

---

# **EAI-Low Level Design Document**

**Version No.:**

**Date:**

**Project Name:**

**Project Code:**

<Address>

---

### Revision History

Version No	Date	Prepared by / Modified by	Significant Changes
1.			

### Glossary

Abbreviation	Description

---

## **Table Of Contents**

<b>1</b>	<b>Introduction.....</b>	<b>5</b>
1.1	Identification .....	5
1.2	Glossary of Terms .....	5
1.3	Referenced Documents .....	5
<b>2</b>	<b>Design Overview .....</b>	<b>5</b>
<b>3</b>	<b>Design Description .....</b>	<b>6</b>
3.1	Business Process workflow.....	6
3.2	Business Process Modeling and Management (as applicable) .....	6
3.3	Interface Design .....	6
3.3.1	Business Logic .....	6
3.3.2	Variables.....	7
3.3.3	Transformation Rules .....	7
3.3.4	Transformer Classes (as applicable) .....	7
3.4	Message / Event Design (as applicable).....	7
3.4.1	Event/ Message Flow Diagram (as applicable).....	<b>Error! Bookmark not defined.</b>
3.4.2	Event / Message Structure (as applicable) .....	<b>Error! Bookmark not defined.</b>
3.4.3	Transactional Event / Messages (as applicable).....	<b>Error! Bookmark not defined.</b>
3.5	Error / Exceptional Handling .....	9
3.6	Activity / Class Diagrams (as applicable) .....	10
3.7	Data Mapping Information .....	11
<b>4</b>	<b>Configuration .....</b>	<b>11</b>
4.1	Adapter / Connector Configuration.....	11
4.1.1	Adapter / Connector Name.....	11
4.1.2	Adapter / Connector Details (as applicable).....	11
4.2	Application Configuration(as applicable).....	12
4.3	EAI Tool Configuration(as applicable).....	12
4.4	Third Party Tool Configuration(as applicable).....	12
<b>5</b>	<b>Customization .....</b>	<b>12</b>
5.1	Application (as applicable) .....	12
5.2	EAI Tool.....	12
<b>6</b>	<b>Assumptions .....</b>	<b>12</b>
<b>7</b>	<b>Dependencies .....</b>	<b>12</b>
<b>8</b>	<b>Cross Reference with SRD / HLD.....</b>	<b>13</b>

---

<b>9</b>	<b>Annexures .....</b>	<b>13</b>
----------	------------------------	-----------

---

## 1 Introduction

### 1.1 Identification

Project Name	
Document Name	
Document Version No.	

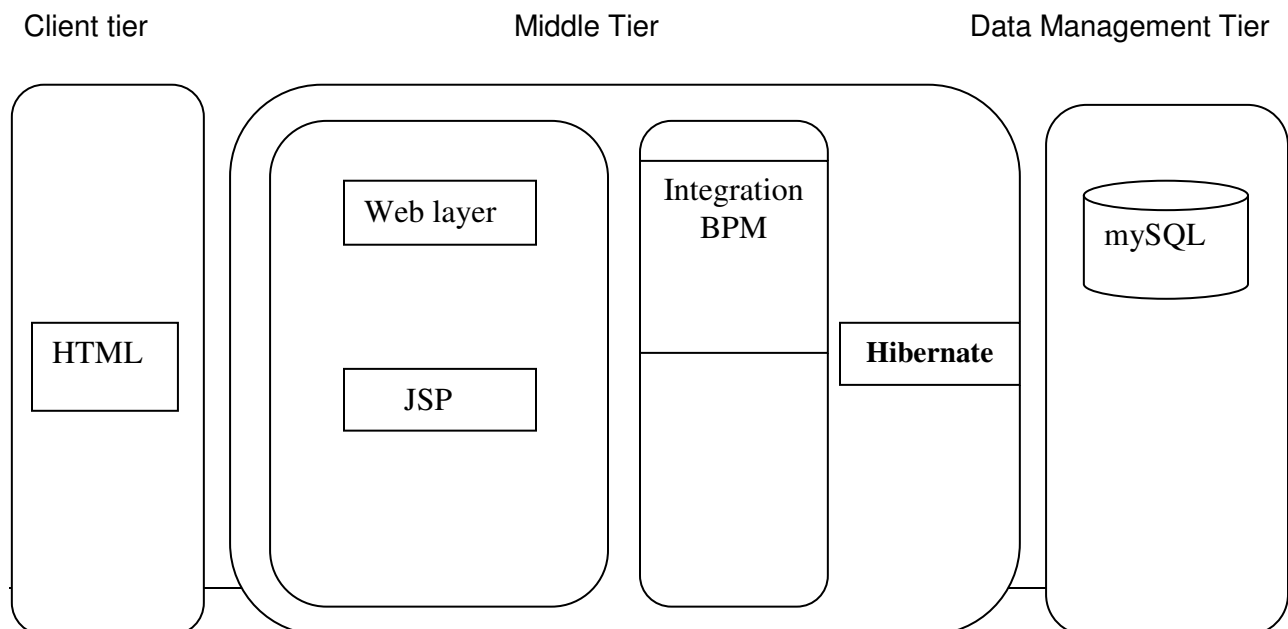
### 1.2 Glossary of Terms

### 1.3 Referenced Documents

## 2 Design Overview

The Business Process workflow used in our system can be broken down into three tiers:-

- 1) The Client Tier with the HTML and the messages that are interchanged
- 2) The middle tier which comprises of service mix and the Server with the business logic
- 3) The Back end tier with the database.



---

## **3 Design Description**

### **3.1 Business Process workflow**

Our scenario and workflow design call for our workflow to interface with the following hypothetical and actual components and applications:

The WorkFlow client used by the HCS, and Government Employee, and the normal user accessing through the web interface.

The Hibernate component that provides database access and update and change of data.

Server-side components, such as servlets, in the front-end Web application with which the customer interacts

A back-end reporting application

An e-mail client to which a confirmation message may be sent

A message passing architecture implemented using service-mix

### **3.2 Business Process Modeling and Management (as applicable)**

### **3.3 Interface Design**

#### **3.3.1 Business Logic**

---

### 3.3.2 Variables

### 3.3.3 Transformation Rules

### 3.3.4 Transformer Classes (as applicable)

## 3.4 Message / Event Design (as applicable)

To interact, in an asynchronous fashion, with other types of server-side components such as servlets or back-end applications, the Integration workflow shares data via XML documents that are published to queues and received from queues.

Thus, to communicate with many of the external and even internal components in our workflows, we will need to compose and send XML messages to either a Worklist client user, or an external or internal topic or queue. XML elements and attributes correspond to workflow variables for which values are provided at run time. We use outgoing XML documents in the sample workflows for the following purposes:

To substitute for the XML document which would be sent by a front-end Web application to trigger the new-individual registration application

To request the government to provide the result of the search performed

#### **Services**

search\_person: Gov → HCS

- input: Query
- output: set of person\_Data

Business Logic :

Search on the database to get the personal data. Receive the search parameter, send the search result.

Used By:

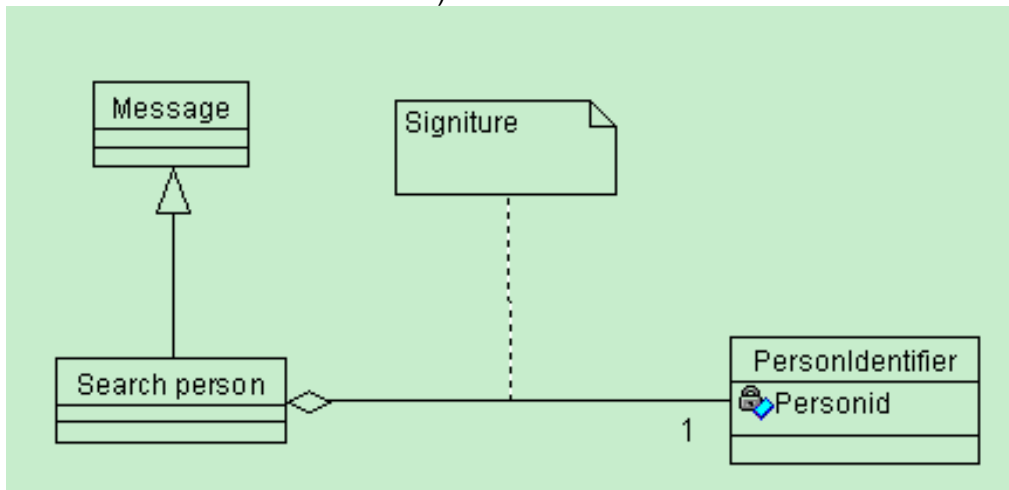
Government Clerk

Variables :

personID

The diagram above demonstrates how to represent in the UML a service that exposes a set of interfaces. By separating the notion of a service which is an actual software entity that exists at some identified end point from the interfaces which are a specification of some set of behavior we gain the flexibility to reuse common interfaces across a

number of services. In this way an interface can be defined as a stand-alone entity, versioned and managed independently from any service that implements it (in UML terms the service realizes the interface).



The diagram above demonstrates how the abstract “Message” structure is used, that the “Journal Content Request” message is a specialization that includes an element from the Journal Content Domain.

As an example, given SOAP 1.2 and a simple UML→XML serialization we could expect to see the following request message in XML. Also note that the constraint {signed} in the model results in the generation of an XML

#### ➤ Soap

```

<env:Envelope xmlns:env="http://www.w3.org/2002/12/soap-envelope">
<env:Header>
...
</env:Header>
<env:Body>
<m:SearchRequest xmlns:m="http://dit.unitn.it/schema/jrnlcnt">
<m:PersonIdentifier>
<m:Person id>. . .</m: Person id >
</m:PersonIdentifier>
<sig:Signature xmlns:sig="http://www.w3.org/2000/02/xmldsig#">
</sig:Signature>
</m:JournalContentRequest>
</env:Body>
</env:Envelope>
  
```

#### ➤ Deploy WSDL

```

<deployment xmlns="http://xml.apache.org/axis/wsdd/"
xmlns:java="http://xml.apache.org/axis/wsdd/providers/java">
<service name="Searchperson" provider="java:RPC">
<parameter name="className" value="server.Searchperson.getdata"/>
<parameter name="allowedMethods" value=""/>
  
```



---

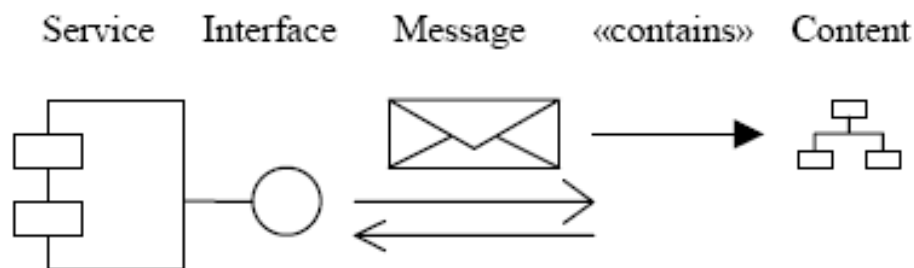
```
</service>  
</deployment>
```

➤ Publish service

deploy.bat , Axis\_Lib is the path of axis.jar :

```
set Axis_Lib=D:\workspace\test\WEB-INF\lib  
set Java_Cmd=java -Djava.ext.dirs=%Axis_Lib%  
set Axis_Servlet=http://localhost:8080/test/servlet/AxisServlet  
%Java_Cmd% org.apache.axis.client.AdminClient -l%Axis_Servlet% deploy.wsdd
```

it is important to separate out the structures that define a message passed in or out of a service action and the domain content that is used as the content of the message. The following diagram demonstrates this separation.

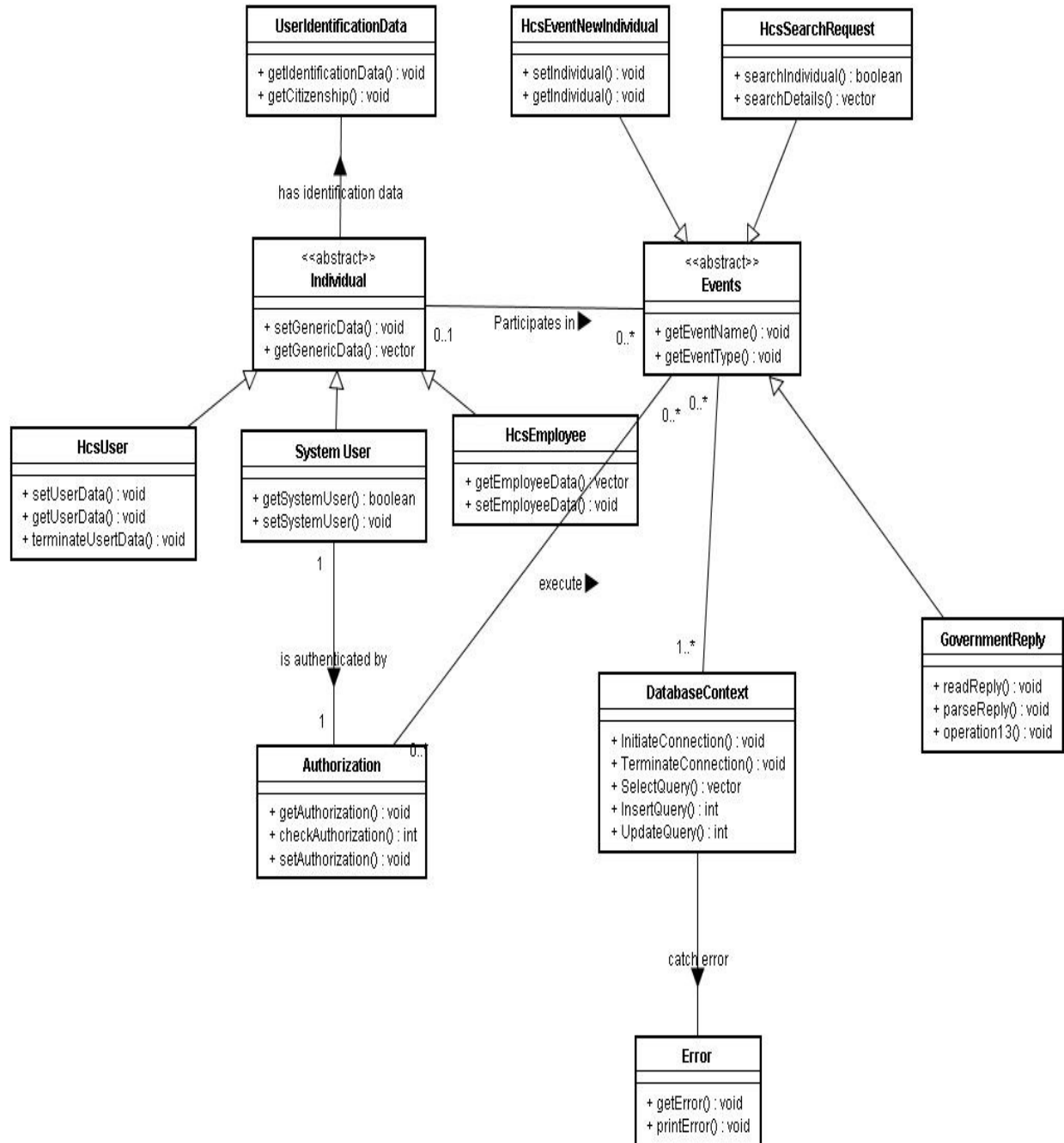


This allows us to separate information about the “instance” of a message from the content in the message, so for example the same content might be sent to two different end points but have different associated message-level information included. This separation also allows the definition of properties that are to be consumed by the HCS-Gov-System aware end-point but not necessarily by the end-user of consumer of the content. This separation also allows us to adhere to a common layered approach to the architecture that clearly defines the content relevant to each layer in a simple model as shown below.

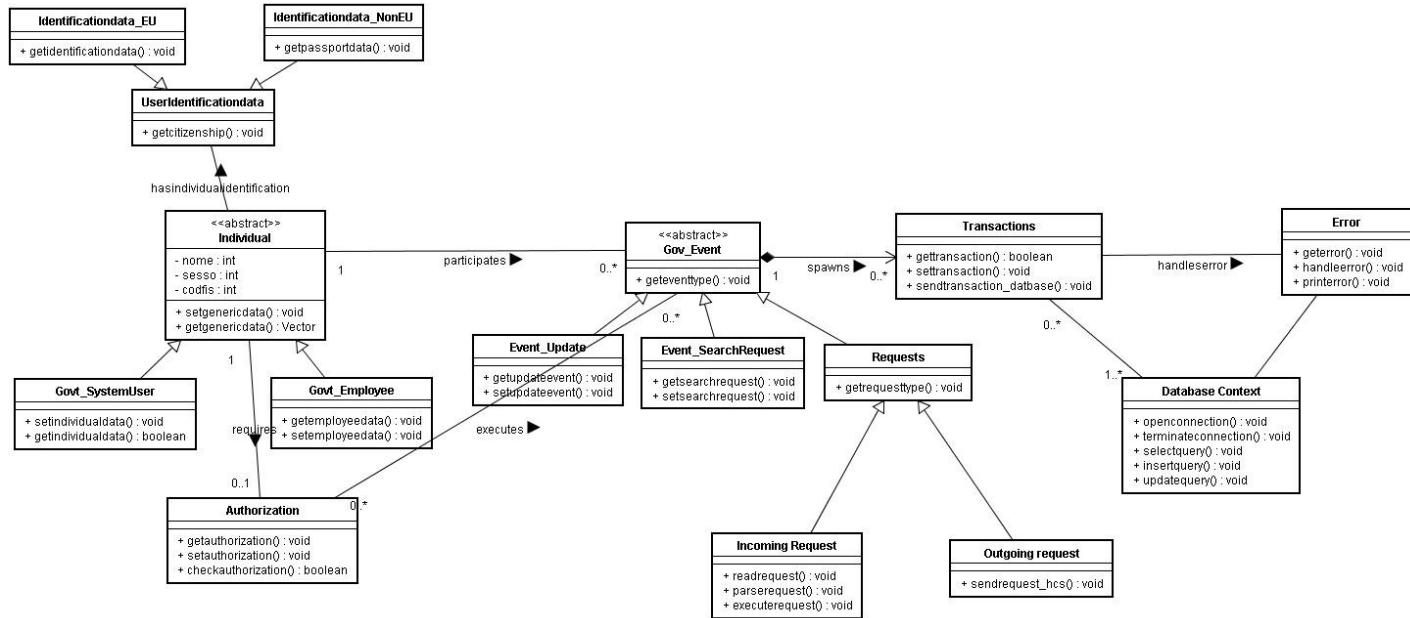
### 3.5 Error / Exceptional Handling

## 3.6 Activity / Class Diagrams (as applicable)

### 3.6.1 Class Diagram for HCS side



### 3.6.2 Class Diagram for Government side



## 3.7 Data Mapping Information

# 4 Configuration

## 4.1 Adapter / Connector Configuration

### 4.1.1 Adapter / Connector Name

### 4.1.2 Adapter / Connector Details (as applicable)

---

## 4.2 Application Configuration(as applicable)

## 4.3 EAI Tool Configuration(as applicable)

## 4.4 Third Party Tool Configuration(as applicable)

# 5 Customization

## 5.1 Application (as applicable)

Serial no	Interface identified	Available out-of-the-box in application	Application module to be used	% customization required

## 5.2 EAI Tool

Serial No	Interface Identified	Available Out-of-the-box in EAI Tool	Connector / Adaptor to be Used	% Customization Required

# 6 Assumptions

# 7 Dependencies

---

## **8 Cross Reference with SRD / HLD**

## **9 Annexures**