

SEVENTH FRAMEWORK PROGRAMME
Challenge 1
Information and Communication Technologies



Trusted Architecture for Securely Shared Services

Document type: Deliverable

Title:	D8.3 - Client / Intalio ServiceRequester ADPEP
---------------	--

Work Package: WP 8

Deliverable Number: D8.3

Editor: Michael Kutscher - University of Koblenz-Landau (DE)

Dissemination Level: Public

Preparation Date: 30 May 2009

Version: 1.2

Legal Notice

All information included in this document is subject to change without notice.

The Members of the TAS3 Consortium make no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Members of the TAS3 Consortium shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

The TAS3 Consortium

Nr	Participant name	Country	Participant short name	Participant role
1	K.U. Leuven	BE	KUL	Coordinator
2	Synergetics nv/sa	BE	SYN	Project Manager
3	University of Kent	UK	KENT	Partner
4	University of Karlsruhe	DE	KARL	Partner
5	Technische Universiteit Eindhoven	NL	TUE	Partner
6	CNR/ISTI	IT	CNR	Partner
7	University of Koblenz-Landau	DE	UNIKOLD	Partner
8	Vrije Universiteit Brussel	BE	VUB	Partner
9	University of Zaragoza	ES	UNIZAR	Partner
10	University of Nottingham	UK	NOT	Partner
11	SAP research	DE	SAP	Partner
12	Eifel	FR	EIF	Partner
13	Intalio	FR	INT	Partner
14	Risaris	IR	RIS	Partner
15	Kenteq	BE	KETQ	Partner
16	Oracle	UK	ORACLE	Partner
17	Custodix	BE	CUS	Partner
18	Medisoft	NL	MEDI	Partner

Contributors

	Name	Organisation
1	Michael Kutscher	University of Koblenz-Landau (DE)
2	Alex Boisvert	Intalio
3		
4		
5		

Table of Contents

1	EXECUTIVE SUMMARY	5
2	INTRODUCTION	7
2.1	SCOPE, OBJECTIVES	7
2.2	STRUCTURE OF THIS DOCUMENT	7
3	SERVICEREQUESTER ADPEP	8
3.1	SERVICES PROVIDED BY THE SERVICEREQUESTER ADPEP-COMPONENT	8
3.1.1	Functions realized	8
3.1.2	Interfaces	8
3.1.3	Pre- and Post-Conditions	8
3.2	ARCHITECTURE OF THE SERVICEREQUESTER ADPEP-COMPONENT	8
3.2.1	Architecture	8
3.2.2	Technologies Used	11
3.3	INTEGRATION	11
3.3.1	Business process Parts for a demonstrator	11
3.3.2	First Approach of the implementation	12
3.4	LIMITATIONS AND KNOWN ISSUES	17
3.4.1	Limitations	17
3.4.2	Known Issues	17
4	ROADMAP FOR FUTURE RELEASES	18
4.1	ENHANCEMENT OF THE INTALIO SERVICEREQUESTER ADPEP	18
4.2	INTALIO SERVICEREQUESTER ADPEP VERSION 2 COMPONENT	18
4.2.1	Services Provided by the Intalio ServiceRequester ADPEP V2 Component	18
4.2.2	Functions planned	19
4.2.3	Interfaces	19
4.2.4	Pre- and Post-Conditions	19
4.2.5	Architecture of the ADPEP V2 Component	19
4.2.6	Technologies Used	19
4.2.7	Integration	19
4.2.8	Known Issues	19
4.3	TEST CLIENT	20
4.4	INTALIO INDEPENDENT CLIENT	20
5	ANNEX	21
5.1	INTALIO BPMS: TAS3 RELATED AND COMPLEMENTARY CAPABILITIES	21
5.1.1	Upload and Download Capabilities	21
5.1.2	Support for User Interface Technologies	22
5.1.3	Integration with Existing Single Sign-On Solutions	24
5.1.4	WS-Security Support	24
5.2	INTERNAL DATA FIELDS PROVIDED BY THE INTALIO BPMS	25
6	GLOSSARY	27
7	DOCUMENT CONTROL	28

Table of Figures

Figure 1: TAS3-architecture with relevant parts for the ServiceRequester ADPEP.....	9
Figure 2: Kenteq-APL business process with focus on 'import of existing PCP-file'	12
Figure 3: Intalio ServiceRequester ADPEP-collector-part for "import of existing PCP-file"	13
Figure 4: Start of the upload process	14
Figure 5: Login into the user interface of the Intalio BPMS.....	14
Figure 6: Selection of the provided upload-form	15
Figure 7: Completion of the upload-form	15
Figure 8: Notification with data from the upload form	16
Figure 9: Drag & drop of controls on a workflow form	22
Figure 10: Example form built with Xforms technology	23
Figure 11: Same example form displayed in Liferay Portal	24

1 Executive Summary

This document provides the description of the client side of the TAS3 Trusted Architecture for Securely Shared Services and its connection to the core components of the infrastructure (the TAS3 stack). The client side is also called the ServiceRequester side and the connection from the client to the TAS3 stack is called the ServiceRequester ADPEP (Application Dependent Policy Enforcement Point). As well as the general concepts for a ServiceRequester ADPEP also the first implementation of a ServiceRequester ADPEP specific adapted for Intalio|BPMS (Intalio ServiceRequester ADPEP) is presented. By using the ServiceRequester ADPEP the Intalio|BPMS and the connected web browser provide the functionality of a TAS3-client for uploading and retrieving data from a TAS3 ServiceProvider. Actually the ServiceRequester ADPEP and the Intalio ServiceRequester ADPEP do not cover security functions.

The ServiceRequester ADPEP in general is the application dependent part of the TAS3 infrastructure on the "ServiceRequester side". Its purpose is to provide functions to connect the client-application to the TAS3 infrastructure for uploading and retrieving data. These functions are needed to realize the pilots of WP9 (Employability and Healthcare Demonstration) in the fields of employability and eHealth. The TAS3 Trusted Application Infrastructure depends on the requirements collected in WP1 (Requirements Analysis), the architecture design provided by WP2 (Framework, Architecture and Semantics) and the secured business process models developed in WP3 (Securely Adaptable Business Processes) in conjunction with WP9. It allows the application independent services developed in WP4 (Information Protection), WP5 (Trust Policy Management) and WP7 (IDIJ Authentication Authorization) to be used in the WP9 pilots. As technical basis for the demonstrators the Intalio|Designer and Intalio|BPMS from the project partner Intalio is used. Therefore the first implementation of a ServiceRequester ADPEP is created for the Intalio|BPMS and is called the 'Intalio ServiceRequester ADPEP'. A short description of the Intalio|Designer and the Intalio|BPMS is provided in the deliverable D3.1.

Within the TAS3 trusted application infrastructure the ADPEP-components provide the application dependent connection from the ServiceRequester and the ServiceProvider to the core components of the TAS3-infrastructure. Therefore two different types of ADPEPs were designed: the ServiceRequester ADPEP (described in this document) at the ServiceRequester side and the ServiceResponder ADPEP (described in the deliverable D8.1) at the ServiceProvider side.

Later on in this document the term *ADPEP* is used for the *ServiceRequester ADPEP*.

This document describes:

- the assumptions on which the development of the ServiceRequester ADPEP-component is based
- the functions provided by ServiceRequester ADPEP
- the architecture of ServiceRequester ADEP
- the first implementation of a ServiceRequester ADPEP for Intalio|BPMS (Intalio ServiceRequester ADPEP)
- limitations and known issues
- Roadmap for future releases of the component (ServiceRequester ADPEP / Intalio ServiceRequester ADPEP / Intalio|BPMS, Clients) in the next phases of the TAS3 project.

NOTE: The software components produced in WP 8 implement application specific adaptors that are required to use the application independent TAS3 infrastructure in the TAS3 pilots in eHealth and eEmployability. The overall architecture, semantically enriched executable business process models with an XForms user interface and the design of the core TAS3 services are the pre-requisites for WP8. The TAS3 architecture has been finalized only recently (see D2.1). While all our results are consistent with and usable for the current TAS3 architecture it cannot be a surprise that some alignments and refinements will be required. Moreover new service needs, for example the request for a service bus, are emerging from the architecture document, which still need to be detailed before they can be implemented and documented in any of the deliverables of WP8.

Technical Note:

All produced components (web services, libraries and clients) of Deliverables D8.1., D8.2. and D8.3. can be found in a binary version at this location:

<http://citrix.uni-koblenz.de:9000/homepage/tas3/default.aspx>

To access this page, please use the following login data:

Login: tas3

Password: z65rf5

2 Introduction

2.1 Scope, Objectives

The focus of the deliverable D8.3 is an Intalio|BPMS based client for TAS3 and the Intalio specific Application Dependent Policy Enforcement Point (ADPEP) on the ServiceRequester side of the TAS3 infrastructure. The ServiceRequester side represents the application people use to work with data which is processed and stored on the ServiceProvider side in a (e.g. fedora-) database. As an interface the ADPEP provides the connection from “existing” applications to the Application Independent Policy Enforcement Point and by that a secured and trusted channel - to the innovative TAS3 infrastructure. The first implementation of a ServiceRequester ADPEP is created for the Intalio|BPMS and is called the ‘Intalio ServiceRequester ADPEP’. By using the ServiceRequester ADPEP the Intalio|BPMS and the connected web browser act as a TAS3 client. The first implementation of this Intalio based TAS3 client is described in this document.

2.2 Structure of this document

Chapter 3 provides detailed information about the ServiceRequester ADPEP-component, the Application Dependent Policy Enforcement Point on the ServiceRequester side as part of a business process and the connection to the TAS3 infrastructure. It contains information about functions, interfaces and the architecture of the component and its main parts (collector, converter). Also there is a description of the first implementation of an Intalio ServiceRequester ADPEP for the Intalio|BPMS (Business Process Management System) and the sample implementation (uploading a file from a process running on the Intalio|BPMS) for one part of the Kenteq-APL process which is provided by WP3 in conjunction with WP9.

Chapter 4 provides a Roadmap for future releases of the planned integration of the Intalio ServiceRequester ADPEP as a module/function in future releases of the Intalio|BPMS, the enhancements of the Service Requester ADPEP in general and its further integration in the TAS3 infrastructure. As well there are mentioned other possibilities to access data via the TAS3 infrastructure (testclient).

Chapter 5 / Annex includes a section that describes capabilities of the Intalio|BPMS that relate to and complement the TAS3 infrastructure in providing broader solutions to the problem domain that TAS3 addresses.

Chapter 6 / Glossary provides a list of abbreviations and the corresponding full text.

Chapter 7 / Document History provides an overview about the changes of this document.

Gender generalization: Wherever the male form (he/him/his) is specified the female is also implied!

3 ServiceRequester ADPEP

3.1 Services Provided by the ServiceRequester ADPEP-Component

3.1.1 Functions realized

The ServiceRequester ADPEP-Component (Application Dependent Policy Enforcement Point) provides the connection between the business process/application of the ServiceRequester and the particular ServiceProvider(s). A further implementation of the ADPEP will provide these connections via the TAS3 infrastructure. Therefore the ServiceRequester ADPEP is able to find a ServiceProvider appropriate for the particular request and trust requirements. The ADPEP also collects data (security, environmental data) which is necessary to fulfill the requirements of the ServiceProvider for the request and the security issues. Also the ADPEP-component provides the status information and the payload data provided by the ServiceProvider for the application of the ServiceRequester. Actually the two ADPEPs (one on the ServiceRequester side and one on the ServiceResponder / ServiceProvider side) communicate directly with each other. Further implementations will connect to the corresponding components of the TAS3 infrastructure (mainly the AIPEPs) which have to be developed in the next phases.

3.1.2 Interfaces

The ServiceRequester ADPEP-component has three interfaces. One is to the business process/the application of the ServiceRequester. From this side the ADPEP gets the request for data and other necessary information (security, environmental data and payload) and later on delivers the retrieved information (e.G. the required payload data). The second interface is to the TAS3 infrastructure or to be more in detail the corresponding AIPEP (Application Independent Policy Enforcement Point). The ServiceRequester ADPEP delivers appropriate formatted data (request, security information, data) to the AIPEP and gets back the delivered information via the AIPEP. Another Interface is towards the Trust Negotiator. This component of the TAS3 infrastructure gets a request/the requested service and the minimum trust level and sends back a list of trustworthy endpoints / ServiceProviders which offer the needed service at the specified trust level or higher.

3.1.3 Pre- and Post-Conditions

The collector-part of the ServiceRequester ADPEP-component is application dependent in particular and for the Intalio ServiceRequester ADPEP realized as a small business process which has to be integrated in the overall business process. Therefore an appropriate business process and business process server have to be available. Although the reference implementation uses the Intalio|BPMS other business process server like Apache ODE (open source version of the Intalio|BPMS) might be used. Also it is possible to implement simple applications that provide the necessary functionality (collect request, security and environmental data and call the AIPEP) to access the TAS3 infrastructure.

3.2 Architecture of the ServiceRequester ADPEP-Component

3.2.1 Architecture

The following diagram shows the integration of the ServiceRequester ADPEP (Application Dependent Policy Enforcement Point) in the overall TAS3-architecture and the relevant modules.

The left side of the TAS3-infrastructure is called the ServiceRequester. The right side of the TAS3-infrastructure is called the ServiceProvider. The modules between the ServiceRequester and the ServiceProvider represent the core TAS3-infrastructure.

The ADPEPs are application-dependent and provide the connection between the application (actually provided by the Intalio|BPMS on the ServiceRequester side and the fedora repository on the ServiceProvider side) to the application-independent parts the TAS3-

infrastructure. The interface to these application independent parts of the infrastructure is the AIPEP (Application Independent Policy Enforcement Point).

The ADPEP on the ServiceRequester side consists of **two major parts**:

- ➔ **the collector-part and**
- ➔ **the converter-part.**

The collector part of the ADPEP collects the request, the payload and the necessary data for the authorization-check of the request by the PDP (Policy Decision Point). The PDP is part of the TAS3 infrastructure.

The following figure shows the main components which are relevant for understanding the function of the ADPEP-component. The ServiceRequester and the ServiceRequester ADPEP are surrounded by a red box.

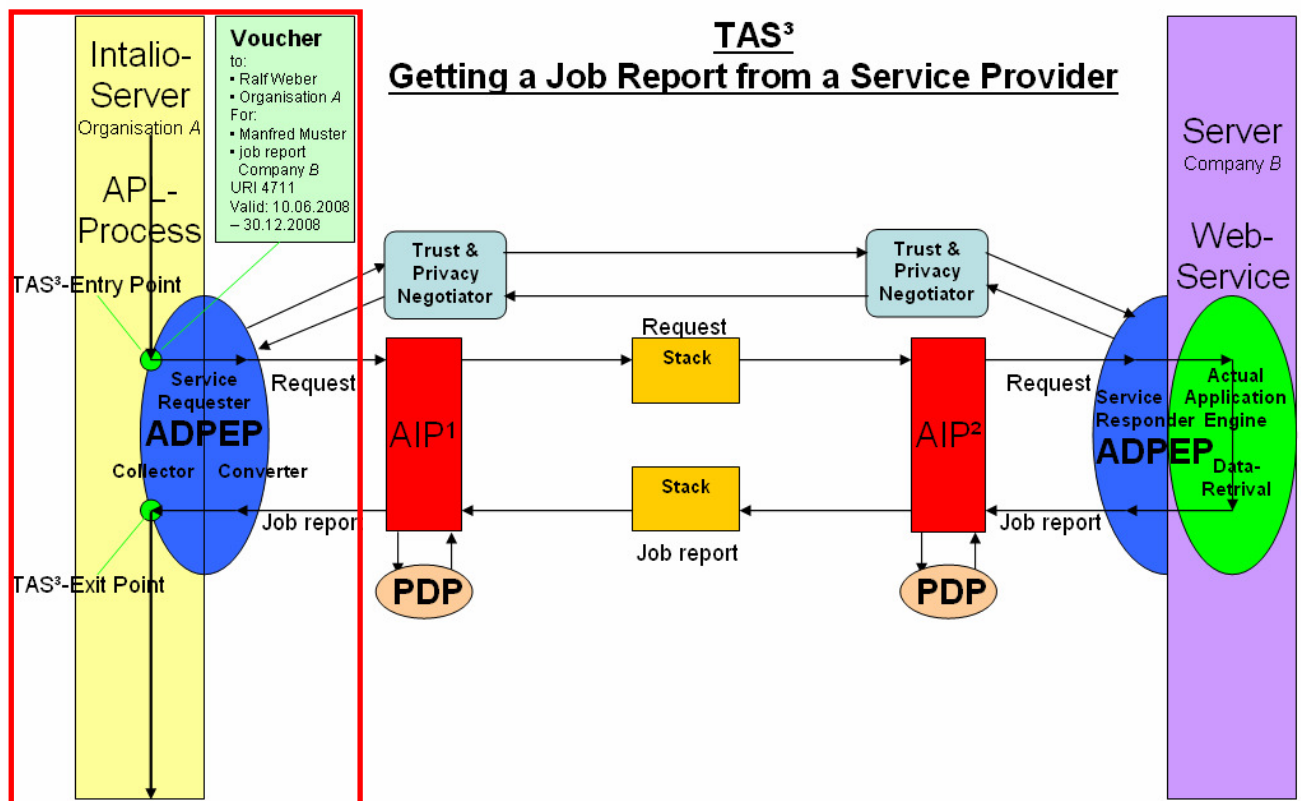


Figure 1: TAS3-architecture with relevant parts for the ServiceRequester ADPEP

The ADPEP is application dependent and provides the connection from the application on the ServiceRequester side via the core TAS3-Infrastructure to the ServiceProvider side.

There are three possible hooks where the ADPEP can be integrated into the application / the application server:

1. Integration into the application / Business-Process (Business Application-Level)
In this case the collector-Part of the ServiceRequester ADPEP is implemented inside of the application. For the sample implementation with the Intalio|BPMS this means the BPMN-representation of the entire business-process. The connection to the core TAS3-infrastructure is implemented as a web service.
2. Integration in Server-Application (Application-Server-Level)
in this case the ServiceRequester ADPEP is implemented as a module at the level of the Application-engine (in the sample implementation the Intalio|BPMS).
3. Integration on Application-Server (Operating system level)
in this case the ServiceRequester ADPEP is implemented as a module on the operating

system level. Perhaps it could be realized as a driver between the OS and the network adapter. This implementation might be very complex but would be the best possibility for the integration of unmodified legacy-Applications.

In the first approach of the TAS3-project the hook number one will be realized. As the basic application we actually use a business process and the Intalio|BPMS. The Intalio ServiceRequester ADPEP is build as an additional part of the business process from the Kenteq APL-process.

In the second approach the Intalio ServiceRequester ADPEP will be implemented inside the Intalio|BPMS. (This is described in chapter 3 of this document.)

The ServiceRequester ADPEP-component consists of two parts:

1. the collector-part
2. the converter-part

The function of the collector-part is to:

- request a trust level from the user, send this data with the request to the trust negotiator and receive the list of URIs with appropriate service-providers.
- accept the request and payload
- collect the relevant environment- and security-Data (direct from the Server or via xforms) and
- transfer this data to the converter
- receive status information and/or data from the converter and provide it to the business process / application.

In the first approach the collector-part for the Intalio ServiceRequester ADPEP is realized as an additional part of the business process.

It collects the necessary data from the server / Intalio|BPMS. The following list shows the data to be collected for an upload of data/payload:

- Request
- User-Identification
- Role
- Process-Identification
- Server-Identification
- Credential from the user (Upload)
- Policy (Upload)
- Payload/data (Upload)
- Timestamp

The converter is a Web-Service and provides the following functions:

- It asks the TPN (Trust and Privacy Negotiator) for a ServiceProvider fitting the actual request and trust-level.
- It converts the request, security- and environment-data collected by the collector-part of the ADPEP into a appropriate format for the AIPEP (Application Independent PEP). Actually the appropriate format for the security-data is XACML.
- transmit this data to the AIPEP for further execution.
- Receives status information and data from the AIPEP
- Converts this information into an appropriate format for the collector-part/business process
- Sends the status information and data to the collector-part of the ServiceRequester ADPEP

In the first approach the ServiceRequester ADPEP (-collector) was implemented at the Business-Application-Level inside of a business process using the Intalio|BPMS (so called Intalio ServiceRequester ADPEP).

In the second approach it is planned that Intalio will extend its Intalio|BPMS so that the functionality of the ServiceRequester ADPEP (-collector and perhaps converter) will be integrated in the server-software. This means that a data request via TAS3 will be a simple call in the business process. Further information about this is included in the chapter 'Roadmap for future releases'.

To collect the necessary data for the AIPEP / PDP inside the business process the Intalio ServiceRequester ADPEP is able to use internal data provided by the Intalio|BPMS. For example it is possible to retrieve the name and the role of the user. A larger list with internal data fields is included in the annex.

At runtime this data can be used to complement security and environmental data which is needed for the PDP to make a proper decision whether a request is allowed or not.

3.2.2 Technologies Used

For the reference implementation we use the Intalio|BPMS on the ServiceRequester side.

The collector-part of the Intalio ServiceRequester ADPEP is build in BPMN using the Intalio|Designer for business process modeling 6.0.0.054 for Windows (free available at intalio.com).

For the processing of the business process and the Intalio ServiceRequester ADPEP (collector-part) the Intalio|BPMS 6.0.0.022 is used (Community edition free available at intalio.com). For the development of the Intalio ServiceRequester ADPEP-converter as a web service eclipse and java and Axis2 is used. The converter is processed by the Axis2-component on Linux server with Apache Tomcat V6.0.

3.3 Integration

3.3.1 Business process Parts for a demonstrator

For a demonstrator it is planned to integrate some parts of the Kenteq APL business process. The business process is provided by the work packages WP9 and WP3. For a demonstrator some tasks were selected which require access to a data store. By using TAS3 it will be possible to securely access external data provider.

The abbreviation PCP is used for *Personal Competency Profile*.

- Task RequestImportOfExistingPCP:
The Candidate is asked to upload a PCP-file into the database.
This is the *insert*-case for the TAS3-Infrastructure
 - The candidate selects the PCP-file to be uploaded into the database. Also he selects a minimum trust-level (e.g. from a drop down list) for the ServiceProvider.
 - The collector-Part of the Intalio ServiceRequester ADPEP combines the PCP-file, the request, the security- and environment-data (including the trust-level) and sends it to the converter.
 - The converter-part of the Intalio ServiceRequester ADPEP sends the trust-level and the request to the trust negotiator and receives a list with URIs of appropriate ServiceProviders.
 - The converter-part of the Intalio ServiceRequester ADPEP converts the security data and environmental data into the appropriate format for the AIPEP and sends it to the AIPEP. The payload data is send as it is.
 - The converter-Part of the Intalio ServiceRequester ADPEP gets back some status information (e.g. request rejected) from the AIPEP and sends it to the collector-Part of the Intalio ServiceRequester ADPEP.
 - The collector-Part of the Intalio ServiceRequester ADPEP provides the status information for the business process.
- Task RequestCandidateToCreate/UpdatePCP:
The candidate is asked to create or update an existing PCP-file from the database.
This is the *select/read* and *update*-case for the TAS3 infrastructure
 - The candidate selects his PCP and the appropriate ServiceProvider.

- The collector-Part of the Intalio ServiceRequester ADPEP combines the PCP-file, the request, the security- and environment-data (including the trust-level) and sends it to the converter.
- The converter-part of the Intalio ServiceRequester ADPEP converts the security data and the environmental data into the appropriate format for the AIPEP and sends it to the AIPEP.
- The converter-Part of the Intalio ServiceRequester ADPEP gets back from the AIPEP the PCP or some status information and sends this to the collector-Part of the Intalio ServiceRequester ADPEP.
- The collector-Part then provides the PCP or the status information to the business process.
- After the modification of the PCP-file the update-function can be processed
- The collector-Part combines the modified PCP-file, the (update-) request and the security- and environment-data and sends it to the converter.
- The converter-part of the Intalio ServiceRequester ADPEP converts the security data and environmental data into the appropriate format for the AIPEP and sends it to the AIPEP.
- The converter-Part gets back some status information from the AIPEP and sends it to the collector-Part of the Intalio of the Intalio ServiceRequester ADPEP.
- The collector-Part of the Intalio of the Intalio ServiceRequester ADPEP provides the status information to the business process.

3.3.2 First Approach of the implementation

As a first approach the Task "RequestImportOfExistingPCPData" was selected from the actual model of the Kenteq-APL business process. It is the first task within the process where access to an external data provider is needed and represents the first of the essential functions concerning data (insert, update, select, delete). Within this task the candidate has to upload his PCP (Personal Competency Profile) which is stored in a zip-file on his client computer. The following figure 2 (screenshot of the used BPM-Tool Intalio|Designer) shows the mentioned part of the actual Kenteq-APL business process in BPMN.

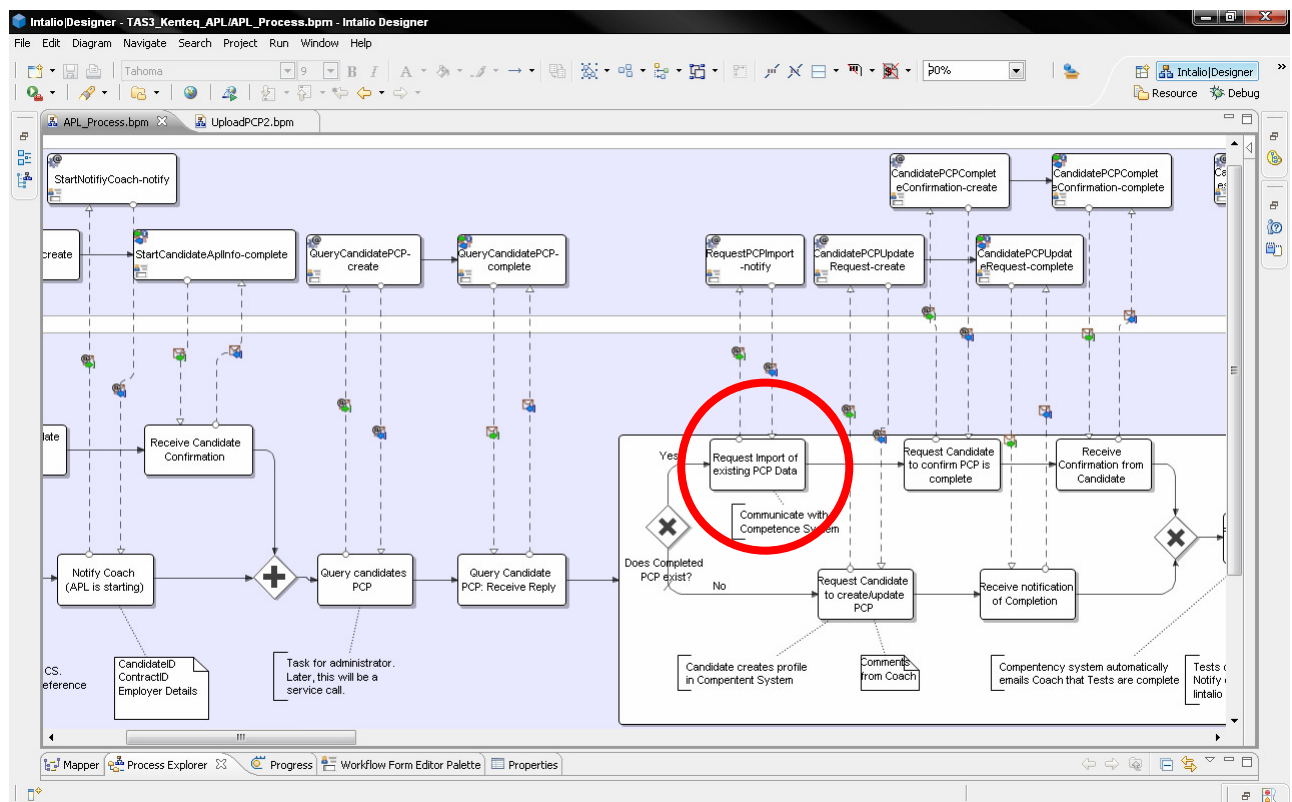


Figure 2: Kenteq-APL business process with focus on 'import of existing PCP-file'

The following diagram shows the actual implementation of the Intalio ServiceRequester ADPEP-collector-part for the upload-process.

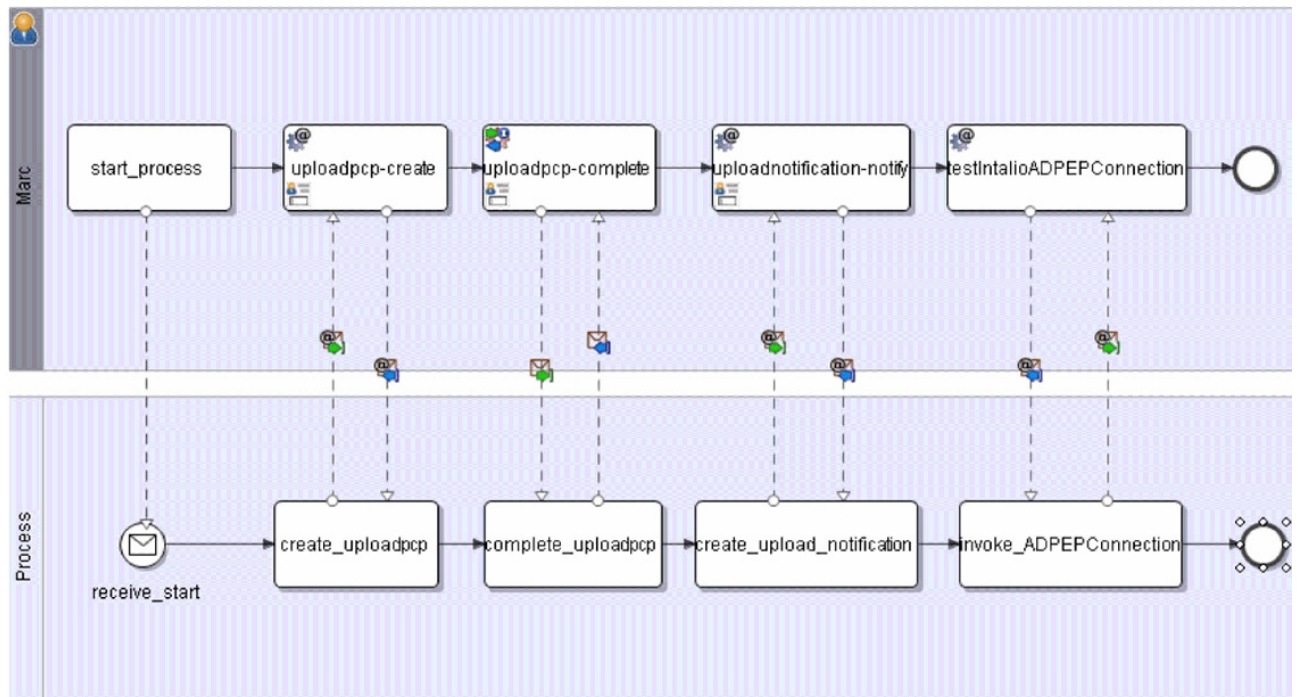


Figure 3: Intalio ServiceRequester ADPEP-collector-part for "import of existing PCP-file"

After starting the process the needed form 'uploadpcp' is generated by the Intalio|BPMS. The user accesses the form fills in the needed data and completes the form. With the notification form generated by the Intalio|BPMS the user is able to check the data that is collected from the server and the 'uploadpcp'-form and that will be sent to the TAS3 infrastructure. Therefore in the task 'invoke_ADPEPConnection' the web service that represents the converter-part of the Intalio ServiceRequester ADPEP is called.

The following section shows a test run using the upload scenario of the Kenteq-APL process. The candidate wants to upload an existing PCP (Personal Competency Profile). "His" APL-process is at the point where the candidate was asked whether a PCP already exists and he answered "yes". The PCP is located as a zip-file on the local PC of the candidate.

The Upload-process is implemented as a test-process and was deployed from the Intalio|Designer to the Intalio|BPMS:

The upload-process is started by the administrator as shown on the next figure 4. For that the administrator uses the Intalio|BPMS server console. He selects the appropriate process (*UploadPCPProcess*) and presses the *start*-button. In the next phase of the project the upload-process is integrated into the higher business process (e.g. the Kenteq APL process) and doesn't have to be started separately.

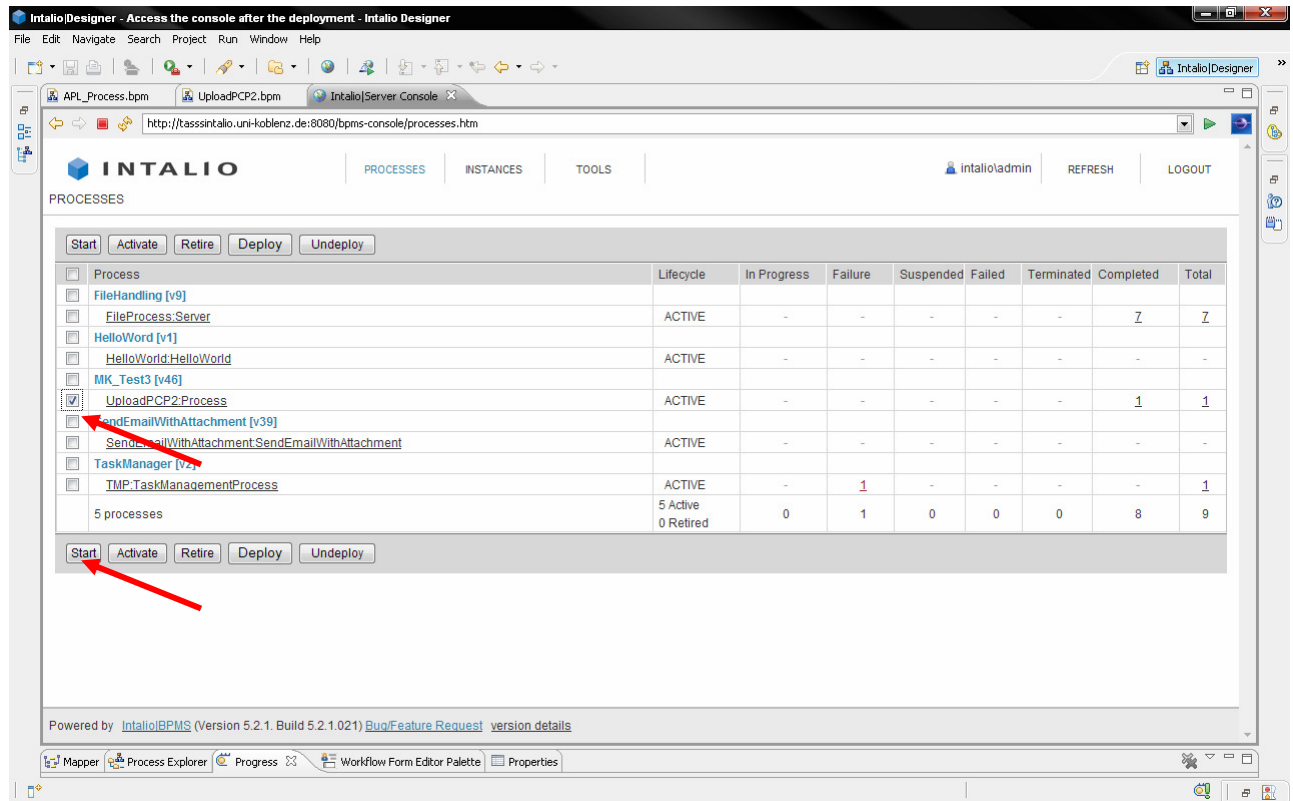


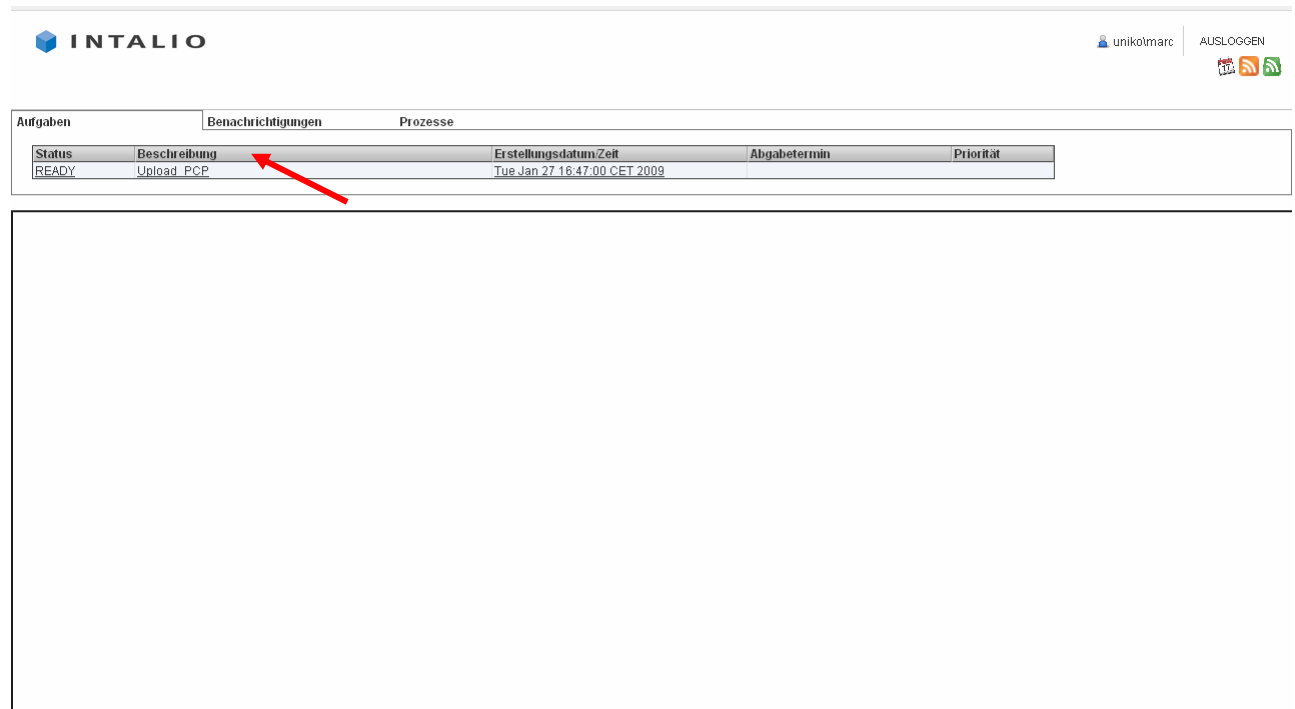
Figure 4: Start of the upload process

The candidate now logs in the user web interface of the Intalio|BPMS. For that he can use a standard web-browser. Actually the users and roles are located in a file on the Intalio|BPMS. Later versions of the Intalio|BPMS will provide access to several single sign on-applications.



Figure 5: Login into the user interface of the Intalio|BPMS

The candidate selects the forthcoming form *Upload_PCP* which is already provided by the running upload process.



The screenshot shows the Intalio web interface. At the top, there is a navigation bar with the Intalio logo, a user profile 'unikolmarc', and a link to 'AUSLOGGEN'. Below this is a table with three tabs: 'Aufgaben', 'Benachrichtigungen', and 'Prozesse'. The 'Aufgaben' tab is active, displaying a table with the following data:

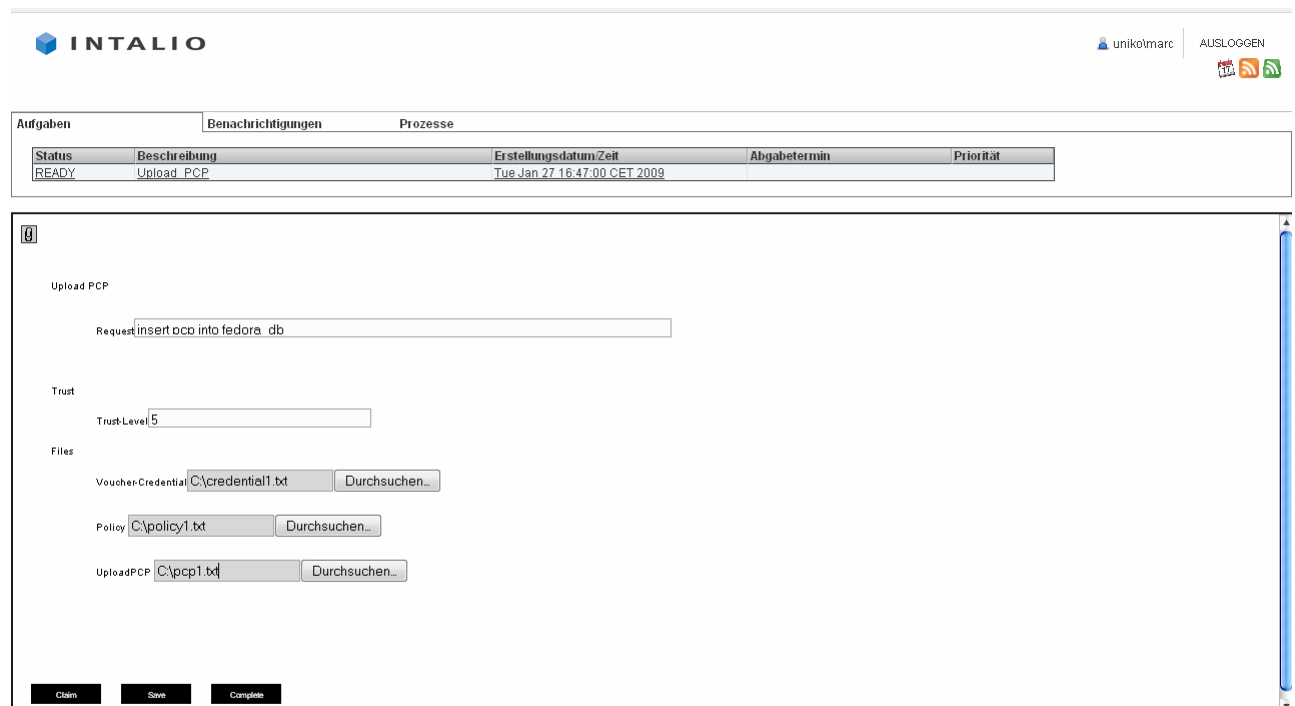
Status	Beschreibung	Erstellungsdatum/Zeit	Abgabetermin	Priorität
READY	Upload_PCP	Tue Jan 27 16:47:00 CET 2009		

A red arrow points to the 'Upload_PCP' task in the table.

Powered by [IntalioBPMS \(Version 5.2.1 Build 021\)](#) [Bug/Feature Anfrage](#)

Figure 6: Selection of the provided upload-form

Now the candidate fills in the missing data in the form (minimum trust level for a ServiceProvider) and he selects the necessary files for the upload.



The screenshot shows the Intalio web interface with the 'Upload_PCP' form. The form contains the following fields and buttons:

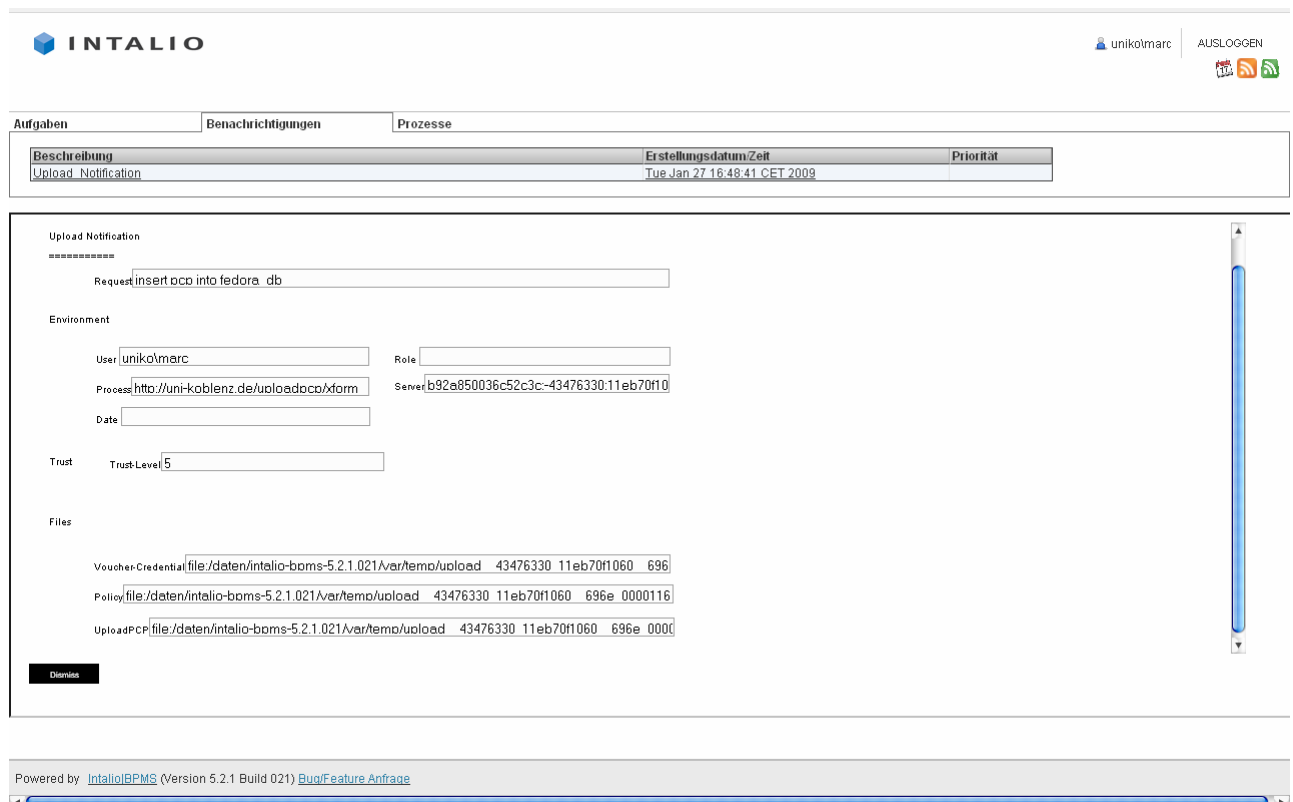
- Request:** A text input field containing 'insert pcp into fedora_db'.
- Trust:** A text input field containing '5'.
- Files:**
 - Voucher-Credential:** A text input field containing 'C:\credential1.txt' and a 'Durchsuchen...' button.
 - Policy:** A text input field containing 'C:\policy1.txt' and a 'Durchsuchen...' button.
 - UploadPCP:** A text input field containing 'C:\pcp1.txt' and a 'Durchsuchen...' button.
- Buttons:** 'Claim', 'Save', and 'Complete' buttons at the bottom.

Powered by [IntalioBPMS \(Version 5.2.1 Build 021\)](#) [Bug/Feature Anfrage](#)

Figure 7: Completion of the upload-form

These files are the *voucher/credential* which later on will prove to the PDP that the candidate has the right to upload that file, the *policy* which contains rules for the further treatment and access to the data and the *PCP-file* which contains the Personal Competency Profile of the candidate. In this case the PCP is the payload for the upload-Task.

In the actual version of this process a notification is available which contains the information that is collected by the form and from the Intalio|BPMS and which is sent to the converter-part (web-service) of the Intalio of the Intalio ServiceRequester ADPEP-component. This notification is for debugging only and will not be part of the further implementation of Intalio of the Intalio ServiceRequester ADPEP-component or the Kenteq APL-process.



INTALIO unikolmarc AUSLOGGEN

Aufgaben **Benachrichtigungen** **Prozesse**

Beschreibung	Erstellungsdatum/Zeit	Priorität
Upload Notification	Tue Jan 27 16:48:41 CET 2009	

Upload Notification

Request:

Environment

User: Role:

Process: Sender:

Date:

Trust

Trust-Level:

Files

Voucher-Credential:

Policy:

UploadPCP:

Powered by [Intalio|BPMS \(Version 5.2.1 Build 021\)](#) [Bug/Feature Anfrage](#)

Figure 8: Notification with data from the upload form

The converter-part is a web service which is reachable by a given web service interface (e.g. using WSDL). This web service is able to

- find a service, which can answer to the user's request
- Negotiate with a so-called 'Trust and Privacy Negotiator', whether the offered service(s) is (are) trusted at the minimum trust-level or not.
- prepare the message by invoking a so called 'Message Preparer' service.
- Fetch information about the required credentials, which are needed to access the system that is connected via TAS3.

Combined this web service converts the data (security and environment) provided by the collector part in the appropriate form for the AIPEP (Application Independent Policy enforcement point) and to transfers the files to the AIPEP/TAS3-infrastructure. In the actual phase of the project the converter only accepts the request, the trust-level and the PCP as a zip-file. Due to missing components the actual target of the converter part of the ADPEP is not the AIPEP but it is the corresponding Service Responder ADPEP on the ServiceProvider side where the fedora repository processes the request.

3.4 Limitations and Known Issues

3.4.1 Limitations

Actually some components of the TAS3-Infrastructure such as AIPEP, PDP, ... are not yet realized. Therefore the two ADPEPs on both sides (ServiceRequester-side and the ServiceProvider-side) directly communicate with each other. This is necessary to be able to show a complete data transfer between these two sides. In the further phases of the project the communication between the ADPEPs will be realized via other TAS3-components such as the AIPEP (Application Independent PEP) and the PDP (Policy Decision Point) which provide functions for application independent authorization and communication to trustworthy partners via the TAS3-infrastructure.

Actually the ServiceRequester ADPEP does not cover security functions. For example the communication between the components is not encrypted yet. This would make it possible to track the information between the Intalio|BPMS and the TAS3-infrastructure and back.

3.4.2 Known Issues

None

4 Roadmap for future releases

4.1 Enhancement of the Intalio ServiceRequester ADPEP

In the second phase of the TAS3-project the collector-part of the Intalio ServiceRequester ADPEP-component will be integrated in the Intalio|BPMS. This is described in the next section. The converter-part of the Intalio ServiceRequester ADPEP -component will also be enhanced to process additional data from the collector-part. In the first approach only request and filename are processed by the converter. In the second approach the converter-part will have to process the voucher/credential- and policy-files as well as the trust- and environment-data. In the first approach the request and files are passed directly to the corresponding ServiceResponder ADPEP of the ServiceProvider. In contrast to that in the second approach the ServiceRequester ADPEP will contact the ServiceResponder / ServiceProvider not directly but via other TAS3-components – the AIPEPs (Application Independent PEP). These components provide – in cooperation with the PDP-components (Policy Decision Point) the authorization-check for the request and then forward request and data to the AIPEP – and later on to the ServiceResponder ADPEP - of the corresponding ServiceProvider. Also a database for policy handling will be attached to the Intalio ServiceRequester ADPEP.

The following table shows the different operations that will be provided by the Intalio ServiceRequester ADPEP and the data which has to be collected and transmitted to the AIPEP. The column 'result' shows the data that will be returned by the AIPEP to the ServiceRequester ADPEP.

Operation	Request	Environment	Trust & Negot.	Voucher / Credential	Policy	Upload-File	Result
Datatype	String	String(s)	Int (0-10)	File	File	File	
Insert Data	+	+	+	+	+	+	Status, URI
Delete Data	+	+	+	+	-	-	Status
Update Data	+	+	+	+	+	+	Status
Select Data	+	+	+	+	-	-	Status, Data

Insert policy	+	+	+	+	+	-	Status, URI ?
Delete policy	+	+	+	+	-	-	Status
Update policy	+	+	+	+	+	-	Status
select policy	+	+	+	+	-	-	Status, Data

Request T&N	+	+	+ TrustLevel	+	?	-	Status, URI
-------------	---	---	--------------	---	---	---	-------------

4.2 Intalio ServiceRequester ADPEP Version 2 Component

4.2.1 Services Provided by the Intalio ServiceRequester ADPEP V2 Component

In this chapter the phrase ADPEP V2 is used for the second realisation of the Intalio ServiceRequester ADPEP V1 component (Application Dependent Policy Enforcement Point) as an integrated module in the Intalio|BPMS.

4.2.2 Functions planned

The Intalio ServiceRequester ADPEP V2 is functionally equivalent to the ADPEP V1 component: it provides the connection between the application of the ServiceRequester and the ServiceProvider. The ADPEP V2 improves on the ADPEP V1 architectural aspects by integrating directly into the Intalio|BPMS instead of being developed as logic in a BPEL business process. This should result in better software design quality characteristics since Java language has better abstraction, composition and extensibility mechanisms than the current BPEL language offers, as well as higher performance since native Java code executes faster than interpreted BPEL code. The design tradeoff is that the ADPEP V2 would not be directly described through BPMN and BPEL and thus would be more opaque from the perspective of a business analyst. However, the ADPEP V1 design could be maintained for the purpose of documenting and communicating the ADPEP logic to non-programmers.

4.2.3 Interfaces

Just like the ADPEP V1, the ADPEP V2 will have three interfaces:

1. one to the application of the ServiceRequester (e.g. business process) where it processes requests on behalf of the application;
2. one to the TAS3 infrastructure, invoking the Application Independent Policy Enforcement Point (AIPEP);
3. one to the TrustAndPrivacyNegotiator to determine a ServiceProvider that is able to fulfill the application request and also satisfies the trustworthiness requirements.

4.2.4 Pre- and Post-Conditions

Beyond the state held during the processing of a given request, the ADPEP V2 is a stateless component. As such, it will require a certain amount of contextual information, such as security policies to be available when new requests are initiated. The current assumption is that all contextual information is carried together with the initial request, either as additional message headers or separate from the message but still available to the ADPEP V2 at the time the request is received (e.g. as message context).

4.2.5 Architecture of the ADPEP V2 Component

The architecture is the same as the ADPEP V1 component except the component is implemented at a lower level, directly in the Intalio|BPMS instead of being implemented in a BPEL process.

4.2.6 Technologies Used

Java 1.5+, Axis2 1.3+, Apache Ode 2.0+

4.2.7 Integration

The ADPEP V2 will be integrated in at the Integration Layer1 level of the Intalio|BPMS. There are currently three integration layer implementations available in Apache Ode: Axis2, JBI and SCA. The integration layer chosen for the ADPEP V2 is Axis2 because this is the default integration layer used in Intalio|BPMS.

The Axis2 architecture is extensible² and supports extensions through modules³ that can introduce message handlers for the processing of requests. The ADPEP V2 will therefore consist of an Axis2 module with one or more message handlers.

4.2.8 Known Issues

Apache Ode does not currently have the capability to implicitly carry an arbitrary context between multiple operations in a business process. This capability is currently under consideration and may be implemented by the time we begin the implementation of the ADPEP V2 component.

¹ <http://ode.apache.org/architectural-overview.html>

² http://ws.apache.org/axis2/1_3/Axis2ArchitectureGuide.html#thearchi

³ http://ws.apache.org/axis2/1_3/Axis2ArchitectureGuide.html#extendingwithmodules

If this capability is not supported, it would require the user to make explicit assignments in the business process to carry the security context across multiple service invocations.

4.3 Test client

WP8 has already developed a test client for the ServiceProvider side. The actual version of this test client is described in the deliverable D8.1. This test client was build to be able to check the correct content in the fedora repository. Therefore it directly connects to the fedora repository and doesn't make use of the TAS3 infrastructure. The test client could be enhanced to upload and retrieve data via the TAS3 stack and act as a simple service-requester-client.

4.4 Intalio independent client

A major advantage of the Intalio|Designer is that the generated xforms and the designed processes can be executed by the Intalio|BPMS very easily. These xforms can also be reused in different business processes on the Intalio|BPMS. Moreover it is possible to separate the generated xforms from the Intalio system and reuse them in completely independent applications. For TAS3 this is the potentiality to build an Intalio independent client application which is able to process these or slightly enhanced xforms generated with the Intalio|Designer enabling this application to use TAS3 for secure storage and retrieval.

5 Annex

5.1 Intalio|BPMS: TAS3 related and complementary capabilities

This section describes capabilities of the Intalio|BPMS that relate to and complement the TAS3 infrastructure in providing broader solutions to the problem domain that TAS3 addresses, as well as to meet the requirement of real-world enterprises that would want to deploy TAS3-based solutions. These functions are needed for the further development of the Intalio ServiceRequester ADPEP and the security requirements that come along with the TAS3 infrastructure.

5.1.1 Upload and Download Capabilities

The workflow framework (Tempo⁴) of the Intalio|BPMS provides a task attachment abstraction that may be used to add file attachments to any workflow task, either as an explicitly required attachment or for ad-hoc attachments in less rigorous processes. The task attachment service (TAS) essentially provides create (upload), read (download) and delete operations to manage attachments.

The Intalio|BPMS now provides two implementations of the attachment service,

1. Database: Attachments are stored in a relational database as binary large objects (BLOBs). This is the default implementation.
2. Alfresco: Attachments are stored in Alfresco⁵ enterprise content-management platform. This implementation may be configured to use an existing Alfresco deployment.

(The task management service is extensible and supports adding different attachment back-ends if the above are not suitable.)

This feature is used in the Kenteq process (described earlier in this document) to allow users to upload their PCP (Personal Competency Profile). The attachments can then be processed by other processes and/or external systems.

At development time, an attachment control can be simply dragged & dropped on a workflow form as illustrated with the figure below.

⁴ Tempo Workflow project: <http://tempo.intalio.org>

⁵ Alfresco Enterprise Content Management: <http://www.alfresco.com>

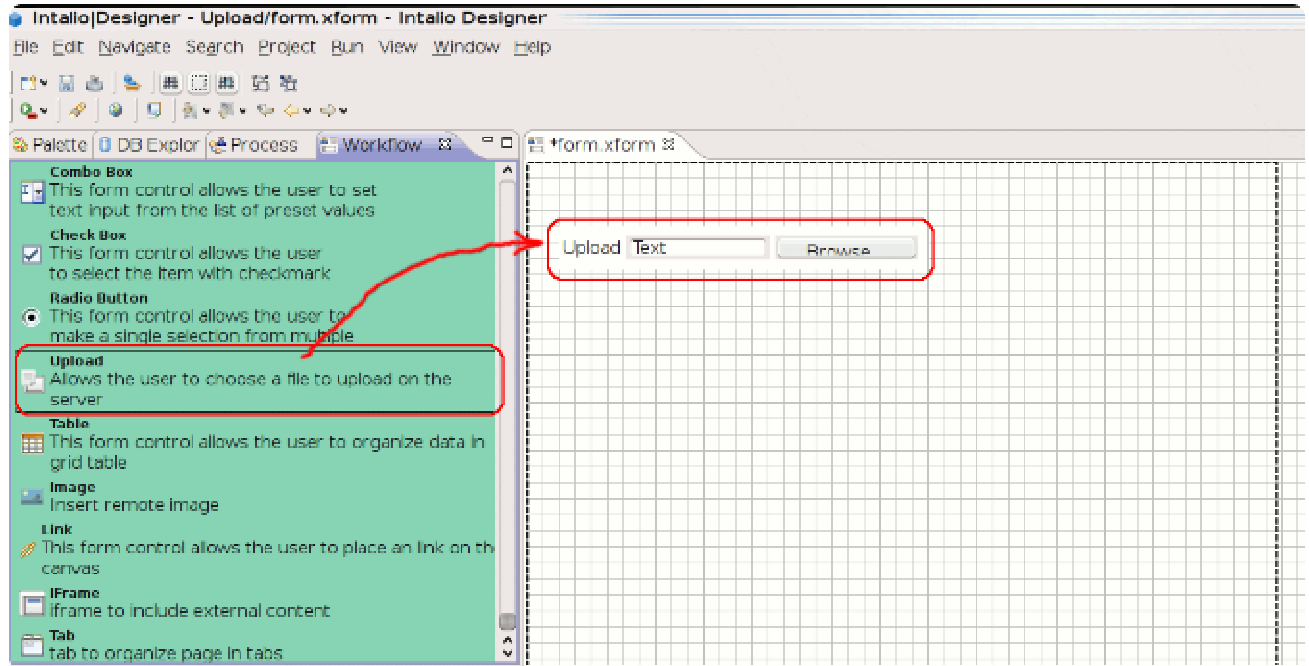
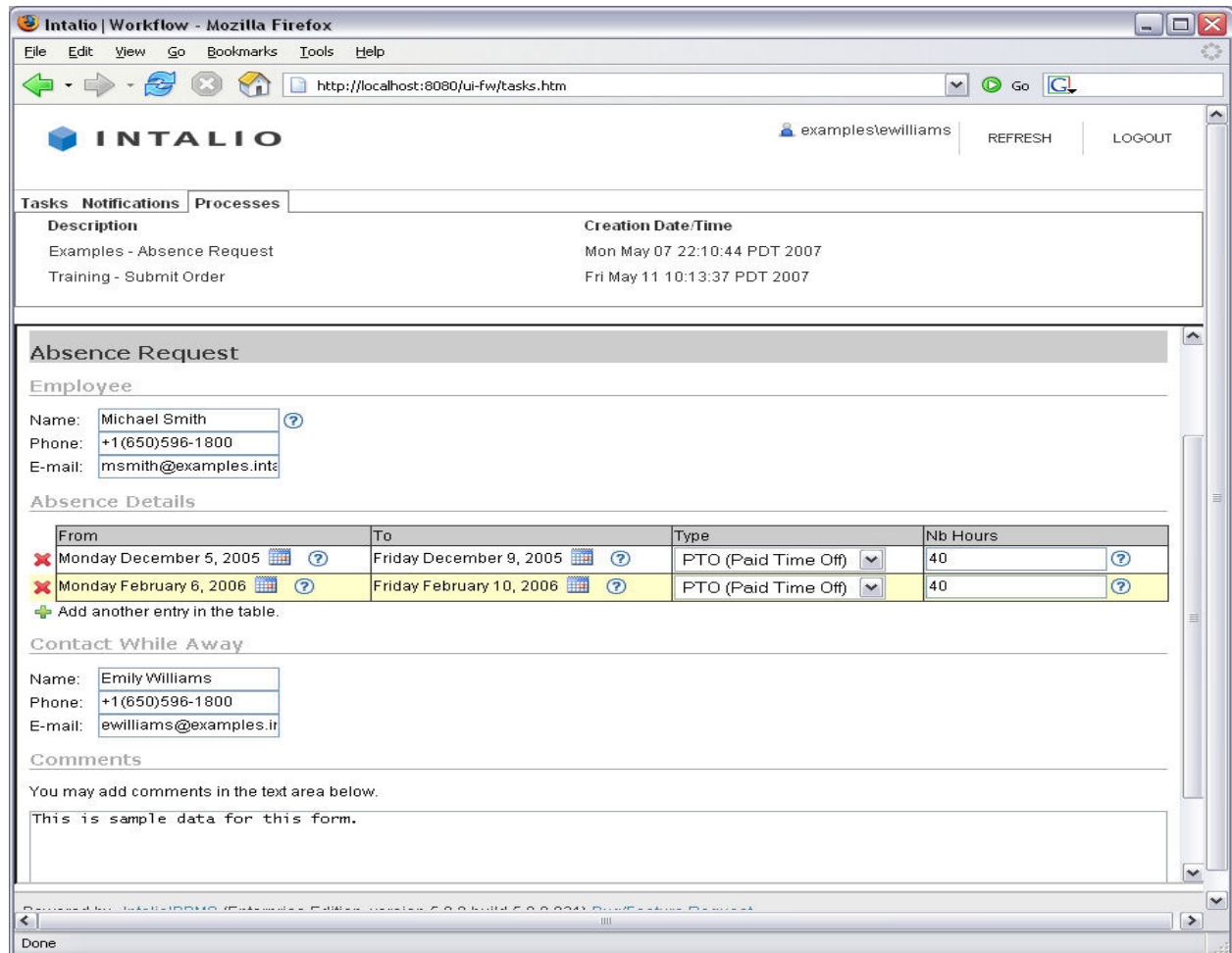


Figure 9: Drag & drop of controls on a workflow form

5.1.2 Support for User Interface Technologies

The Tempo workflow project supports pluggable user-interface technologies. The framework has separate back-end services and front-end user-interfaces.



INTALIO examples\ewilliams REFRESH LOGOUT

Tasks Notifications Processes

Description	Creation Date/Time
Examples - Absence Request	Mon May 07 22:10:44 PDT 2007
Training - Submit Order	Fri May 11 10:13:37 PDT 2007

Absence Request

Employee

Name: Michael Smith
 Phone: +1 (650) 596-1800
 E-mail: msmith@example.com

Absence Details

From	To	Type	Nb Hours
Monday December 5, 2005	Friday December 9, 2005	PTO (Paid Time Off)	40
Monday February 6, 2006	Friday February 10, 2006	PTO (Paid Time Off)	40

+ Add another entry in the table.

Contact While Away

Name: Emily Williams
 Phone: +1 (650) 596-1800
 E-mail: ewilliams@example.com

Comments

You may add comments in the text area below.

This is sample data for this form.

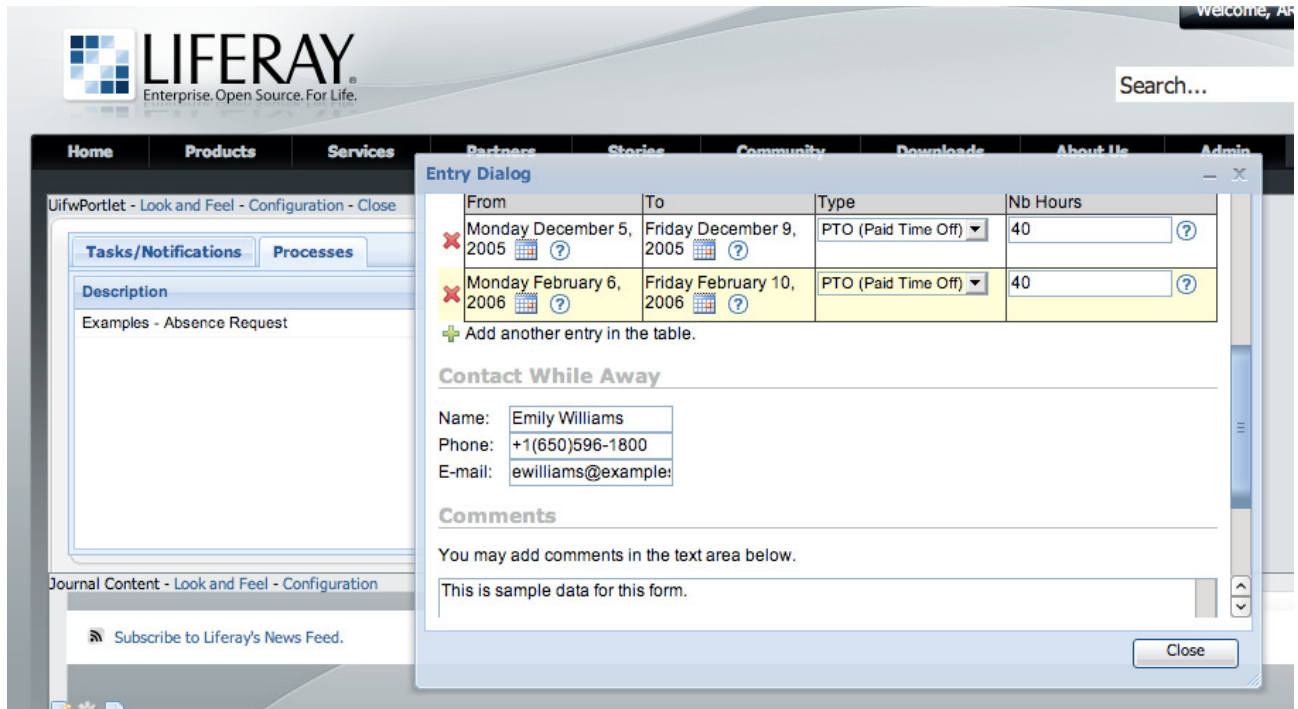
Figure 10: Example form built with Xforms technology

Both task list and forms may be developed with different technologies and used together. Tempo provides built-in support for the XForms⁶ technology by embedding the Orbeon XForm Engine. The Intalio|BPMS also provides integration with TIBCO General Interface⁷ which provides rich web-based interface capabilities.

Other technology integration have been demonstrated, such as with Java/JSP, Ruby on Rails, and integration into the Liferay Enterprise Portal.

⁶ <http://en.wikipedia.org/wiki/XForms>

⁷ <http://www.tibco.com/devnet/gi>



The screenshot shows the Liferay Portal interface with an 'Entry Dialog' form open. The form is titled 'Entry Dialog' and contains a table for adding absence requests. The table has columns for 'From', 'To', 'Type', and 'Nb Hours'. Two entries are shown: one for Monday December 5, 2005 to Friday December 9, 2005, and another for Monday February 6, 2006 to Friday February 10, 2006. Both entries are marked as 'PTO (Paid Time Off)' and have '40' hours. Below the table is a section for 'Contact While Away' with fields for Name, Phone, and E-mail. The 'Name' field is filled with 'Emily Williams', 'Phone' with '+1(650)596-1800', and 'E-mail' with 'ewilliams@example.com'. There is also a 'Comments' section with a text area for adding comments. The background shows the Liferay Portal navigation menu and a search bar.

From	To	Type	Nb Hours
Monday December 5, 2005	Friday December 9, 2005	PTO (Paid Time Off)	40
Monday February 6, 2006	Friday February 10, 2006	PTO (Paid Time Off)	40

Contact While Away

Name: Emily Williams
Phone: +1(650)596-1800
E-mail: ewilliams@example.com

Comments

You may add comments in the text area below.

This is sample data for this form.

Figure 11: Same example form displayed in Liferay Portal

5.1.3 Integration with Existing Single Sign-On Solutions

As part of the TAS3 project, the Tempo project was integrated with two dominant single sign-on frameworks: Central Authentication Service⁸ (CAS) and OpenSSO⁹.

The CAS integration allows for single sign-on across different web application and specifically with the Liferay Portal¹⁰ (shown above) and the Alfresco content management system (mentioned in the previous section). This integration also allows the task list to be embedded this in popular enterprise portal, as well adding attachments to tasks and transparently storing them into Alfresco for management. CAS supports a number of authentication forms, including LDAP, RADIUS, and X.509 certificates. It also supports a variety of protocols including both OpenID and SAML 1.1/2.0 open standards.

OpenSSO is an identity and single sign-on middleware that offers interoperability with many additional systems, of particular interest is interoperability with the Liberty Alliance Project Specifications¹¹.

This integration is part of Intalio|BPMS v6.0 and already under testing and deployment by some customers.

5.1.4 WS-Security Support

The Intalio|BPMS is based on Web Service standards to communicate with external services and uses Apache Axis2¹² as its web services stack to support various web service security standards. As part of the TAS3 project, the Rampart¹³ security module of Axis2 was integrated to support the following interoperability standards,

- WS-Security 1.0 and 1.1 (incl. relevant parts in WS-Policy)
- WS-Security UsernameToken Profile 1.1
- WS-Security X.509 Certificate Token Profile 1.0
- WS-Security SAML Token Profile 1.0

⁸ <http://www.iasig.org/cas>

⁹ <https://opensso.dev.java.net/>

¹⁰ <http://www.liferay.com/>

¹¹ <https://opensso.dev.java.net/public/use/docs/pdf/index.html>

¹² <http://ws.apache.org/axis2/>

¹³ http://ws.apache.org/axis2/modules/rampart/1_3/security-module.html

- Web Services Secure Conversation - February 2005
- WS-Security Policy 1.1
- WS-Trust - February 2005
- WS-Basic Security Profile 1.1

The above specify on how integrity and confidentiality can be enforced on Web services messaging. WS-Security describes how to attach signatures and encryption headers to SOAP messages. In addition, it describes how to attach security tokens, including binary security tokens such as X.509 certificates and Kerberos tickets, to messages.

These functionalities are essential for the further development and the connection of the Intalio|BPMS to the TAS3 infrastructure.

5.2 Internal data fields provided by the Intalio|BPMS

The following list was extracted from the implemented test process using the Data Mapper which is a module of the Intalio|Designer. It shows some fields from internal objects of the implemented test-process and the corresponding object-type. At runtime they can be used to complement security and environmental data which is needed for the PDP to make a proper decision whether a request is allowed or not.

➔ \$inputCreateTaskRequestMsg.root	
➔ taskMetaData	
➔ taskId	abc text: string
➔ taskState	abc text: token
➔ taskType	abc text: token
➔ description	abc text: string
➔ processId	abc text: string
➔ creationDate	abc text: dateTime
➔ userOwner[*]	abc text: string
➔ roleOwner[*]	abc text: string
➔ claimAction[?]	
➔ user[*]	abc text: string
➔ role[*]	abc text: string
➔ revokeAction[?]	
➔ user[*]	abc text: string
➔ role[*]	abc text: string
➔ saveAction[?]	
➔ user[*]	abc text: string
➔ role[*]	abc text: string
➔ completeAction[?]	
➔ user[*]	abc text: string
➔ role[*]	abc text: string
➔ formUrl	abc text: anyURI
➔ failureCode	abc text: string
➔ failureReason	abc text: string
➔ priority	abc text: int
➔ scheduledActions	
➔ expiration	
➔ until[?]	abc text: dateTime
➔ for[?]	abc text: duration
➔ deferActivation	
➔ until[?]	abc text: dateTime
➔ for[?]	abc text: duration
➔ userProcessCompleteSOAPAction	abc text: string
➔ isChainedBefore	abc text: boolean
➔ previousTaskId	abc text: string
➔ userProcessEndpoint	abc text: string
➔ userProcessNamespaceURI	abc text: string

- ➔ participantToken abc text: string
- ➔ taskInput
- ➔ \$inputCreate TaskResponseMsg.root
 - ➔ isChainedAfter[?]
 - ➔ taskMetaData
 - ➔ status abc text: string
 - ➔ errorCode[?] abc text: string
 - ➔ errorReason[?] abc text: string
- ➔ \$inputNotifyTaskCompletionRequestMsg
 - ➔ taskMetaData
 - ➔ taskOutput
 - ➔ output
 - ➔ @formUrl abc text: string
 - ➔ @participantToken abc text: string
 - ➔ @taskId abc text: string
 - ➔ @user abc text: string
 - ➔ Result 1 abc text: string
 - ➔ Param2 abc text: string
 - ➔ Param1 abc text: string
 - ➔ InputForm_for_ADPTTestservice abc text: string
 - ➔ Status abc text: string
 - ➔ \$inputNotifyTaskCompletionResponseMsg
 - ➔ isChainedAfter[?]
 - ➔ taskMetaData
 - ➔ status abc text: string
 - ➔ errorCode[?] abc text: string
 - ➔ errorReason[?] abc text: string
 - ➔ \$nsTestIntalioADPEPConnectionRequestMsg
 - ➔ Param0[?] abc text: string
 - ➔ Param1[?] abc text: string
 - ➔ \$xformNotifyTaskCompletionRequestMsg.root
 - ➔ taskMetaData
 - ➔ taskOutput
 - ➔ output
 - ➔ @formUrl abc text: string
 - ➔ @participantToken abc text: string
 - ➔ @taskId abc text: string
 - ➔ @user abc text: string
 - ➔ Upload-Credential
 - ➔ @filename abc text: string
 - ➔ @mediatype abc text: string
 - ➔ @upload-id abc text: default="Upload_Credential"
 - ➔ anyURI abc text: string
 - ➔ Texteingabe abc text: string
 - ➔ UploadTest abc text: string
 - ➔ Status abc text: string

6 Glossary

ADPEP:	Application Dependent Policy Enforcement Point
AIPEP:	Application Independent Policy Enforcement Point
BPEL:	Business Process Execution Language
BPMN:	Business Process Modelling Notation
BPMS:	Business Process Management System
PCP:	Personal Competency Profile
PDP:	Policy Decision Point
TAS3:	Trusted Architecture for Securely Shared Services

7 Document Control

Amendment History

Version	Baseline	Date	Author	Description/Comments
0.1		10. Nov. 2008	Kutscher	First Draft
0.5		30. Jan. 2009	Kutscher Boisvert	Including Intalio Chapter
0.6		30. Mar. 2009	Kutscher Boisvert	Release for Koblenz Conference
0.7.2		14. Apr. 2009	Kutscher	Final release for project-internal review.
0.8		12. May 2009	Kutscher	Modifications based on comments from internal reviewers.
0.9		13. May 2009	Kutscher	Modifications based on comments from internal reviewers.
1.0		15. May 2009	Kutscher	Integration TAS3-Template
1.1		23. May 2009	Kutscher	Modifications based on the Bruxelles review
1.2		28. May 2009	Kutscher	Minor renamings