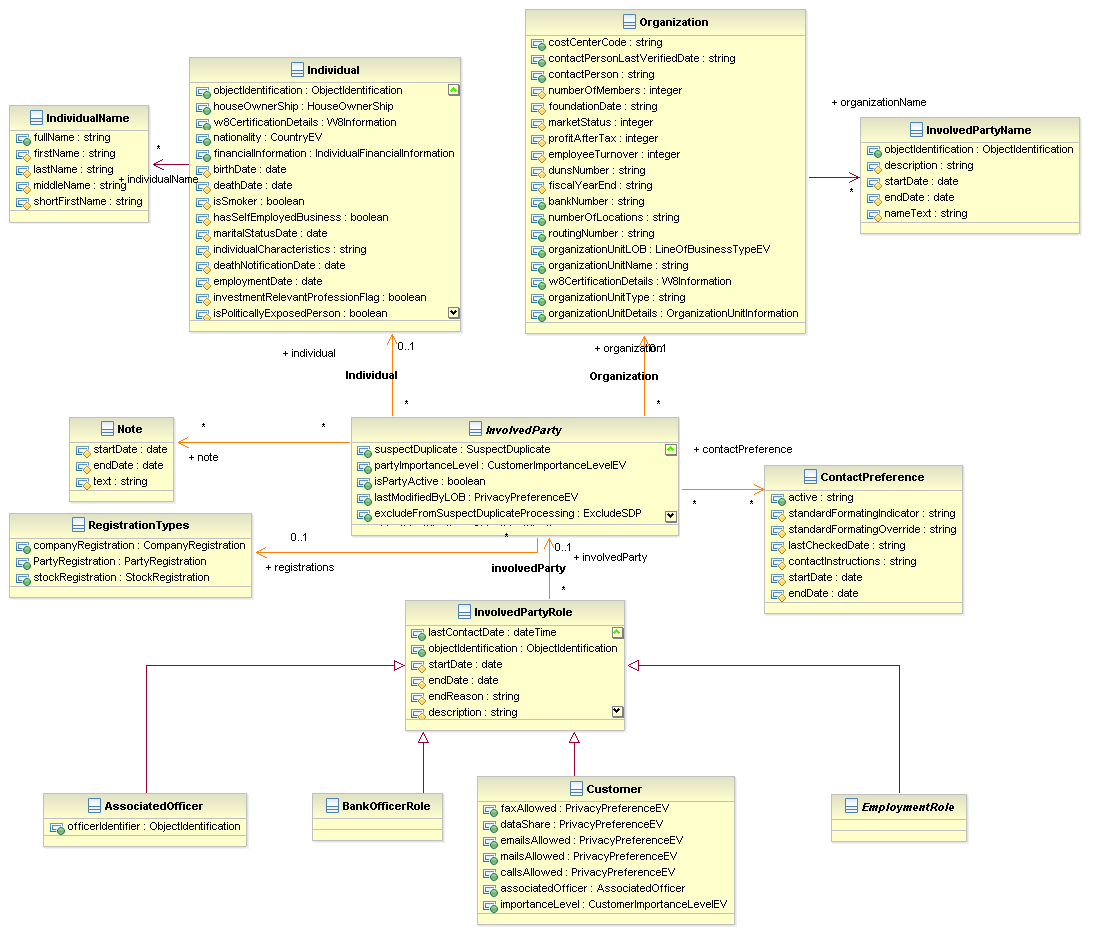
**Request/Response:**

****

*Service Data Design*

The following diagram shows the service data design for the service operations. This data represents the fraction of the enterprise data model that applies to the operations.

**Qualified Name: InvolvedPartySearch**



### *Layout*

|  |  |
| --- | --- |
| Owning App ID | SNE |
| Owning Application | Enterprise Service Network |
| Transports | HTTPS |
| Type | Atomic Service |
| Format | XML |
| Endpoint Type | Webservice |
| Transaction volumes per day | TBD per day |
| Interface Dependencies | EC |

### *WSDL*



## Service Binding or Protocol

Argo – SNet : SOAP over HTTP

Pega – SNet : SOAP over HTTP

SNet – EC : RMI/IIOP

## Service Data Mapping

### *Request and Response Mapping*



### *Legacy System Data Definition*

N/A

# VI Unit Testing Strategy

This service conforms to the standard unit testing best practices set forth by the enterprise architecture guidelines. The service developer(s) will build automated test scripts for each service operation to exercise the operation.

## Definitions of Processing Cycles

## Interface Dependencies

The functionality of this service depends on the following:

* The Enterprise CIS(WCC) Services needs to be up and running.

## Testing Tool(s)

## SOAP UI



## Environmental Considerations

## Test Considerations

Unit test transactions will be determined by the functional requirements and the mappings of functions to transactions. Both positive and negative tests should be made in this test practice.

Test scenarios:

* Retrieve involved party for Person based on Name
* Retrieve involved party for Person based on Name and Address
* Retrieve involved party for Person based on Last Name, TIN, and DOB
* Retrieve involved party for Organization based on Name
* Retrieve involved party for Organization based on Name and Address
* Retrieve involved party for Person/Organization based on TIN
* Retrieve involved party for Person/Organization based on CIN
* Retrieve involved party for Person/Organization based on Address
* Retrieve involved party for Person/Organization based on Name & TIN
* Retrieve involved party for Person/Organization based on Telephone
* Retrieve involved party for based on Account
* Mandatory input unavailable.
* All Error conditions

Below test scenarios for alias name search

* Full Last Name
* Partial Last Name
* Full Last Name and Full First Name
* Full Last Name and Partial First Name
* Partial Last Name and Partial First Name
* Partial Last Name and Partial First Name
* Full Organization Name
* Partial Organization Name

# VII Appendix

## SNET Error Conditions Summary

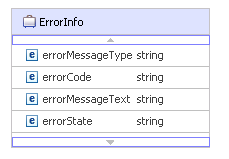
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| System | UC Name | Source Document | Error Trigger | Error Outcome |
| SNET |  |  | The input request message is Invalid. |  |
| SNET |  |  | Transformation errors while mapping the request/response to/from source system request/response message formats |  |
| SNET |  |  | Errors occurred while invoking the source system. |  |
| SNET |  |  | The source system returns an error message. |  |
| SNET |  |  | Errors occurred while sending the message back from atomic operation to composite. |  |

## Source System Error Conditions Summary

NOTE: A complete list of CIS Error Messages can be found at the following link:

[http://teamsites.suntrust.com/sites/pgg/enterprisecustomer/CRFRetire/Shared%20Documents/04.%20Design%20and%20Activation%20Planning/Technical%20Design%20Document/(DSD)Design%20Specification%20Documents/WCC/Component\_Error\_Ids.xls](file:///C:\Users\a1ry3\.cascm\workspace\.metadata\pgg\enterprisecustomer\CRFRetire\Shared%20Documents\04.%20Design%20and%20Activation%20Planning\Technical%20Design%20Document\(DSD)Design%20Specification%20Documents\WCC\Component_Error_Ids.xls)

## Error Info Model



## Supporting Documentation

|  |  |
| --- | --- |
| **Document Name** | **Location** |
|  |  |

NOTE: Please reference the link below for the most current version of the CIS Code Table Values document.

<http://teamsites.suntrust.com/sites/WkstrCrfrp/Workstream%20Information/Forms/AllItems.aspx?RootFolder=%2fsites%2fWkstrCrfrp%2fWorkstream%20Information%2fApplications%2fCIS%2fReference%20Documents%2fCode%20Table%20Updates%20Information&FolderCTID=0x0120000DADA7B87864DF46B6E515E909EC2CE6&View=%7bBC7F122A%2dF9DA%2d49E1%2dBB09%2dCD35050602AF%7d>

**Service Name Changes**:

This service is refactored from IFW Model to Service Design Model and following are the operation name changes:

|  |  |  |  |
| --- | --- | --- | --- |
| **Old IFW/SOM Service Name** | **Old IFW/SOM Operation Name** | **New Service Design Model Service Name** | **New Service Design Model Operation Name** |
| InvolvedPartySearch | retrievePartiesInvolvedInArrangement | InvolvedPartySearch | searchinvolvedParties |
| InvolvedPartyInquiryBiz | searchInvolvedPartyAccountBySafeBox | InvolvedPartySearch | searchInvolvedPartiesBySafeBox |