

## IST PROJECT 2001-35399



### A Governmental Knowledge-based Platform for Public Sector Online Services

<b>Project Number:</b>	IST-2001-35399
<b>Project Title:</b>	A Governmental Knowledge-based Platform for Public Sector Online Services
<b>Deliverable Type:</b>	Public

<b>Deliverable Number:</b>	D81
<b>Contractual Date of Delivery:</b>	31-8-2003
<b>Actual Date of Delivery:</b>	17-10-2003
<b>Title of Deliverable:</b>	Trials configuration and evaluation process
<b>WP contributing to the Deliverable:</b>	WP8
<b>Nature of the Deliverable:</b>	Report
<b>Editor(s):</b>	Pablo Fernández Pardo, Tomás Pariente Lobo
<b>Author(s):</b>	Rafael Cañadas, Pablo Fernández Pardo, Tomás Pariente Lobo, Segundo Heredia, Stelios Gerogianakis, Eleni Fatourou, Pavlos Kattoulas, Gregory Kavadias, Ann Macintosh, Angus Whyte, Andy McKay-Hubbard, Nick Adams, Costas Vassilakis, George Lepouras, Akrivi Katifori, Vassilis Stoumpos, Andreas Pipis, Leonidas Bogiatzis, Simon Haston

**Abstract:** After the integration and deployment done in WP8, a testing framework is set up to assess the SmartGov platform. The aim of D81 is to describe the SmartGov trials configuration and the evaluation process framework that is applicable for the assessment. Thus it introduces real scenarios and use cases for the pilot services that are run in CEC and GSIS.

#### Project funded by the European Community under the "Information Society Technologies" Programme (1998-2002)

© Copyright by the SmartGov Consortium.  
The SmartGov Consortium consists of:

Partner's Name	Acronym	Role	Country
University of Athens	UoA	Project Coordinator	Greece
T-Systems Nova	TNB	Partner	Germany
Indra Sistemas S.A.	Indra	Partner	Spain
Archetypon S.A.	ARC	Partner	Greece
Napier University	NU	Partner	UK
General Secretariat for Information Systems	GSIS	Partner	Greece
City of Edinburgh Council	CEC	Partner	UK

## Table of Contents

1	Introduction .....	7
2	The integration of the SmartGov platform .....	9
2.1	Integration teams, components and tests.....	12
2.1.1	Integrator – Validator .....	12
2.1.2	Integrator – XML Repository .....	13
2.1.3	Front End – XML Repository .....	14
2.1.4	Front End – Integrator .....	15
2.1.5	Integrator - Agent .....	15
2.1.5.1	Description.....	15
2.1.5.2	Preparation .....	16
2.1.5.3	Actions .....	16
2.1.5.4	Expected results .....	17
2.1.5.5	Notes .....	21
2.1.6	IIG - XML Repository.....	21
2.1.6.1	Description.....	21
2.1.6.2	Preparation .....	21
2.1.6.3	Actions .....	22
2.1.6.4	Expected results .....	22
2.1.6.5	Notes .....	23
2.2	Integrated common repository .....	23
2.2.1	Front-end versions.....	23
2.2.2	XML-Repository versions .....	24
2.2.3	Integrator tool versions.....	24
2.2.4	Communication services versions .....	25
3	The deployment in CEC and GSIS .....	27
3.1	Available infrastructure in the PAs .....	27
3.1.1	Available infrastructure in CEC .....	30
3.1.2	Available infrastructure in GSIS .....	31
3.2	Deployment planning .....	32
3.2.1	Deployment implementation in CEC .....	33
3.2.2	Deployment planning in GSIS.....	34
3.3	User Training .....	36
4	The trials configuration .....	38

4.1	Setting up of pilot e-services scenarios .....	38
4.2	Pilot service requirements .....	42
4.2.1	The GSIS pilot.....	42
4.2.2	The CEC pilot .....	42
4.3	Mapping to SmartGov platform entities .....	43
4.3.1	The GSIS pilot.....	43
4.3.1.1	Transaction service .....	43
4.3.1.2	Forms.....	43
4.3.1.3	TSEs .....	43
4.3.1.4	KUs.....	46
4.3.2	The CEC pilot .....	47
4.4	Service implementation.....	58
4.4.1	The GSIS pilot.....	58
4.4.1.1	TSEs, TSE groups and validation checks.....	58
4.4.1.2	KUs.....	62
4.4.1.3	Form layout .....	63
4.4.1.4	Linkage of form layout to the SmartGov platform items .....	64
4.4.1.5	Definitions of statistics.....	65
4.4.1.6	Linkage to external information systems.....	65
4.4.1.7	Service deployment.....	66
4.4.1.8	Service testing and improvement.....	66
4.4.2	The CEC pilot .....	67
4.4.2.1	Locations .....	67
4.4.2.2	Volumes (Only includes North sector) .....	67
5	The Evaluation Process .....	68
5.1	The Principles Behind the Evaluation.....	68
5.2	SmartGov Overall Success Criteria .....	69
5.2.1	Conformance to Functional Requirements: .....	69
5.2.1.1	Platform specific user requirements.....	70
5.2.2	System acceptability.....	77
5.3	CEC Baseline Measurements .....	80
5.4	The User Sites Success Criteria and Metrics .....	80
5.4.1	Development Environment User Acceptability.....	81
5.4.1.1	Utility and relevance.....	81
5.4.1.2	Usability .....	81

5.4.2	Pilot Service User Acceptability.....	84
5.4.2.1	Utility and relevance.....	84
5.4.2.2	Usability .....	84
5.4.3	Social & Organizational Acceptability .....	86
6	Conclusions.....	87
7	References .....	88
Appendix A.	CEC Baseline Measurements .....	89
A.1	Design and Deployment .....	89
A.2	Service Delivery .....	89
A.3	Average time taken to process one 305 form .....	89
A.4	Average number days to process 305 forms .....	90
A.5	Error Rate .....	90

## Table of Figures

Figure 1 Integration phase.....	9
Figure 2 Interoperation among the SmartGov components .....	11
Figure 3. Pilot Application Architecture .....	28
Figure 4 – CEC pilot service overview .....	47
Figure 5 – Logical diagram for the CEC pilot service part 1.....	49
Figure 6 - Logical diagram for the CEC pilot service part 2 .....	50
Figure 7 - CEC pilot welcome screen .....	55
Figure 8 - CEC pilot order page #1 .....	56
Figure 9 - CEC pilot order page #2 .....	56
Figure 10 - CEC pilot clientpage #1.....	57
Figure 11 - CEC pilot clientpage #2.....	57
Figure 12 - CEC pilot final summary page .....	58
Figure 13 – Header form for the e-Vies service.....	63
Figure 14 – Detail form for the e-Vies service.....	64
Figure 15 - Linkage of visual form layout to the SmartGov platform items .....	65
Figure 17 Adjusted Nielsen model .....	78

## List of Acronyms

Acronym	Explanation
API	Application Programming Interface
BEAN	Java Bean
DSN	Data source name
JDBC	Java Database Connectivity
JSP	Java Server Page
KU	Knowledge unit
LDAP	Lightweight Directory Access Protocol
MVC	Model-View-Controller
PA	Public Authorities
RDBMS	Relational Database Management System
RUP	Rational Unified Process
SGA	SmartGov agent
TS	Transaction service
TSE	Transaction service element
TSE Group	Group of transaction service element
UML	Unified Modeling Language
WAP	Wireless Application Protocol
WML	Wireless Markup Language
XHTML	eXtensible Hypertext Markup Language
XML	Extensible Markup Language
XSLT	Extensible Style sheet Language Template

# 1 Introduction

The assessment of the SmartGov platform is the latest major activity in WP8, following the integration and deployment of its hardware and software infrastructure in the participating PAs CEC and GSIS. Therefore this document tries to provide a global view of the configuration and evaluation processes as well as the foundations, i.e. the principles, concepts, guidelines, techniques, methods selected in order to evaluate the Smartgov system, how the methods are executed and how the result is analysed.

For a better understanding of the accomplished work during this Work Package and of the scope of the deployment of the platform in the PAs, the trials configuration and the evaluation process, Chapter 2 shows an overview of the Integration of the Smartgov platform. It refers to relevant issues concerning the structure and organisation of the SmartGov common repository, e-services and SKDB contents and the integration teams, components and tests done during the integration phase.

Chapter 3 describes the deployment of the platform in CEC and GSIS. A detailed list of of the hardware and software requirements for the development and production environments, the deployment tasks and the scheduled time is provided. Additionally non technical issues are addressed as usually they may prove critical for a successful introduction of a new software system such as the user training, learning and support tasks during deployment, and the population of the system contents (KUs, TSEs, etc.). The setting up of the scenarios in the PAs is described in Chapter 4. In order to demonstrate and measure how the SmartGov fits the user needs produced in WP4, two pilot services have been populated for the trials configuration and evaluation: in CEC: the Equipment and Adaptation service; in GSIS: the e-VIES, the electronic submission of VIES documents (recapitulative statement of intra-community supplies and acquisitions).

A work plan for the setting up and the performing of the of the pilot services and the trials is provided. The application domains and the expertise needed for the implementation of the pilot services are described in detail: workgroups with suitable expertise and skills, SmartGov stakeholders, i.e. managers, domain experts, service workers, administrators, IT staff, and Service end-users. Finally details on the mappings to SmartGov platform entities for each service are described.

Once established the deployment of the platform and the work plan to carry out the complete fulfilment of the pilot services, the evaluation process for the SmartGov

platform is introduced in the chapter 5. Once outlined the process and its principles, Overall Smartgov platform success criteria are defined, as specific criteria for each of the pilot sited (GSIS and CEC).



## 2 The integration of the SmartGov platform

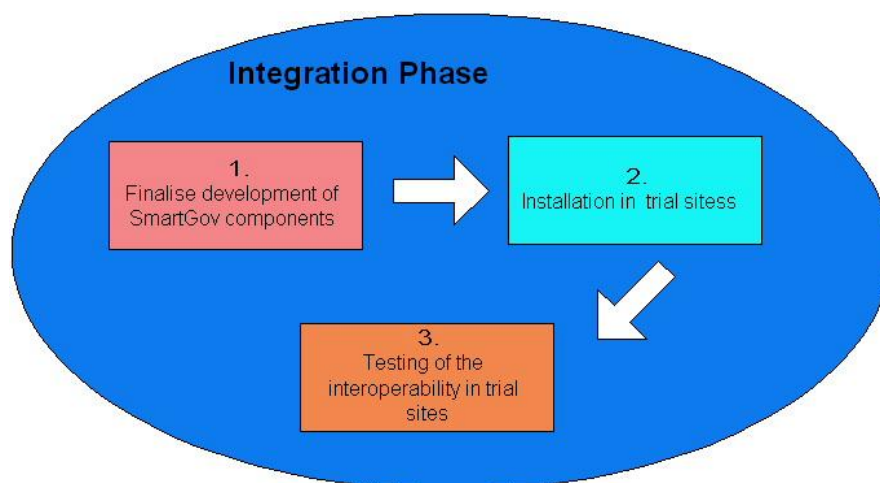
Workpackage 8 builds on the results and complements the development work realised in Work packages 5 and 6. The different components designed and implemented in Work packages 5 and 6 must be put together to produce the integrated SmartGov platform. Before the resulting system is demonstrated and validated by its potential users during the trials, the system has to be installed in both trial sites, namely CEC and GSIS and its technical solidity has to be guaranteed.

Following the installation of the SmartGov platform, the end users at the trial sites should create the SmartGov platform e-services, namely the Equipment and Adaptation service in CEC and the e-VIES or VAT in e-commerce in GSIS. The population of the SKDB (including service rules, online assistance, and other forms of knowledge) will allow these e-services be operational.

The integration methodology for the SmartGov platform will progress gradually from the component level to the SmartGov platform level and consists of the following phases:

- Finalisation of the SmartGov components under development;
- Installation of all the software components in both trial sites GSIS and CEC.
- Testing of the interoperability of both installed platforms

The integration of the SmartGov platform is illustrated in Figure 1.



**Figure 1 Integration phase**

Before the beginning of the installation phase, each SmartGov module should comply with the functional specifications elaborated in deliverables D51-D61. As functional

parts of an integrated system, the different components of the SmartGov architecture should be ready to communicate with their neighbouring components.

The platform integration methodology is used in order to ensure that:

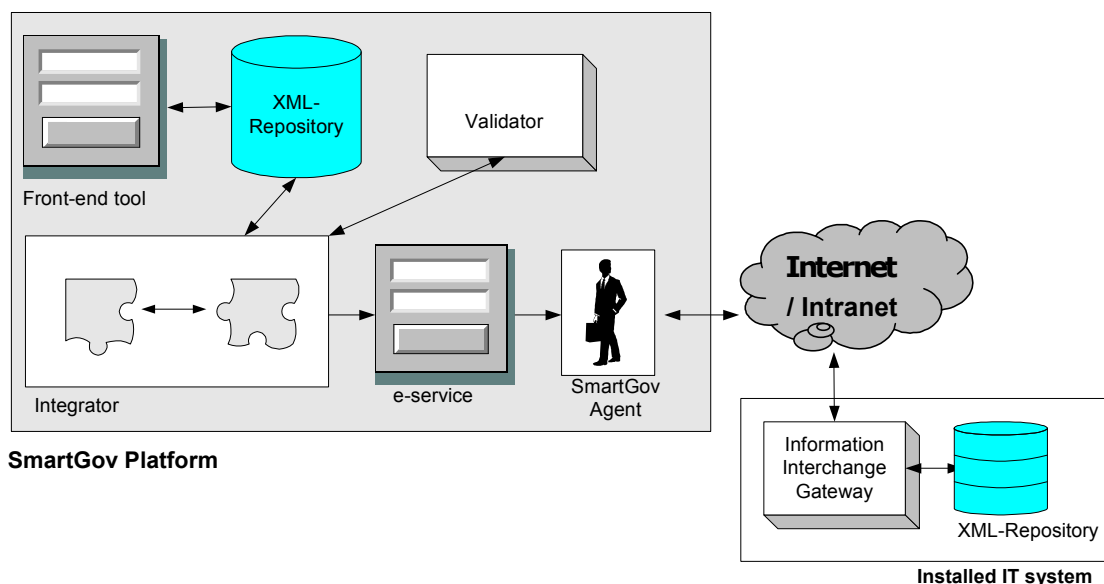
- each component is installed and functions successfully,
- the platform is fully integrated and consistent.

The major software components of the SmartGov platform and the interactions among them are illustrated in Figure 2:

- The front-end tool: This tool is the graphical interface between the end-user and the XML-Repository. It enables end-users (domain experts and managers of the platform) to manipulate transaction services and their components: the forms of the service, the validation checks that apply to these forms and the knowledge units that will be presented to the user. The front-end tool interoperates and should be integrated with the XML-Repository.
- The XML repository: It is the component, which is responsible for the storage of the documents manipulated by the front-end. It exposes an API to front-end component, to enable it to manipulate all the XML documents, which comprise the Transaction services (TS). It interoperates and should be integrated with the front-end tool as well as the integrator component.
- The validator component: It is the component, which translates the SmartGovLang language, a high-level description language defined within SmartGov and useful for defining simple rules, during the instantiation of TSs. The validator translates the simple rules introduced by the user using the SmartGovLang language, and will produce all necessary programming language fragments, in order to implement the back-end functionality (fragments of java language, understandable by the server) and where appropriate the front-end functionality (fragments of javascript language, understandable by the browser). Validator interoperates with integrator component. More specifically, the integrator passes to the validator the appropriate arguments and receives the corresponding code fragments, which it inserts into the produced e-service.
- The integrator component: It is the heart of the SmartGov development platform. Its task is to automatically generate all the necessary files, objects, and components in order to create a fully operational e-forms web application. Its output is a web application (war file) deployed to the application server e.g. to Tomcat of the Jakarta project (<http://jakarta.apache.org/tomcat/>). Integrator component retrieves all the building modules of an e-service

(Transaction Services, Forms, Transaction service elements etc) from the XML-Repository and eventually creates the actual web application with all necessary files. Moreover, it interoperates with the validator component in order to produce the front-end / back-end code fragments and incorporates them into the final service.

- The SmartGov agent (SGA): It is an integral part of the SmartGov platform, enabling the submission of *requests* to external systems and the retrieval of the respective results. SmartGov agent interoperates with two components: the e-service created by the SmartGov platform and the IIG component. SmartGov agent receives requests from the e-service and forwards them to IIG component, which can be considered as plugged into the installed IT system / external to SmartGov platform.
- The Information Interchange Gateway: The Information Interchange Gateway is attached to the installed IT system and arranges for interception of the requests originating from the SmartGov agent, their execution and the returning of the appropriate results back to the SmartGov agent. IIG interoperates with the SmartGov agent, and an XML repository, located by the side of the installed IT system. This XML-Repository may communicate with an Excel spreadsheet, an RDBMS, even with the installed IT system.



**Figure 2 Interoperation among the SmartGov components**

## ***2.1 Integration teams, components and tests***

### **2.1.1 Integrator – Validator**

**Description:** Integrator and validator components should interoperate in order to incorporate all the validation rules to the final Transaction Service. Integrator reads from the XML repository the XML representations of the validation checks and then makes the appropriate API calls to validator component, in order to receive all the necessary fragments of code for the actual e-service. The code fragments of the front-end and the back-end will be in the form of javascript and java code respectively.

The interaction between integrator and validator is as follows:

**Preparation:** For the interoperability testing of the integrator with the validator, the XML-Repository must be installed and populated previously. Moreover, the following software must be installed on the machine that the interoperability tests will take place:

- Operating System: Windows 2000, with service pack greater than 3
- Servlet engine: Tomcat 4.1 or later
- JDK: Java2 SE 1.4.2 or greater.

#### Action 1

Use each one of the actual validation checks analysed in the specifications of the SmartGov language compact rules and make API calls from the integrator component to the validator component by passing to the validator as arguments each validation check, namely:

- *between(fieldId, lowerLimit, upperLimit).*
- *requires(fieldId1, fieldId2)*
- *precludes(fieldId1,fieldId2)*
- *requiresMulti(fieldId1, fieldIdList2)*
- *checkRow(fieldIdList)*
- *checkRelation(fieldId1, operator, fieldId2, c)*

For each validation check the validation should take place:

- at back-end
- at back-end and front-end

#### **Expected result**

The expected generated code should be produced and function as expected from the syntactic constructs of the SmartGovLang compact rules.

## Action 2

Use each validation check in multiple languages. For each validation check the validation should take place:

- at back-end
- at back-end and front-end

### **Expected result**

The appropriate code is generated and multilingual messages (information, warnings or errors) are displayed properly.

## **2.1.2 Integrator – XML Repository**

**Description:** Integrator interoperates with the XML repository in order to produce the final e-Service, which will be available to end-users (citizens, public servants etc). To do so, it retrieves all the necessary components to build the e-service from the XML repository: These components are:

- transaction services (TS),
- forms,
- transaction service elements (TSEs),
- TSE groups,
- validation checks
- knowledge units (KU).

**Preparation:** For the interoperability testing, the integrator and the XML-Repository must be installed previously. Moreover, the XML-Repository must be populated.

The following software must be installed on the machine that the interoperability tests will take place:

- Operating System: Windows 2000, with service pack greater than 3
- Servlet engine: Tomcat 4.1 or later
- JDK: Java2 SE 1.4.2 or greater.
- MS sql server 2000
- Mysql 4.x

## Action 1

The XML-Repository is installed in both RDBMSs: MS Sql server and Mysql.

### **Expected result**

The integrator works properly with both RDBMSs

## Action 2

The integrator builds an e-service, provided that this service consists of all kind of building elements: TS, forms, TSEs TSE groups, validation checks and KUs. The validation checks should concern forms and TSE groups.

**Expected result**

The e-Service is fully functional and works without a problem. All the service components stored within the XML-Repository are transformed to components of the actual service (e.g. Jsp pages containing javascript, java classes etc) and are deployed successfully to the servlet engine.

### 2.1.3 Front End – XML Repository

**Description:** The Front-end uses the XML repository to store in it two types of objects: Transactional Service elements and Knowledge elements.

All the elements involved in the development of a Transactional Service can be created or modified with the Front-end. These elements are stored in the Xml Repository and, once the service is completely defined, the Integrator will retrieve the created elements, in order to build an e-service.

Moreover, the Front-end also interoperates with the XML Repository to store the elements related with knowledge: Taxonomies and SmartGov-platform Knowledge Units (KUs). These elements are used to improve the usability of the platform.

**Preparation:** For the interoperability testing, the XML-Repository and the Front-end must be installed previously.

The following software must be installed on the machine that the interoperability tests will take place:

- Operating System: Windows 2000, with service pack greater than 3
- Servlet engine: Tomcat 4.1 or later
- JDK: Java2 SE 1.4.2 or greater.
- MS sql server 2000 or Windows Mysql 4.x

**Action 1**

The Front-end can store and retrieve all the elements from the XML-Repository, installed in both RDBMSs: MS Sql server and Mysql.

**Expected result**

The Front-end interoperated properly with the XML Repository, using both RDBMSs.

**Action 2**

The Front-end is able to retrieve lists of elements (last modified elements, related with other specific elements), using the index functionality provided by the XML Repository.

**Expected result**

The expected results are retrieved without problem from the XML Repository.

**2.1.4 Front End – Integrator**

**Description:** Front-end and Integrator components interact implicitly. The Front-end is used to populate the elements of a Transactional Service in a XML Repository, and the Integrator takes these elements and builds a new e-service. Therefore, the Front-end must enable the users to create all the elements of a service, so that the integrator is able to build successfully a deployable e-service, ready to be used.

**Preparation:** For the interoperability testing, the XML-Repository, the Integrator and the Front-end must be installed previously.

The following software must be installed on the machine that the interoperability tests will take place:

- Operating System: Windows 2000, with service pack greater than 3
- Servlet engine: Tomcat 4.1 or later
- JDK: Java2 SE 1.4.2 or greater.
- MS sql server 2000 or Windows Mysql 4.x

**Action 1**

A complete Transactional Service is created using the Front-end. Once the definition is complete, the TS is processed by the integrator.

**Expected result**

All the TS elements created using the Front-end can be processed appropriately by the Integrator, and the generated e-Service reflect completely all the data introduced through the Front-end.

**2.1.5 Integrator - Agent****2.1.5.1 Description**

The Integrator module, though it does not directly communicate with the SmartGov agent, needs to know certain details on how services provided by the SmartGov agent may be invoked, since this knowledge will enable the Integrator module to embed into the compiled e-services appropriate requests to the SmartGov agent, which will be executed in the context of the service runtime. The Integrator needs to embed into any e-service it compiles invocations to three SGA services delivering the following functionality:

1. *User authentication.* The electronic service presents the user with a screen prompting for a user name and a password. When the user enters the requested data, the electronic service submits a request to a specific SmartGov Agent service to validate whether the authentication credentials entered by the user are valid.
2. *Document storage.* When the electronic service user submits a document, the electronic service invokes a specific SmartGov Agent service to store the document. The document storage service should also offer provisions for storing the XML document into external (legacy) information systems, through an appropriate mapping.
3. *Document retrieval.* The SmartGov agent provides a service enabling the retrieval of documents previously stored through the document storage facility. The document retrieval service should also offer provisions for retrieving data from external (legacy) information systems to formulate initial documents.

Additionally, the SmartGov API should be documented, in order to be accessible by the PAs IT staff willing to enhance the logic of compiled services with additional invocations to services provided by the SmartGov agent.

#### **2.1.5.2 Preparation**

The SmartGov API, needed both by the PAs' IT staff and the developers of the integrator was promptly standardised and documented in D61. No amendments to the API were found to be required during the integration and pilot implementation phases. Regarding the functionality that the Integrator needed in order to compile a running service (i.e. user authentication, document storage and document retrieval), technical meetings were held between ARC and UoA, in order to determine the actual format of the XML documents exchanged in the context of the invocation and reply collection of the SmartGov Agent services. During these technical meetings the XML schemata and the semantics of the XML document elements were derived.

#### **2.1.5.3 Actions**

The Integrator implementation team relied on the SmartGov Agent API to perform invocations to the services delivered through the SmartGov Agent; preparation of XML documents to be submitted along with the requests and handling of replies were based on the decisions regarding the schemas and semantics of exchanged XML documents.



The implementation team that undertook the development of the SmartGov agent services was similarly based on the standardised XML schemata and semantics to realise and deliver the required service functionality. Once both development actions were complete and the individual components were tested, integration tests were conducted to determine whether the developed components could interoperate.

#### **2.1.5.4 Expected results**

The results of these integration tests, for each of the SmartGov Agent services, are summarised in the following paragraphs.

##### **2.1.5.4.1 User authentication service**

<b>Id</b>	<b>Test case</b>	<b>Expected test result</b>	<b>Actual test result</b>
1	Valid user name and password	Return of a "successful authentication" indication	A "successful authentication" indication was returned
2	Valid user name, invalid password	Return of an "unsuccessful authentication" indication	An "unsuccessful authentication" indication was returned
3	Invalid user name, invalid password	Return of an "unsuccessful authentication" indication	An "unsuccessful authentication" indication was returned
4	Malformed XML request message	An exception is raised on the service submitting the request	An exception was raised on the service submitting the request
5	Malformed XML response message	An exception is raised on the service submitting the request	An exception was raised on the service submitting the request
6	User credential repository misconfiguration	An exception is raised on the service submitting the request	An exception was raised on the service submitting the request
7	Communications failure	An exception is raised on the service submitting the request	An exception was raised on the service submitting the request

<b>Id</b>	<b>Test case</b>	<b>Expected test result</b>	<b>Actual test result</b>
8	Misconfigured SGA	An exception is raised on the service submitting the request	An exception was raised on the service submitting the request
9	Misconfigured IIG	An exception is raised on the service submitting the request	An exception was raised on the service submitting the request

#### **2.1.5.4.2 Document storage service**

<b>Id</b>	<b>Test case</b>	<b>Expected test result</b>	<b>Actual test result</b>
1	Valid XML document	Successful storage of the document in the repository	The document was successfully stored in the repository
2	Valid XML document using capabilities for external system interconnection	Successful storage of the document in the repository and successful storage of the document within the external information system	The document was successfully stored both in the repository and in the external information system
3	Invalid XML document	An exception is raised on the service submitting the request	An exception was raised on the service submitting the request
4	Malformed XML request message <sup>1</sup>	An exception is raised on the service submitting the request	An exception was raised on the service submitting the request
5	Malformed XML response message	An exception is raised on the service submitting the request	An exception was raised on the service submitting the request
6	Communications failure	An exception is raised on the service submitting the request	An exception was raised on the service submitting the request

---

<sup>1</sup> The XML request message encapsulates the XML document to be stored

<b>Id</b>	<b>Test case</b>	<b>Expected test result</b>	<b>Actual test result</b>
7	Misconfigured SGA	An exception is raised on the service submitting the request	An exception was raised on the service submitting the request
8	Misconfigured IIG	An exception is raised on the service submitting the request	An exception was raised on the service submitting the request
9	Misconfigured XML repository	An exception is raised on the service submitting the request	An exception was raised on the service submitting the request
10	XML repository lacking indices	The document is normally stored in the XML repository	The document was normally stored in the XML repository

#### **2.1.5.4.3 Document retrieval service**

<b>Id</b>	<b>Test case</b>	<b>Expected test result</b>	<b>Actual test result</b>
1	Valid request, asking for the retrieval of all documents submitted by a designated user	Retrieval of all documents submitted by the designated user	All documents submitted by the designated user were retrieved.
2	Valid request, asking for the retrieval of all documents submitted by a designated user through a specific service	Retrieval of all documents submitted by the designated user through the specified service	All documents submitted by the designated user through the specified service were retrieved.
3	Valid request, asking for the an existing specific document	Retrieval of the designated document	The designated document was retrieved

<b>Id</b>	<b>Test case</b>	<b>Expected test result</b>	<b>Actual test result</b>
4	Valid request, asking for the a specific, non-existing document, without the option of creating a new one	Return of an empty reply	An empty reply was returned
5	Valid request, asking for the a specific, non-existing document, setting the option of creating a new one	Return of a new document with pre-populated fields	A new document with pre-populated fields was created.
6	Invalid XML document	An exception is raised on the service submitting the request	An exception was raised on the service submitting the request
7	Malformed XML request message	An exception is raised on the service submitting the request	An exception was raised on the service submitting the request
8	Malformed XML response message	An exception is raised on the service submitting the request	An exception was raised on the service submitting the request
9	Communications failure	An exception is raised on the service submitting the request	An exception was raised on the service submitting the request
10	Misconfigured SGA	An exception is raised on the service submitting the request	An exception was raised on the service submitting the request
11	Misconfigured IIG	An exception is raised on the service submitting the request	An exception was raised on the service submitting the request

<b>Id</b>	<b>Test case</b>	<b>Expected test result</b>	<b>Actual test result</b>
12	Misconfigured XML repository	An exception is raised on the service submitting the request	An exception was raised on the service submitting the request
13	XML repository lacking indices	An exception is raised on the service submitting the request	An exception was raised on the service submitting the request

#### **2.1.5.5 Notes**

The tested scenarios cover all cases of invocations to the user authentication service. It must be noted that cases involving malformed XML request and response messages in particular are not bound to occur in the running SmartGov environment since request and reply messages are formulated by controlled modules. They have been included however in the test suite, in order to verify module robustness under unexpected circumstances.

All test were successful in the sense that they produced the expected results.

### **2.1.6 IIG - XML Repository**

#### **2.1.6.1 Description**

The IIG interacts with the XML repository in the context of the SmartGov Agent services providing facilities for document storage and retrieval. The XML repository was chosen as a storage provider for XML documents submitted by the user, since it offers rich functionality including document validation against XML schemata, creation of indices against designated document elements, index-based querying for optimised document retrieval etc.

#### **2.1.6.2 Preparation**

Initially the available versions of the XML repository were examined. It was decided that the document storage and retrieval services would employ the XML repository v2.0 as underlying infrastructure, since (a) it provides richer functionality than the previous versions and (b) in this way a single XML repository version would be used throughout the platform. Since the XML repository API had been standardised, service coding could commence immediately without the need to wait for a definitive version of the actual code. Technical meetings were held to define the XML schema for

documents and determine the indices that had to be created for more efficient service operation.

### 2.1.6.3 Actions

The XML repository was set up, as an underlying infrastructure and was prepared for use by the document storage and retrieval service. XML repository preparation for this task included the declaration of the document type and the definition of the indices on specific document elements, which were used in queries. Service coding commenced, based on the XML repository v2.0 API specifications. Once developed, the code of the service was tested, firstly as an autonomous component and subsequently within the full SmartGov platform context to determine whether it could interoperate with the XML repository.

### 2.1.6.4 Expected results

The results of the integration tests between the IIG and the XML repository are described in the following table.

<b>Id</b>	<b>Test case</b>	<b>Expected test result</b>	<b>Actual test result</b>
1	Request for storage of a valid document	Storage of the document	The document was stored
2	Request for retrieval of all documents	Retrieval of all documents	All documents were retrieved.
3	Request for retrieval of documents having a specific value on an indexed field	Retrieval of the designated documents	The designated documents were retrieved
4	Request for storage of an invalid document	Raising of an exception	An exception was raised

<b>Id</b>	<b>Test case</b>	<b>Expected test result</b>	<b>Actual test result</b>
5	Request for retrieval through a non-existing index	Raising of an exception	An exception was raised
6	Invalid XML document	Raising of an exception	An exception was raised
7	Misconfigured XML repository	Raising of an exception	An exception was raised

#### **2.1.6.5 Notes**

All test were successful in the sense that they produced the expected results.

## ***2.2 Integrated common repository***

In this section, a small history of the successive versions of the SmartGov software components is listed. Namely, the components analysed are:

- Front-end tool
- XML-Repository
- Integrator tool
- Communication services (SmartGov agent, IIG component)

### **2.2.1 Front-end versions**

- Front-end v1.0:
  - The first version of the Front-end.
  - This version (and later) interoperates with the XML Repository v1.0 and support Windows MySql 4.x RDBMS.
- Front-end v1.1:
  - Includes a process to import Kus from XML files so that they are available in the Front-end.
  - Taxonomy and Taxonomy node editor has been added.
  - This version also corrects some bugs appeared in the first version.
- Front-end v1.2:
  - This version adds support for Microsoft Sql Server 2000.
  - It also correct some bugs appeared in the previous version.

- Front-end v2.0:
  - This version (and later) interoperated with the Xml Repository v2.0.
  - It's also the first version compatible with the Integrator. Therefore, the Front-end is able to create complete TS that will be built by the integrator, to generate a deployable e-service.
  - A process to create the XML Repository with the configuration required for the Integrator and the Front-end has been developed.
  - Some bugs detected in previous versions have been corrected.

## 2.2.2 XML-Repository versions

- XML Repository v1.0:
  - The first version of the XML Repository.
  - This version supports MySQL 3.x only.
  - Documents and indexes are not organized per document type.
- XML Repository v2.0:
  - Supports MySQL 4.x and MS SQL Server 2000.
  - Documents and indexes are organized per document type.
  - Performance improvements
  - Internal code re-factoring
  - Small changes in public API
  - Inclusion of visual Repository model management tool (XML Repository Manager)
- XML Repository v2.01:
  - XPath index expressions can now evaluate to elementary data types (non-node graph objects)

## 2.2.3 Integrator tool versions

- Integrator v0.4:
  - The first stand-alone working version of the Integrator.
  - This version works against a file system containing service description files.
  - Elementary data type validation is available.
  - No custom validation rules supported.
  - Works as an off-line, command-line process.
  - User-submitted documents are saved as individual XML files
- Integrator v0.5:



- User-submitted documents can be searched and retrieved.
- User login screen with no back-end checking
- Generated service placed inside frames
- UI enhancements in generated service
- Form-level statistics
- Integrator v0.6:
  - Works with XML Repository v.2.0
  - Works with SGA and IIG
  - Distributed as a self-installing package
  - Repeating TSEGroups
  - Delivered with a demo service (TaxService)
  - Web UI for the Integrator
- Integrator v1.0:
  - Corrected bugs discovered during integration with SGA/IIG
  - Delivered with eVies and TaxService demo services
- Integrator v1.01:
  - Corrected bugs found during integration with Front-End
  - Integrated with Front-End v.2.0

## **2.2.4 Communication services versions**

- Version 0.7
  - Basic functionality for posting requests and receiving replies. Alternate execution paths are supported.
- Version 0.8
  - Support for spooling non-real time requests has been added. Pending action queue daemons included in distribution
  - Security features for the IIG have been incorporated
- Version 0.9
  - Pending action queues may operate on any DBMS, instead of being bound to Oracle
  - Added support for SSL communications
  - Logger facilities included in distribution
- Version 1.0
  - First complete release of communication services
  - Incorporates notification mechanisms

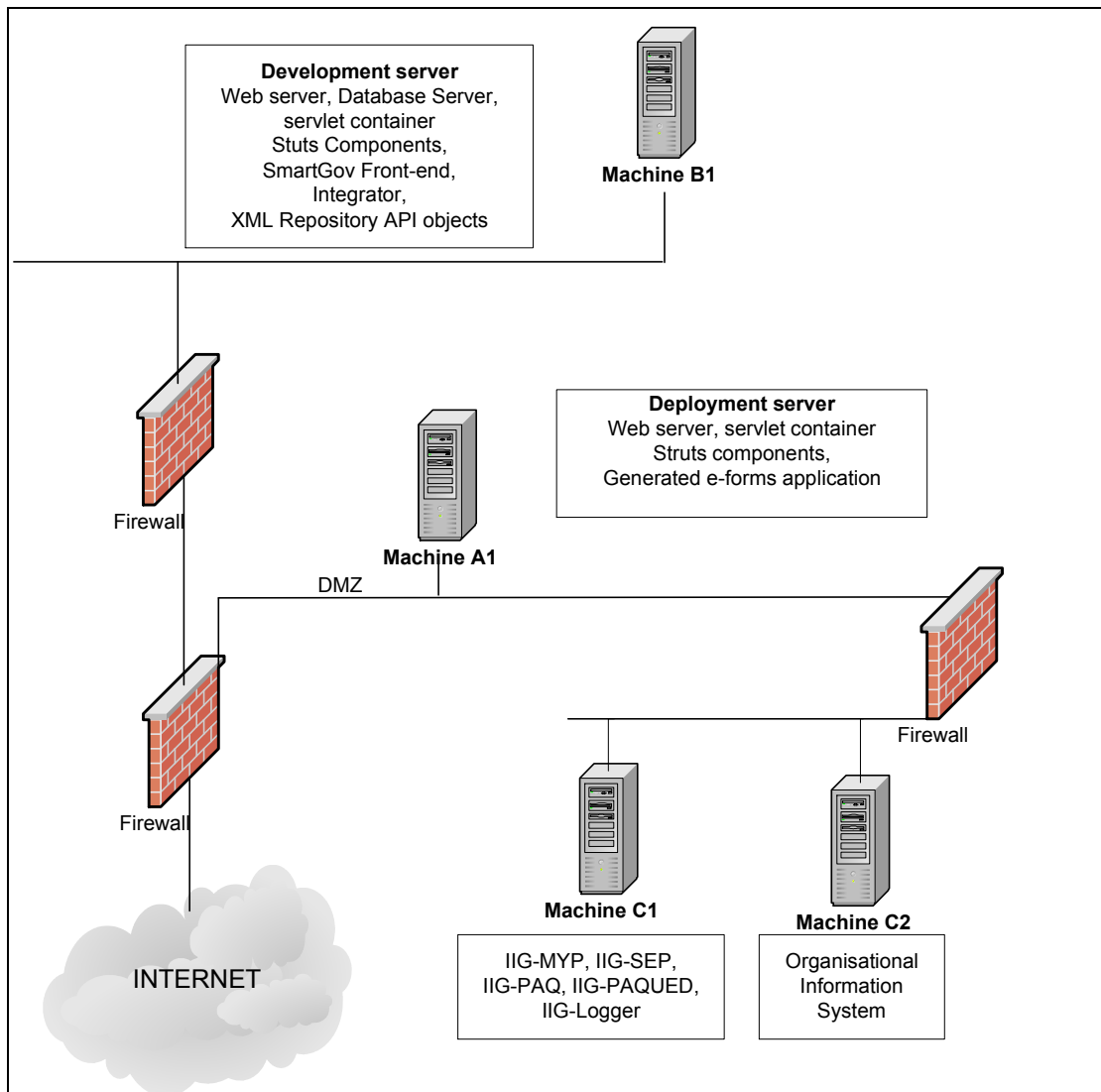
- Interoperability tests with the integrator module have been successfully passed
- Documentation for installing and configuring the services are available
- Version 1.1
  - Tools to create self-signed SSL certificates have been added
  - Added feature for validating XML requests
  - First version of GUI installer
- Version 1.2
  - Incorporation of login validation and document storage and retrieval services into the distribution
  - Interoperability with XML repository has been tested
  - Second version of GUI installer, complemented with installation instructions
- Version 2.0
  - Final release
  - Provisional facilities for creating initial pre-populated documents on-the-fly
  - Provisional facilities for flattening submitted documents for communication with external information systems
  - Final version of GUI installer plus installation instructions

### **3 The deployment in CEC and GSIS**

Primarily the deployment of the SmartGov platform comprises of technical issues such as the installation and the configuration of the software in the target domains. In addition to these technical issues there exist other, non technical issues, which may prove critical for a successful introduction of a new software system: determination of the necessary steps and the sequence they have to be carried out, user support tasks during deployment, user training as well as the population of the system with existing information and/or with newly developed content. All the above-mentioned issues and the approach taken to tackling them are described in this section.

#### ***3.1 Available infrastructure in the PAs***

The architecture of the available infrastructure in the participating Public Administrations is illustrated in the next diagram. According to it there exist two main servers: one development server and one deployment server. This architecture is described in detail in SmartGov deliverable D51-D61. Although more than one development and deployment servers can be used for stability and redundancy in a production environment, the pilot trials allow to reduce these needs. Thus only one server is used for the development testing and one server for the deployment of services.



**Figure 3. Pilot Application Architecture**

The minimum requirements for the installation of the SmartGov platform are outlined in the next table. The first table summarises the minimum and recommended hardware configurations for the development platform, the second table summarises minimum and recommended hardware configurations for the deployment platform while the third the hardware configurations for the developer's client environment.

**Table 1. Development Environment Hardware Configurations**

	<b>Minimum Configuration</b>	<b>Recommended Configuration</b>
Processor	PIII 500 MHz	PIV 1GHz
Memory	256 MB	512 MB
Hard Disk	4 GB	8 GB
Network connectivity	Ethernet	Fast Ethernet

**Table 2. Deployment Environment Hardware Configurations**

	<b>Minimum Configuration</b>	<b>Recommended Configuration</b>
Processor	PIII 500 MHz	PIV 1.5 GHz
Memory	512 MB	1 GB
Hard Disk	8 GB	16 GB
Network connectivity	Ethernet	Fast Ethernet

**Table 3. Developer's Client Environment Hardware Configurations**

	<b>Minimum Configuration</b>	<b>Recommended Configuration</b>
Processor	PIII 500 MHz	PIV 1.5 GHz
Memory	512 MB	1 GB
Hard Disk	8 GB	16 GB
Network connectivity	Ethernet	Fast Ethernet
Monitor	17" CRT	17" TFT

In regard to the software that has to be installed to the servers both environments require a web/application server and a database server as a means for storing information and communicating with the users, either public administrators or end-users. It has to be noted that hardware requirements are largely dependent on and dictated by the database system that will be installed. Currently the SmartGov platform has been successfully tested with MySQL database server and Microsoft SQL Server.

**Table 4. Necessary software for the Development Environment**

<b>Necessary Software</b>	<b>Recommended (tested) software</b>
J2EE-compliant web server	Apache 2 + Tomcat 4.1
Database server	MySQL 4.x, MS SQL 7 or 2000
Java runtime environment, JDBC	Java 1.4.1 Runtime Environment, JDBC 3.0 API + JDBC Driver
Struts	v2.3
Castor	V0.9.4.3

**Table 5. Necessary software for the Deployment Environment**

<b>Necessary Software</b>	<b>Recommended (tested) software</b>
J2EE-compliant web server	Apache 2 + Tomcat 4.1
Database server	MySQL 4.x, MS SQL 7 or 2000
Java runtime environment, JDBC	Java 1.4.1 Runtime Environment, JDBC 3.0 API + JDBC Driver
Struts	v2.3
Castor	V0.9.4.3

For the pilot application back end connectivity will be tested on the same server as the deployment server. Oracle 8.1.7 has been selected and tested as the database server for the back-end system.

### 3.1.1 Available infrastructure in CEC

The next table summarises the hardware and software configurations for the development, deployment platform and client system configuration, present at CEC for the trials.

**Table 6. Developer's Client Environment Hardware Configurations**

	<b>Development Server</b>	<b>Deployment Server</b>	<b>Developer Client</b>
Processor	PIV 1.8 GHz	PIV 1.8 GHz	Pentium III
Memory	1GB	1GB	128 MB
Hard Disk	8 GB	8 GB	2 GB
Network connectivity	Network Card 10/100	Network Card 10/100	Ethernet

	<b>Development Server</b>	<b>Deployment Server</b>	<b>Developer Client</b>
Software	<ul style="list-style-type: none"> <li>• Apache Web server</li> <li>• Jakarta Tomcat web container</li> <li>• MS SQL</li> <li>• STRUTS Framework</li> <li>• Java SDK</li> <li>• Castor</li> <li>• ANT</li> <li>• Xerces</li> <li>• Xalan</li> <li>• JDBC driver type 3 or 4 (comes with the RDBMS)</li> <li>• OS: Windows 2000 server</li> </ul>	<ul style="list-style-type: none"> <li>• Apache Web server</li> <li>• Jakarta Tomcat web container</li> <li>• STRUTS Framework</li> <li>• Java SDK</li> <li>• JDBC driver type 3 or 4 (comes with the RDBMS)</li> <li>• OS: Windows 2000 server</li> </ul>	Web browser

### 3.1.2 Available infrastructure in GSIS

The next table summarises the hardware and software configurations for the development, deployment platform and client system configuration, present at GSIS for the trials.

**Table 7. Developer's Client Environment Hardware Configurations**

	<b>Development Server</b>	<b>Deployment Server</b>	<b>Developer Client</b>
Processor	Pentium IV 2.4GHz	Pentium IV 2.4GHz	Pentium III, 1 GHz
Memory	1 GB	1 GB	256 MBytes
Hard Disk	16 GB	16 GB	6 GB
Network connectivity	Fast Ethernet	Fast Ethernet	Fast Ethernet
Monitor	17" TFT	17" TFT	17" TFT
Database Server	MySQL 4.0.5	MySQL 4.0.5	Not needed
Software	<ul style="list-style-type: none"> <li>• Apache Web server</li> </ul>	<ul style="list-style-type: none"> <li>• Apache Web</li> </ul>	Internet Explorer

	<b>Development Server</b>	<b>Deployment Server</b>	<b>Developer Client</b>
	<ul style="list-style-type: none"> <li>• Jakarta Tomcat web container</li> <li>• MySQL v 4.0.5</li> <li>• STRUTS Framework</li> <li>• Java SDK</li> <li>• Castor</li> <li>• ANT</li> <li>• Xerces</li> <li>• Xalan</li> <li>• JDBC driver type 3 or 4 (comes with the RDBMS)</li> <li>• OS: Windows 2000 server</li> </ul>	<ul style="list-style-type: none"> <li>server</li> <li>• Jakarta Tomcat web container</li> <li>• STRUTS Framework</li> <li>• Java SDK</li> <li>• JDBC driver type 3 or 4 (comes with the RDBMS)</li> <li>• OS: Windows 2000 server</li> </ul>	<ul style="list-style-type: none"> <li>and Netscape Communicator</li> <li>DreamWeaver MX (for form designers only)</li> <li>J2SDK (for IT staff only)</li> </ul>

### ***3.2 Deployment planning***

Towards deploying the SmartGov platform at the trials sites a number of steps and milestones were identified. These steps are listed in the next table.

**Table 8. Identification of critical paths and milestones**

<b>Task No</b>	<b>Task</b>
1	Hardware in place
2	OS and software for both servers Windows 2000 server DB (Oracle 8.1.7 or mysql, or MS SQL server) JDK 1.4 Tomcat v4.1 STRUTS v2.3 Castor
3	Installation of XML repository



<b>Task No</b>	<b>Task</b>
4	Installation of User Interface
5	Testing between User Interface and XML Repository
6	Installation of Integrator
7	Testing of Integrator and XML Repository
8	Testing of Integrator and User Interface
9	Back end connectivity tests
10	HTML form integration with SmartGov entities (KUs, TSEs etc)
11	Populate SKDB, create pilot e-services
11.1	Design service
11.2	Create HTML forms
11.3	Extract knowledge from WP7
11.4	Identify service KUs and TSEs
11.5	Develop service TSEs and KUs
11.6	Link KUs to TSEs, forms etc
11.7	Develop domain ontologies
11.8	Develop platform KUs
11.9	Define and code validation checks
11.10	Define statistics
11.11	Define trials plan and configure system
12	Training

### 3.2.1 Deployment implementation in CEC

The SmartGov platform was deployed at CEC according to the next time schedule.

**Table 9. Implementation of deployment in CEC**

<b>Task No</b>	<b>Task</b>	<b>Date</b>
1	Hardware in place	15 July 2003
2	OS and software for both servers	15 Aug. 2003

<b>Task No</b>	<b>Task</b>	<b>Date</b>
	Windows 2000 server MS SQL server JDK 1.4 Tomcat v4.1 STRUTS v2.3 Castor	
3	Installation of XML repository	30 Aug. 2003
4	Installation of User Interface	15 Sep. 2003
5	Testing between User Interface and XML Repository	10 Oct. 2003
6	Installation of Integrator	30 Aug. 2003
7	Testing of Integrator and XML Repository	10 Oct. 2003
8	Testing of Integrator and User Interface	10 Oct. 2003
9	Back end connectivity tests	10 Oct. 2003
10	HTML form integration with SmartGov entities (KUs, TSEs etc)	10 Oct. 2003
11	Populate SKDB, create pilot e-services	
11.1	Design service	30 Sep. 2003
11.2	Create HTML forms	30 Sep. 2003
11.3	Identify service KUs and TSEs	30 Sep. 2003
11.4	Develop service TSEs and KUs	31 Oct 2003
11.5	Link KUs to TSEs, forms etc	31 Oct 2003
11.6	Develop domain ontologies	30 Sept 2003
11.7	Develop platform KUs	31 Oct 2003
11.8	Define and code validation checks	31Oct 2003
11.9	Define statistics	31 Oct 2003
11.10	Define trials plan and configure system	31Oct 2003
12	Training	15 Nov. 2003

### 3.2.2 Deployment planning in GSIS

The SmartGov platform was deployed at GSIS according to the next time schedule.

**Table 10. Implementation of deployment in GSIS**

<b>Task No</b>	<b>Task</b>	<b>Date</b>
1	Hardware in place	06/07/2003
2	OS and software for both servers Windows 2000 server My SQL JDK 1.4 Tomcat v4.1 STRUTS v2.3 Castor	21/07/2003
3	Installation of XML repository	23/07/2003
4	Installation of User Interface	v1: 29/07/2003 v1.1: 22/09/2003 v2: 11/10/2003
5	Testing between User Interface and XML Repository	11/08/2003
6	Installation of Integrator	V0.8: 08/09/2003 V0.9: 13/10/2003
7	Testing of Integrator and XML Repository	V0.8: 11/09/2003 V0.9: 14/10/2003
8	Testing of Integrator and User Interface	V0.8: 10/09/2003 V0.9: 14/10/2003
9	Back end connectivity tests	phase 1: 12/08/2003 phase 2: 14/08/2003
10	HTML form integration with SmartGov entities (KUs, TSEs etc)	08/10/2003
11	Populate SKDB, create pilot e-services	
11.1	Design service	08/09/2003
11.2	Create HTML forms	11/09/2003
11.3	Identify service KUs and TSEs	14/09/2003
11.4	Develop service TSEs and KUs	22/09/2003
11.5	Link KUs to TSEs, forms etc	08/10/2003
11.6	Develop domain ontologies	11/10/2003

Task No	Task	Date
11.7	Develop platform KUs	06/10/2003
11.8	Define and code validation checks	06/10/2003
11.9	Define statistics	09/10/2003
11.10	Define trials plan and configure system	15/10/2003
12	Training	25/10/2003

### ***3.3 User Training***

Training on the platform started before the completion of the deployment phase. Once users had the user interface of the system installed they could start learning its functionality. To this end, users first learned how to create Knowledge Units (KUs) and Transaction Service Elements (TSEs). Once the XML repository was in place, users could get accustomed to saving and retrieving knowledge units and transaction service elements.

Users used the interface to initiate population of the XML repository with knowledge units and transaction service elements identified previously during the design of the pilot e-services. The domain ontologies developed for the public administrations were used in categorising knowledge units to ease saving and retrieval.

The training phase also gave an opportunity for testing the integrated platform with real data and users. Several minor alterations and fine-tunings were carried out in response to user requests.

Finally, when the integrator was installed and tested, users could fully explore the completed platform to code validation checks and to define statistics. At this point users had been accustomed to all the objects and functions of the platform and could recognise their importance and mode of usage. They had also learned the correct sequence for performing tasks and they could synthesize simple actions to perform more complex goals.

The first version of the user manual was provided in the start of the training phase to be used as a basis for the training. User manual evolved with the training and was enhanced and optimised to cover all the aspects of the SmartGov platform. Testing and revisions were regularly performed since platform success is closely coupled to the documentation quality.

Especially for the City of Edinburgh Council pilot service, for which end-users will mainly be public sector employees there will be an extension of the training period in order to train not only the direct users of the platform, that is domain experts, IT personnel and system administrators, but also the service users. This will enable the acquisition of further assessment data, directly from the users of the resulting services, evaluating thus indirectly the SmartGov platform.

## 4 The trials configuration

The SmartGov trials configuration is based on the user requirements that have been extracted during WP4 and documented in D41 [D41]. The trials configuration task has produced an elaborate workplan for setting up and performing the trials; this workplan is detailed in the following paragraphs.

### ***4.1 Setting up of pilot e-services scenarios***

In order to set up the pilot e-service scenarios, a number of aspects of the electronic services were examined. Firstly, the user requirements documented in D4.1 [D41] were consulted to identify the application domains and the expertise needed for the implementation of the pilot services. This step enabled the formulation of workgroups with suitable expertise and skills, including all types of SmartGov stakeholders, i.e. managers, domain experts, service workers, administrators and IT staff. Service end-users also participated in the workgroups. Then, the tasks required for the development of the services were analysed in detail and assigned to specific teams within the workgroups, depending on the skills required for each task. The task list for service development using the SmartGov platform is as follows:

<b>Task Id</b>	<b>Description</b>	<b>Expertise required</b>
1	Derivation of service requirements, which includes the identification of data that must be collected and knowledge that must be available, the flow that the collected data will follow within the organisation, the business rules that apply to the service and the statistics that need to be collected. The service entry may be created within the SmartGov platform.	Domain experts, managers, service workers, IT staff

Task Id	Description	Expertise required
2	Within this task, the requirements documented in task (1) are mapped to SmartGov platform entities: the data that need to be collected are mapped to TSEs and TSE groups; the forms on which TSEs and TSE groups will be placed are drafted; validation checks are expressed more formally; connectivity requirements with other systems are determined. It must be noted that within this step the qualitative characteristics of the items are determined, while implementation details will be addressed in the following tasks. Naming conventions (e.g. TSE names) are also developed to provide for an unambiguous "communication glossary" between workgroup members. An initial population of the SmartGov platform with item descriptions may be conducted within this task. Taxonomies are also derived in the context of this task.	Domain experts, managers, service workers, IT staff
3	Task 3 is the main implementation task during which SmartGov platform items are made concrete, by having all their details filled in. Task 3 is divided into the following subtasks:	
3.1	Development of TSEs. Each data item that must be either presented to or collected from the user is mapped to a TSE. Data types, value ranges and validation checks are also derived and entered to the platform.	Domain experts; possible assistance from IT staff

<b>Task Id</b>	<b>Description</b>	<b>Expertise required</b>
3.2	Development of TSE groups. Data items that must be grouped together are clustered into TSE groups. A TSE group may have its knowledge units and validation rules that apply to the TSE group as a whole; these are entered into the platform within this subtask.	Domain experts; possible assistance from IT staff
3.3	Development and approval of KUs. KUs may apply to the service as a whole, to a specific TSE group or an individual TSE; moreover a KU may pertain either to the <i>item semantics</i> , i.e. what the item represents and the rules governing it, or to the <i>interaction with the item</i> , e.g. how individual values may be entered, whether multiple selections are allowed etc.	Domain experts, service workers
3.4	Forms layout implementation. Within this task the forms that will be presented to the service end-users are created. Provided that sufficiently user-friendly tools for form layout creation are available (e.g. DreamWeaver, Front Page etc), this task may be undertaken by domain experts. Service workers and service end-users may contribute by providing constructive comments on how the layout can be improved.	Domain experts with possible assistance from IT staff; service workers and service end-users play a consultative role
3.5	Linkage of form layout to the SmartGov platform items. Visual elements of forms (input boxes, text areas, help item anchors etc) are linked with their SmartGov platform item counterparts (TSEs, TSE groups, KUs).	Domain experts



<b>Task Id</b>	<b>Description</b>	<b>Expertise required</b>
3.6	Validation checks. Remaining validation checks pertaining to forms or to the service as a whole are implemented.	Domain experts; possible assistance from IT staff
3.7	Statistics definition. Each user group defines the statistics that are needed for service assessment and evaluation.	Managers, domain experts
3.8	Communication with IT systems. The IT staff provides the facilities for further communication with back-end or third-party IT systems	IT staff
4	Service deployment. The integrator is run to create and deploy the service.	No specific expertise required; any role authorised to run the integrator is sufficient
5	Service testing and improvement. Final tests are performed on the service before it is made publicly available and comments on its operation are made; SmartGov platform stakeholders take into account the comments and the collected statistics and make improvements to the service.	Service workers and service end users in cooperation with domain experts and IT staff

The tasks were carried out by each individual team, under the coordination of managers; it must be noted that no integration step is necessary, since the SmartGov platform structure enables the automation of the integration step.

The following paragraphs provide details on how the specific tasks were conducted within the scope of the pilots' implementation. It has to be noted that task 1 has been performed within work package 4 and has been reported in D41, thus only brief descriptions are included here for completeness purposes; the interested reader is referred to SmartGov deliverable D41 - User Requirements, Services and Platform Specifications. Whenever appropriate, separate sub-sections are introduced for pilots in the two participating Pas, namely CEC and GSIS.

## **4.2 Pilot service requirements**

In the following subsections the pilot service requirements are described in a high-level of abstraction. More details are included in deliverable D41, to which the interested reader is referred.

### **4.2.1 The GSIS pilot**

In WP4 two services have been analysed for the GSIS, namely the electronic submission of VIES documents (recapitulative statement of intra-community supplies and acquisitions) and the electronic submission of e-Commerce statements, a service that is designed to commence within one year. By analysing two services, rather than one, a more complete picture of the requirements from the platform was drawn. In the pilot implementation and evaluation phase, however, only the first service will be implemented, therefore in the following paragraphs only the VIES document submission service is discussed.

The recapitulative statement of intra-community acquisitions and supplies has to be submitted from taxable persons that are identified by a valid VAT number and supply, acquire or transfer goods to other Member States without charging VAT, after having verified through VIES the VAT number of the payer. The supply or acquisition of services to other Member States, the exports to other countries not belonging to the EU and the local supply of goods are not included.

The Recapitulative Statement contains detailed information for each supplier or buyer within EU, including the peer's country and VAT number, the taxable value of the transactions conducted with the specific peer, the total value of "triangular" acquisitions or supplies (a special form of transactions including an intermediate and an ultimate destination) and discounts and the rebates for intra-Community supplies that have been already reported in a previous invoice.

Statements are submitted on a trimester basis to the local tax administration.

### **4.2.2 The CEC pilot**

For the CEC, the Equipment and Adaptations service was chosen as the pilot service to be implemented. The Equipment and Adaptations service is provided by the Social Work department to supply, service and maintain and uplift any equipment required in the care of their customers. A large number of users are involved in this service: the customers for example could be an elderly person, disabled person or a carer acting on

their behalf; occupational Therapists or Social Workers employed by CEC or external Healthcare Professionals have responsibility for assessing the client and ordering the equipment, and there is a network of stores staff who actually supply it; social worker staff, directly employed by the Social Work Department of the City of Edinburgh Council, and stores workers (storemen, stores administrators or the stores managers) are also involved in this service.

The existing procedure for procuring equipment for a customer involves either a Social Worker, Occupational Therapist or Healthcare professional assessing the client's need and creating a Care Plan. Having then agreed it with the Customer their equipment needs, if any, are identified and then ordered using paper forms. These are then passed to stores who are responsible for the administration and moving of the equipment to the customer and reclaiming it when it is no longer required.

### ***4.3 Mapping to SmartGov platform entities***

Within this task, the requirements for each service are mapped to SmartGov platform entities. The following paragraphs provide details on the mappings for the two pilot services.

#### **4.3.1 The GSIS pilot**

##### **4.3.1.1 Transaction service**

A single transaction service is defined, named **EVAT\_AQ**. The service has a single form set for the HTML platform, including two forms namely FORM\_EVAT\_AQ\_HEADER and FORM\_EVAT\_AQ\_DETAIL.

##### **4.3.1.2 Forms**

The form FORM\_EVAT\_AQ\_HEADER includes the TSEs listed in section 4.3.1.3.1. The form FORM\_EVAT\_AQ\_DETAIL includes a single TSE group, as described in section 0.

##### **4.3.1.3 TSEs**

###### ***4.3.1.3.1 Header form***

<b>TSE name</b>	<b>TSE semantics</b>
TSE_EVAT_DCL_NO	A number uniquely identifying the statement. The number is system-assigned.

<b>TSE name</b>	<b>TSE semantics</b>
TSE_EVAT_SUBM_DATE	The date on which the statement is submitted.
TSE_EVAT_RECEPTION_DATE	The date on which the statement is received. In a paper-based environment this may be different than the submission date; in an electronic environment it is always the same, it was however decided to retain the field for uniformity purposes.
TSE_EVAT_TAX_OFFICE	The tax office to which the submitting taxable person has been registered.
TSE_EVAT_RECEIVING_TAX_OFFICE	The tax office by which the statement is being received. A special code has to be standardised to represent the "electronic tax office"
TSE_EVAT_PERIOD_BEGIN	The beginning of the period for which the statement reports.
TSE_EVAT_PERIOD_END	The end of the period for which the statement reports.
TSE_EVAT_CURRENCY	An indication on whether the currency used in the statement is "Euro". Since all initial statements (as opposed to corrective statements) should be filled-in using the Euro currency, the indication should be always set to reflect this fact.
TSE_EVAT_IS_CORRECTIVE	An indication on whether the statement is corrective. Since the pilot will only accept initial statements, the indication will be always set to indicate that the statement is an initial one.
TSE_EVAT_TRIMESTER	The serial number of the trimester reported on (1 = Jan to Mar, 2 = Apr to Jun and so on).
TSE_EVAT_YEAR	The year reported on.
TSE_EVAT_REG_AFM	The VAT number of the taxable person.
TSE_EVAT_REG_COMPANY_TITLE	The company title of the taxable entity; for individuals, the surname and name are entered in this field.

<b>TSE name</b>	<b>TSE semantics</b>
TSE_EVAT_REG_ADDRESS	The postal address of the taxable entity.
TSE_EVAT_REG_TK	The zip code of the taxable entity
TSE_EVAT_REG_AREA	The area of the taxable entity's premises.
TSE_EVAT_REG_PHONE	The phone number of the taxable entity.
TSE_EVAT_REG_FAX	The fax number of the taxable entity.
TSE_EVAT_FILE_NO	The page serial number, when a paper document comprising of multiple pages is submitted. In the electronic version, a single page will be always used, and this number will be set to one.

The values of the fields on the first page are system-assigned or retrieved for registries, and the submitting taxable person should not be able to change the values provided; they are displayed however, to ascertain the submitting person that the correct values are used.

#### **4.3.1.3.2Detail form**

The detail form contains the data for the transactions conducted by the taxable person within the reporting period. For each transaction peer, a number of inputs should be provided, which are mapped into TSEs as follows:

<b>TSE name</b>	<b>TSE semantics</b>
TSE_EVAT_DETAIL_COUNTRY_PREFIX	The two-letter abbreviation of the country to whose taxation authorities the transaction peer is registered.
TSE_EVAT_DETAIL_AFM	The VAT number of the transaction peer.
TSE_EVAT_DETAIL_SUPPLIES	The net value of all transactions conducted with the specific peer within the reporting period.
TSE_EVAT_DETAIL_TRIANGULAR_SUPPLIES	The net value of all triangular transactions involving the specific peer as an intermediate destination.

Since these inputs should be provided multiple times (once for each transaction peer), the respective TSEs are placed within a TSE group that is named TSEG\_EVAT\_DETAIL.

For each transaction peer reported on, the country prefix and the VAT identification number are mandatory; additionally at least one of the net values should be provided, since there is no meaning in reporting on an entity with zero transactions value and zero triangular transactions value. Moreover, a specific transaction peer should be reported at most once within a statement, i.e. there should not be two "lines" in the statement with the same values **both** for the country prefix **and** the VAT identification number.

Additionally, the detail form should contain two auto-calculated fields giving the sum of the columns in which the values of supplies and triangular supplies are declared. The taxable person's VAT number is also repeated in this form for the user's reference and convenience. These details are mapped to the following TSEs:

<b>TSE name</b>	<b>TSE semantics</b>
TSE_EVAT_DETAIL_SUPPLIES_SUM	The sum of the net transaction values for all transaction peers.
TSE_EVAT_DETAIL_TRIANGULAR_SUPPLIES_SUM	The sum of the net triangular transaction values for all transaction peers.
TSE_EVAT_DETAIL_REG_AFM	The taxable entity's VAT number.

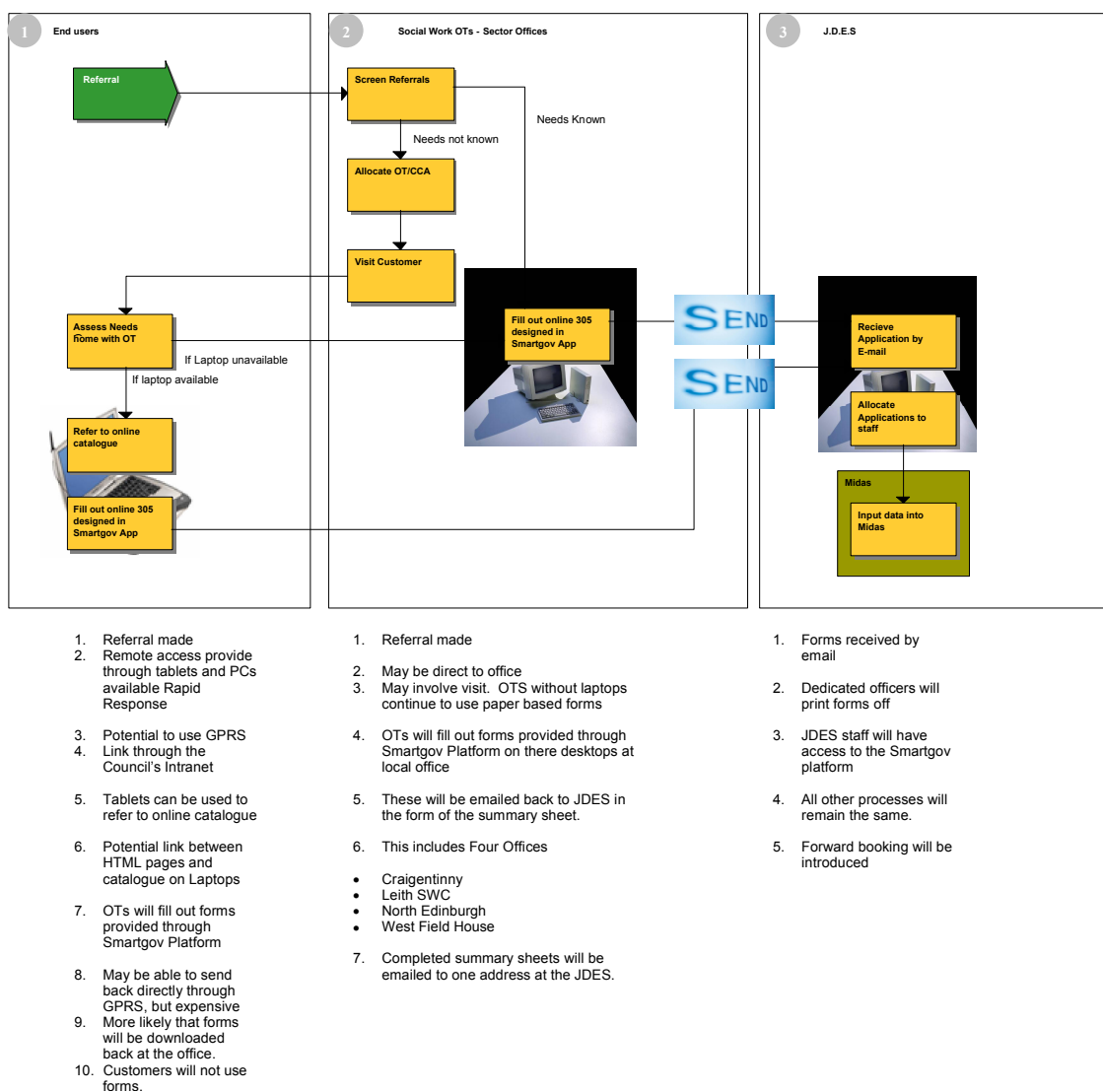
#### 4.3.1.4 KUs

The service as a whole will be associated with a single KU, which will contain the updated version of the legislation in effect that pertains to the service. For the header form, only one KU is needed, stating that the individual fields cannot be changed through this service and providing brief explanations for the field semantics. For the detail form, each field that can be filled in by the user should have its own KU explaining its semantics, while an additional KU may be placed on the form giving a thorough example of how this form is filled in.

Regarding the naming conventions, for elements (TSEs, TSE groups, forms or the service as a whole) needing to have a KU associated with them, the respective KU identifier should be formed by appending the string `_KU` to the TSE name. For instance, the KU associated with the TSE `TSE_EVAT_DETAIL_AFM` TSE should be named `TSE_EVAT_DETAIL_AFM_KU`.

### 4.3.2 The CEC pilot

The CEC pilot will be an electronic implementation of the “305 form” used by the CEC. The pilot will only include the offices based in the North Sector and the JDES (Joint Domiciliary Equipment Stores). Only CEC staff will take part in the pilot although referrals maybe received from any of the Council’s associated partners in the delivery of the equipment services. The pilot overview is illustrated in Figure 4.



**Figure 4 – CEC pilot service overview**

For the design phase of the CEC pilot, the following human resources were involved:

- 1 JDES (Joint Domiciliary Equipment Stores) Manager
- 2 Senior OTs (Occupational Therapists)

- 2 e-Government Consultants
- 1 SmartGov Specialist – Napier University

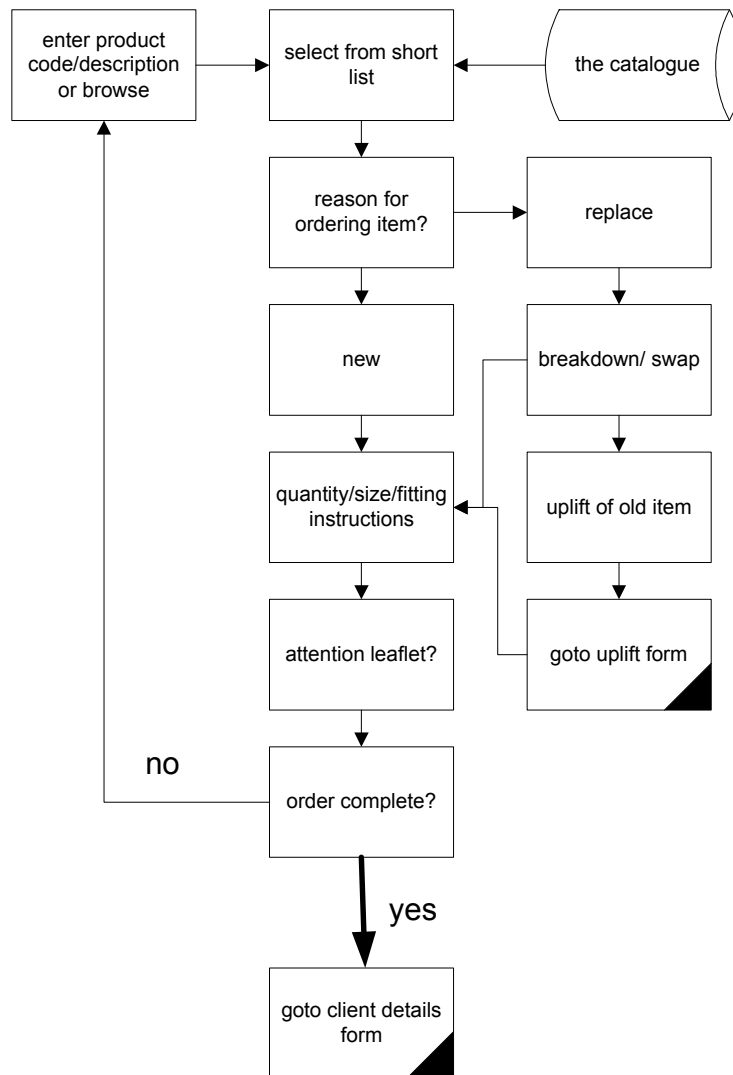
The CEC pilot has an workflow comprising of three major phases, and each phase includes a number of actions. At least one action per phase will be supported by the SmartGov platform and more specifically:

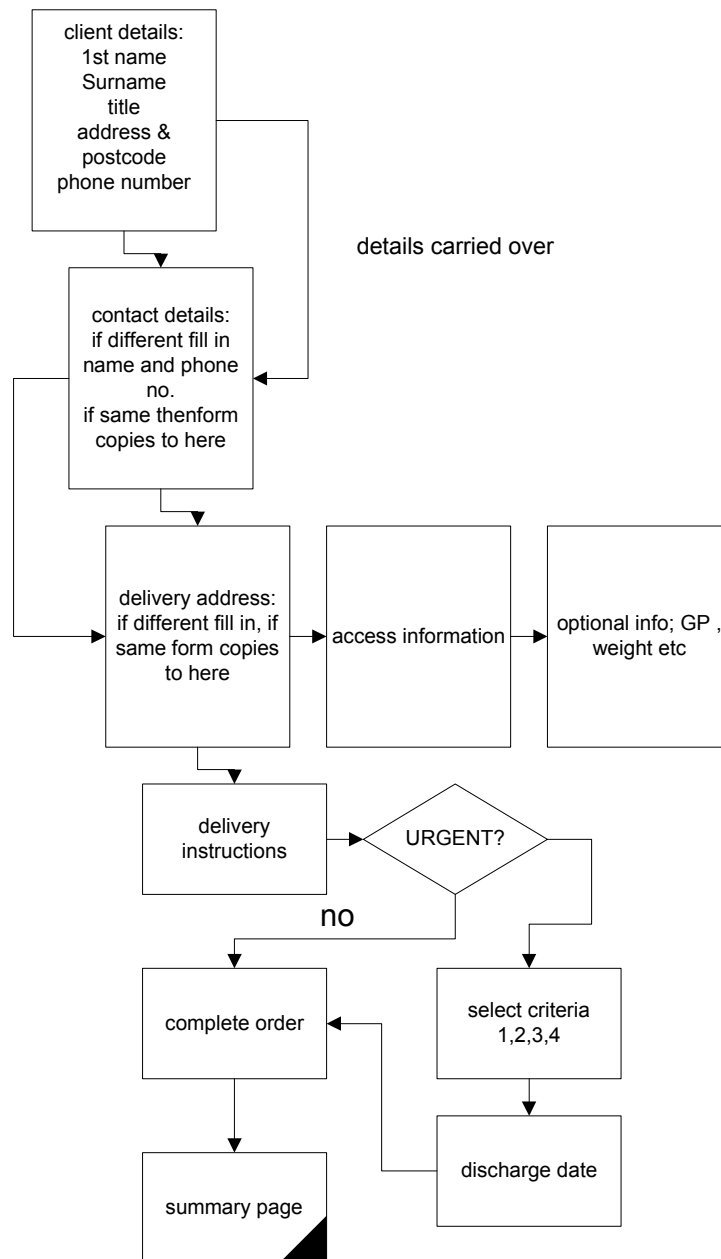
1. In phase (1), visiting occupational therapists will fill out forms provided by the SmartGov platform.
2. In phase (2), social work occupational therapists will fill out forms provided by the SmartGov platform.
3. In phase (3), JDES staff will fill out forms provided by the SmartGov platform the SmartGov platform.

It is important to note that forms filled-in in some phase will process data from forms submitted in previous stages; e.g. forms filled-in in stage (2) will use data from forms filled-in in phase (1).

The important TSEs and the alternative paths involved in the service design are depicted in Figure 5 and Figure 6 (Figure 6 begins at the stage that Figure 5 ends). Alternative paths are important in the service design, since they will provide input for the implementation of the validation checks in the phase of implementation.



**Figure 5 – Logical diagram for the CEC pilot service part 1**



**Figure 6 - Logical diagram for the CEC pilot service part 2**

The full list of TSEs is given in Table 11:

**Table 11 - TSEs for the Edinburgh pilot**

<b><i>TSE #</i></b>	<b><i>TSE name</i></b>	<b><i>Form</i></b>	<b><i>Semantics</i></b>
1.	ITEM_NUMBER	Order page #1	The sequence number of an ordered item
2.	ITEM_DESC	Order page #1	The description of the ordered item
3.	ITEM_QUANTITY	Order page #1	The quantity in which an item is ordered
4.	ITEM_INFO	Order page #1	Any information regarding the order of an item
5.	ITEM_ATTN_LEAFLET	Order page #1	An indication whether the ordered item has attention leaflet
6.	SUM_ITEM_NUMBER	Order page #2	The sequence number of an ordered item in the summary page
7.	SUM_ITEM_DESC	Order page #2	The description of an ordered item in the summary page
8.	SUM_ITEM_QUANTITY	Order page #2	The quantity in which an item is ordered in the summary page
9.	SUM_ITEM_INFO	Order page #2	Any information regarding the order of an item in the summary page
10.	SUM_ITEM_ATTN_LEAFLET	Order page #1	An indication whether the ordered item has attention leaflet in the summary page
11.	CLIENT_TITLE	Client page #1	The title of the client (Mr, Ms, etc)
12.	CLIENT_FIRSTNAME	Client page #1	The first name of the client
13.	CLIENT_LASTNAME	Client page #1	The last name of the client
14.	CLIENT_ADDRESS1	Client page #1	The first portion of the client's address
15.	CLIENT_ADDRESS2	Client page #1	The second portion of the client's address
16.	CLIENT_POSTCODE	Client page #1	The client's postcode
17.	CLIENT_CITY	Client page #1	The client's city
18.	CLIENT_TEL	Client page #1	The client's telephone
19.	CONTACT_TITLE	Client page #1	The title of the contact (Mr, Ms, etc)
20.	CONTACT_FIRSTNAME	Client page #1	The first name of the contact
21.	CONTACT_LASTNAME	Client page #1	The last name of the contact

<b>TSE #</b>	<b>TSE name</b>	<b>Form</b>	<b>Semantics</b>
22.	CONTACT_ADDRESS1	Client page #1	The first portion of the contact's address
23.	CONTACT_ADDRESS2	Client page #1	The second portion of the contact's address
24.	CONTACT_POSTCODE	Client page #1	The contact's postcode
25.	CONTACT_CITY	Client page #1	The contact's city
26.	CONTACT_TEL	Client page #1	The contact's telephone
27.	DELIVERY_TITLE	Client page #1	The title of the delivery (Mr, Ms, etc)
28.	DELIVERY_FIRSTNAME	Client page #1	The first name of the delivery
29.	DELIVERY_LASTNAME	Client page #1	The last name of the delivery
30.	DELIVERY_ADDRESS1	Client page #1	The first portion of the delivery's address
31.	DELIVERY_ADDRESS2	Client page #1	The second portion of the delivery's address
32.	DELIVERY_POSTCODE	Client page #1	The delivery's postcode
33.	DELIVERY_CITY	Client page #1	The delivery's city
34.	DELIVERY_TEL	Client page #1	The delivery's telephone
35.	CONTACT_COPY_CLIENT	Client page #1	A convenience TSE for using the client's details as contact details, instead of typing them in again
36.	DELIVERY_COPY_CLIENT	Client page #1	A convenience TSE for using the client's details as delivery details, instead of typing them in again
37.	URGENCY_0_INDICATION	Client page #2	An indication whether the order is characterised as "not urgent"
38.	URGENCY_0_CRITERIA	Client page #2	The grounds on which the order is characterised as "not urgent"
39.	URGENCY_0_DISCH_DATE	Client page #2	The discharge date for a "not urgent" order
40.	URGENCY_1_INDICATION	Client page #2	An indication whether the order is assigned an urgency rating equal to 1
41.	URGENCY_1_CRITERIA	Client page #2	The grounds on which the order is assigned an urgency rating equal to 1

<b>TSE #</b>	<b>TSE name</b>	<b>Form</b>	<b>Semantics</b>
42.	URGENCY_1_DISCH_DATE	Client page #2	The discharge date for an order that has been assigned an urgency rating equal to 1
43.	URGENCY_2_INDICATION	Client page #2	An indication whether the order is assigned an urgency rating equal to 2
44.	URGENCY_2_CRITERIA	Client page #2	The grounds on which the order is assigned an urgency rating equal to 2
45.	URGENCY_2_DISCH_DATE	Client page #2	The discharge date for an order that has been assigned an urgency rating equal to 2
46.	URGENCY_3_INDICATION	Client page #2	An indication whether the order is assigned an urgency rating equal to 3
47.	URGENCY_3_CRITERIA	Client page #2	The grounds on which the order is assigned an urgency rating equal to 3
48.	URGENCY_3_DISCH_DATE	Client page #2	The discharge date for an order that has been assigned an urgency rating equal to 3
49.	URGENCY_4_INDICATION	Client page #2	An indication whether the order is assigned an urgency rating equal to 4
50.	URGENCY_4_CRITERIA	Client page #2	The grounds on which the order is assigned an urgency rating equal to 4
51.	URGENCY_4_DISCH_DATE	Client page #2	The discharge date for an order that has been assigned an urgency rating equal to 4
52.	FIN_SUM_ORD_REF	Final summary	The order reference
53.	FIN_SUM_CLNT_NAME	Final summary	The title, first name and surname of the client in the final summary.
54.	FIN_SUM_CLNT_ADDRESS1	Final summary	The first portion of the client's address in the final summary.

<b><i>TSE #</i></b>	<b><i>TSE name</i></b>	<b><i>Form</i></b>	<b><i>Semantics</i></b>
55.	FIN_SUM_CLNT_ADDRESS2	Final summary	The second portion of the client's address in the final summary.
56.	FIN_SUM_CLNT_CITY	Final summary	The city and the zip code of the client in the final summary.
57.	FIN_SUM_CLNT_PHONE	Final summary	The client's phone in the final summary.
58.	FIN_SUM_CONTACT_NAME	Final summary	The title, first name and surname of the contact in the final summary.
59.	FIN_SUM_CONTACT_ADDRESS1	Final summary	The first portion of the contact's address in the final summary.
60.	FIN_SUM_CONTACT_ADDRESS2	Final summary	The second portion of the contact's address in the final summary.
61.	FIN_SUM_CONTACT_CITY	Final summary	The city and the zip code of the contact in the final summary.
62.	FIN_SUM_CONTACT_PHONE	Final summary	The contact's phone in the final summary.
63.	FIN_SUM_DELIVERY_NAME	Final summary	The title, first name and surname of the delivery in the final summary.
64.	FIN_SUM_DELIVERY_ADDRESS1	Final summary	The first portion of the delivery's address in the final summary.
65.	FIN_SUM_DELIVERY_ADDRESS2	Final summary	The second portion of the delivery's address in the final summary.
66.	FIN_SUM_DELIVERY_CITY	Final summary	The city and the zip code of the delivery in the final summary.
67.	FIN_SUM_DELIVERY_PHONE	Final summary	The delivery's phone in the final summary.
68.	FIN_SUM_ACCESS_INSTR	Final summary	Any access instructions for the order.
69.	FIN_SUM_DELIV_INSTR	Final summary	Any delivery instructions for the order.
70.	FIN_SUM_ITEM_NUMBER	Final summary	The sequence number of an ordered item
71.	FIN_SUM_ITEM_DESC	Final summary	The description of an ordered item
72.	FIN_SUM_ITEM_QUANTITY	Final summary	The quantity in which an item is ordered

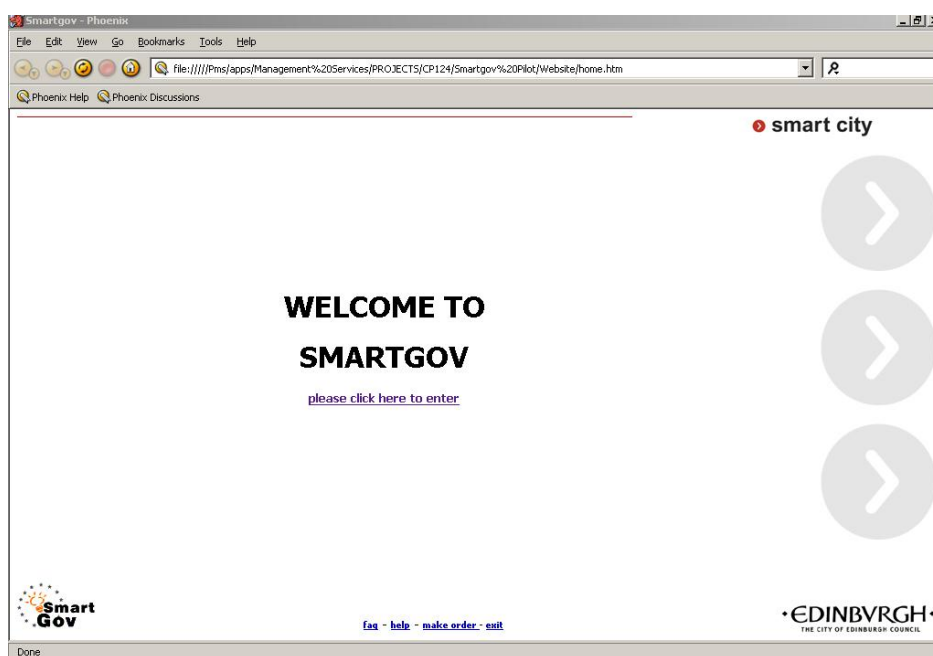
<b>TSE #</b>	<b>TSE name</b>	<b>Form</b>	<b>Semantics</b>
73.	FIN_SUM_ITEM_INFO	Final summary	Any information regarding the order of an item
74.	FIN_SUM_ITEM_ATTN_LEAFLET	Final summary	An indication whether the ordered item has attention leaflet

TSEs (#1, #2, #3, #4, #5) may repeat in "Order page 1" as many times as the number of items that are ordered; to this end they constitute a repeating TSE group. The same arrangement is made for TSEs (#6, #7, #8, #9, #10) and (#70, #71, #72, #73, #74).

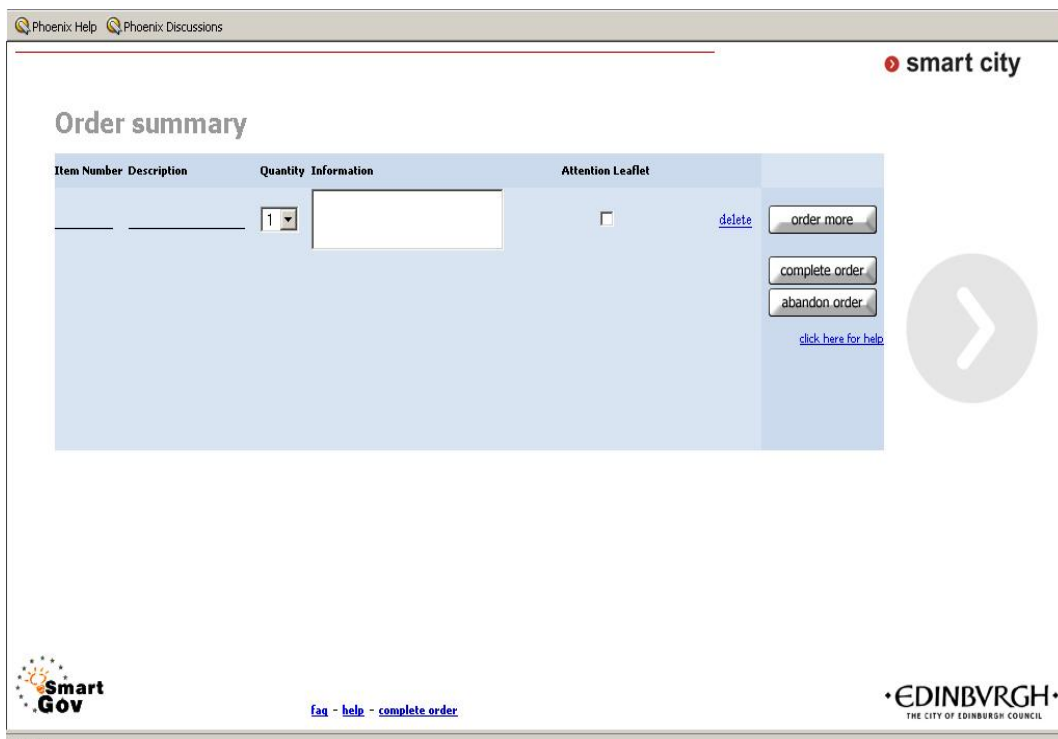
As far as knowledge units are concerned, all forms will be linked with a number of generic KUs describing general aspects of the service. More specific KUs will be available in selected forms or form items.

Regarding connectivity to back-end information systems, the code items and descriptions should be retrievable from a database.

The design of HTML forms, finally, has been completed. Illustrations from the HTML forms are provided in Figure 7 to Figure 11.



**Figure 7 - CEC pilot welcome screen**



Phoenix Help Phoenix Discussions

smart city

### Order summary

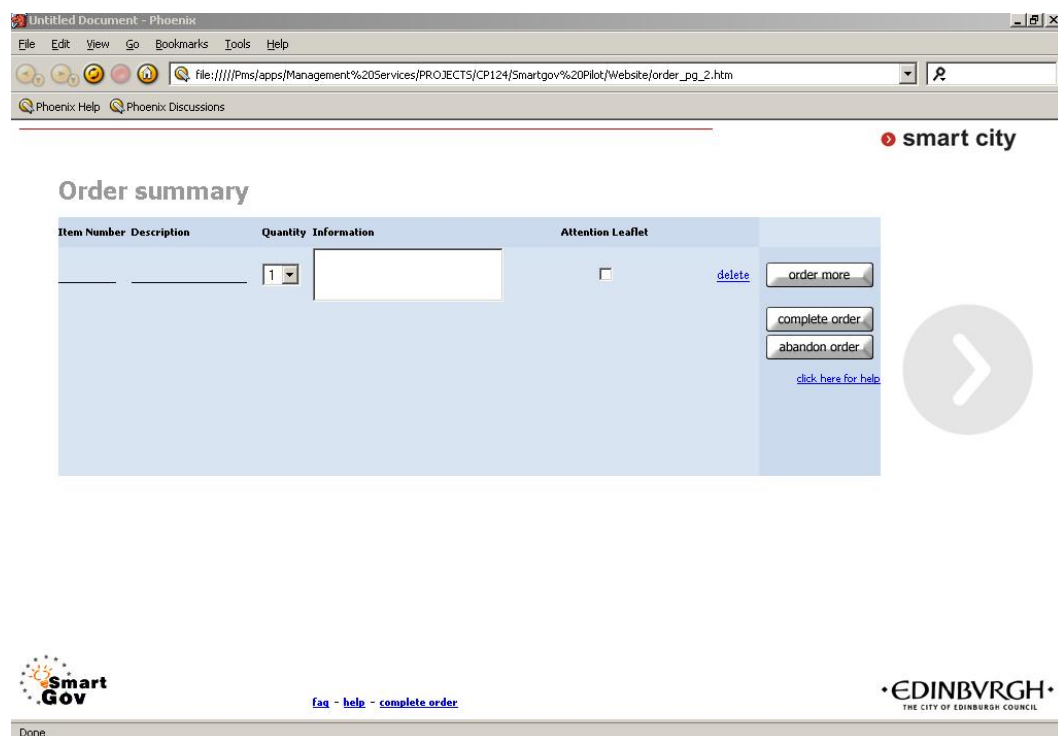
Item Number	Description	Quantity	Information	Attention Leaflet	
		1		<input type="checkbox"/>	<a href="#">delete</a> <a href="#">order more</a> <a href="#">complete order</a> <a href="#">abandon order</a> <a href="#">click here for help</a>

Smart Gov

[faq](#) - [help](#) - [complete order](#)

EDINBURGH  
THE CITY OF EDINBURGH COUNCIL

Figure 8 - CEC pilot order page #1



Untitled Document - Phoenix

File Edit View Go Bookmarks Tools Help

File://///Pms/apps/Management%20Services/PROJECTS/CP124/Smartgov%20Pilot/Website/order\_pg\_2.htm

Phoenix Help Phoenix Discussions

smart city

### Order summary

Item Number	Description	Quantity	Information	Attention Leaflet	
		1		<input type="checkbox"/>	<a href="#">delete</a> <a href="#">order more</a> <a href="#">complete order</a> <a href="#">abandon order</a> <a href="#">click here for help</a>

Smart Gov

[faq](#) - [help](#) - [complete order](#)

EDINBURGH  
THE CITY OF EDINBURGH COUNCIL

Done

Figure 9 - CEC pilot order page #2



Untitled Document - Phoenix

File Edit View Go Bookmarks Tools Help

file:///Pms/apps/Management%20Services/PROJECTS/CP124/Smartgov%20Pilot/Website/client\_pg\_1.htm

Phoenix Help Phoenix Discussions

smart city

Client details

Title  Firstname  Surname

Address 1

Address 2

Postcode  City/Town

Tel.

Contact details copy client details: ☐

Title  Firstname  Surname

Address 1

Address 2

Postcode  City/Town

Tel.

Delivery details copy client details: ☐

Title  Firstname  Surname

Address 1

Address 2

Postcode  City/Town

Tel.

Done

Figure 10 - CEC pilot clientpage #1

Untitled Document - Phoenix

File Edit View Go Bookmarks Tools Help

file:///Pms/apps/Management%20Services/PROJECTS/CP124/Smartgov%20Pilot/Website/client\_pg\_2.htm

Phoenix Help Phoenix Discussions

smart city

Other details

	Criteria	Discharge date
<input type="radio"/> Not urgent	<input type="text"/>	<input type="text"/>
<input type="radio"/> 1	<input type="text"/>	<input type="text"/>
<input type="radio"/> 2	<input type="text"/>	<input type="text"/>
<input type="radio"/> 3	<input type="text"/>	<input type="text"/>
<input type="radio"/> 4	<input type="text"/>	<input type="text"/>

Level of urgency

submit order

Smart Gov

[faq](#) - [help](#) - [abandon order](#)

EDINBURGH  
THE CITY OF EDINBURGH COUNCIL

Done

Figure 11 - CEC pilot clientpage #2

Untitled Document - Phoenix

File Edit View Go Bookmarks Tools Help

file:///Pms/apps/Management%20Services/PROJECTS/CP124/Smartgov%20Pilot/Website/client\_pg\_2.htm

Phoenix Help Phoenix Discussions

smart city

Other details

Level of urgency

☐ Not urgent  
☐ 1  
☐ 2  
☐ 3  
☐ 4

Criteria	Discharge date

submit order

Smart Gov

[faq](#) - [help](#) - [abandon order](#)

EDINBURGH  
THE CITY OF EDINBURGH COUNCIL

Done

Figure 12 - CEC pilot final summary page

## 4.4 Service implementation

In the following paragraphs relevant comments and issues regarding the population of the services data (KUs, TSEs, TSE groups etc) in the PAs are reported. This includes tasks 3.1 to 3.8. Reporting does not rigidly follow the task list, since some tasks have been interweaved by the implementers (e.g. while a domain expert implemented a TSE or a TSE group, she usually entered the respective validation checks as well, as an indivisible part of the TSE/TSE group creation).

### 4.4.1 The GSIS pilot

#### 4.4.1.1 TSEs, TSE groups and validation checks

##### 4.4.1.1.1 Header form

The details for the TSEs appearing in the header form are summarized in the following table:

<b>TSE name</b>	<b>TSE details</b>
TSE_EVAT_DCL_NO	Type = number (assigned dynamically by the system) read only
TSE_EVAT_SUBM_DATE	Type = date (assigned dynamically by the system) Read only
TSE_EVAT_RECEPTION_DATE	Type=date (assigned dynamically by the system) Read only
TSE_EVAT_TAX_OFFICE	Type = string Maxlen = 4 Alphabet = 0123456789 Read only
TSE_EVAT_RECEIVING_TAX_OFFICE	Type = string Maxlen = 4 Alphabet = 0123456789 Read only
TSE_EVAT_PERIOD_BEGIN	Type = date (assigned dynamically by the system) Read only
TSE_EVAT_PERIOD_END	Type = date (assigned dynamically by the system) Read only
TSE_EVAT_CURRENCY	Type = Boolean (fixed value = "true") Preferred widget = check box Read only
TSE_EVAT_IS_CORRECTIVE	Type = Boolean (fixed value = "false") Preferred widget = check box Read only
TSE_EVAT_TRIMESTER	Type = integer Min value = 1 Max value = 4 Read only

<b>TSE name</b>	<b>TSE details</b>
TSE_EVAT_YEAR	Type = integer Min value = 2003 Read only
TSE_EVAT_REG_AFM	Type = Tax record number Maxlen = 9 Alphabet = 0123456789 <i>Check-digit check</i> Read only
TSE_EVAT_REG_COMPANY_TITLE	Type = string Maxlen = 60 Read only
TSE_EVAT_REG_ADDRESS	Type = string Maxlen = 60 Read only
TSE_EVAT_REG_TK	Type = string Maxlen=5 Alphabet = 0123456789 Read only
TSE_EVAT_REG_AREA	Type = string Maxlen=40 Alphabet = 0123456789 Read only
TSE_EVAT_REG_PHONE	Type = string Maxlen=20 Alphabet = 0123456789 Read only
TSE_EVAT_REG_FAX	Type = string Maxlen=20 Alphabet = 0123456789 Read only
TSE_EVAT_FILE_NO	Type = integer Fixed value = 1 Read only

**4.4.1.1.2Detail form**

The detail form contains a TSE group named TSEG\_EVAT\_DETAIL. This group contains four TSEs, namely TSE\_EVAT\_DETAIL\_COUNTRY\_PREFIX, TSE\_EVAT\_DETAIL\_AFM, TSE\_EVAT\_DETAIL\_SUPPLIES and TSE\_EVAT\_DETAIL\_TRIANGULAR\_SUPPLIES. The TSE group repeats with *minOccurrences* = 0, *maxOccurrences* = 1000, *initialRows* = 0, *uniqueColumn* = {TSE\_EVAT\_DETAIL\_COUNTRY\_PREFIX, TSE\_EVAT\_DETAIL\_AFM}, *groupControls*=true, *rowStep* =1. The following validation checks are defined for this TSE group:

## 1. Full rule:

```

CONDITION: (TSE_EVAT_DETAIL_SUPPLIES      <>      0)      AND
            (TSE_EVAT_DETAIL_TRIANGULAR_SUPPLIES  <>      0)      AND
            (length(TSE_EVAT_DETAIL_COUNTRY_PREFIX) = 0)      OR
            length(TSE_EVAT_DETAIL_AFM) = 0)

ACTION:     errorMessage("en_gb:: You must provide both the
                        country prefix and the VAT number||el: Πρέπει να
                        εισάγετε τη συντομογραφία χώρας και τον Α.Φ.Μ.")

```

## 2. Custom code check: implement the validation checks for the pair (TSE\_EVAT\_DETAIL\_COUNTRY\_PREFIX, TSE\_EVAT\_DETAIL\_AFM)

Recall that the detail form contains two auto-calculated fields giving the sum of the columns in which the values of supplies and triangular supplies are declared, while the taxable person's VAT number is repeated again in this form. The detail form TSE data are summarised in the following table.

TSE name	TSE details
TSE_EVAT_DETAIL_COUNTRY_PREFIX	Type = String Allowable values <sup>2</sup> : AT - Austria BE - Belgium DE - Deutschland DK - Denmark ES - Spain FI - Finland FR - France GB - Great Britain IE - Ireland IT - Italy LU - Luxemburg NL - Netherlands PT - Portugal SE - Sweeden Mandatory
TSE_EVAT_DETAIL_AFM	Type = string Maxlen=20
TSE_EVAT_DETAIL_SUPPLIES	Type = money (or float)
TSE_EVAT_DETAIL_TRIANGULAR_SUPPLIES	Type = money (or float)
TSE_EVAT_DETAIL_SUPPLIES_SUM	Type = money (or float) Computation rule: sum(TSE_EVAT_DETAIL_SUPPLIES)
TSE_EVAT_DETAIL_TRIANGULAR_SUPPLIES_SUM	Type = money (or float) Computation rule: sum(TSE_EVAT_DETAIL_TRIANGULAR_SUPPLIES)
TSE_EVAT_DETAIL_REG_AFM	Type = Tax record number Maxlen = 9 Alphabet = 0123456789 Check-digit check Read only

#### 4.4.1.2 KUs

The texts for the KUs described in the previous section were extracted, entered into the platform and associated with the respective SmartGov platform items (TSEs, TSE groups, forms and service). Domain experts liaised with service workers to determine the most appropriate form of the KU texts, so as service end-users would easily perceive the information communicated to them. KUs were also linked to the organisational taxonomy.

---

<sup>2</sup> The descriptive texts (i.e. the full names for the countries) are actually provided in Greek, English and Spanish; only the English resource is given here for brevity reasons.

#### 4.4.1.3 Form layout

In order to determine the form layout several parameters were examined, with the most important being the nature of the form (static vs. dynamic) and the respective paper form layout. It was decided to follow the paper form layout as closely as possible, since this would allow users to capitalise on the experience gained from filling in and submitting paper forms, decreasing thus the time needed for end-users to learn how to interact with the service. The header form thus, which is static in nature, was modelled very closely after the respective area of the paper-based form. The detail form, on the other hand, is more dynamic, since the number of detail lines is not a priori known. The option to include a large number of lines, sufficient for all cases, was rejected, since the resulting XHTML form would be very large, increasing thus download time, and impractical. A small, fixed number of lines, on the other hand, would preclude some taxable entities from using the service. It was consequently decided to provide a small initial number of lines together with the ability to dynamically add and remove lines as needed. The order of the elements within each line should remain identical to the order in the paper form, to avoid user confusion. Forms were prepared by domain experts, using the DreamWeaver MX tool. The header form layout is depicted in Figure 13, while the detail form layout is depicted in Figure 14.

**Smart Gov**

Αναζήτηση  
Αρχική Σελίδα  
Στατιστικά στοιχεία

Χρήστης: Stelios Gerogiannakis  
Δήλητ. Υποβολής: 31 Δεκ 2999

**Επιβεβαίωση E-VAT**

Φόρμα επιβεβαίωσης αποκτήσεων E-VAT

Αριθμός δήλωσης 003 123456789  
Ημερομηνία υποβολής της δήλωσης 004 02/10/2003  
Ημερομηνία παραλαβής 005 02/10/2003  
Ενδειξη διορθωτικής δήλωσης 007 ☐

Κωδικός φορολογικής υπηρεσίας 001 233  
ΔΟΥ παραλαβής 002 455  
Νόμισμα του E-VAT 024 ☒  
Τρίμηνο υποβολής / Έτος υποβολής 008 1 / 2003  
Περίοδος υποβολής 006 Από 02/10/2003 Έως 02/10/2003

**Στοιχεία υπόχρεου**

Επωνυμία επιχείρησης 010 SmartGov  
Διεύθυνση 014 Εγγυρά 126 Καλλιθέα  
Τηλέφωνο 017 2101234567  
ΑΦΜ 021 9730280  
Αριθμός φακέλου 022 1

Περιοχή 015 Καλλιθέα  
Φαξ 018 2107654321  
Ταχυδρομικός κώδικας 016 18450

**Figure 13 – Header form for the e-Vies service**



Αναζήτηση  
Αρχική Σελίδα  
Στατιστικά στοιχεία

Χρήστης: Stelios Gerogiannakis  
Λήξη Υψηρείας: 31 Δεκ 2999

Φόρμα λεπτομερειών αποκτήσεων E-VAT

Φόρμα λεπτομερειών αποκτήσεων E-VAT

Ομάδα TSE λεπτομερειών για το E-VAT			
Πρόθεμα χώρας BE - Βέλγιο	ΑΦΜ του προμηθευτή 123456	TSE για τις αποκτήσεις 10	TSE για τριγωνικές αποκτήσεις 20
FI - Φινλανδία	111111	15	30
<input type="button" value="Προσθήκη"/> <input type="button" value="Αφαίρεση"/>			
Αθροισμα αποκτήσεων			0,0
Αθροισμα τριγωνικών αποκτήσεων			0,0

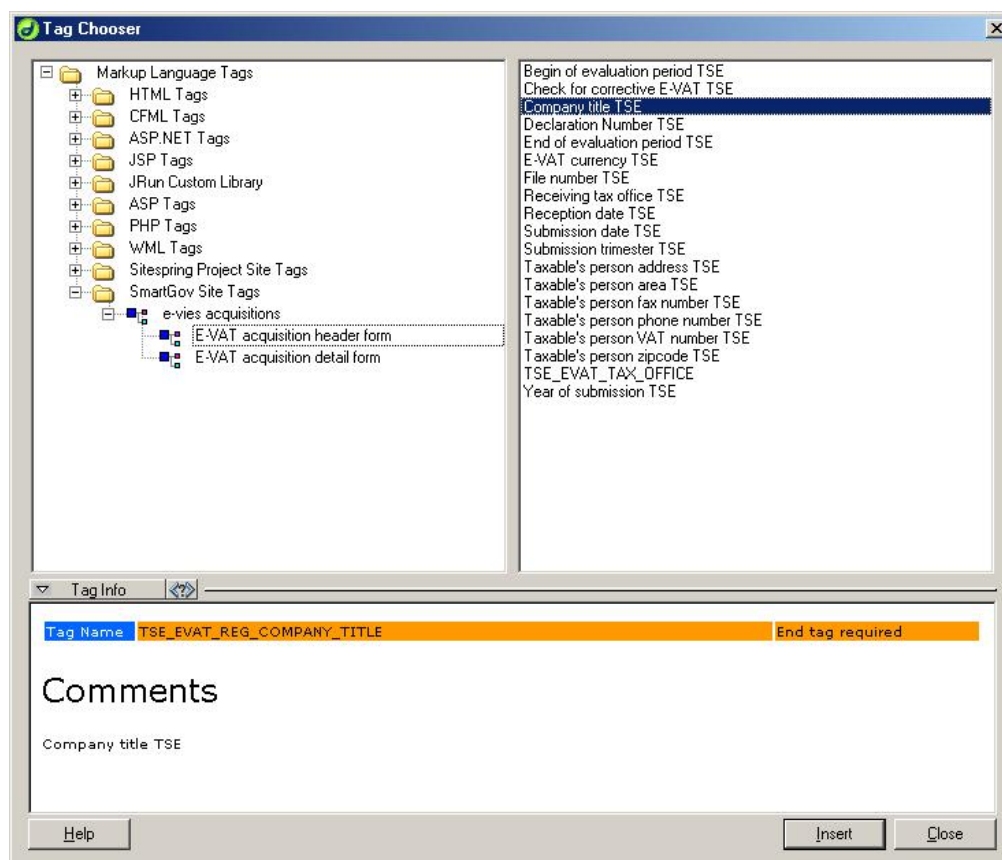
**Figure 14 – Detail form for the e-Vies service**

#### 4.4.1.4 Linkage of form layout to the SmartGov platform items

This portion of the work was performed by domain experts using the DreamWeaver MX tool and was completed in a straightforward manner. The users were able to select the visual part of the form using the mouse and then establish the correspondence to the respective SmartGov platform item by selecting the latter from a tree hierarchy. The only negative comment received is that the DreamWeaver MX tool automatically switches to HTML source view when a tag is inserted, thus the users would have to switch back to the rendered view. This issue, however, is simply a “nuisance” owing to the behaviour of the DreamWeaver MX tool, and does not preclude users from performing the work. It has to be noted that one domain expert, who was familiar with HTML syntax, performed the work entirely within the HTML code view (switching to the rendered view only for testing purposes) and thus did not find this issue annoying.

Figure 15 provides a sample screen from the linkage phase.





**Figure 15 - Linkage of visual form layout to the SmartGov platform items**

#### 4.4.1.5 Definitions of statistics

The statistics that were requested by the managerial level included only the number of submitted documents. In order to further test the platform's functionality, statistics have been defined to calculate the number of failed submissions (due to violation of constraints in the values provided), and the number of failures for specific validation checks. The statistics were easily defined by the respective SmartGov platform stakeholders through the SmartGov front-end.

#### 4.4.1.6 Linkage to external information systems

The GSIS pilot needs to retrieve data from external registries as well as populate relational DBMSs with the data submitted by the users. More specifically:

1. When a user requests to fill in a new document, certain information have to be extracted from the taxation registry, including the taxable entity's VAT number, its name (company name or family and given name as appropriate), its postal

address, phone and fax numbers etc. This information is used to pre-populate elements of the various service forms.

2. When a user submits a document, the information contained within the document should be stored in a relational DBMS in a specific format. The DBMS contents are then used for further back-office processing.

The requested functionality has been implemented by the IT staff of GSIS as two Java modules that attach to the IIG. More specifically, the `IIGCreateEVAT_AQDocument` has been coded to query the registry database and prepare an initial document, in which certain fields will be pre-populated with the values retrieved from the registry. The module `IIGStoreEVAT_AQDocument` module, on the other hand, provides the functionality of storing the information contained within the submitted document into a relational DBMS. It has to be noted that the naming conventions used by the IIG simplify the attachment of the modules to the runtime environment, since only the location of the files containing the implementations have to be made known to the environment.

#### **4.4.1.7 Service deployment**

Once all aspects of the service had been put in place, the integrator was run to deploy the service to a test environment within the GSIS. The integration and deployment procedure was performed rapidly (less than one minute) and error-free.

#### **4.4.1.8 Service testing and improvement**

The service was tested by GSIS service workers and selected users (mainly professional accountants plus three individuals). The overall operation of the service was smooth with a small number of comments for improvement received. The comments were as follows:

1. A misspelling in the coding of a validation check precluded the submission of certain documents that were correct. The issue was corrected by editing the validation check and redeploying the service.
2. Two error messages were characterised as "difficult to comprehend" by service end-users. The messages were re-stated.
3. A number of comments on the aesthetic issues of the service were received, mainly related to field alignment and size as well as the use of colour. This was anticipated to some extent, since the domain experts that implemented the HTML forms had little or no experience in the usage of DreamWeaver (with the

exception of one domain expert who was rather fluent with the tool). The issues were easily corrected with the assistance of IT staff, who provided also some general tips for better form design, to be used by domain experts in their future projects. Colour use was also inconsistent (and, in some cases, unfortunate colour choices were made). This was tackled by providing a set of guidelines to the XHTML form designers, together with a library of pre-defined styles (a *cascading style sheet* file), which groups all formatting parameters into easily usable entities (named styles), facilitating uniform formatting and ease of adaptation.

#### **4.4.2 The CEC pilot**

The implementation of the CEC pilot is underway, thus definitive reports will be provided as an addendum to this deliverable. The CEC however has determined the details of the pilot deployment phase, which are as follows:

##### **4.4.2.1 Locations**

The pilot will be deployed in the North Sector and at the JDES premises with the following number of participants:

1. North Sector
  - a. 8 OTs (occupational therapists)
  - b. 4 Rapid Response Services
2. JDES
  - a. 1 dedicated officer for sorting referrals
  - b. 6 Clerical staff processing forms.

##### **4.4.2.2 Volumes (Only includes North sector)**

- Estimated number of customers per week – 250
- Estimated number of forms per week - 290

## 5 The Evaluation Process

We now consider the evaluation process for the SmartGov Platform and the framework for e-government services. As the full methodology behind the evaluation process shall be given in D9.1 we outline the process specifying the criteria for success.

We start in Section 5.1 by outlining the principles behind the evaluation process before giving the criteria. This itself constitutes a number of levels. Firstly there are overall success criteria for the SmartGov Platform as whole, that will be concerned with core aspects such as its technological implementation, reliability, and how well it meets the requirements for each of the five generic user roles defined in [D4.1]. These are Managers, Domain Experts, IT Staff, Service Workers and End Users.

In addition to the overall criteria there will be specific success criteria for each of the pilot sites (GSIS and CEC). These complement the generic technological and functional criteria with analysis of usability and user acceptance, and thus provide a complete picture of the performance of the platform. These criteria are elucidated in Sections 5.2 et seq.

### ***5.1 The Principles Behind the Evaluation***

The evaluation will adopt a two fold approach: Conformance to functional requirements and System Acceptability. The first set of criteria will be derived directly from the user requirements of Deliverable D4.1, of which there are 83. Thus the evaluation in this sense will be a measure of the extent to which each of the requirements has been met. These requirements were weighted; Mandatory or desirable, and this will be shown in the evaluation. The evaluation will show where each criterion has been, either "met in full", "met partially" or "not met". The second approach, system acceptance, is based on Nielsen's [Nielsen 1993] model of attributes of acceptance. The actual success criteria are developed in the two sub sections specific to the local situation of the pilot sites, where applicable the baseline metrics from which the success criteria are developed are included in Appendix B 0. Thus the two approaches of the evaluation are:

1. **Conformance to functional requirements:** based wholly on the 83 user requirements of D 4.1 this is an evaluation of the smartgov platform and framework in

terms of the extent to which those requirements are fulfilled.

2. **System Acceptance:** This is an evaluation of the platform and framework, based on the theoretical foundation of the Nielsen model of attributes of acceptance in combination with the success criteria developed at the pilot sites.

## ***5.2 SmartGov Overall Success Criteria***

### **5.2.1 Conformance to Functional Requirements:**

Here the evaluation will take the 83 Platform specific user requirements presented in deliverable D4.1, and examine the extent to which they have been met.

The extent is indicated by

- Fully met
- Partially met
- Not met

In D4.1 the requirements are weighted by importance (mandatory or desirable) and this is indicated for each requirement. The relevance of these requirements to the final evaluation will also be discussed in each case.

The user requirements are reproduced below.

**5.2.1.1 Platform specific user requirements.**

<div> <div>User Groups</div> <div>FUNCTIONAL SPECIFICATIONS</div> </div>	Managers		Domain Experts		IT Staff		Service workers		End users inside PA (Run.)	End users outside PA (Run.)	Mandatory /desirable
	Dev.	Run	Dev.	Run .	Dev.	Run .	Dev .	Run .			
User-friendly interface	√	√	√	√	√	√	√	√	√	√	Mandatory
Context-sensitive help functionality	√	√	√	√	√	√	√	√	√	√	Mandatory
Multilingual interface and content	√	√	√	√	√	√	√	√	√	√	Desirable
Multiple access	√		√		√	√	√	√	√	√	Mandatory
Authentication and access control mechanism	√	√	√	√	√	√	√	√	√	√	Mandatory
Back up facilities	√	√	√	√	√	√	√	√			Mandatory
View automated reports	√	√					√	√			Mandatory
Management of statistics	√	√	√	√	√	√	√	√			Mandatory
Management of predefined TSEs			√		√						Mandatory
Reuse and adaptation of previous work			√								Mandatory
Creation of new TSEs			√		√						Mandatory
Definition of constraints and validation checks			√		√						Mandatory
Modelling of inter-element relations			√		√						Mandatory

<div> <div>User Groups</div> <div>FUNCTIONAL SPECIFICATIONS</div> </div>	Managers		Domain Experts		IT Staff		Service workers		End users inside PA (Run.)	End users outside PA (Run.)	Mandatory /desirable
	Dev.	Run	Dev.	Run .	Dev.	Run .	Dev .	Run .			
Attachment of domain knowledge on form elements			√		√						Mandatory
Composition of manual and instructions for the end users			√		√						Mandatory
Composition of documentation about the implemented service			√		√						Mandatory
Connections with third party systems					√	√					Mandatory
Management of user accounts							√	√			Mandatory
Definition of the information to be recorded in the log files					√	√	√	√			Mandatory
Definition and editing of service process models	√		√				√				Desirable
Browsing of service process models	√		√				√				Desirable
Web-enabled interfaces	√	√	√				√	√	√	√	Desirable
Design, store and use standard look and feel for forms			√				√				Mandatory
Detect data duplication within a service or across services			√	√			√	√			Desirable

<div> <div>User Groups</div> <div>FUNCTIONAL SPECIFICATIONS</div> </div>	Managers		Domain Experts		IT Staff		Service workers		End users inside PA (Run.)	End users outside PA (Run.)	Mandatory /desirable
	Dev.	Run	Dev.	Run .	Dev.	Run .	Dev .	Run .			
Associate forms with time periods and specify related actions			√				√				Mandatory
Specify deadlines for document submission			√				√				Mandatory
Specify trigger conditions when certain dates arrive			√				√				Mandatory
Provide checks for dates that appear on forms (special case of "definition of constraints & validation checks")			√				√				Mandatory
Store PA worker details and associate them with services			√				√				Mandatory
Leverage client awareness for the new services			√					√			Desirable
View material associated with a form			√					√	√	√	Mandatory
Provide context sensitive guidance on form-filling			√		√			√			Mandatory
Associate help and guidance with forms and their uses			√				√				Mandatory
Provide guidance on scannability of forms			√				√				Desirable



<div> <div>User Groups</div> <div>FUNCTIONAL SPECIFICATIONS</div> </div>	Managers		Domain Experts		IT Staff		Service workers		End users inside PA (Run.)	End users outside PA (Run.)	Mandatory /desirable
	Dev.	Run	Dev.	Run .	Dev.	Run .	Dev .	Run .			
Automatically check forms for scannability			√				√				Desirable
View process model details									√	√	Mandatory
Access advice and guidance on good form design			√				√				Desirable
Assess actual use of data items			√	√			√	√			Desirable
Identify data on forms that may already be known			√								Mandatory
Design form formats readable by people with sight impairment			√								Mandatory
Maintain versions of forms			√				√				Mandatory
Access form-editing facilities			√				√				Mandatory
Access simple rules of thumb for identifying complicated text			√								Desirable
Select colours in user interfaces	√	√	√	√	√	√	√	√	√	√	Mandatory
Access guidance on readable colour combinations			√								Mandatory
Edit colours in form design			√								Mandatory
Select font sizes in end user interfaces	√	√	√	√	√	√	√	√	√	√	Mandatory

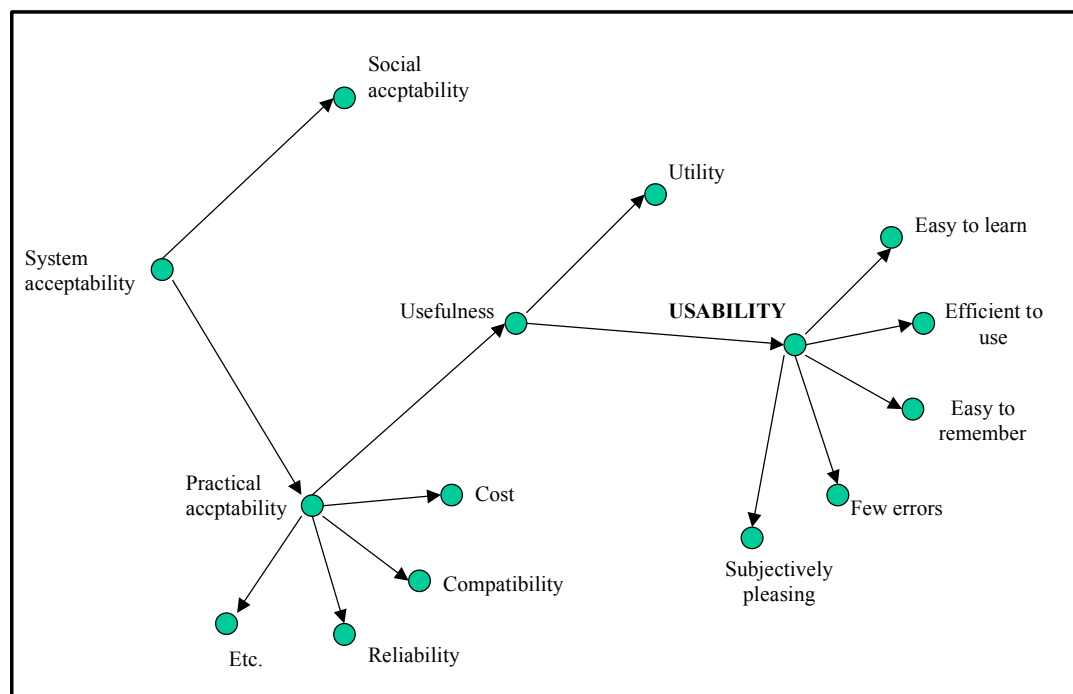
<div> <div>User Groups</div> <div>FUNCTIONAL SPECIFICATIONS</div> </div>	Managers		Domain Experts		IT Staff		Service workers		End users inside PA (Run.)	End users outside PA (Run.)	Mandatory /desirable
	Dev.	Run	Dev.	Run .	Dev.	Run .	Dev .	Run .			
Define font sizes in form design			√								Mandatory
Enter indications clearly show which data are required at each stage			√								Mandatory
View which data has been entered and which remains to be entered									√	√	Mandatory
Tag data input against each stage in a process			√								Mandatory
Receive email messages from end users								√			Mandatory
Include e-mail facilities in services			√								Mandatory
Specify notifications for interested parties (service workers and end users) about approaching deadlines			√					√			Mandatory
Provide means for users to enter dates without typing			√								Mandatory
Provide details of where to go for extra help if the user gets stuck			√					√			Mandatory
Provide print facilities so that paper versions of forms look the same as the online version			√					√			Mandatory

<div> <div>User Groups</div> <div>FUNCTIONAL SPECIFICATIONS</div> </div>	Managers		Domain Experts		IT Staff		Service workers		End users inside PA (Run.)	End users outside PA (Run.)	Mandatory /desirable
	Dev.	Run	Dev.	Run .	Dev.	Run .	Dev .	Run .			
Define/view how each piece of data will be used	√		√								Mandatory
Include graphics in form design			√				√				Mandatory
Include offline activities in process models			√								Mandatory
Save sessions to be resumed later									√	√	Mandatory
Switch from on-line to off-line mode and vice versa									√	√	Mandatory
Define when electronic signatures will be produced, accepted or validated			√								Mandatory
Define user classes and authority to access forms	√		√								Mandatory
Specify shortcut codes and their meanings			√								Desirable
Define system use metrics	√		√		√		√				Mandatory
View system use reports in comprehensible form (graph, tables etc)		√		√		√		√			Desirable
Design end-user surveys	√		√								Desirable
Define efficiency and effectiveness metrics	√		√								Desirable

<div> <div>User Groups</div> <div>FUNCTIONAL SPECIFICATIONS</div> </div>	Managers		Domain Experts		IT Staff		Service workers		End users inside PA (Run.)	End users outside PA (Run.)	Mandatory /desirable
	Dev.	Run	Dev.	Run .	Dev.	Run .	Dev .	Run .			
View efficiency and effectiveness metrics in comprehensible forms (charts, tables etc)		√		√							Desirable
Schedule notifications and announcements for end-users			√					√			Mandatory
Schedule notifications and announcements for PA users			√					√			Mandatory
Access and use data entered by the end-users								√			Mandatory
Get guidance on what to do next								√			Mandatory
State which documents are required from end users			√								Mandatory
Access records for submitted forms		√						√			Mandatory
Check if required documents have been submitted								√			Mandatory
Specify which forms should have hard copies retained			√					√			Mandatory
Specify lists of permissions for data	√		√								

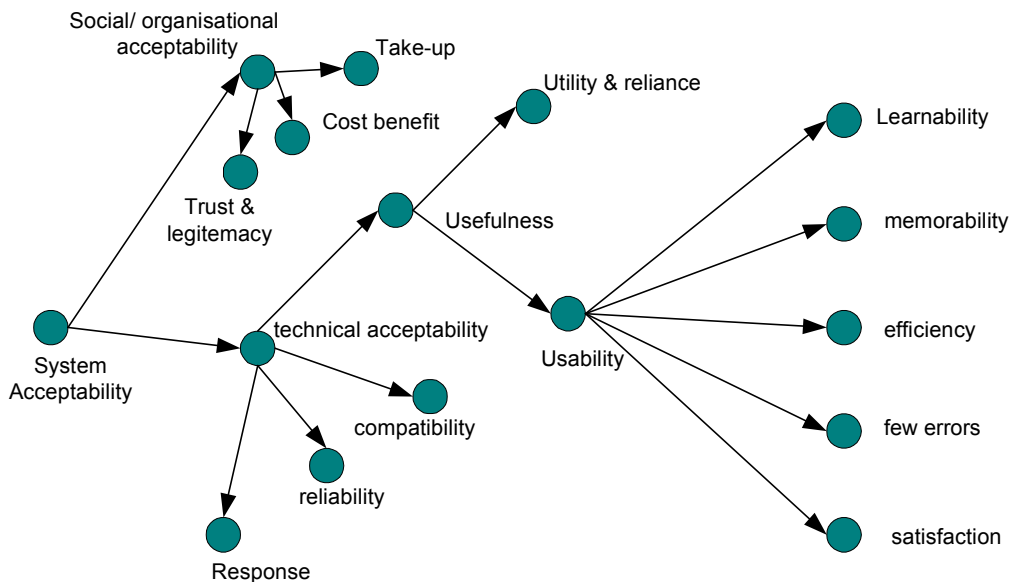
### 5.2.2 System acceptability.

The second approach of the evaluation is based on the Nielsen "attributes of acceptability" model [Nielsen 1993 Page 25] as shown in the figure below. This is a model for evaluating overall system acceptability through the evaluation of sub-concepts such as practical acceptability, usability etc.



**Figure 16 Nielsen's model of attributes of acceptance , Nielsen 1993, Page 25**

We have adjusted the Nielsen model to better fit the requirements of this evaluation as follows. Figure 2 below is our model used in the development of the success criteria.



**Figure 17 Adjusted Nielsen model**

Success Criteria are developed under each of the node titles as headings. The technical acceptance success criteria are outlined here, whereas the success criteria that refer directly to users or organisations, are outlined in section 5.4 relating to each of the two pilot sites.

#### **Technical acceptability:**

this is the level to which the system performs, within general parameters.

#### **Compatibility**

1. Does the platform adhere to HTML standards for browser compatibility with IE and Netscape?
2. Backend connectivity, does the platform conform to ODBC and JDBC standards?

#### **Reliability**

1. Mean time between failures should be no more than 10% higher than current system, given that this is a prototype.

**Response**

System response. Mean execution time at server level for actions should be as follows:

Front end actions	75% of actions less than 1 second
Smartgov agent	75% of actions less than 5 seconds
Integrator	75% of actions less than 60 seconds

**Social/ organisational acceptability:**

This is the level of acceptance of the platform and framework by the organisation that will be running the pilot.

The success criteria for evaluating this will be developed at the local level as they are specific to the individual pilot site organisation.

**Cost benefit**

This can only be partially explored within the context of this pilot evaluation.

**Take-up**

The level to which the system is adopted and utilised by the organisation.

**Trust & Legitimacy**

Reliance on and confidence in the system

**Usefulness:**

this broad term essentially addresses what the platform and framework offer the users as a tool to perform their tasks. It is constructed of the sub-terms below. Success criteria addressing these terms are given in the section 5.3 concerning the success criteria and metrics for the each of the pilot sites.

**Utility & Reliance:**

This is the level to which the platform and framework are relevant to the domain of the user and offer real solutions to domain issues.

**Usability:**

this is constructed of the following sub terms.

**Learnability:**

the ease with which users can learn to use the system.

**memorability :**

the ease with which users retain skills and knowledge about using the system.

**efficiency :**

how well the system performs as a tool for the user's needs.

**few errors:**

how well the system performs in terms of presenting few errors for users.

**satisfaction:**

how aesthetically pleasing, intuitive and enjoyable to use, the system is.

### ***5.3 CEC Baseline Measurements***

Measurement of the existing service is primarily focused around the 305 form, as this is the current vehicle for requesting equipment within the City of Edinburgh Council. The following metrics have been defined:

1. Numbers of errors of various types. These will take form of count boxes for each type of error to be completed by staff over a specified period to ascertain existing error rates.
2. Number of forms requiring extra information/needling to be returned.
3. Mean time for an OT/Service Professional to complete a paper 305 form.
4. Tracking of the path that 305 forms will typically take in their journey from initiator until the equipment is finally delivered.
5. Mean time for a form to complete its full journey.

These metrics were used to collect baseline data for the Equipment and Adaptations Service at the City of Edinburgh council. This data is presented in Appendix A.

### ***5.4 The User Sites Success Criteria and Metrics***

Evaluation of the performance of the SmartGov platform at the user sites consists of two stages. The first stage consists of measurements of the service prior to deployment of the pilot. The results are then used as the baseline for assessing the performance of the SmartGov platform, which is measured during the second stage,



using the following success criteria and metrics for the two SmartGov pilot sites, CEC and GSIS. The criteria are described under these headings: -

- Development Environment User Acceptability
- Pilot Service User Acceptability
- Overall Social & Organizational Acceptability

The criteria and metrics are defined below at a generic level, and will be localized in Wp9 to fit each pilot site's service environment.

## **5.4.1 Development Environment User Acceptability**

### **5.4.1.1 Utility and relevance**

1. Reusability of elements (TSEs, KUs etc)

*Metric:* Stakeholders should rate the reusability "satisfactory" or better in 70% of test cases.

2. Relevance of elements (TSEs, KUs etc)

*Metric:* Stakeholders should rate the relevance "satisfactory" or better in 70% of test cases.

3. Minimal IT input needed for service design and development

- a. time involved

*Metric:* Less than 10% of design time requires input from IT professionals.

- b. tasks involved

*Metric:* More than 80% of design tasks can be undertaken by non-IT professionals.

### **5.4.1.2 Usability**

#### Ease of learning

4. Time to learn how to create a service

*Metric:* 75% of users (domain experts) should be proficient in creating a specified test service after one days training or self-practice.

## 5. Time to learn how to modify a service

*Metric:* 75% of users (domain experts) should be proficient in modifying a specified test service after one days training or self-practice.

## 6. Time to learn how to retrieve statistics and view results

*Metric:* 75% of users (managers) should be proficient in creating a specified test service after 2 hours training and self-practice.

## 7. Friendliness/intuitiveness of development environment

*Metric:* Over 70% of users who have received training should rate the service development environment as "more helpful than current methods for developing the service".

Efficiency in use

## 8. Time to create a service

## a. Time required to elicit user needs

*Metric:* 25% reduction in person-hours required by service development staff

## b. Design and deployment time:

*Metric:* Service designed and deployed within 2 months

## 9. Effort to create a service

## a. HTML forms.

*Metric:* HTML form development requires no additional effort; simple forms should be capable of development in less than one working day; most complex forms should be capable of development in six working days.

## b. Service knowledge aspects (TSEs, KUs, validation checks).

*Metric:* Effort needed to consider and act on the relevant aspects is estimated to be less than with current working practices for translating business logic into a working service.

## c. Service technical parameters (communication with back-end systems and changes in back-end processing).

*Metric:* Effort needed to implement communications with back-end systems is estimated to be less.

- d. Statistics definitions.

*Metric:* Less than effort to code statistics now.

10. Time to Modify a Service:

*Metric:* Service changes implemented within 2 days

11. Effort to modify a service

- a. HTML form editing.

*Metric:* HTML form modification requires no additional effort.

- b. Service knowledge aspects (TSEs, KUs, validation checks)

*Metric:* Effort needed to modify the relevant aspects is estimated to be less than using the current working practices.

- c. Service technical parameters (communication with back-end systems, changes in back-end processing etc)

*Metric:* Effort needed to modify the relevant aspects is estimated to be less than using the current working practices.

- d. Statistics definitions

*Metric:* Effort needed to modify statistics is estimated to be less than using existing methods.

Memorability

12. Ease of remembering how to develop a service

*Metric:* Users should be able to recall how to perform 75% of specific tasks, following one week of non-use.

Few errors

13. Errors when creating a service

*Metric:* Fewer than 20% of trained users should experience serious problems when creating specific test services

14. Errors when modifying a service

*Metric:* Fewer than 20% of trained users should experience serious problems when modifying specific test services

15. Errors when defining statistics

*Metric:* Fewer than 20% of trained users should experience serious problems when defining specific test statistics

Satisfaction

16. Preference for the SmartGov development environment

*Metric:* 70% or more of the development environment users should rate it as “preferable to the current methods”.

## 5.4.2 Pilot Service User Acceptability

### 5.4.2.1 Utility and relevance

1. Improvement to delivered service

*Metric:* The delivered service should be rated by service users as “equivalent to former practices” or better in 80% of the cases.

2. Availability of performance statistics to managers

a. Real time performance information

*Metric:* 100% of defined statistics

b. Real time statistics on number of forms completed

*Metric:* 100% of forms

c. Real time statistics on the number of errors on forms

*Metric:* 100% of forms

### 5.4.2.2 Usability

Ease of learning

3. Time to learn how to use the runtime environment

*Metric:* 75% of service users should be proficient in using a specific test service after one hours training or self-practice.

Efficiency in use

4. Mean time to complete a form

*Metric:* Average completion time is 15% less

5. Number of forms successfully/unsuccessfully completed.

*Metric:* More than 50% forms error free

6. Mean time to access the service

*Metric:* Average time to complete a transaction is reduced by 50% on similar services

7. Processing time improvements in back office (through receiving better quality data)

*Metric:* Through validation rules forms are completed accurately in 98% cases

Memorability

8. Ease of remembering how to use a service

*Metric:* Users should be able to recall how to perform 75% of specific tasks, following one week of non-use.

Few errors

9. Errors when using a service

*Metric:* Fewer than 20% of users should experience serious problems when using specific test services

Satisfaction

10. Overall user satisfaction from the service

- a. On-line form completion (ease of use, intuitiveness, help topics, explanatory messages)

*Success criterion:* 70% of the service users should rate the on-line form completion procedure as "equivalent to former practices" or better.

- b. Back-end processing and return of results

*Success criterion:* 70% of the service users should rate the quality of service as “equivalent to former practices” or better.

### 5.4.3 Social & Organizational Acceptability

The overall assessment focuses on criteria that are relevant to the service area management, rather than specific to individual users.

#### Take-up

1. Number of registered service users

*Metric:* The number of registered users should increase by 10% over two months.

2. Number of registered users that actually used the service

*Metric:* 20% increase in the take-up by the registered target group

3. Number of submitted documents

*Metric:* 20% higher than currently

#### Cost-benefits

4. Overall (actual or projected) cost savings from the delivery of a service

*Success criterion:* The overall cost savings from the delivery of a pilot service should be 15% or higher.

5. Benefits from the introduction of the SmartGov platform and services framework

*Success criterion:* At least three of the following benefits should be realisable: cost savings, reduced development time, increase in reusability, better working conditions for PA employees, better services for the citizens.

#### Trust and legitimacy

6. Employee attitude towards the SmartGov platform

*Success criterion:* Fewer than 20% of the SmartGov platform stakeholders should say they think the SmartGov platform is not a worthwhile development.

7. Service user trust

Fewer than 20% of service users should say they have less confidence that their data will be properly processed.

## 6 Conclusions

This deliverable has described all the accomplished work during this Work Package. Firstly, the integration process, with the tests among the different components and the history of all the released version of each one of these components. Later, the methodology applied to install and deploy the final platform in the Public Authorities, responsible for the development and deployment of the Pilot services. The whole process to develop these Pilot Services is also described in this deliverable, from the scenario and the requirements for each of the services, to the service implementation, including the required step of adapting the requirement to SmartGov platform entities, in order to adapt the 'in paper' description of the service to a 'real' implementation.

To complete the objectives of this Work Package and set the principles to be used in the Work Package 9 (Trials evaluation), the Evaluation Process is outlined, as the SmartGov platform success criteria, regarding the overall platform and the specific criteria for each Pilot service. This Evaluation Process will be defined in complete detail during the WP9, applying the principles and basis set in this deliverable.

## 7 References

- [D31] State-of-the-Art and Current Situation at Public Authorities, SmartGov Project Deliverable 31, Stelios Gorilas, George Boukis, Giorgos Lepouras, Kostas Vassilakis, Akrivi Katifori, John Fraser, Heredia Larios Segundo, Rafael Canadas Martinez, Gerald Weiss, Kirstin Karasz, Spyros Argyropoulos and Hilary Coyne (May 31, 2002) available at <http://www.smartgov-project.org/index.php?category=results&langid=eng>
- [D41] User Requirements, Services and Platform Specifications, SmartGov Project Deliverable 41, Akrivi Katifori, Anna Charissi, George Lepouras, Stathis Rouvas, Costas Vassilakis, Nick Adams, John Fraser, Segundo Heredia Larios, George Boukis, Stelios Gorilas, Rafael Canadas Martinez and George Laskaridis, available (July 31, 2002) at <http://www.smartgov-project.org/index.php?category=results&langid=eng>
- [D51-61] Low-level Specifications of SmartGov Services and Applications and the Knowledge-Based Core Platform, SmartGov Project Deliverable 51-61, Stelios Gorilas, Pablo Fernandez Pardo, Tomas Pariente Lobo, Costas Vassilakis, Akrivi Katifori, Anna Charissi, George Lepouras, Stathis Rouvas, Nick Adams, John Fraser, Ann Makynthos (February 28, 2003) available at <http://www.smartgov-project.org/index.php?category=results&langid=eng>
- [D62] Implementation of SmartGov Services and Applications, SmartGov Project Deliverable 62, Stelios Gorilas, Pablo Fernandez Pardo, Tomas Pariente Lobo, Costas Vassilakis, Akrivi Katifori, Anna Charissi, George Lepouras, Nick Adams, John Fraser, Ann Makynthos, Vassilis Stoumpos, available (July 31, 2003) at <http://www.smartgov-project.org/index.php?category=results&langid=eng>



## **Appendix A.CEC Baseline Measurements**

### ***A.1 Design and Deployment***

The Council has undertaken research into a number of other new services developed using 'off the shelf packages'. Time and costs for the development of these services varies considerably depending on the service being developed. More work is required to ensure that Smartgov is compared against similar service deployments. CEC has been and will continue to monitor costs, time and expertise required to develop the Smartgov platform and compared these against similar services.

### ***A.2 Service Delivery***

In order to facilitate the evaluation process a number of key baseline measurements were collated. These measurements focussed on three key areas:

- I. The time taken to complete each step of the process
- II. The number of days it took to complete the whole process and the various domain experts and admin staff involved in the process
- III. % errors contained on completed forms.

These measurements will be revisited during the evaluation phase. CEC will run the current service in parallel with the new service developed using the Smartgov pilot providing an opportunity to compare the levels of service delivery.

### ***A.3 Average time taken to process one 305 form***

Number of forms received annually = 27,112

Stages	Time taken (mins)
Complete 305 form	34.5
Progress chase	8
Coding of forms	0.6
Setting routes	1
Logging details	8
Taking enquiries	1.2
Phoning clients	1.6
Average minutes spent on one form	55.2

Total cost of service = £261,075

#### ***A.4 Average number days to process 305 forms***

Stages	Lapsed Days
OTs send form to JDES	0
Stores receive order	2
Codify Mail	2
Arrange Delivery/enter into comp	4
Check Completion of delivery	8
Send Conformation to OTs	10
OTs receive conformation of delivery	12
Average number days to complete process	12

#### ***A.5 Error Rate***

A sample exercise was undertaken to determine the number of errors contained on each form that needed to be corrected before the form could be processed. On average the service received 21 forms with errors a week.

Number of forms per annum	27,112
Number of forms with errors per annum	1100
Error rate	4%