EUROPEAN COMMISSION
Research Executive Agency (REA)
Inclusive, Innovative and Reflective Societies

Project acronym:	SIMPATICO
Project full title:	SIMplifying the interaction with Public Administration Through Information technology for Citizens and cOmpanies
Call identifier:	EURO-6-2015
Type of action:	RIA
Start date:	1 March 2016
End date:	28 February 2019
Grant agreement no:	692819

D5.1 – SIMPATICO Platform Requirements and Architecture v1

WP5	Integration and Environment Setup
Task 5.1	Technical specification and architecture of the SIMPATICO platform
Due Date:	31/08/2016
Submission Date:	31/10/2016
Responsible Partner:	Engineering – Ingegneria Informatica SpA (ENG)
Version:	2.0
Status:	Final
Author(s):	Antonio Filograna (ENG), Raúl Santos de la Cámara (HIB), Iván Pretel (DEUSTO), Giuseppe Di Modica (BENG), Neetu Agrawal (SPARTA), Michele Trainotti (FBK), Gustavo H Paetzold (USFD)
Reviewer(s):	Raman Kazhamiakin (FBK), Fabio Perossini (SPA)
Deliverable Type:	R: Report
Dissemination Level:	PU: Public

Version History

Version	Date	Author	Partner	Description
0.1	18/04/2016	A. Filograna F. Arigliano	ENG	First version of ToC
0.2	21/06/2016	A. Filograna F. Arigliano V. Savarino	ENG	Second version of ToC
0.3	23/06/2016	A. Filograna U. Lopez I. Pretel	ENG DEUSTO DEUSTO	First Contribution to Introduction
0.4	01/06/2016	M. Pistore L. Specia V. Savarino	FBK USFD ENG	First Contribution to Section 2 and 4
0.5	06/07/2016	R. Santos	HIB	First contribution of HIB to sections 2 and 4
0.6	07/07/2016	I. Pretel U. Lopez	DEUSTO DEUSTO	First contribution of HIB to sections 2
0.7	13/07/2016	A. Filograna G. Di Modica	ENG BENG	Feedback after first internal review
0.8	27/09/2016	R. Santos de la Cámara	HIB	Revised contents for HIB component cards in section 2.2 (LOG, AQ, DA, EE). First version of the SoA in section 4.2
0.9	14/10/2016	All	All	Update architecture after Palermo GA meeting (sections 2 and 3)
1.0	15/10/2016	G. Di Modica	BENG	Conclusion added
1.1	21/10/2016	R. Kazhamiakin F. Perossini	FBK SPARTA	Internal review
1.2	26/10/2016	A. Filograna	ENG	Final version
2.0	31/10/2016	M. Pistore	FBK	Final quality check

Statement of originality

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Simpatico

Table of contents

List of figures

List of tables

Glossary

Acronym	Definition
CDV	Citizen Data Vault
UP	User Profile
TAE	Text Adaptation Engine
WAE	Workflow Adaptation Engine
IFE	Interactive Front-End
SF	Session Feedback
DA	Data Analysis
EE	Enrichment Engine
QAE	Question Answering Engine
CPD	Collaborative Procedure Design
CKB	Collective Knowledge Base
GE	Gamification Engine
DB	Dashboard
IL	Integration Layer
RO	Research Objectives
API	Application Programming Interface
REST	Representational State Transfer
PA	Public Administration
SME	Small and Medium Enterprise
UI	User Interface
HTML	HyperText Markup Language
GUI	Graphical User Interface

UML	Unified Modeling Language
DB	Data Base
DOM	Document Object Model
XML	EXtensible Markup Language
AJAX	Asynchronous JavaScript and XML
JSON	JavaScript Object Notation
LS	Lexical Simplification
CBOW	Continuous Bag of Words
ERNESTA	Enhanced Readability through a Novel Event-based Simplification Tool
SMT	Satisfiability modulo theories
PDS	Personal Data Store
RDF	Resource Description Framework
OASIS	Advancing open standards for the information society
XDI	eXtensible Data Interchange
SaaS	Software as a Service
PaaS	Platform as a Service
IaaS	Infrastructure as a Service
W3C	World Wide Web Consortium
XHTML	Extensible HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IE	Interner Explorer
MBUI	Model-Based UI Design
ISTI-CNR	Information Science and Technologies - National Research Council

Simpatico

Executive summary

This document is the deliverable **"D5.1 – SIMPATICO platform requirements and architecture"** of the European project "SIMPATICO - SIMplifying the interaction with Public Administration Through Information technology for Citizens and cOmpanies" (hereinafter also referred to as **"SIMPATICO"**, project reference: 692819).
This report includes the results of project task T5.1 "**Technical specification and architecture of the SIMPATICO platform**". The aim of this document is to give an overview on the **SIMPATICO Architecture** describing the its **Requirements**, the **methodology** used to collect them, and **each component** that builds the platform.

The SIMPATICO architecture requirements have been collected taking into account the IT systems of the Public Administrations involved in the project, the nature of SIMPATICO components that will be used and in case modified from those one reused, the data privacy and security to be taken into account during the development of use cases.

The SIMPATICO architecture has been designed in order to guarantee the interoperability with the legacy systems of PA and across platforms, operating systems and programming languages. All the SIMPATICO **technical components** were described highlighting their input and output in order to allow the interaction among all the other SIMPATICO components.

Starting from **use cases scenarios** defined in the three PAs involved in the project, how SIMPATICO tools and techniques are integrated in Trento, Sheffield and Galicia pilots was described. For each pilot a general overview of the integration schema and identifies possible specific constraints and issues are defined.

Besides, a **comparative analysis** of existing solutions and components was carried out in order to select, on the one hand, the more suitable technology to implement the SIMPATICO components, and on the other hand, already existing tools which could be integrated, reused or extended.

1 Introduction of Simpatico

This deliverable presents the outcomes of SIMPATICO project task T5.1 "*SIMPATICO platform requirements and architecture*" in the scope of WP5 "*Integration and environment setup*". During the first 6 months of project execution, this task has worked to the specification of the SIMPATICO requirements and architecture.
To better understand the aim and scope of the project architecture, we provide in this introductory section a short description of the SIMPATICO project (Section 1.1) and of the validation strategy that we intend to adopt (Section 1.2). We conclude the section with a description of the structure of the rest of this deliverable.

1.1 SIMPATICO project

SIMPATICO's goal is **to improve the experience of citizens and companies in their daily interactions with the public administration** by providing a personalized delivery of **e-services** based on advanced **cognitive system technologies** and by promoting an active engagement of people for the continuous improvement of the interaction with these services. The SIMPATICO approach is realized through a platform that can be deployed on top of an existing PA system and allows for **a personalized service delivery** without having to change or replace its internal systems: a process often too expensive for a public administration, especially considering the cuts in resources imposed by the current economic situation.
The goal of SIMPATICO is accomplished through a solution based on the **interplay of language processing, machine learning and the wisdom of the crowd** (represented by citizens, business organizations and civil servants) **to change for the better the way citizens interact with the PA. SIMPATICO adapts the interaction process** to the characteristics of each user; **simplifies** text and documents to make them understandable; **enables feedback for the users** on problems and difficulties in the interaction; **engages civil servants, citizens and professionals** so as to make use of their knowledge and integrate it in the system (Figure 1).

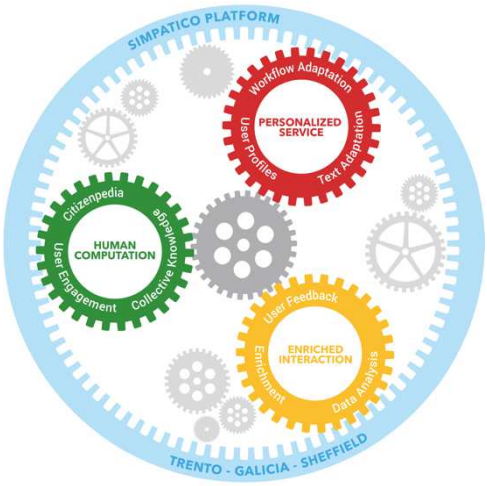


Figure 1: SIMPATICO concept as a glance

The project aims can be broken down into the following **smaller research objectives (ROs)**.

RO1. Adapt the interaction process with respect to the profile of each citizen and company (PA service consumer), in order **to make it clear, understandable and easy to follow**.

- A **text adaptation** framework**,** based on a **rich text information layer** and on machine learning algorithms capable of **inducing general text adaptation operations** from **few examples, and of customizing these adaptations to the user profiles.**
- **A workflow adaptation engine** that takes user characteristics and tailor the interaction according to the user's profile and needs.
- A feedback and annotation mechanism that **gives users the possibility to visualize, rate, comment, annotate, document the interaction process** (e.g., underlying the most difficult steps), so as to provide valuable feedback to the PA, further refine the adaptation process and enrich the interaction.

RO2. Exploit the wisdom of the crowd to enhance the entire e-service interaction process.

- An **advanced web-based social question answering engine (Citizenpedia)** where citizens, companies and civil servants **discuss and suggest potential solutions and interpretation for the most problematic procedures and concepts.**
- A **collective knowledge** database on e-services used to simplify these services and improve their understanding.
- An **award mechanism** that **engages users and incentivizes them to collaborate** by giving them **reputation** (a valuable asset for professionals and organizations) and **privileges** (for the government of Citizenpedia – a new public domain resource) according to their contributions.

RO3. Deliver the SIMPATICO Platform, an open software system that can interoperate with PA legacy systems.

- A platform that **combines consolidated e-government methodologies with innovative cognitive technologies** (language processing, machine learning) at different level of maturity, enabling their experimentation in more or less controlled operational settings.
- An interoperability platform that enables an **agile integration of SIMPATICO's solution with** PA legacy systems and that allows the exploitation of data and services from these systems with the SIMPATICO adaptation and personalization engines.

RO4. Evaluate and assess the impact of the SIMPATICO solution

- Customise, deploy, operate and evaluate the SIMPATICO solution on **three use-cases in two EU cities** – Trento (IT) and Sheffield (UK) – **and one EU region** – Galicia (ES).
- **Assess the impact** of the proposed solution in terms of **increase in competitiveness, efficiency of interaction and quality of experience.**

This deliverable focuses in particular on the RO4, covering both the description of the plan to customize, integrate, deploy and operate the SIMPATICO solution in the three pilot sites, and the definition of the objectives and KPIs for the evaluation and assessment of the SIMPATICO effectiveness and impact.

1.2 SIMPATICO Architecture

As first step, we want to provide the reader with a simple general scenario to better understand how the platform works and what the interactions among the components are. Below a brief description on how users (e.g. citizen, civil servant, SME, professional) can use the platform and benefit from it.

When a citizen wants to use a SIMPATICO e-service s/he has to access to the service authenticating via Municipality Identification System. After authentication, the citizen interacts with the needed e-service provided by Public Administration thanks to the Interactive Front-End that makes easier the

way to interact with the service and web site in general. If the user has to compile a web form to request or receive something to PA, the SIMPATICO platform (Figure 2) can provide him/her with a prefilled web form, through the interaction between the Interactive Front-End and the Citizen Data Vault. This latter allow citizen to store his/her personal data in order to be used whenever s/he wants without re-filling in the form. During the usage of the e-service the citizen can ask for some simplification of the text or suggest and annotate those parts of text really difficult to be understood.

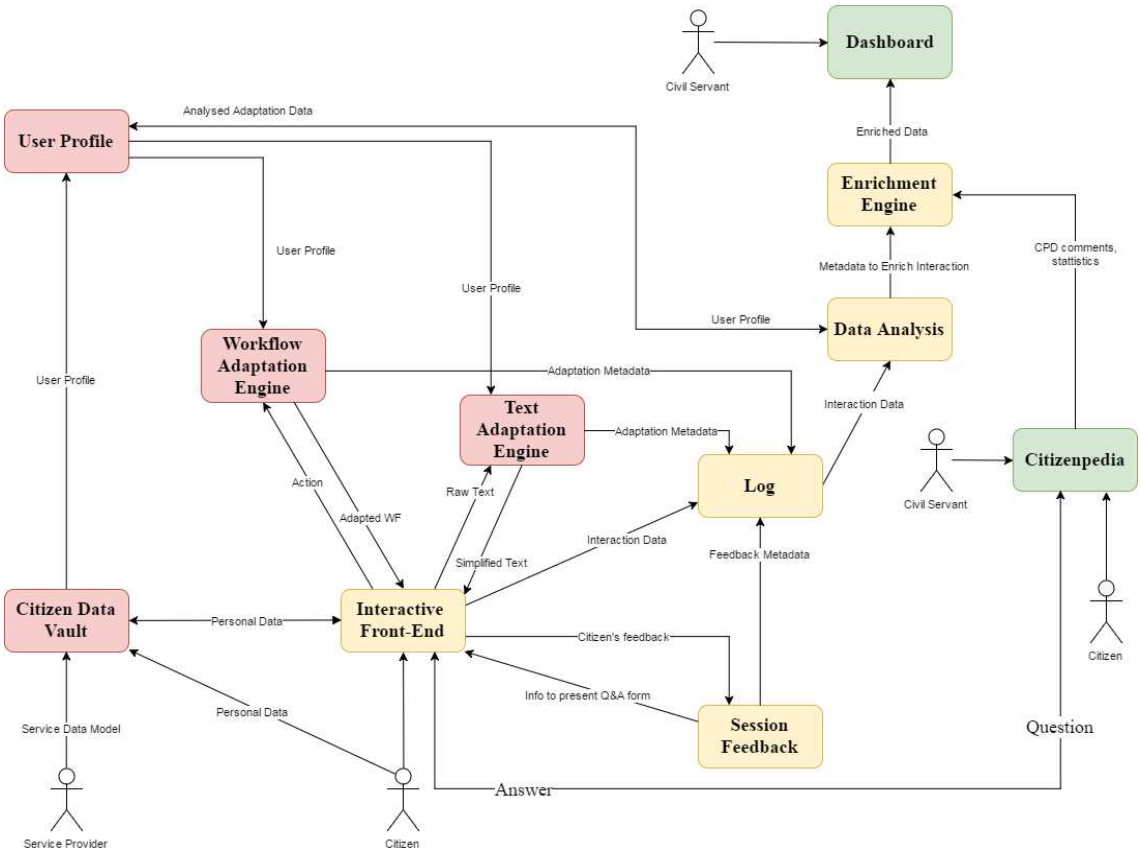


Figure 2: SIMPATICO Architecture

The Text Adaptation Engine suggests some changes to be made to the text depending on the user's profile, which will be saved in a dedicated database interacting with Text Adaptation Engine. In the same way, according to the user profile of the citizen the Workflow Adaptation Engine adapts the workflow of the interaction basing also on the action done by the user. Both the Interactive Front-End, Workflow Adaptation Engine and Text Adaptation Engine send data (e.g. how long is a service was used, how long it is used to fill out the form) to the Log component. After the interaction with the system, the citizens can give their feedback about the interaction answering to the question provided by Session Feedback component.

The data from User Profile and Log component are analysed by Data Analysis component and sent to the Enrichment Engine that receives also data from Citizenpedia. The Enrichment Engine processes the data and sends them to Dashboard.

The Citizenpedia component has its own model of user roles, defined in (D4.1 - Citizenpedia Framework Specification and Architecture). In order to manage them an authentication system is

needed (it is under investigation the possibility to have a single authentication system for all SIMPATICO components that request an authentication). It is the place where users (citizen, professional, civil servant, etc.) can consult it in order to understand and comprehend how to perform a specific procedure. The knowledge is shared among users through the Question and Answer Engine that is in charge of collecting and visualizing the questions and answers received from the users. The latter will be encouraged to the use of Citizenpedia through the use of Gamification techniques. During the using of the e-service, the citizen can make a question or send a comment to the Question and Answer Engine (Q&AE) through the Interactive Front-End. The Q&AE sends back the answer if exists, otherwise the citizen can create a new question whom the community can answer to.
In addition, the civil servants can draw the operation process of the procedures by the Collaborative Procedures Design in an easy way. On the other hand, citizens or professionals can comment, giving suggestions or propose changes to the diagram.
Finally, civil servants through the use of the Dashboard can manage the on-line service, viewing data from various sources/tools that, for example, tell them how easy is a text to be understood, how to improve it, how many users have used the e-service, and so on. In a few words, a dashboard to facilitate the work of the civil servant.
During the usage of the Dashboard, once a decision on how to improve or adapt the service is taken, this will have impact on both the e-service itself (changing the text, the process or the interaction model) and also on the Citizenpedia in order to modify and improve the information contained therein.

1.3 Structure of the deliverable

The rest of the deliverable is organized as follows.

Section 2 describes the overall requirements of SIMPATICO Architecture and the related Methodology used to define them.

Section 3 will then describe all the Component Cards for each component belonging to the SIMPATICO Architecture, their functionalities and the input and output information in order to interact with the other components of Architecture. In addition to this information, the Sequence Diagram related to a set of components will be provided.

Section 4 describes the actual use-case scenario in which they will be used, providing the integration of the SIMPATICO component with the three use-cases (Trento, Sheffield, Galicia).

Section 5 give an overview on the state of the art of technologies that SIMPATICO components are going to adopt.

Section 6 completes the deliverable with some concluding remarks.

2 SIMPATICO platform requirements

In this section the high level integration requirements for the SIMPATICO platform will be presented and the methodology used to define them.

2.1 Description of the Volere Methodology

The requirements analysis mostly relies on the Volere methodology (Robertson, 2012). The Volere methodology comprises a general template for presenting the layout and structure of the requirements specification document, as a result of the requirements elicitation and analysis processes. Not all elements of the Volere template are considered to be suitable for our project. Therefore, a number of Sections from the Volere template about the description of functional and non-functional specifications have been chosen. The sections that were considered more suitable to the SIMPATICO requirements are the following:

- Functional Requirements
    - Functional and Data Requirements
- Non-Functional Requirements
    - Look and Feel Requirements
    - Usability and Humanity Requirements
    - Performance Requirements
    - Operational and Environmental Requirements
    - Maintainability and Support Requirements
    - Security Requirements
    - Cultural and Political Requirements
    - Legal Requirements

This deliverable adopts the aforementioned structure by reporting functional and non-functional requirements for the main components of the envisaged solution at proposal time. For the description of each atomic requirement that belongs to each one of the aforementioned subsections we provide a tabular template based mainly on the Volere requirements shell, after applying the following modifications:

The following fields have been added to the default template:

- *Name*, this field has been added in order to provide in addition to the ID field, a short name that describes the specific requirement in human readable format.
- *Difficulty*, which indicates the level of difficulty for the implementation of this requirement (estimation from a technical point of view). Difficulty ranges on a scale from 1 (=low difficulty) to 5 (=extreme difficulty).
- *Actors*, indicating either those persons or things that interact externally with the system or one of its components, or react through these components.

Removal/replacement of fields:

- *Event/use case#*: This field has been omitted from the original Volere requirements shell because the use cases, to which it refers, will be defined later within the project, therefore this field cannot be filled in at this stage.
- *Supporting materials*: This field has been also removed because some of the documents that are related to these requirements are subject to IPR (e.g. the companies that have conducted the surveys keep them as internal reports with restricted access).
- *Originator* (the person who raised this requirement), has been replaced by the Author field (the owner of each recorded requirement).
- *History* field has been replaced by Revision (indicates versioning).

Table 1: Requirements table template summarises the SIMPATICO "requirements shell" (i.e., requirements gathering template) in tabular format, after the application of the aforementioned changes to the Volere shell.

Table 1: Requirements table template

ID	*A unique identifier*
Name	*Title of the requirement.*
Requirement Type	*Whether it is a functional or non-functional requirement and in case of non-functional requirements the specific type of requirement according to the Volere notation.*
Description	*A requirement must say exactly what is required.*
Rationale	*A justification of the requirement*
Fit Criterion (Measurable)	*By measurable we mean is it possible, once the system has been constructed, to verify that this requirement has been met. In other words, this means the tests which must be performed in order to satisfy the requirement*
Customer satisfaction	*Degree of stakeholder happiness if this requirement is successfully implemented (Scale from 1=uninterested to 5=extremely pleased).*
Customer dissatisfaction	*Degree of stakeholder unhappiness if this requirement is not implemented (Scale from 1=hardly matters to 5=extremely displeased).*
Priority	*The requirement is ranked according to the customer value. (Scale from 1=low priority to 5=highest priority).*

Conflicts	*Any requirements whose implementation is blocked by this one.*
Constraints (Attainable)	*An attainable requirement will usually answer the question:* *How can the requirement be accomplished?* *Hence, here we provide any constraints / conditions for the requirement to be executed.*
Difficulty	*Level of difficulty for requirement implementation (estimation). (Scale from 1=low difficulty to 5=extreme difficulty).*
Actors	*An actor is someone or something outside the system that interacts with it or with one of its components (primary actor). If the actor interacts with the system or one of its components is a secondary actor*
Author	*The owner of each requirement that was recorded.*
Revision	*This section lists when a version of the requirement was created.*

2.2 SIMPATICO Requirements definition

Starting from the analysis of the main objectives of the SIMPATICO project these requirements have been identified during the first stage of the project. The focus of the requirements listed below is not defining detailed features or technologies to adopt for implementation but to give reader a high level overview. This approach has been followed in order to, on the one hand, maintain the definition of the platform more general, in particular in the first phase of the project where some concrete aspects, related to the role of the platform and its usage, are not completely defined; on the other hand the detailed description of the components requirements and the related technical choice will be described in other specific deliverables (e.g. D2.1, D3.1, D4.1, etc.).

Below the detailed description of every requirements of the Integration Layer (IL), namely all the standards and formalisms that guarantee an easy and effective interaction among all the components.

ID	IL.1
Name	Interoperability with legacy systems
Requirement Type	Non functional (Security)

Description	It has to be possible to connect the SIMPATICO platform with the existent PA legacy systems (e.g. databases, web services). Secure and reliable communication with the existing public administration information systems have to be provided without requiring changes in these systems.
Rationale	The existent PA services are based on different and probably legacy technologies. Some parts of these services will continue to be fully functional because are related to critical functionalities that cannot be easily changed or outsourced.
Fit Criterion (Measurable)	The platform is able to provide connectors towards the PA's legacy systems to enable the interoperability.
Customer satisfaction	5 (Scale from 1=uninterested to 5=extremely pleased).
Customer dissatisfaction	5 (Scale from 1=hardly matters to 5=extremely displeased).
Priority	5 (Scale from 1=low priority to 5=highest priority).
Conflicts	The IFE and WE requirements are blocked by this one.
Difficulty	3 (Scale from 1=low difficulty to 5=extreme difficulty).
Actors	Municipality IT department and SIMPATICO IT team
Author	ENG team
Revision	V1.0, 04/07/2016

ID	IL.2
Name	Open API access
Requirement Type	Non Functional (Operational and Environmental)
Description	Data and services available in the SIMPATICO platform have to be accessible via a set of APIs using standardized approaches(e.g. RESTful

	API).
Rationale	The SIMPATICO platform should provide tools, data and services to several actors, PAs, SMEs external systems etc. For this reason it is necessary that all the necessary platform functionalities have to be accessible through standard and well documented open APIs.
Fit Criterion (Measurable)	All SIMPATICO platform components provide a set of Open API to interact with them; All Open APIs of the SIMPATICO platform are documented in a standard and uniform way.
Customer satisfaction	4 (Scale from 1=uninterested to 5=extremely pleased).
Customer dissatisfaction	3 (Scale from 1=hardly matters to 5=extremely displeased).
Priority	4 (Scale from 1=low priority to 5=highest priority).
Conflicts	No other requirement is blocked by this one
Difficulty	2 (Scale from 1=low difficulty to 5=extreme difficulty).
Actors	SIMPATICO component owners
Author	ENG Team
Revision	V1.0, 04/07/2016

ID	IL.3
Name	Secure storage
Requirement Type	Non Functional (Security)

Description	SIMPATICO platform components has to provide secure storage functionalities in order to record data needed for their execution.
Rationale	It will be useful to have storage functionalities available in the SIMPATICO platform to manage and save in a secure way the information of the citizen and the usage of PA service. The SIMPATICO platform should not permanently store any sensitive data in order to reduce security and privacy issues.
Fit Criterion (Measurable)	All SIMPATICO components have implemented proved expedients to secure the data they store and manage.
Customer satisfaction	4 (Scale from 1=uninterested to 5=extremely pleased).
Customer dissatisfaction	5 (Scale from 1=hardly matters to 5=extremely displeased).
Priority	5 (Scale from 1=low priority to 5=highest priority).
Conflicts	The IL.4 requirement is blocked by this one.
Difficulty	4 (Scale from 1=low difficulty to 5=extreme difficulty).
Actors	SIMPATICO component owners
Author	ENG Team
Revision	V1.0, 04/07/2016

ID	IL.4
Name	Privacy and Data Protection
Requirement Type	Non-Functional (Security)
Description	SIMPATICO platform has to be compliant with the EU legislation regarding

	privacy and data protection. It should adopt all the necessary technologies, standards and methods to protect privacy of the users of the platform services and to secure stored information that could be considered private.
Rationale	Despite the fact that the platform should not permanently store any sensitive data, the services that will run on it could handle private data. For this reason it is necessary that the SIMPATICO platform provides functionalities to protect personal information in line with the current legislation.
Fit Criterion (Measurable)	The SIMPATICO platform is compliant with Data Privacy and Data Protection EU regulations.
Customer satisfaction	5 (Scale from 1=uninterested to 5=extremely pleased).
Customer dissatisfaction	5 (Scale from 1=hardly matters to 5=extremely displeased).
Priority	5 (Scale from 1=low priority to 5=highest priority).
Conflicts	Although no requirement is blocked by this one, it is to be considered mandatory.
Difficulty	4 (Scale from 1=low difficulty to 5=extreme difficulty).
Actors	SIMPATICO platform providers
Author	ENG Team
Revision	V1.0, 04/07/2016

ID	IL.5
Name	Accessibility
Requirement Type	Non-Functional (Security)

Description	SIMPATICO platform will be compliant with the accessibility regulations.
Rationale	SIMPATICO platform will modify the behaviour of official institutional websites that already meet the accessibility guide lines of their own Country.
Fit Criterion (Measurable)	The SIMPATICO platform is compliant with accessibility regulations for each Country.
Customer satisfaction	5 (Scale from 1=uninterested to 5=extremely pleased).
Customer dissatisfaction	5 (Scale from 1=hardly matters to 5=extremely displeased).
Priority	4 (Scale from 1=low priority to 5=highest priority).
Conflicts	Although no requirement is blocked by this one, it is to be considered mandatory.
Difficulty	3 (Scale from 1=low difficulty to 5=extreme difficulty).
Actors	SIMPATICO platform providers
Author	ENG Team
Revision	V1.0, 04/07/2016

3 SIMPATICO architecture and components

In this Section each component of the SIMPATICO platform will be described by using the following Component Card (Table 2). The aim is not to give a complete description of the component but its overview, mainly focused on the interaction with other components to better understand how the integration among all components work.

Table 2: Component Card

ID	A unique identifier
Name	The component name
Description	A general description of the component indicating its main scope and the role the component has in the architecture
Functionalities	A list of the main functionalities offered by the component
Input Data	Input data accepted by the component (if applicable).
Output Data	Output data produced by the component (if applicable).
Interaction with other components	This field lists interactions with other components of the SIMPATICO platform.
Related Requirements	The requirements that are satisfied or partially covered by the functionalities provided by the component

This Section includes four sub-sections, one for each Architecture layer where the belonging component will be defined and an additional sub-section where the cross-layer components will be described.

3.1 Interaction adaptation and personalization

One of the core objectives of the SIMPATICO project is to provide with new technologies that assist users in comprehending the content of public administration websites and documents. For that purpose, SIMPATICO features the Text and Workflow Adaptation Engine, which is responsible for producing simplified versions of webpages, forms, questionnaires and documents in the public administration domain.
Through the Interactive front-end, the Text and Workflow Adaptation Engine are able to interact with the user, and this way outline their profile with respect to their personal information as well as

their interaction adaptation needs. With this information at hand, the Text and Workflow Adaptation Engine can help the user in a personalized fashion that tends to their profile specifically. By sending simple on-demand requests to the Text Adaptation Engine through the Interactive front-end, the user can require simplified versions of words, expressions and sentences present in public administration websites. When completing official forms and questionnaires, the Workflow Adaptation Engine can also help by either auto-completing fields for which the answers have already been provided by a given user, or by simply omitting fields that they do not need to fill, simplifying the process.

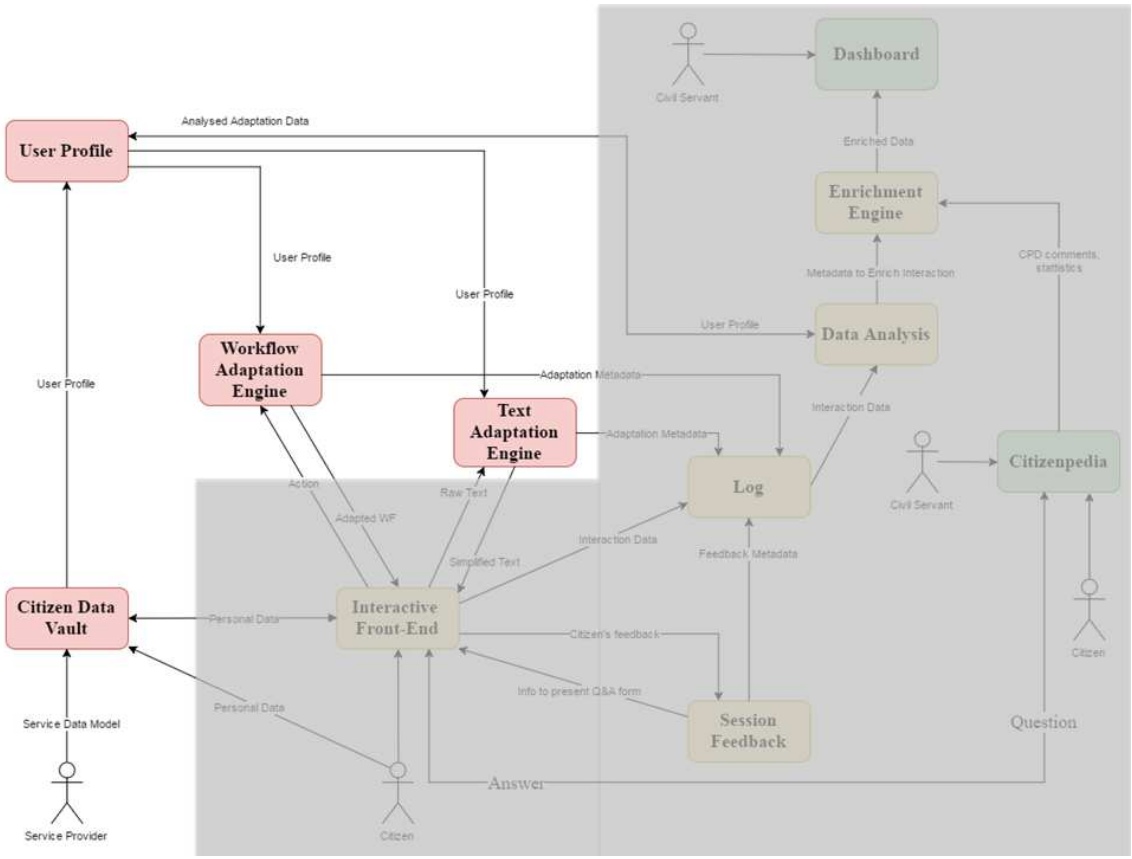


Figure 3: Interaction adaptation and personalization components

3.1.1 Citizen Data Vault

Table 3: CDV Component Card

ID	CDV
Name	Citizen Data Vault
Description	The Citizen Data Vault is the repository of the citizen (or company) personal data, profile and information. It is continuously updated through

	each citizen interaction and is used to automatically fill forms. In this way the citizen will give to the P.A. each information only once, as the information will be stored in the vault and used in all the following interactions.
Functionalities	• Register Service and Data Mapping: Each service that wants to interact with the CDV has to be registered by providing a mapping between the service data model and the Personal Data Meta-model of CDV. The Service Provider is in charge of the service registration and this is a back-end process. • Store Personal Data: the CDV provides the functionality to store citizen personal data to be reused when the citizen needs. Retrieve Personal Data: the CDV provides the functionality to retrieve citizen personal data to be reused in order to have a pre-filled web form. • Manage Personal Data: The CDV provides a data dashboard enabling users to manage and control their personal data
Input Data	• Service Data Model-Personal Metadata mapping: The data model of each registered service will be mapped with concepts and data fields belonging a Personal Data Meta-model. • Personal Data: A document representing a set of key-values related of personal data stored during the interaction of user with the e-services (through the Interactive Front End) or provided directly by the user by means of CDV data dashboard.
Output Data	• Personal Data: A document representing a set of key-values related to Personal data according to a "GetData" interaction with the other components • User Profile: A document representing a specific subset key-values of personal data referring to user profile datamodel
Interaction with other components	The component interacts with: • **Interactive Front-End**: users through the Interactive Front-End can save the citizen personal data in the CDV. Once done, next time users can retrieve their personal data from CDV to have a pre-filled web form. • **User Profile**: CDV provides User Profile with the user information needed to other platform components in order to provide users with profiled information

3.1.2 User Profile

Table 4: UP Component Card

ID	UP
Name	User Profile
Description	The User Profile will be responsible for storing the user profile data, as well as any data pertaining to their needs with respect to Text Adaptation.
Functionalities	The User Profile will make two types of information available: • **User Information:** refers to any information that pertains to the users' personal profile, such as age, native language, language proficiency and education level. • **Simplification Needs:** refers to the information gathered by the Interaction Model which say respect to the type of information that the users find most challenging to understand, as well as the types of adaptation that is most/least suitable for them. Some practical examples of information on simplification needs that could be found in the UP are: complex words, expressions, and sentences selected by the user for Text Adaptation requests, as well as the users' feedback on the output provided by the Text Adaptation Engine.
Input Data	New user profile data. A document containing key-value pieces of information about the user.
Output Data	Stored user profile data. A document containing key-value pieces of information about the user.
Interaction with other components	The component interacts with: • **Interactive Front-End:** the feedback provided by the users on the output produced by the Text Adaptation Engine through the Interaction Model will be stored into the User Profile database. • **Citizen Data Vault:** the personal information available through the CDV will complement the data in the User Profile database. • **Text Adaptation Engine:** will request for data from the User Profile in order to customize simplifications and enhancements. • **Workflow Adaptation Engine:** will provide with user profile data in order for workflow to be adapted for the user. • **Data Analysis:** will received analysed data collected from interactions between the user and the interactive front-end.

3.1.3 Text Adaptation Engine

Table 5: TAE Component Card

ID	TAE
Name	Text Adaptation Engine
Description	As its name suggests, the Text Adaptation Engine is the component responsible for providing with Text Adaptation solutions to the city council services and documents being addressed in the Simpatico project. Its goal is to modify or enhance texts, forms and services that may challenge the users targeted in the project.
Functionalities	The TAE will feature various distinct Text Adaptation techniques: • **Lexical Simplification:** Replaces complex words and expressions with more familiar alternatives equivalent in meaning. • **Syntactic Simplification:** Modifies the syntactic structure of sentences in order to make them more easily comprehensible, such as through sentence splitting or anaphoric resolution. • **Text Enhancement:** Provides with complementary pieces of information, such as synonyms, translations and images, about words and expressions unknown to the reader.
Input Data	User profile data (such as age, native language, etc.). A document containing key-value pieces of information about the user. Simplification requests in form of raw text (made by the user). A document fields specifying the type and context of the simplification being requested.
Output Data	Simplified words, expressions and sentences. A document specifying the details of the simplification made, as well as any enhancement information such as synonyms, images, etc. Adaptation metadata. A document containing details about the simplifications made by the engine.
Interaction with other components	The component interacts with: • **Interactive Front-End:** users will be able to select certain portions of forms and documents in order to request for specific simplifications. Users will also be able to judge the quality of simplifications and enhancements. Such judgments will then be fed back to the Text Adaptation Engine in order for the models to continuously improve on how they adapt content based on the

	users' profiles. ● **User Profile:** the user profile data available through the User Profile will allow for the users' needs to be more easily outlined, which will then allow for the Text Adaptation technologies to better customize the simplifications and enhancements produced. ● **Interaction Data Log:** the engine will send metadata about the simplifications made in order for interaction information to be inferred.

3.1.4 Workflow Engine

Table 6: WE Component Card

ID	WE
Name	Workflow Engine
Description	The workflow engine will help the users to more easily navigate, comprehend and interact with city council services. It will do so by modifying, completing or enhancing the information present in the fields of online forms and questionnaires found in city council webpages.
Functionalities	The WE will offer: **Auto-Completion:** form fields which request for information that have already been provided by the user will be auto-completed in order to assist the user in completing the form more quickly. **Text Adaptation:** the WE will be able to interact with the Text Adaptation Engine in order to provide with simplifications and enhancements for words and expressions present in interactive city council services.
Input Data	User profile data (such as age, native language, etc.). A document containing key-value pieces of information about the user.
Output Data	Customised workflow components, such as forms and questionnaires. A document containing instructions on which form/questionnaire components to be modified and how. Adaptation metadata. A document containing details about the adaptations made by the engine.
Interaction with other	The component interacts with: • **User Profile:** the user profile data available through the User Profile

components	will allow for the Workflow Engine to more effectively customise interactive fields that request for the users' profile information. • **Interactive Front-End:** the workflow adaptation engine will provide with adapted workflow components back to the user through the interactive front-end. • **Interaction Data Log:** the engine will send metadata about the adaptations made in order for interaction information to be inferred.

3.2 Front-end interaction and enrichment

SIMPATICO has as one of its core objectives the improvement of the interaction process for each involved actor in the system (citizens, civil servants, companies). In order to do this, an interactive layer that enables the adaptation mechanisms proposed in Section 3.1 is required. Thus, in this section of the architecture we propose a system to (a) present the user with a fully featured Interactive Front-End for the SIMPATICO e-services, (b) provide the system's analysis modules with data gathered from the interaction to propose improvements in future versions of the e-services and finally (c) a method of summarizing these proposed changes and provide the civil servants with useful information to improve the services.

These modules, outlined in the Figure 4, comprise the main interactive loop for the SIMPATICO system. As a whole, they interact with the Adaptation components to execute simplifications on the front-end and with the Human Computation subsystem to gather more information from the users.

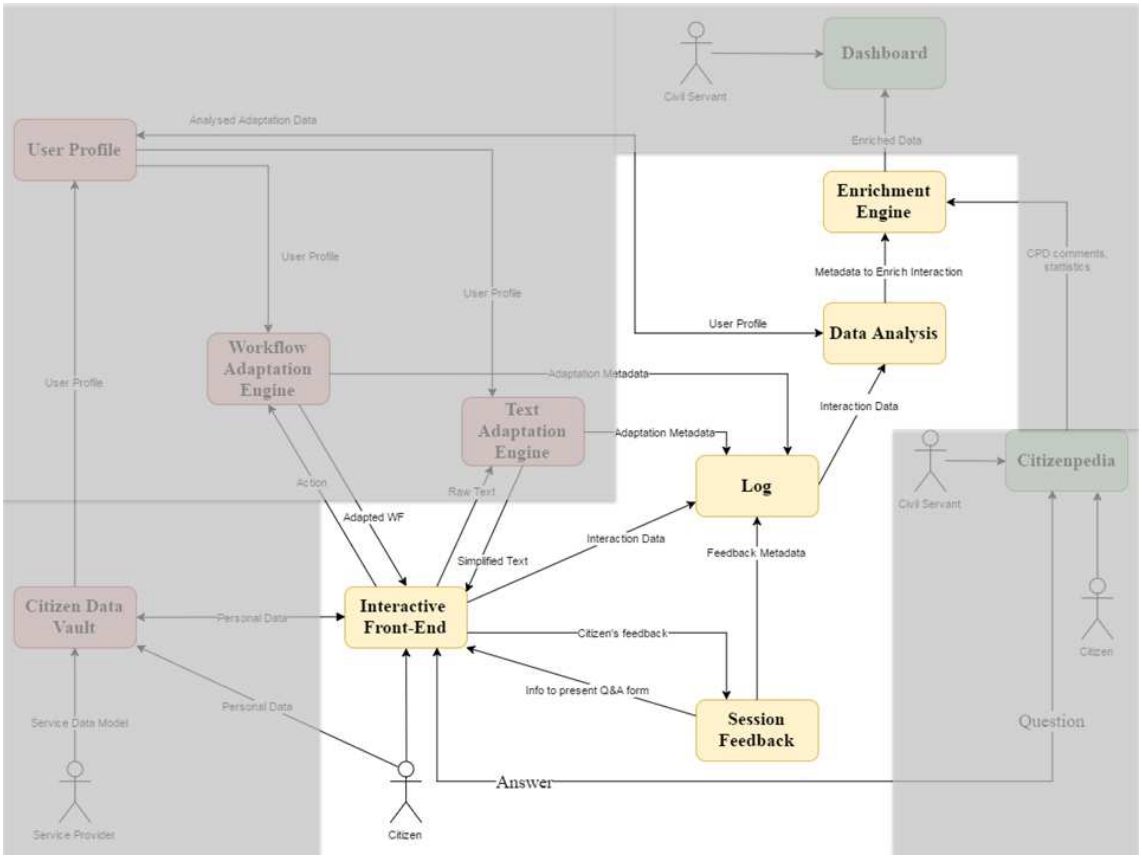


Figure 4: Front-end interaction and enrichment Components

3.2.1 Interactive Front-End

Table 7: IFE Component Card

ID	IFE
Name	Interactive Front-End
Description	A piece of software installed in a web browser that acts a proxy between the user and the Simpatico platform. It will enable a user interacting with a given e-service to annotate, enquire or obtain content and workflow adaptations based on his/her profile and preferences detected in previous interactions.
Functionalities	The IFE will provide a minimal UI (e.g. as a pop-up with buttons) over the existing e-service form and documents. Using this UI, the text in the legacy e-service will be replaced with simplified text provided by the SIMPATICO

	Adaption Engine.
Input Data	• Raw text & interaction: The text used as input to IFE (and consequently to Simpatico) is selected by the user on the web browser. The text will be paragraphs that describe bureaucratic processes (e.g. build permit request steps) and interactions between the citizens and the e-service (e.g. time filling a form). • Adapted workflow: Information that has been modified or enhanced by the WE, i.e. the information present in the fields of online forms found in e-services. • Simplified text: Simplified versions of a portion of text selected by the user in an e-service, produced by TAE. • Personal data: explicit information related to personal data (e.g. address, gender…) stored during the interaction of user with the e-services. • Answers: Doubts related to the exposed e-service coming from the Citizenpedia
Output Data	• Personal data: explicit information related to personal data (e.g. address, gender…) stored during the interaction of user with the e-services. Structured equally to the personal data used as input. • Raw text: The text to be simplified and adapted by the TAE and WE respectively. This texts is the propagation of the selected text in the e-service. • Interaction data: Interaction metrics and information gathered and generated during the dialog produced by the user and the e-service (e.g. time filling a form, click locations, interaction errors and satisfaction feedback). • Citizen's feedback: Doubs and comments related to the content of the e-service. They will be added as side comments and propagated to the Session Feedback component. • Questions: Doubts related to the exposed e-service published in the Citizenpedia
Interaction with other components	The component interacts with: • **Citizen Data Vault:** in order to fetch the personal data of the active user and update it if necessary. • **Text adaptation engine:** it will provide simplified versions of the texts in each selected section/paragraph. • **Workflow adaptation engine**: it will provide the adapted workflow for the current user taking into account its needs.

	• **Interaction Data Log:** in order to store and use the conducted actions, interactions and usability data of each user session. • **Session Feedback:** in order to fetch/store the annotations/comments made by the user. • **Citizenpedia:** to create and retrieve questions and answers based on the e-service text

3.2.2 Interaction Data Log

Table 8: Interaction Data Log Component Card

ID	LOG
Name	Interaction data log
Description	In order to optimize interaction, an analysis of past interaction by citizens is made to detect potential problems, bottlenecks and steps in which users may be less confident or dissatisfied with the results. This is done asking the citizens and also analysing the user interaction using automated means. The Interaction Data Log is the element in which information from the user's interaction is captured. The information is captured using a common data model as put forward (as a draft) in (D3.1 - User Interactions Modelling and Design).
Functionalities	The Interaction Data Log stores data according to a data model and provides an API for external modules (such as Data Analysis) to query for data.
Input Data	Interaction data collected automatically from the Interactive Front-End using instrumentation embedded in the code. This includes timing of steps of a procedure, click events, mouse movements, etc. This is described as Implicit data sources in the Interaction Data Model (D3.1). It also gets input from other components which produce data for the interaction process: • Session Feedback: data coming from the users' responses to questions at the end of an interactive session. • Text adaptation Engine and Workflow Adaptation Engine: metadata from the text and workflow adaptation requests that are required in interactive sessions.

Output Data	Previously collected data from past sessions. It interacts with modules that require this past data, in particular the Data Analysis. Querying will be available using REST interfaces (still to be defined). Data representation will most likely be JSON objects.
Interaction with other components	**Interactive Front-End, Session Feedback, Text Adaptation Engine and Workflow Adaptation Engine** SEND data to Interaction Data Log. **Data Analysis** REQUESTS data to the Interaction data Log component.

3.2.3 Session Feedback

Table 9: AQ Component Card

ID	SF
Name	Session Feedback
Description	In order to optimize interaction, an analysis of past interaction by citizens is made to detect potential problems, bottlenecks and steps in which users may be less confident or dissatisfied with the results. This is done asking the citizens and also analysing the user interaction using automated means. Session Feedback collects the data which is explicitly provided by the citizen to assess the quality of an interaction at the end of an interactive session.
Functionalities	At the end of an interactive session, Session Feedback produces a summary of an interactive form for the users to fill and sends it to the Interactive Front End. At the completion of the form, it stores the collected data in the Interaction Data Log according to the concepts and relationships in the Interaction Data Model.
Input Data	Direct feedback by the citizen executing an e-service: • Predefined, non-standard answers to questions: 'How would you rate your interaction? (Poor/Fair/Good/Excellent)' OR (☺,☹)' • Predefined, graded answers to questions (e.g., Likert scale): 'How would you rate your interaction? (-3 being strongly dissatisfied, 0 being neutral and 3 being strongly satisfied)' • Open text-based answers:

	'How would you rate your interaction? Free text: (500 characters)' to be further analysed by a human operator.
Output Data	Summary of the form to be presented to the user via the Interactive Front-End. Data collected from the user through the presented form.
Interaction with other components	**SF** sends a summary of an interactive form to IFE (e.g., as an HTML snippet). **Interactive Front-End** SENDS data to SF (answers from users). **SF** SENDS data to the Interaction Data Log.

3.2.4 Data Analysis

Table 10: DA Component Card

ID	DA
Name	Data Analysis
Description	In order to optimize interaction, an analysis of past interaction by citizens is made to detect potential problems, bottlenecks and steps in which users may be less confident or dissatisfied with the results. This is done asking the citizens and also analysing the user interaction using automated means. DA uses the basic data elements captured by SIMPATICO and stored in the Interaction Data Log to provide higher level metadata: aggregated average data are derived from individual timings in the Interaction Data Log data, and potential interaction problems from averaged timings in the Interaction Data Log data.
Functionalities	Provide statistical analysis of interaction data in the system. Produce reports that can be used to inform the Civil Servants in the Dashboard and using other SIMPATICO components such as the Enrichment Engine to enforce changes in the front-end.
Input Data	Data stored in Interaction Data Log. User profiles stored in the User Profile mirrored from the CDV.

Output Data	Data to use in the Enrichment Engine to produce interaction analysis reports so that the Civil Servants can improve the process. Summaries of interaction issues about users to store in the User Profile so that future interactive sessions can take advantage.
Interaction with other components	DA REQUESTS data from **Interaction Data Log** and **User Profile** (basic information for analysis such as timestamps and user profiles). DA SENDS data to the **User Profile** . DA SENDS data to the **Enrichment Engine** (summaries of metadata to enrich the interaction).

3.2.5 Enrichment Engine

Table 11: EE Component Card

ID	EE
Name	Enrichment Engine
Description	The Enrichment Engine gets the available processed data from the interaction analysis pipeline (stored in the Interaction Data Log, processed by the DA) and produces summaries of interaction issues that are presented to the Civil Servant so that the future versions of the interaction are optimized.
Functionalities	EE gathers the available information from the interaction and issues that are detected either by users (via the Citizenpedia and comments gathered by the SF) and based on the knowledge of the e-service produces a summary of interaction issues to be presented by the administrator (in this case the Civil Servant using the dashboard).
Input Data	Processed interaction data coming from the Data Analysis, information coming from the Citizenpedia (e.g., common questions by the citizens).
Output Data	Summary of recommendations to the Civil Servants to be presented in the dashboard.
Interaction with other	EE gets processed metadata about the interaction from the **DA**.

components	EE gets comments and statistic from the **Citizenpedia**. EE sends summaries of required changes to the **DB**.

3.3 Human computation

The human computation framework, that leverages the Simpatico project with the collaborative knowledge, provided by citizens and civil servants. It will complement the Simpatico environment with a place where citizens can solve their doubts and actively take part in the enhancement of the e-services. It exposes several tools for citizens and civil servants. Through the Question Answering Engine, citizens can post doubts and look up for terms and definitions. From the civil servants' side, this component is an easy channel to rapidly solve doubts. The Collaborative Procedure Designer describes current e-services in the form of flowcharts/diagrams and citizens are able to comment on them. Using the comments of the citizens posted in Collaborative Procedure Designer and the analytics provided by the Decision Support Tool component, civil servants can enhance e-services. Moreover, the Gamification Engine is a tool that can engage citizens and civil servants in their own interaction through games; this tool will be used in the second phase of experimentation with its related scenarios of usage.
Finally, the Dashboard provides Civil Servants with a tool where understanding what are the main problems for citizens in the usage of e-services and how they can correct to improve the user experience.

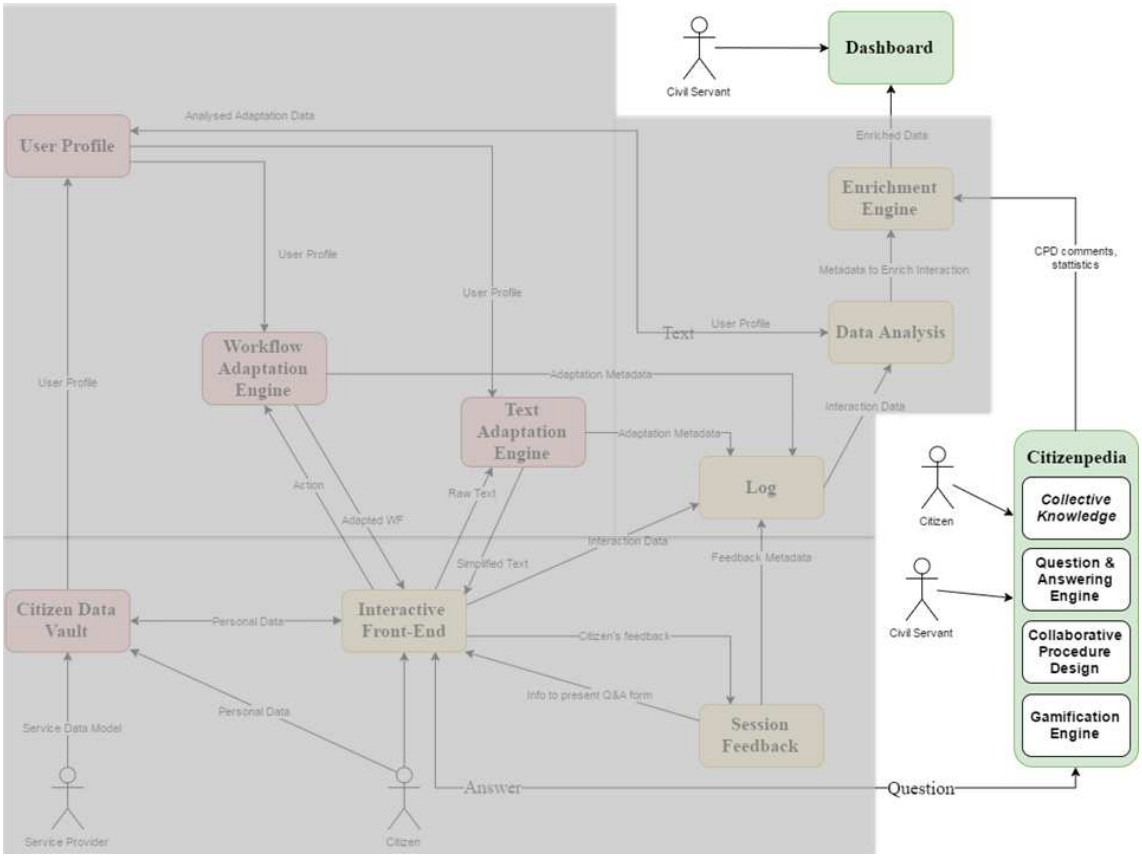


Figure 5: Human computation components

3.3.1 Citizenpedia

Table 12: Citizenpedia Component Card

ID	CIT
Name	Citizenpedia
Description	A human computation portal, included in the Simpatico platform as support to the interaction with public administration e-services. It will provide a place where citizens will be able to interact with other citizens and civil servants in order to solve their public administration-related doubts and collectively improve the user experience of e-services.
Functionalities	Citizenpedia is composed by different modules, each providing a functionality: • Question answering engine (QAE): a portal where citizens can post and resolve doubts regarding e-services and public

	administrations. • Collaborative procedure designer(CPD): a portal where citizens can collaborate in the definition of new e-services, which are represented as diagrams. • Collective knowledge base(CKB): a data storage system for the information generated in the Citizenpedia. It will include an indexing engine to increase searching capabilities. • Gamification engine(GE): a mechanism to manage badges and rewarding to Citizenpedia users, in order to increase engagement.
Input Data	It will be mainly of two types: • Questions, answers and comments created in the QAE. • Collaborative Procedure Designer comments
Output Data	It will be mainly of two types: • Questions, answers and comments created in the QAE by citizens and civil servants. • CPD Comments: comments left by citizens on diagrams created by civil servants. • Statistics: metrics taken within the Citizenpedia related to questions/answers/comments created over a particular e-service (e.g. number of comments, questions…).
Interaction with other components	• **Enrichment engine:** the EE will fetch from Citizenpedia questions/answers and CPD Comments created over a particular e-service and all the statistics described as output are analysed by this component • **Interactive Front-End**: doubts corresponding to an e-service are handled. Questions are posted in Citizenpedia through IF and answers are retrieved to expose them on the e-service GUI

3.3.2 Question Answering Engine

Table 13: QAE Component Card

ID	QAE
Name	Question answering engine
Description	Portal within Citizenpedia that allows the posting and resolution of

	questions and terms with ranking capabilities.
Functionalities	This component is built with a responsive web interface to enable the interaction of users (citizens and civil servants) from any type of device, e.g. smartphone or PC. Some moderator roles will be defined. Moderators will be in charge of managing and maintaining the quality of the generated knowledge through ranking and moderator/roles tasks.
Input Data	Questions, answers and comments created by the citizens.
Output Data	QAE contents (Questions, answers and comments) and statistics.
Interaction with other components	This component will interact with other Citizenpedia components: • **Collective knowledge base**: to store, retrieve QAE content, and make advanced searches on them. • **Gamification engine**: to retrieve the game score of each Citizenpedia user.

3.3.3 Collaborative Procedure Designer

Table 14: CPD Component Card

ID	CPD
Name	Collaborative Procedure Designer
Description	The tool that will enable to create graphical representations of public procedures in the form of diagrams. These diagrams will represent both e-services and (non-digital) services to be performed by citizens to achieve a specific goal. Thanks to this component, civil servants will be able to initially model and publish public administration procedures, implemented by both digital and non-digital services, while citizens could enrich procedures diagrams with feedbacks and annotations.
Functionalities	We identify two main functionalities: • Diagram editor: enables to create and edit a workflow diagram, using UML-like symbols. • Social/collaborative-like: that will enable to post comments over the diagram.

Input Data	There will be two main roles creating content, as input data: • Civil Servants: who will create and modify (draw) a diagram representing a procedure • Citizens: who will create comments on the diagrams.
Output Data	CPD Content: diagrams representing procedures, and comments left on them.
Interaction with other components	This component will interact with: • **Collective knowledge base**: to store and retrieve CPD content. • **Enrichment Engine**: with one way CPD comments fuelling.

3.3.4 Collective Knowledge Base

Table 15: CKB Component Card

ID	CKB
Name	Collective Knowledge Base
Description	This component manages and stores the data generated in the Citizenpedia, excluding the scores and reputation skills which are stored in the Gamification Engine. It defines the data model of such knowledge base, including entries, contributors, and revision entities among many others. Besides, it will undertake the integration of such knowledge base with the other two components of Citizenpedia that feed information to it, i.e. the question/answer engine and the collaborative procedures designer.
Functionalities	• Store and manage the data created in the QAE and CPD • Index the text-based data to provide advanced searching capabilities • Provide an API for other Simpatico components and 3rd party applications to query and fetch data
Input Data	QAE and CPD content. There are two ways of storing it: • User interface. • Programmatically through a REST API.
Output Data	QAE and CPD content. There are two ways of fetching it:

	• User interface. • Programmatically through a REST API.
Interaction with other components	This component will interact with other Citizenpedia components: • **Question answering engine**: to fetch or serve QAE content. • **Collaborative procedure designer**: to fetch or serve CPD content.

3.3.5 Gamification Engine

Table 16: GE Component Card

ID	GE
Name	Gamification Engine
Description	The Gamification Engine (GE) is a software component responsible of the implementation and execution of the logics associated to the rewarding mechanisms adopted to ensure user engagement. The usage of the GE is in the scope of Citizenpedia, to reward the active participation of the users (i.e., civil servants, professionals and citizens) in the collaborative development of Citizenpedia knowledge base. More precisely, actions are defined (e.g., contributing answers to questions, reviews, endorsements) that allow users to advance in the defined games. Game status is reported both to the single users and to the managers of Citizenpedia.
Functionalities	• Definition of the game logic. This functionality, executed by the Citizenpedia managers, allows the specification of actions done by the users that are relevant for game, and of the rules that define how these actions are rewarded with points, badges, and similar concepts. • Execution of the game. Based on the collected information from the other components of the system about the executed actions, this functionality is responsible of the execution of the game logic and of the update of the status of the users in the game. • Presentation of the game status (e.g., scoreboards, acquired badges, winners of the challenges…), both to single users and to the Citizenpedia managers (both single-user view and aggregated views).
Input Data	• Game Logics (game concepts, game actions, game rules), uploaded to the GE by the Citizenpedia managers. • Executed Game Actions (e.g., answers contributed to questions, reviews, endorsements), from the Citizenpedia.

Output Data	● Game Status (both single-user and aggregated)
Interaction with other components	The component interacts with: ● **Citizenpedia**: operations performed by users on the Citizenpedia knowledge base are forwarded to the GE in the form of Game Actions, and make the game evolve. ● **Citizenpendia**: the game status of users are acquired from the GE and suitably represented in the Citizenpedia front-end. ● **User Profile**: (optionally) information on the user profile is collected by the GE for personalizing the game for the user; information on the game activities and status of the user are stored in the User Profile for further analysis.

3.3.6 Dashboard

Table 17: DB Component Card

ID	DASH
Name	Dashboard
Description	The Simpatico Dashboard is a decision support system that provides different layers of information to be used by civil servants to enrich the interaction with the users. Three layers are initially foreseen for this component. The first layer provides information on text complexity. The second displays statistics coming from the Interaction Data Log (Section 3.2.2). The third layer shows the output of the analysis of the Enrichment Engine (Section 3.2.5).
Functionalities	The **text complexity layer** provides linguistic analysis that evaluates the comprehensibility and the readability of the texts. It is language-dependent. Given a text as input, the civil servant receives as an output some metrics calculated on the input text: ● A language detector that guesses the language of the input text (Italian, Spanish, English). ● A general readability index for the language (Gulpease for Italian, Flesch-Szigriszt for Spanish, Flesch for English). It does not take into consideration the content of the text, but quantitative information such as the length of sentences and words. ● Three indices that assign the text to three levels of education (elementary, middle, high school level). ● Part-of-speech tags distribution over the words.

	• General statistics on the text (number of sentences, tokens, words, content words). The second layer shows **statistics from the Interaction Data Log**, displaying information that is useful to the civil servants for analyzing past interactions and for enriching future interactions. The last layer outputs **information from the Enrichment Engine**, i.e. summaries of interactions and issues that will help the civil servants improving the interactions.
Input Data	• Enriched data from the EE. • A text to be analyzed inserted by the civil servant.
Output Data	For what concerns the text analysis layer, linguistic annotations for the text and a list of numeric values (only to allow the civil servant to adapt the text, no interaction with other components is provided as output). For what concerns the other layers, detailed and aggregated representations (including graphical views) of the interaction logs and of the outputs of the EE. The outcome of the dashboard is also provided in a machine-readable form, so that other applications (even external ones) can use it easily.
Interaction with other components	The component interacts with: • **Enrichment engine**

3.4 Use cases and Sequence Diagram

In this section an overview is given of the main use cases that will be addressed in the course of the project. For each use case, it will be clearly identified the *participating actor*, i.e., the role responsible to trigger the case and, at the same time, the beneficiary of the actions taken within the case. Use cases are labeled with names which in some cases recall the action taken by the actor to trigger the use case, in some others they identify the primary action taken by the SIMPATICO framework. In the Figure 6 a **UML use case diagram** of the main cases addressed in the project is depicted:

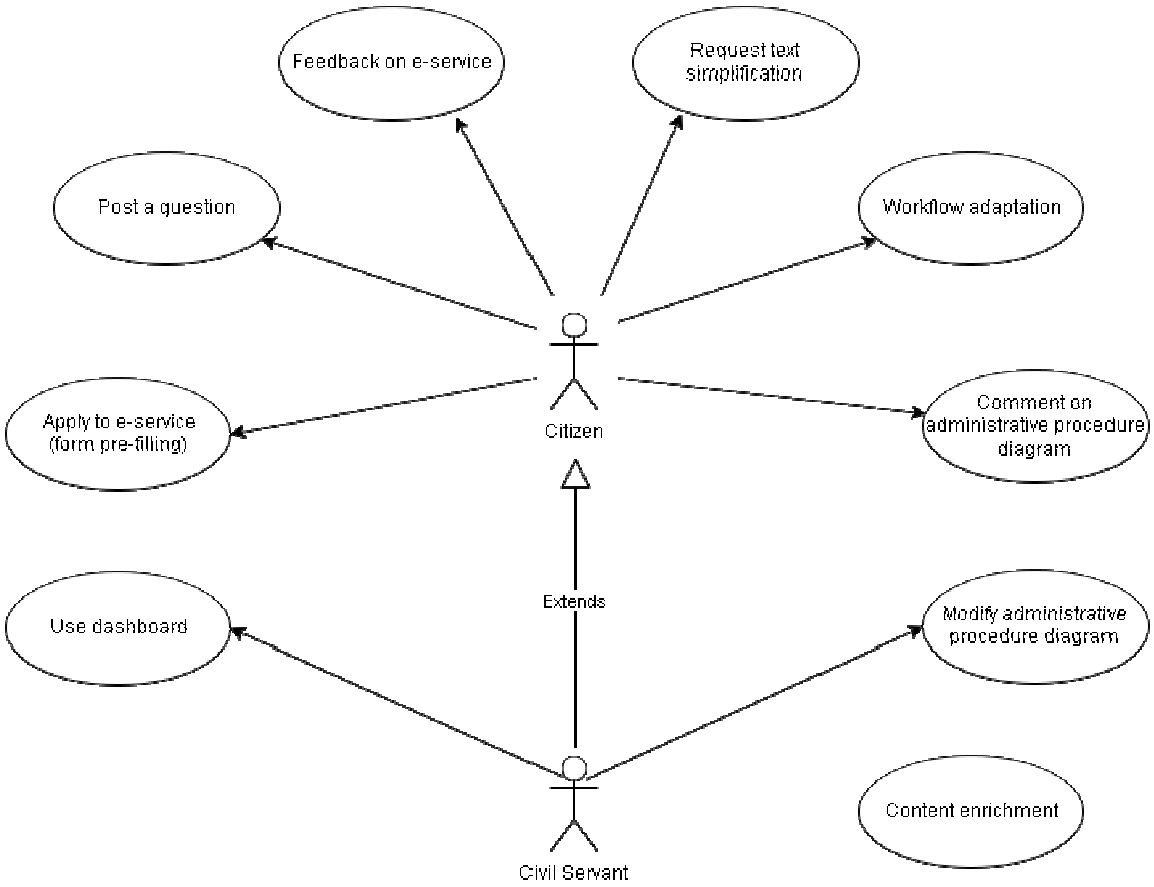


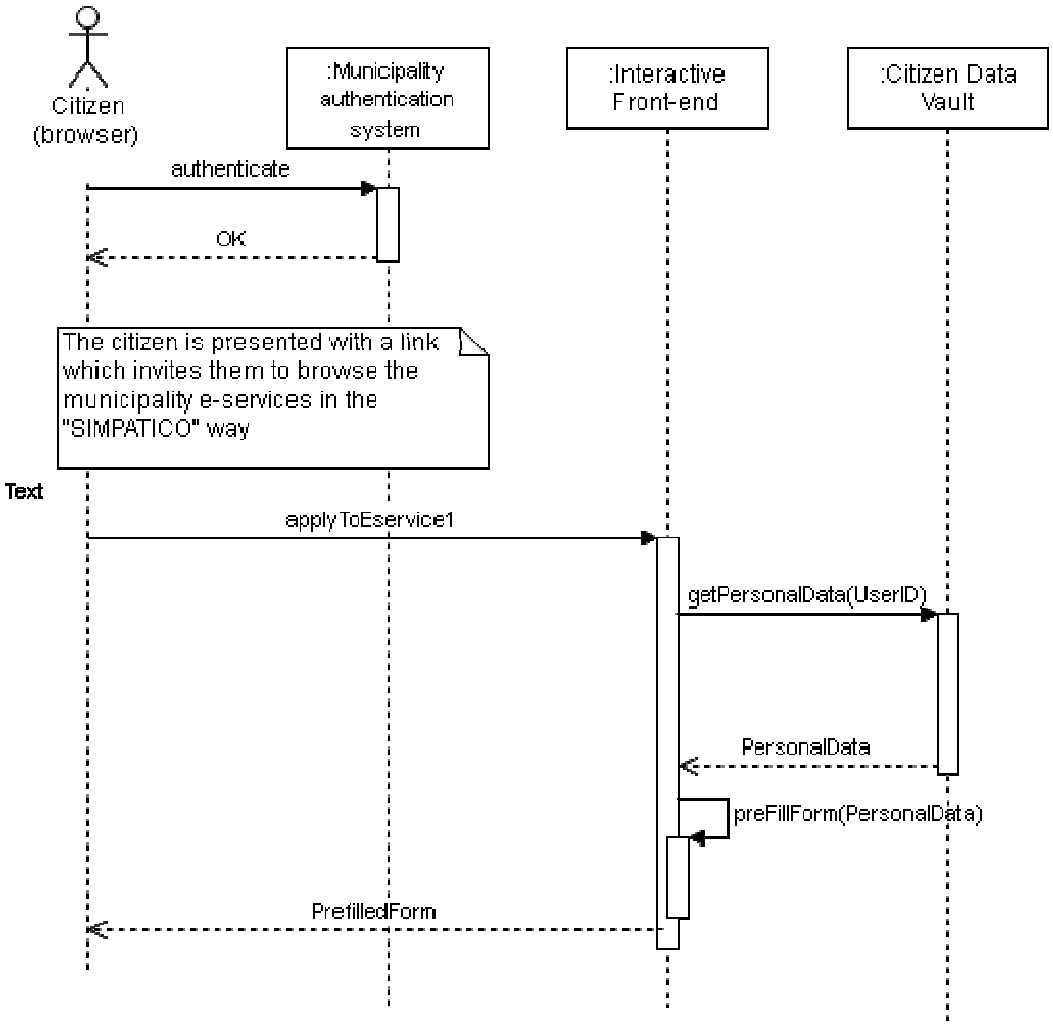
Figure 6: UML use case diagram

All cases are triggered by human actors with the exception of the "Content Enrichment" use case. Content enrichment, in fact, is triggered by batch procedures which are periodically run by the SIMPATICO framework.

In the following, we report a list of **UML sequence diagrams** corresponding to each of the use cases depicted in the Figure 6. Sequence diagrams highlight how the SIMPATICO framework's components interact to each other's in order to carry out a specific action requested by the participating actor.

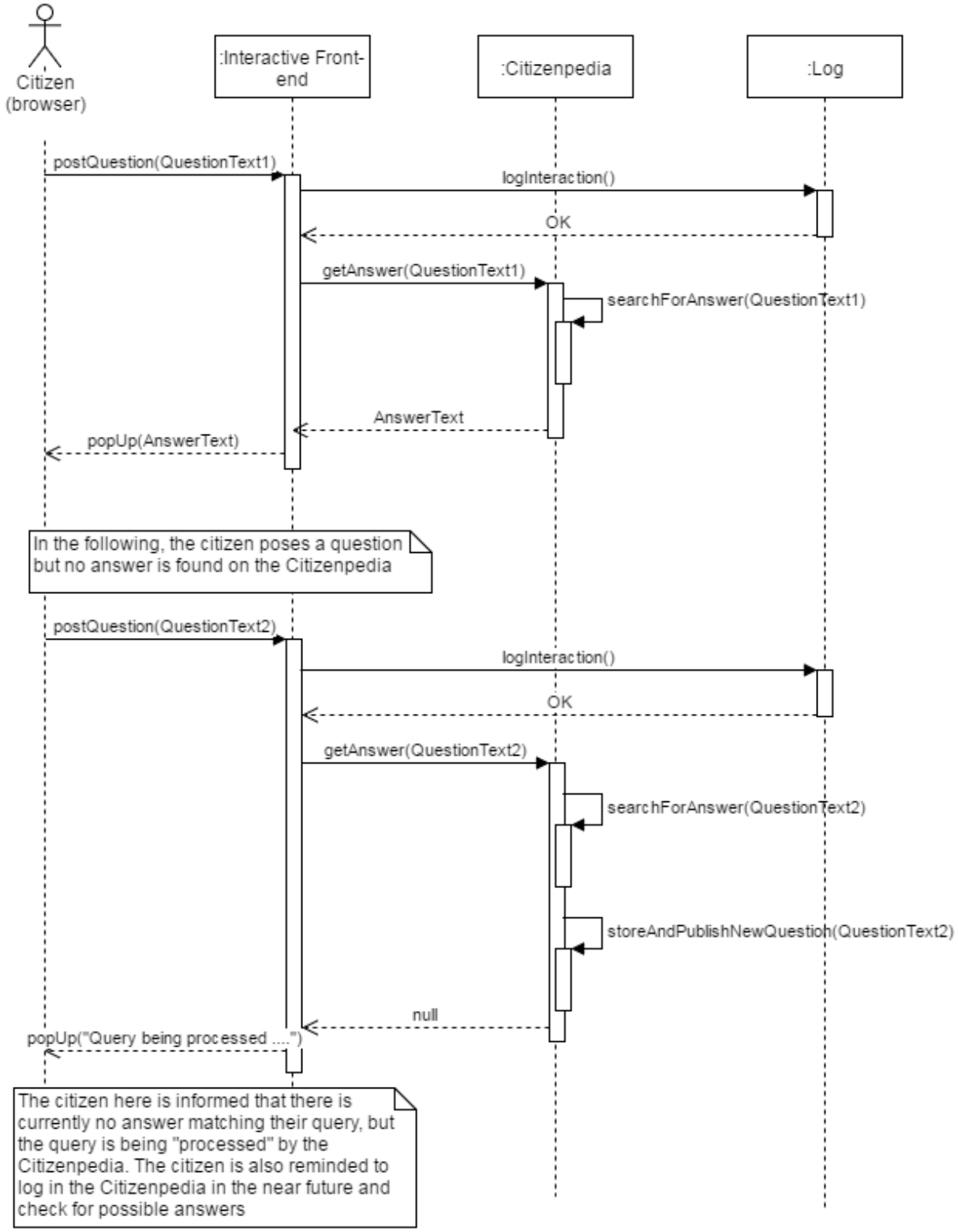
3.4.1 Citizen – Apply to e-service

Role responsible for the use case	The Citizen
Use case pre-conditions	The Citizen is provided with credentials which grant them access to both the Municipality e-services and the SIMPATICO services
Use case post-conditions	The Citizen is presented with a form where the fields requiring their personal data have been automatically pre-filled



3.4.2 Citizen – Post a question

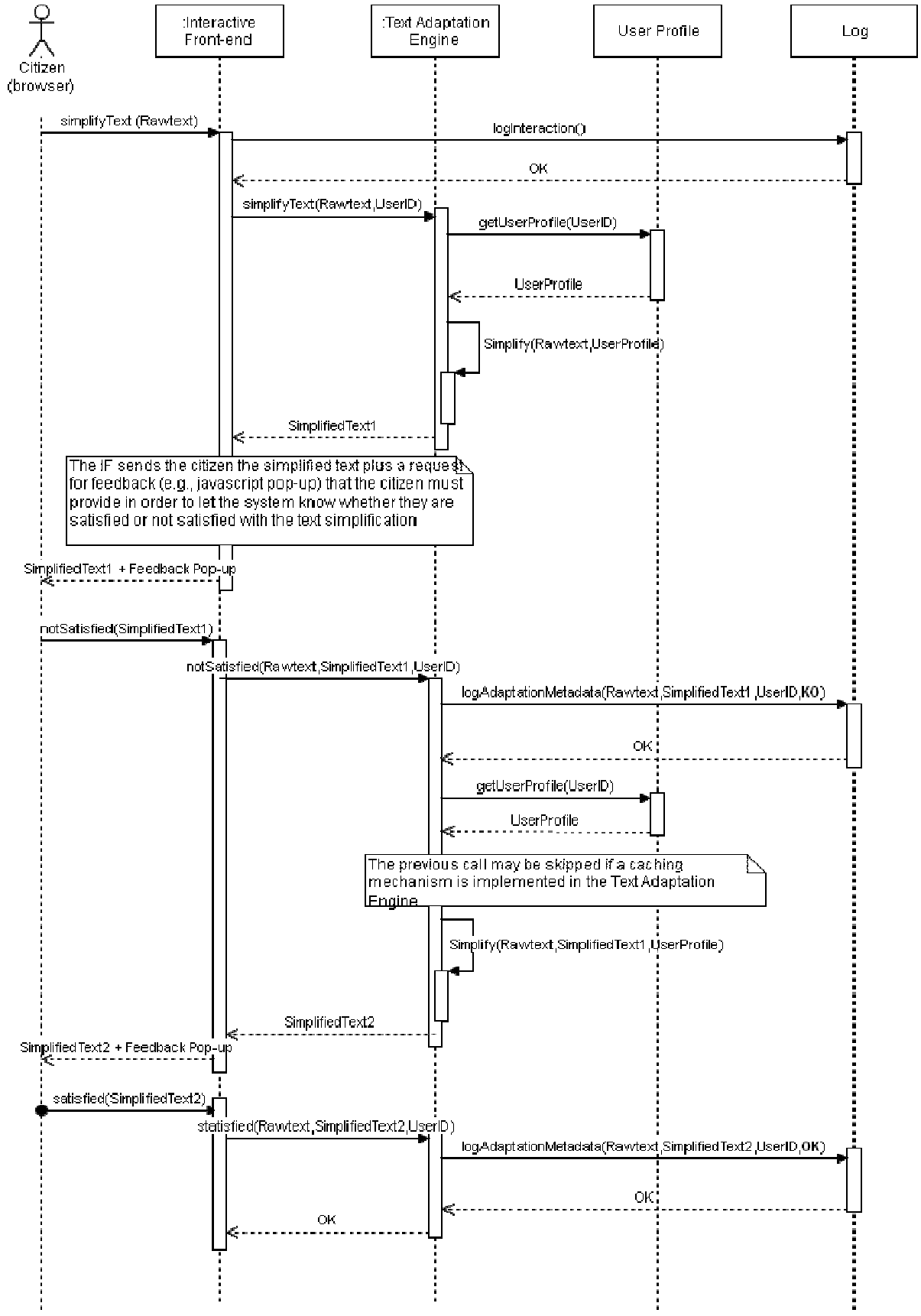
Role responsible for the use case	The Citizen
Use case pre-conditions	The Citizen has already been authenticated and has accessed a Municipality e-service. There is a step in the e-service workflow which is not clear to them, so they do not know how to proceed
Use case post-conditions	The Citizen has a clear idea of how to proceed



3.4.3 Citizen – Request text simplification

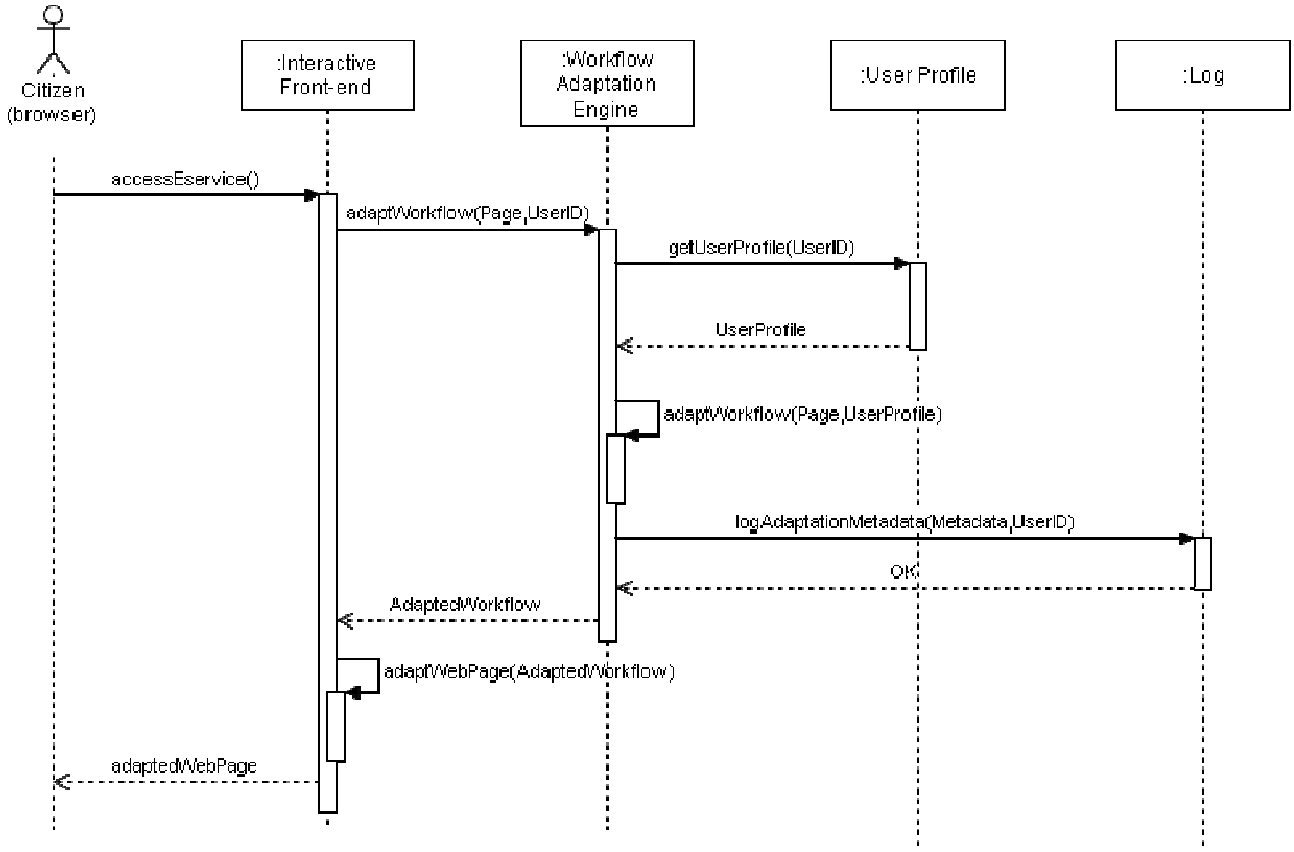
Role responsible for the use	The Citizen

case	
Use case pre-conditions	The Citizen has already been authenticated and has accessed a Municipality e-service. There is a sentence which is not fully comprehensible to them
Use case post-conditions	The Citizen has clearly understood the text meaning


Citizen (browser)
:Interactive Front-end
:Text Adaptation Engine
User Profile
Log
simplifyText (Rawtext)
logInteraction()
OK
simplifyText(Rawtext,UserID)
getUserProfile(UserID)
UserProfile
Simplify(Rawtext,UserProfile)
SimplifiedText1
The IF sends the citizen the simplified text plus a request for feedback (e.g., javascript pop-up) that the citizen must provide in order to let the system know whether they are satisfied or not satisfied with the text simplification
SimplifiedText1 + Feedback Pop-up
notSatisfied(SimplifiedText1)
notSatisfied(Rawtext,SimplifiedText1,UserID)
logAdaptationMetadata(Rawtext,SimplifiedText1,UserID,KO)
OK
getUserProfile(UserID)
UserProfile
The previous call may be skipped if a caching mechanism is implemented in the Text Adaptation Engine
Simplify(Rawtext,SimplifiedText1,UserProfile)
SimplifiedText2
SimplifiedText2 + Feedback Pop-up
satisfied(SimplifiedText2)
satisfied(Rawtext,SimplifiedText2,UserID)
logAdaptationMetadata(Rawtext,SimplifiedText2,UserID,OK)
OK
OK

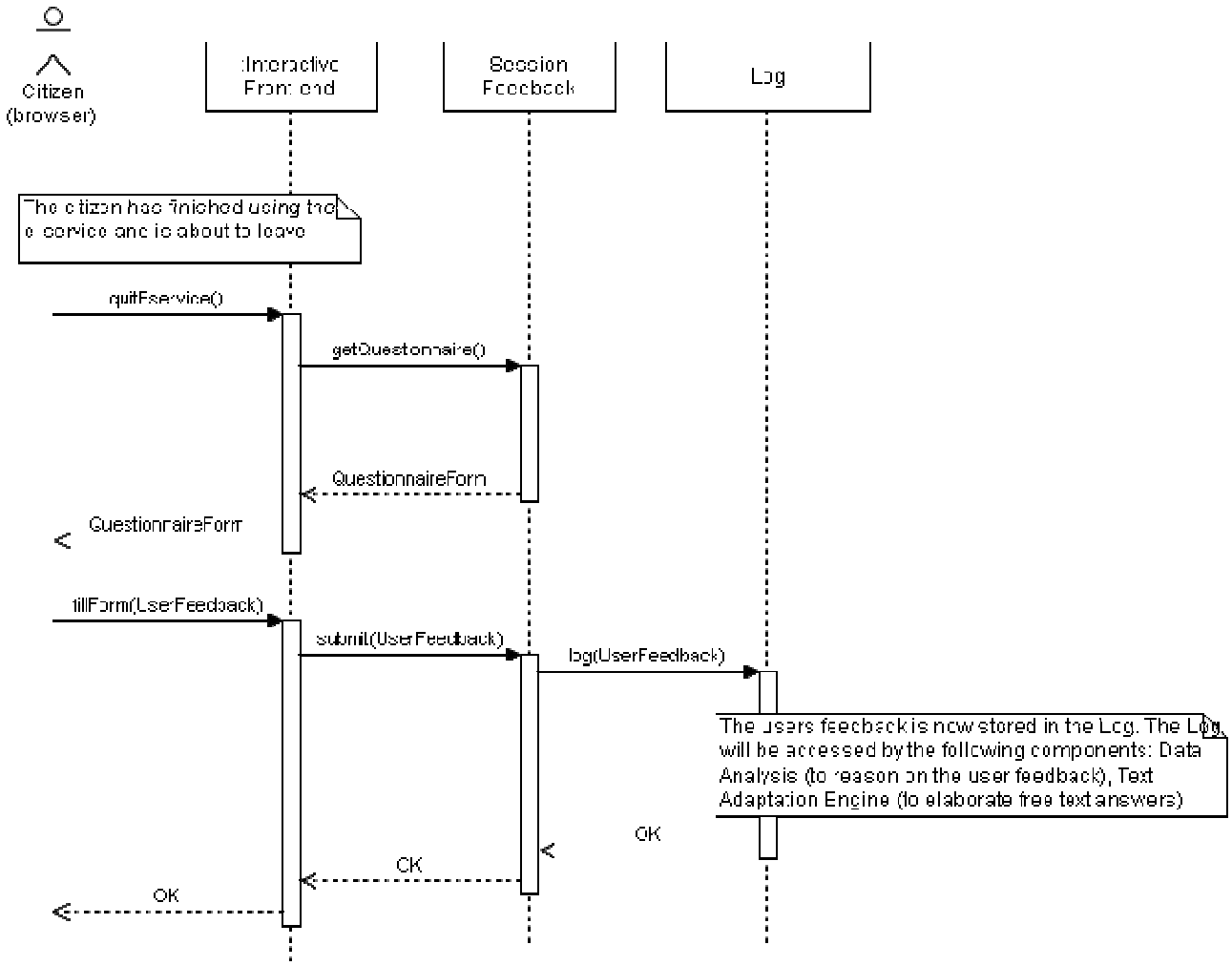
3.4.4 Citizen – Workflow adaptation

Role responsible for the use case	The Citizen
Use case pre-conditions	The Citizen has already been authenticated
Use case post-conditions	The Citizen is presented with a workflow properly adapted to their profile



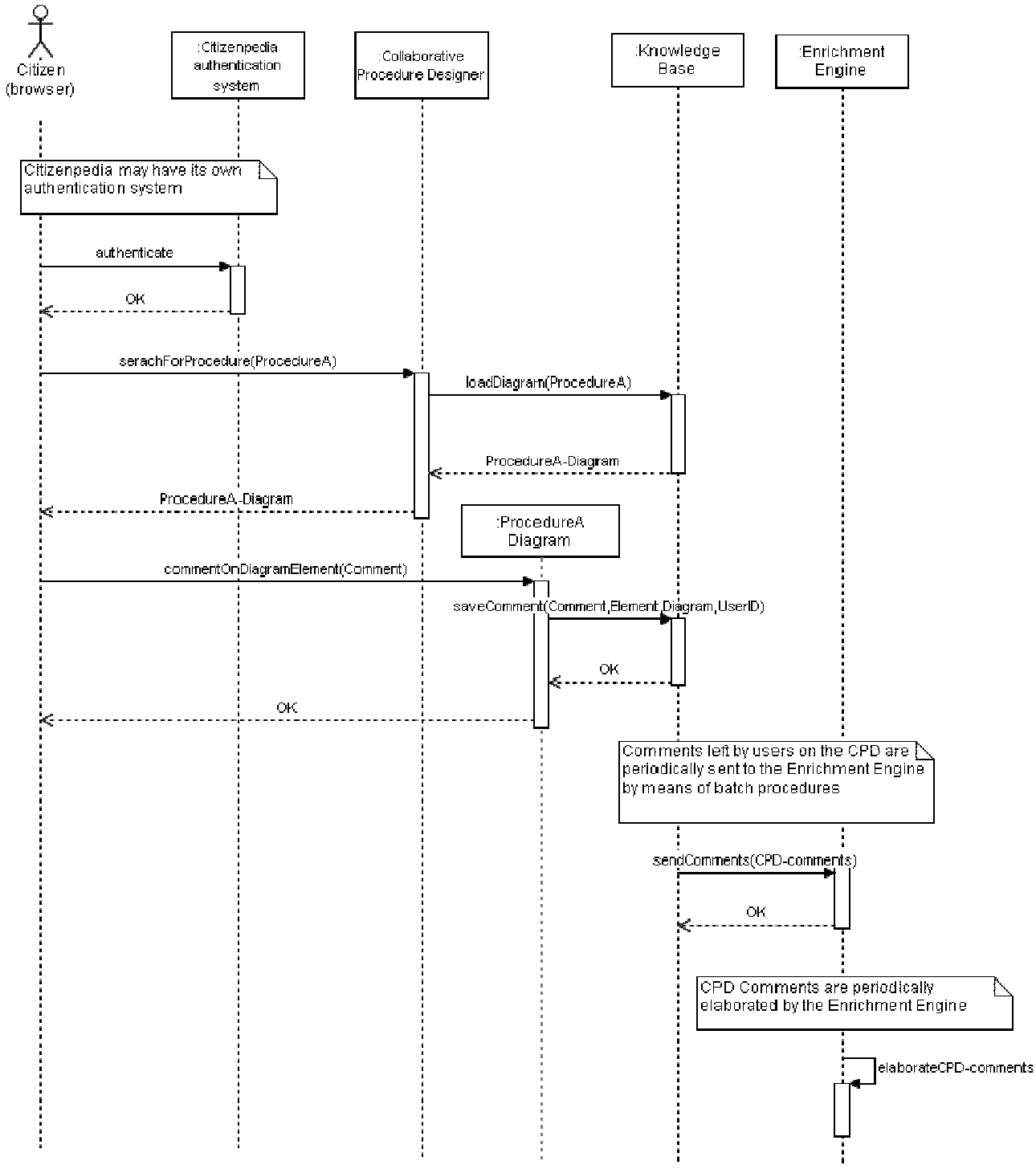
3.4.5 Citizen – Feedback on e-service

Role responsible for the use case	The Citizen
Use case pre-conditions	The Citizen has finished using the e-service and is quitting
Use case post-conditions	The Citizen's feedback is acquired and ready to be further elaborated



3.4.6 Citizen – Comment on administrative procedure diagram

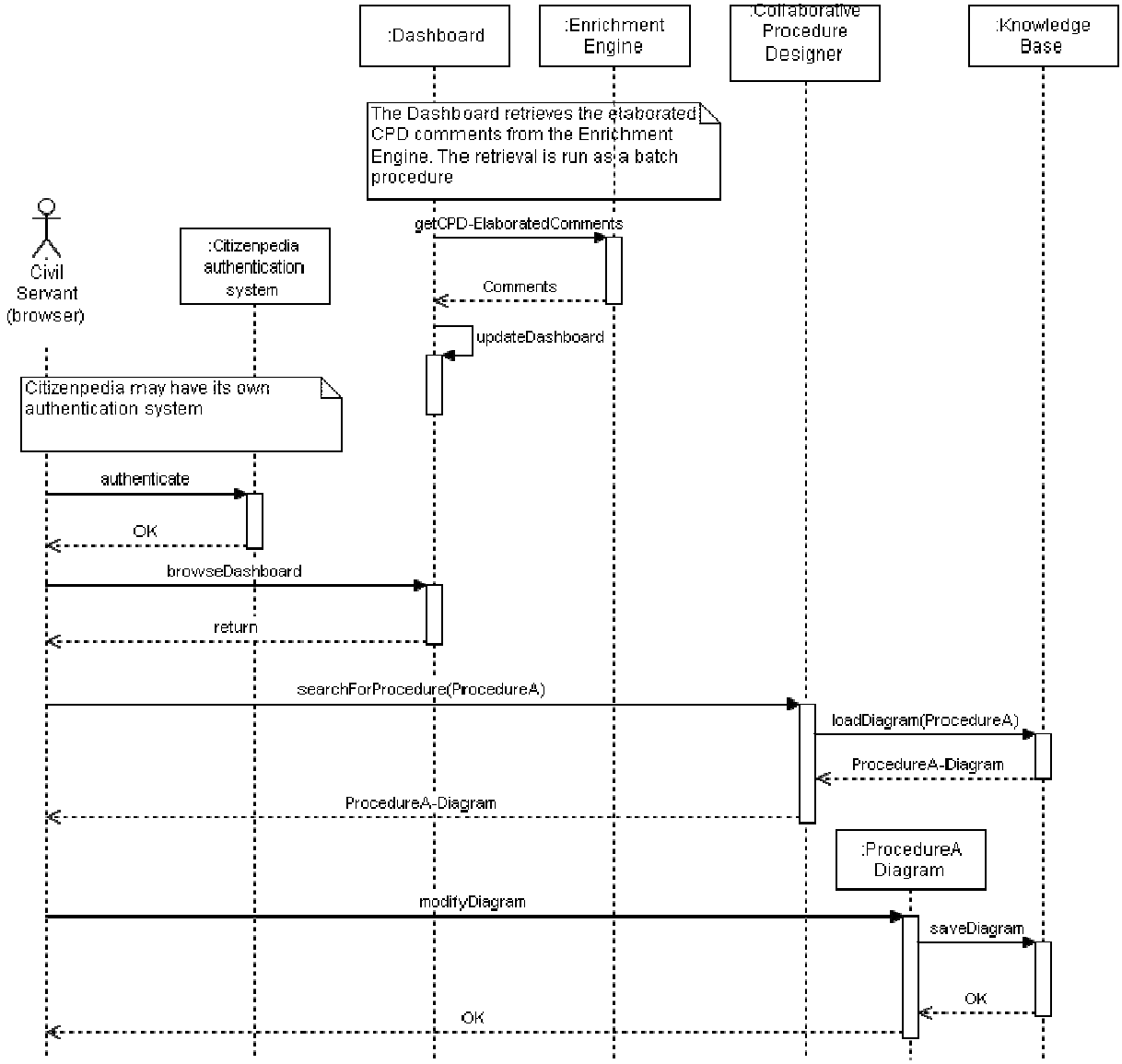
Role responsible for the use case	The Citizen
Use case pre-conditions	The Citizen is provided with credentials which grant them access to the Citizenpedia
Use case post-conditions	The Citizen's comment is acquired and ready to be further elaborated



3.4.7 Civil Servant – Modify administrative procedure diagram

Role responsible for the use case	The Civil Servant

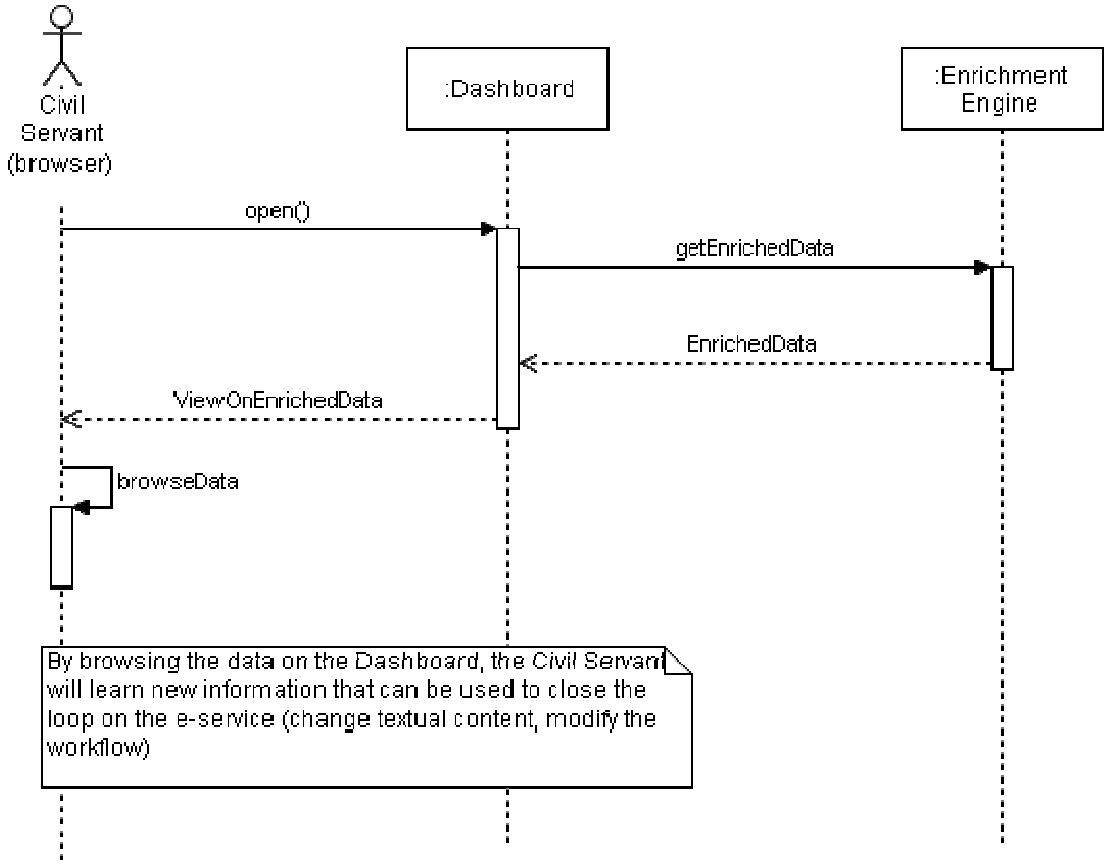
Use case pre-conditions	The Civil Servant is provided with credentials which grant them access to the Citizenpedia
Use case post-conditions	The administrative procedure reflects the changes made by the Civil Servant



3.4.8 Civil Servant – Use dashboard

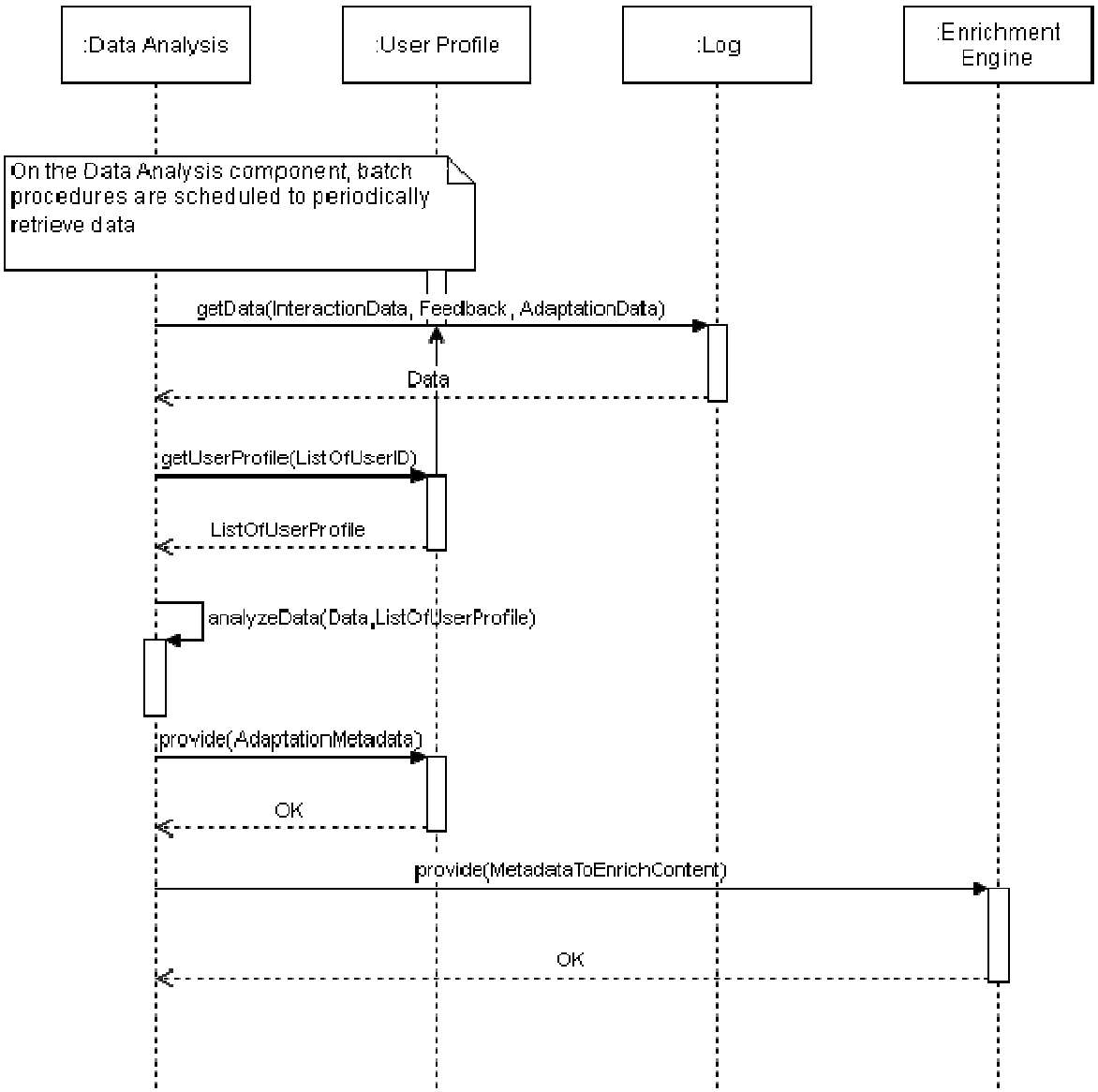
Role responsible for the use case	The Civil Servant
Use case pre-conditions	The Civil Servant has already gained access to the Dashboard

Use case post-conditions	The Civil servant has a clue on how to modify the e-service's textual content and workflow in order to meet the Citizens' demand



3.4.9 Simpatico system – Content enrichment

Role responsible for the use case	The Simpatico system
Use case pre-conditions	-
Use case post-conditions	The Enrichment engine stores information on how to further enrich the e-services content


:Data Analysis
:User Profile
:Log
:Enrichment Engine
On the Data Analysis component, batch procedures are scheduled to periodically retrieve data
getData(InteractionData, Feedback , AdaptationData)
Data
getUserProfile(ListOfUserID)
ListOfUserProfile
analyzeData(Data,ListOfUserProfile)
provide(AdaptationMetadata)
OK
provide(MetadataToEnrichContent)
OK

4 Components interaction scenarios

This section describes how SIMPATICO tools and techniques are integrated in Trento, Sheffield and Galicia pilots. For each pilot the section gives a general overview of the integration schema and identifies possible specific constraints and issues. Critical points are the integration of the Interactive Front-end with the public body system, the different roles of the Civil Servant and the final user (Citizen/Professional) in the use of the tools provided by SIMPATICO and finally the authentication and authorization schema and issues. As for the interaction with legacy systems, the advanced interaction functionalities of SIMPATICO are realized by injecting code that manipulates the "legacy" front-end as required. This can be accomplished through single JavaScript files enforced server-side or by providing a plug-in for browser-side adaptation through a dedicated SIMPATICO browser extension.

4.1 Trento Pilot

Trento pilot is based on the Enrolment to day nursery service and on the Permit on acoustic derogation for temporary activities. The table below matches the TRENTO e-services with SIMPATICO components and usage scenarios.

Table 18 Service to SIMPATICO feature application mapping

E-Service	Scenario	Features	Description
Enrolment to day nursery service	Citizen needs to enrol his child to day nursery service. Citizen, using the municipality e-service, accesses to the integrated SIMPATICO functionalities and tools. More precisely he uses the text and workflow simplification instruments; he uses the information automatically provided by the Citizen data Vault; and eventually he proposes question through the Query and Answer Engine.	Interactive Front End	Through the Interactive Frontend the user will access to all the SIMPATICO components and tools provided.
		Text Adaptation Engine	Complex words and phrases are enlighten. When a user clicks on one enlighten phrase or word, a pop-up within a simplified version of it is showed according to the user profile.
		Text Adaptation Engine	The user selected module phrases and words are automatically translated in the user language/profile.
		Text Adaptation Engine + Workflow Adaptation Engine	Near the most difficult fields to fill (understood thanks to logs or other evidences such as Q&A) will be present a symbol/icon. Clicking on it will be present a pop-up

			within a text explaining what it is asked to insert for that specific field according to the user profile
		Workflow Adaptation Engine	Parts of the digital module are shown/hidden on the basis of an optimized compilation process defined according to the user profile.
		Workflow Adaptation Engine	Compilation support: When the citizen chooses an option that changes the workflow (with the rules of the "Sportello Telematico" Solution), the parts of the module that are no longer be compiled (because disabled) will be hidden, while the parts of the module that return fillable will be shown again. In addition to this the next section to compile will be highlighted.
		Citizen Data Vault	All the useful information filled in the module (as the information on the degrees of relationship of persons) are stored in the CDV and available for future usage. The CDV data will work on citizen information space different from the one provided by the municipality internal Civil Register. If any information requested in the module is already present in the CDV, it will be retrieved and used to pre-fill the module in a different format than

			the information retrieved from the DB that provides reliable information from Administration (such as the Civil Register). In this way we are going to mark the different level of trust between the data stored in the CDV and the one stored in the Civil Register.
		Question and Answer Engine	The citizen can select a part of the digital module and ask for clarification.
Permit on acoustic derogation for temporary activities	A citizen asks for a permit derogation for temporary activities. Citizen, using the municipality e-service, accesses to the integrated SIMPATICO functionalities and tools. More precisely s/he uses the text and workflow simplification instruments; s/he uses the information automatically provided by the Citizen data Vault; and eventually s/he proposes question through the Query and Answer Engine.	Interactive Front-End	Through the Interactive Front-End the user will access to all the SIMPATICO components and tools provided.
		Text Adaptation Engine	Complex words and phrases are enlighten. When a user clicks on one enlighten phrase or word, a pop-up within a simplified version of it is showed according to the user profile.
		Text Adaptation Engine + Workflow Adaptation Engine	Near the most difficult fields to fill (understood thanks to logs or other evidences such as QAE) will be present a symbol/icon. Clicking on it will be present a pop-up within a text explaining what it is asked to insert for that specific field according to the user profile
		Workflow Adaptation Engine	Parts of the digital module are shown/hidden on the basis of an optimized compilation process defined according to the user profile.

		Citizen Data Vault	All the useful information filled in the module are stored in the CDV and available for future usage. The CDV data will work on citizen information space different from the one provided by the municipality internal Civil Register. If any information requested in the module is already present in the CDV, it will be retrieved and used to pre-fill the module in a different format than the information retrieved from the DB that provides reliable information from Administration (such as the Civil Register). In this way we are going to mark the different level of trust between the data stored in the CDV and the one stored in the Civil Register.
		Workflow Adaptation Engine	Compilation support: When the citizen chose an option that change the workflow (with the rules of the "Sportello Telematico" Solution), the parts of the module that are no longer be compiled (because eg disabled) will be hidden, while the parts of the module that will return fillable will be shown again. In addition to this the next section to compile will be highlighted.
		Question and Answer Engine	The professional can select a part of the digital module and ask for clarification.

	A professional asks for a permit derogation for temporary activities. Professional, using the municipality e-service, accesses to the integrated SIMPATICO functionalities and tools. More precisely s/he uses the text and workflow simplification instruments; s/he uses the information automatically provided by the Citizen data Vault; and eventually s/he proposes question through the Query and Answer Engine.	Interactive Front-End	Through the Interactive Front-End the user will access to all the SIMPATICO components and tools provided.
		Text Adaptation Engine	Complex words and phrases are enlighten. When a user clicks on one enlighten phrase or word, a pop-up within a simplified version of it is showed according to the user profile.
		Text Adaptation Engine	The user selected module phrases and words are automatically translated in the user language/profile.
		Text Adaptation Engine + Workflow Adaptation Engine	Near the most difficult fields to fill (understood thanks to logs or other evidences such as Q&A) will be present a symbol/icon. Clicking on it will be present a pop-up within a text explaining what it is asked to insert for that specific field according to the user profile
		Workflow Adaptation Engine	Parts of the digital module are shown/hidden on the basis of an optimized compilation process defined according to the use profile.
		Workflow Adaptation Engine	Compilation support: When the citizen chose an option that change the workflow (with the rules of the “Sportello Telematico” Solution), the parts of the module that are no longer be compiled (because eg

			disabled) will be hidden, while the parts of the module that will return fillable will be shown again. In addition to this the next section to compile will be highlighted.
		Citizen Data Vault	All the useful information filled in the module are stored in the CDV and available for future usage. The CDV data will work on citizen information space different from the one provided by the municipality internal Civil Register. If any information requested in the module is already present in the CDV, it will be retrieved and used to pre-fill the module in a different format than the information retrieved from the DB that provides reliable information from Administration (such as the Civil Register). In this way we are going to mark the different level of trust between the data stored in the CDV and the one stored in the Civil Register.
		Question and Answer	The user can select a part of the digital module and ask for clarification.
All Services	Civil Servants through the SIMPATICO Dashboard analyse the e-services user interactions. The analysis can result both in changing the e-service interaction model or in adding interaction information or	Dashboard Session Feedback Interaction Data Log Data Analysis Enrichment Engine	The Civil servants can receive reports about the use of the front-end
		Collaborative	The Civil servants will be

	specification in Citizenpedia.	Procedure Design	able to graphically design administrative procedures. Each administrative procedure will have to clearly state the interactions between the citizen requesting a service and the PA offering that service. Interactions may be in the form of on-line form filling, paper form to be filled and sent to the PA by ordinary email, telephone conversation, face-to-face meeting
	Citizen, when required, access to Citizenpedia in order to have information about e-services interaction process.	Collaborative Procedure Design	The citizen can access the administrative procedures' diagrams and comment on the interaction elements in the case they do not fully understand the steps to take in order to receive the service
	Professional, when required, access to Citizenpedia in order to have information about e-services interaction process.	Collaborative Procedure Design	The professional can access the administrative procedures' diagrams and comment on the interaction elements in the case they do not fully understand the steps to take in order to receive the service

All the SIMPATICO tools and techniques are provided to the e-service users in a convenient and transparent way. Most of the functionalities are provided through the Interactive Front-End component which is fully integrated with the e-services.

For what concerns the **integration strategy** that will be adopted for the Trento use-case, the intention is to **validate SIMPATICO solutions in integration with the Municipality e-service portal** currently under delivery. Since "Sportello Telematico" is provided by an external service provider not directly involved in the SIMPATICO project, the integration has to be as nonintrusive as possible: the integration of SIMPATICO tools and techniques must **exploit the integration models supported by "Sportello Telematico"**. These include:

1) The possibility to inject Javascript in the different digital modules;
2) To invoke REST web services to handle interaction with legacy systems.

The injection of Javascript in the digital module is possible only if it does not interfere with the module interaction logic. This means that the injected Javascript can operate on the DOM (Document Object Model) modifying only the static elements of the document (in particular, the labels and text descriptions), but cannot operate in the fields that the user shall fill.

“Sportello Telematico” supports operations on the fields, e.g. pre-filling their values, via explicit external calls. These operations are handled via a unique REST web service, which is responsible of routing the proper request to the proper external service and/or data source and to compose the reply. The web service request and response must implement a specific simple grammar: the request message formatted in XML and contains an array of key-value couple (request params); the response message in formatted in XML and contains an array of key-value couple (relations) or a set of key-value couples (record).

The above constraints have impacts on the main components interaction schema depicted in the Figure 7. As far as the user interaction schema the entry point for Citizens and Professionals are the e-services HTML page provided by Sportello Telematico and the SIMPATICO Citizenpedia. The e-service HTML page, instrument with specific Javascript, sends User Auth Data (eg. user digital identity), User Personal Data (eg. data inserted in the form) and Interaction Data (es. time to complete part of the form) to the SIMPATICO Interactive Front-end component. The received data is propagated by Interactive Front-end to the other SIMPATICO components. The Auth Data, passed from the e-service to the Interactive Front-end, are passed to Trento Authorization and Authentication Service in order to check user identity. During the user-e-service interaction, the Interactive Front-end gains personal information and data which are stored in the Citizen Data Vault. Eventually, when possible, the e-service HTML form will be going to be prefilled with information and data previously stored in Citizen Data Vault (CDV). Finally the interaction between the user and Citizenpedia is direct and asynchronous in respect to the interaction with the e-services. The user can use Citizenpedia when s/he requires to better understand the way s/he has to interact with the e-services.

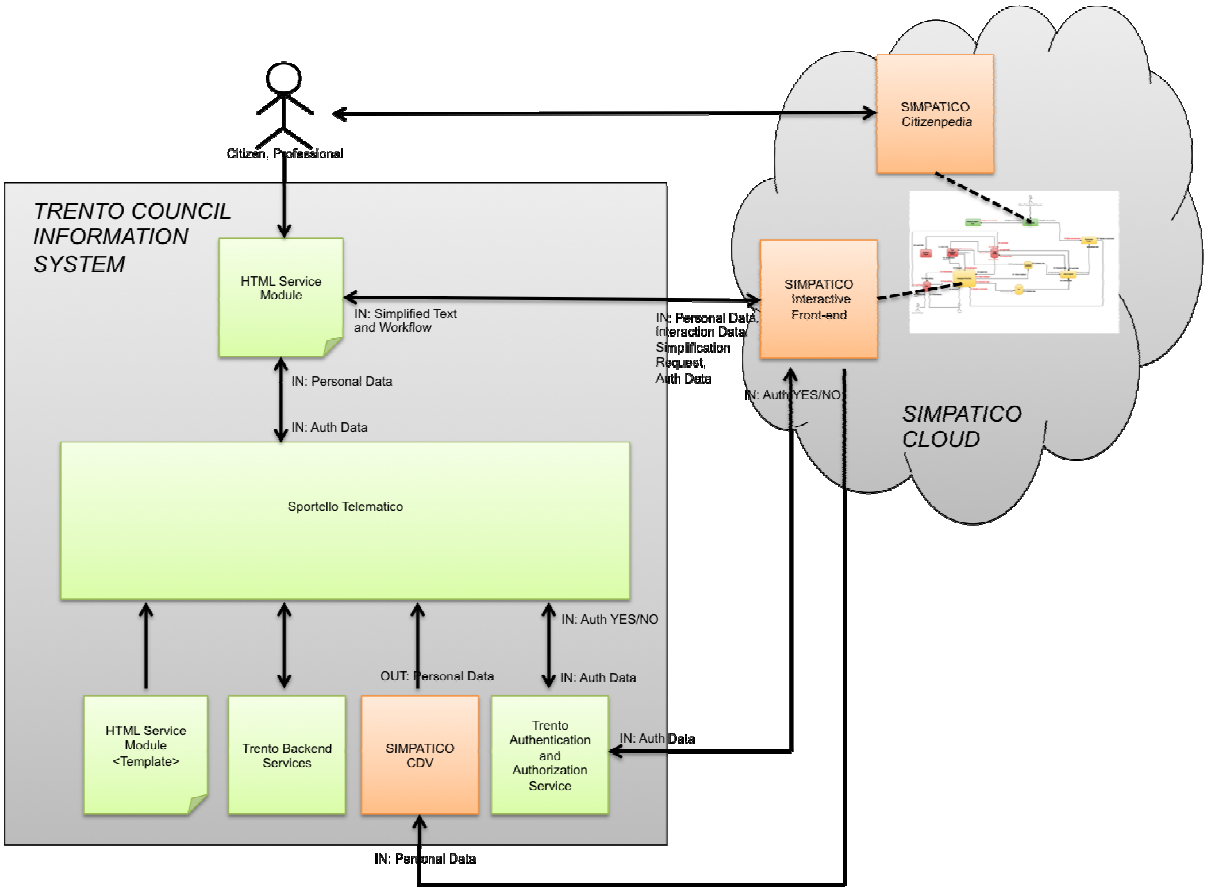


Figure 7: Sportello Telematico and SIMPATICO citizen interaction schema

As far as the Civil Servants, the interaction schema - as reported in Figure 8 - provides three different entry points. The Sportello Telematico which gives the Civil Servant a specific view in the use of the e-services; the SIMPATICO Dashboard which gives the Civil Servant a specific view of the user-e-service interaction complexity and problems; the Citizenpedia which gives the Civil Servant the possibility to document e-service procedure and interaction and to respond worldwide the problems and questions raised by the users. The use of the three tools gives the Civil Servant an aggregated and vertical complete overview of the users-e-services interactions and of the interaction critical points.

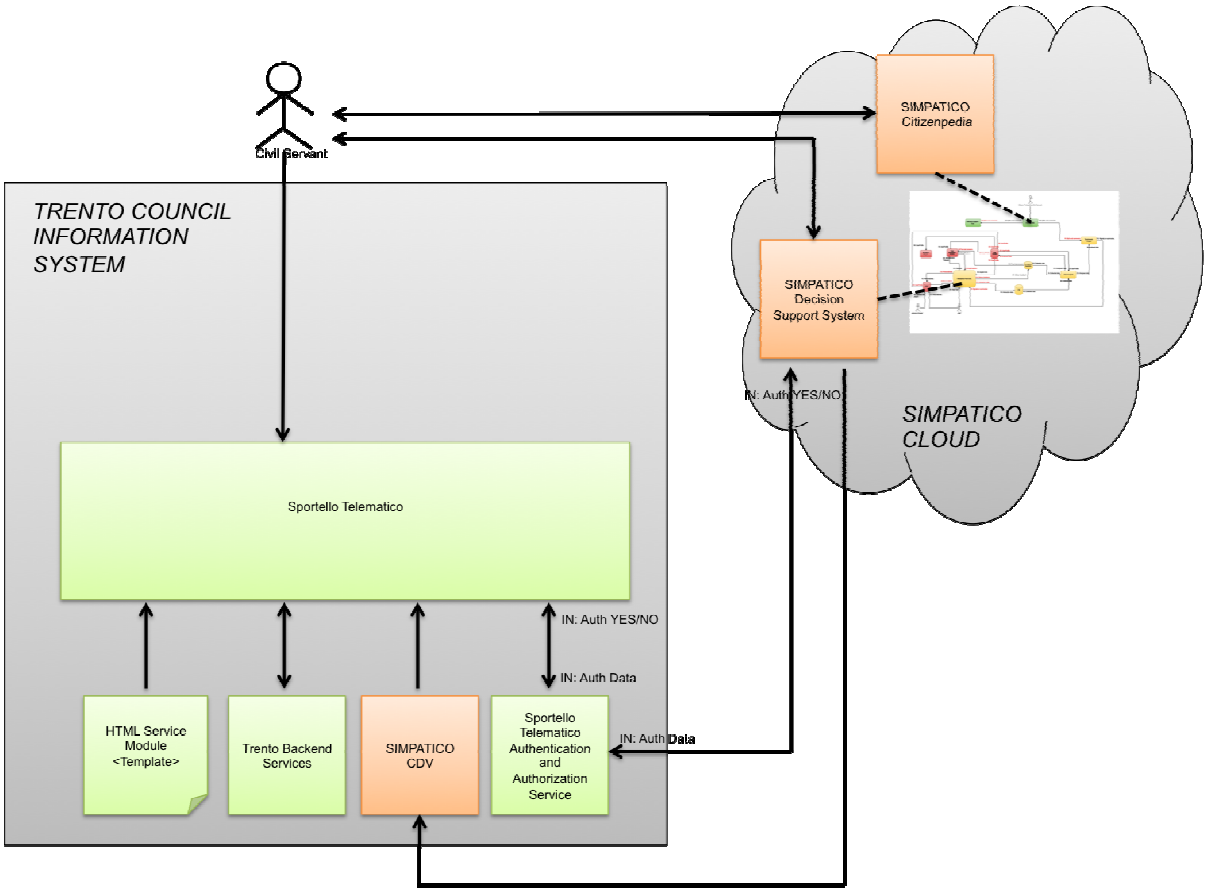


Figure 8: Sportello Telematico and SIMPATICO civil servant interaction schema

As reported in the Figure 8, this first version of the pilot requires SIMPATICO components being deployed on the SIMPATICO cloud. The municipality information system is going to be the container just for Sportello Telematico and for CDV. The tightly interaction schema between Sportello Telematico and CDV is mainly required to the constraints bound to the fact the CDV contains user personal data. This solution will prevent voluntary or accidental access to the personal data. This solution simplifies the integration security and authentication issues between Sportello Telematico and CDV. One important point to stress is that some interactions take place between Interactive Front-end and CDV and between Interactive Front-end and Trento Authentication and Authorization Service. This is mainly due to the need pass to the Citizen Data Vault the user personal data gained during the interaction, and to the need to check and verify the user identity before using SIMPATICO tools.

In the end, some specific words must be said about the authentication aspect. The civil servant, citizen and professional access to Sportello Telematico is handled via the Autenticazione del Cittadino Service provided by the Provincia Autonoma di Trento. The service provides an authentication mechanism based on Shibboleth which returns an user specific authorization token. In the use of SIMPATICO tools we will have to put in place a mechanism to verify the user identity coherent with the Shibboleth authentication model. On the other hand as specified above the citizen, professional and civil servant can access to Citizenpedia also with different authentication

mean. Finally the Civil Servant can also access to Sportello Telematico with a specific authorization service. This may lead to situations where users have different digital identities (corresponding to the different authentication services). This specific problem is not explicitly address in this first version of the architecture, but needs to be taken in consideration for the future project activities.

4.2 Sheffield Pilot

Sheffield Pilot is facing the challenges in engaging migrants in using the website and the services provided by Council. SIMPATICO provides an opportunity to the council to extend the reach of the website to people new to Sheffield and having English as their second language. Sheffield Pilot is based on School Attendees and Parenting skills courses. The table below matches the Sheffield e-services with SIMPATICO components and usage scenarios.

Table 19: Sheffield services to SIMPATICO feature application mapping

E-services	Scenarios	Components	Descriptions
School Attendees	Information advising why school attendance is important	Text Simplification	Complex sentences/phrases/words are pre-highlighted so that it's clear to the user that other alternatives are available. Upon request, highlighted sentences/phrases/words are replaced with a simpler version.
	Form to report suspected truancy	Citizen Data Vault	Information about the citizen and their family is stored and can be used to pre-fill forms and adapt the information presented to the user
	Form to report a case of school absence	Citizen Data Vault	
		Text Simplification + Workflow adaptation	Near the most difficult fields to fill in the form (based on the user experience and CDV information) an icon will be presented. By clicking on it a pop-up will be presented within a text explaining what it is asked to insert for that specific field. Depending on the user's background, fields that do not need to be filled by them will be omitted.
Parenting Skills	Parents wants to find information	Text Simplification	Complex sentences/phrases/words are pre-highlighted so that it's clear to the user that

Course	about support activities provided by the city council and external partners		other alternatives are available. Upon request, highlighted sentences/phrases/words are replaced with a simpler version.
	Professionals want to find information to assist parents.	Text Simplification	Complex sentences/phrases/words are pre-highlighted so that it's clear to the user that other alternatives are available. Upon request, highlighted sentences/phrases/words are replaced with a simpler version.
	Parents register for a parenting skills course through an online form	Text Simplification + Workflow adaptation	Near the most difficult fields to fill in the form (based on the user experience and CDV information) a icon will be presented. By clicking on it a pop-up will be presented within a text explaining what it is asked to insert for that specific field. Depending on the user's background, fields that do not need to be filled by them will be omitted.
		Citizen Data Vault	Information about the citizen and their family is stored and can be used to pre-fill forms and adapt the information presented to the user
	Parents look for information about previous courses and comments of previous attendees	Citizenpedia	An advanced web-based social question answering engine where citizens, companies and civil servants will discuss and suggest potential solutions and interpretation for the most problematic procedures and concepts A link to Citizenpedia is provided (it can point directly to the topic)

Like the web services described in the Trento pilot, the services for the Sheffield pilot are also provided by an external provider that is not directly involved in the SIMPATICO project. The web services are also under repagination, which means that the pilot will have to be set in their legacy version. As the Sheffield city council website is in the process of transformation so the decision has made to use plug-in based approach (Google Chrome Extension) to integrate Simpatico Solution to legacy system for Phase one. In order to speed up the process of deployment and avoid having to re-structure the legacy services, we resort to a Google Chrome extension.

The extension will represent the Interactive Front-End component for Phase one and will hence be interacting with the other components. The extension is developed entirely in JavaScript and can be

easily installed in any version of Google Chrome. The extension will interact with other components of SIMPATICO via REST APIs

Citizens and professionals will interact with e-services HTML page on Sheffield Website. The User will be asked to install the plug-in to get advantage of SIMPATICO solution. From the interaction, Google Chrome Extension will collect the data and propagate it to Citizen Data Vault and Interaction Data Log Component. At the moment Sheffield Pilot doesn't has any user identification or authentication process in place so our proposal is to either ask information about the user when s/he installs the plug-in into the browser and keep this information in a database or use cookie based approach to store the User information.

The plug-in will interact with following SIMPATICO components :

1. Text Simplification Engine
2. Workflow Adaptation Engine
3. Citizen Data Vault
4. Interaction Data Log
5. User Profile
6. Session Feedback

The extension works as a Text Adaptation layer on top of the content of a web page, which allows for the Simpatico technologies to be tested in any domain. The extension allows for the user to input their personal information, such as illustrated in Figure 9.

Figure 9: User profile interface of extension

With the user's profile available, the Simpatico technologies can more effectively customize the adaptations produced based on their needs. Once the user provides their personal information, they can make adaptation requests by selecting portions of text that they find challenging to understand.

There are two types of adaptations that the extension provides: enhancement and simplification. The enhancement capabilities of the Simpatico Chrome extension allow the user to retrieve complementary information about a word they are not familiar with. There are four types of information that the extension can retrieve:

- **Definitions**: Queried from the Merriam Dictionary (http://www.merriam-webster.com).
- **Similar words**: Queried from the Merriam Dictionary (http://www.merriam-webster.com).
- **Translations**: Queried from the Yandex API (https://www.yandex.com).
- **Descriptive images**: Queried from the Getty Images API (http://www.gettyimages.com).

The interface used by the Simpatico extension to showcase these resources is illustrated in Figure 10 and Figure 11. It is represented by a DOM object that becomes visible as soon as the adaptation is requested, and becomes invisible as soon as the "X" button in the upper right corner is clicked. Queries for enhancements are made through asynchronous AJAX requests.

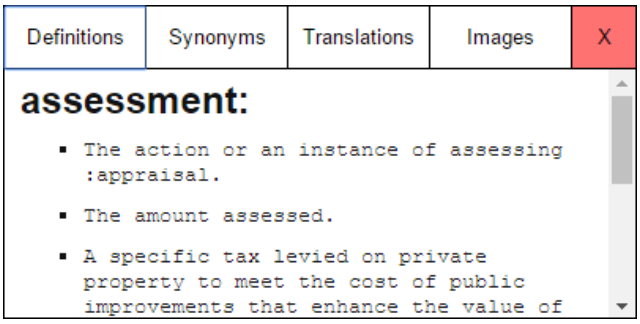


Figure 10: Simpatico extension interface showcasing definitions of "assessment"

Figure 11: Simpatico extension interface showcasing images for "assessment"

By selecting a complex word, phrase or sentence, the user can request a simplification. The request is stored in a JSON object and sent to a server where the REST API of the Simpatico simplifier is operating from. The simplifier receives the request, selects which type of simplification best suits the needs of the user, and returns a JSON object containing a simplified version of the content selected. The Simpatico extension interprets the JSON object and temporarily replaces the original content by its simplified version. Figure 12 shows an example of simplification produced by the Simpatico extension.

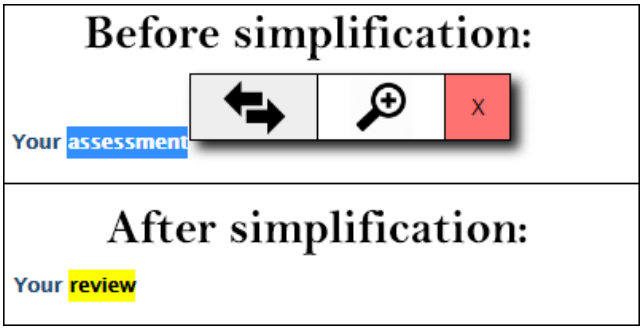


Figure 12: Simpatico extension replacing "assessment" with "review" upon request

4.3 Galicia Pilot

As documented in deliverable (D6.1 - Use-case Planning & Evaluation), the Galicia Pilot for the first iteration is based around two core e-services:

- BS607A: Wellness and spas program
- BS613B: Individual grants for personal autonomy and complimentary personal assistance for disabled people

These services are going to be replicated and analysed in a controlled environment in order to study all the components which conforms the SIMPATICO platform. In the Table 20 the services are analysed component-wise to gather more information about the required integration of components. It has to be noted that this is a specific vision at the writing of this document (M6 - August 2016) which is influenced by both the ongoing discussions about how should be an easy integration with the legacy Xunta back-end and the agreements about the interaction between all the SIMPATICO components.

Table 20: Service to SIMPATICO feature application mapping

E-Service	Scenario	Components	Description
BS607A Wellness and Spas program	Citizen wishes to do a request of stays in Spas/wellness centres within the Galician community.	Interactive Front-End	The IFE enhances the HTML view of the service which includes JavaScript functions which enable the interactive functions provided by the rest of SIMPATICO functionalities. This is the main and core component to be taken into account during the integration with the e-service. The citizen can select a part of a form and provided documents inside this e-service and ask for clarification to the user community. The civil servant can provide clarification and responses of the digital e-service to the user community. All this clarifications are stored in the Citizenpedia component.
		Citizenpedia	This component provides features related to the questions posted by citizens and comments inside diagrams created by civil servants. This component does not represent technical integration issues but all the content should be in Spanish.
		Interaction Data Log Session Feedback	These components will produce their work in a standard manner for

		Data Analysis User Profile Enrichment Engine Dashboard	this scenario (i.e., no particular changes are needed for the Galicia Pilot needs). All the interaction data produced by the dialog between citizens and the Galicia e-services is captured by IFE and stored in Interaction Data Log. SF produce forms which are shown by IF to gather user feedback after the execution of a session. The feedback metadata is also stored in Interaction Data Log. DA processes the captured information in Interaction Data Log and the user profile data stored in UP, which manages implicit information of users. As a result, an analysis is created. EE enhances the resulting analysis with statistics of the Citizenpedia and propagates the result to the Dashboard. Dashboard presents the enhanced analysis to the civil servant.
		Text Adaptation Engine	Change text from Spanish to Spanish enabling the following capabilities: Complex words and phrases are enlightened. When a user selects one enlighten phrase or word, a simplified version of it is shown according to the user profile. This requires the detection of hard words from inputs in Spanish and will be available in the first iteration of the SIMPATICO platform.
		Workflow Adaptation Engine	Selecting difficult fields in a form, a text explaining what it is asked to insert for that specific field is presented according to the citizen profile. Furthermore, fields and sections of forms are shown/hidden according to the user profile. In order to perform these tasks the process of this e-service should be modelled.

		Citizen Data Vault	The information filled in the module is stored in the CDV and available for future usage.
BS613B Individual grants for personal autonomy and complimentary personal assistance for disabled people	The citizen wishes to request a grant for personal autonomy, i.e. promotion services for disabled and elderly people to live as autonomous as possible.	Interactive Front-End	The IFE enhances the HTML view of the service which includes JavaScript functions which enable the interactive functions provided by the rest of SIMPATICO functionalities. This is the main and core component to be taken into account during the integration with the e-service. The citizen can select a part of a form and provided documents inside this e-service and ask for clarification to the user community. The civil servant can provide clarification and responses of the digital e-service to the user community. All this clarifications are stored in the Citizenpedia component.
		Citizenpedia	This component provides features related to the questions posted by citizens and comments inside diagrams created by civil servants. This component does not represent technical integration issues but all the content should be in Spanish.
		Interaction Data Log Session Feedback Data Analysis User Profile Enrichment Engine Dashboard	These components will produce their work in a standard manner for this scenario (i.e., no particular changes are needed for the Galicia Pilot needs). All the interaction data produced by the dialog between citizens and the Galicia e-services is captured by IFE and stored in Interaction Data Log. SF produce forms which are shown by IF to gather user feedback after the execution of a session. The feedback metadata is also stored in Interaction Data Log. DA processes the captured

			information in Interaction Data Log and the user profile data stored in UP, which manages implicit information of users. As a result, an analysis is created. EE enhances the resulting analysis with statistics of the Citizenpedia and propagates the result to the Dashboard. Dashboard presents the enhanced analysis to the civil servant.
		Text Adaptation Engine	Change text from Spanish to Spanish enabling the following capabilities: Complex words and phrases are enlightened. When a user selects one enlighten phrase or word, a simplified version of it is shown according to the user profile. This requires the detection of hard words from inputs in Spanish and will be available in the first iteration of the SIMPATICO platform.
		Workflow Adaptation Engine	Selecting difficult fields in a form, a text explaining what it is asked to insert for that specific field is presented according to the citizen profile. Furthermore, fields and sections of forms are shown/hidden according to the user profile. In order to perform these tasks the process of this e-service should be modelled.
		Citizen Data Vault	The information filled in the module is stored in the CDV and available for future usage.

In this preliminary analysis we see how the components of SIMPATICO are put to use in the Galicia Pilot for the first stage demonstrators. The details and open questions of the integration are mainly related to the particularities of this use case compared to Trento and Sheffield: in order to not disrupt the normal functioning of the current e-services in Galicia, they are to remain as-is while a distinct and separated SIMPATICO replica of the system is developed.

The development of these replicas of the current services is done from the presentation layer and up (HTML results produced by the Xunta backend). The presentation generated by the replicas will be integrated with the Interactive Front-End. The results of this integration will be a new environment

that has SIMPATICO capabilities and can connect to the rest of the components to provide these functionalities (i.e., the substitution of complex words) and test the majority of the SIMPATICO features. The most critical aspects of this integration refer to the need to replicate real services (i.e. to be almost identical to the real e-service). A significant amount of effort to obtain almost identical e-services will be invested.

5 A snapshot on the technologies state of the art (to be used)

In the following sections an overall snapshot of the state of art of the technologies that SIMPATICO components are going to adopt. The main goal of this snapshot is to better identify the main technologies that will influence the entire SIMPATICO Platform.

5.1 Interaction adaptation and personalization

The Adaptation Engine of SIMPATICO has the goal of adapting the content of webpages, documents, forms and questionnaires pertaining to city council services in order to make them more easily comprehensible to users. When it comes to Text Adaptation, the engine will mainly exploit three techniques: Lexical Simplification, Syntactic/Semantic Simplification and Text Enhancement. The state-of-the-art for each of these techniques are described in the Sections that follow.

5.1.1 Lexical Simplification

The goal of a Lexical Simplification (LS) system is to replace the complex words in a text with simpler alternatives, without compromising its meaning or grammaticality. The LS task is often addressed as the series of steps in Figure 1.

The steps in Figure 1 can be described as:

- **Complex Word Identification:** Task of deciding which words of a given sentence may not be understood by a given target audience and hence must be simplified.
- **Substitution Generation:** Task of finding words or expressions that share at least one meaning with the target complex word.
- **Substitution Selection:** Task of deciding which of the generated candidate substitutions can replace the complex word without compromising the sentences grammaticality or meaning.
- **Substitution Ranking:** Task of ranking the remaining candidate substitutions of a given complex word by their simplicity.

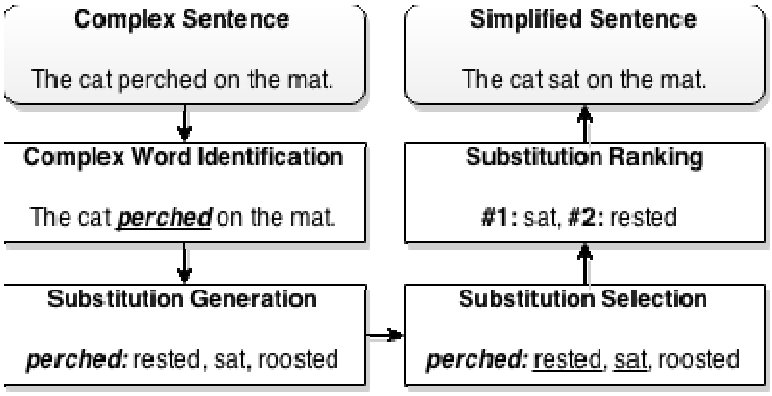


Figure 1 - Lexical Simplification pipeline

Researchers in the field have been addressing each of those tasks in various different ways. The most popular approaches from literature have been surveyed and compared in performance by (Paetzold & Specia, Benchmarking Lexical Simplification Systems, 2016). We take as state-of-the-art solutions those which have performed best in their benchmarks. In the Sections that follow, we describe the most effective solutions in literature for each individual task.

5.1.1.1 Complex Word Identification

The current state-of-the-art Complex Word Identification approach is the one introduced by (Paetzold & Specia, 2016). It combines several Complex Word Identification strategies by weighting their predictions according to their overall performance in a validation dataset. Among these strategies are Machine Learning techniques, as well as threshold-based and lexicon-based systems. This approach obtained the highest G-score (harmonic mean between Accuracy and Recall) in the Complex Word Identification task of SemEval 2016.

The LEXenstein framework (Paetzold & Specia, LEXenstein: A framework for lexical simplification, 2015), which will be employed in the development of all Lexical Simplification technologies used in the Simpatico project, gives access to dozens of Complex Word Identification methods, including the previously mentioned winner of the SemEval 2016 shared task.

5.1.1.2 Substitution Generation

The current state-of-the-art Substitution Generation approach is the one introduced by (Paetzold & Specia, 2016), and produces candidates using a context-aware word embeddings model. This strategy retrieves the 10 words for which the embeddings vector has the highest cosine similarity with that of the target complex word, except for its morphological variants. Candidate substitutions are retrieved with a model trained with the word2vec toolkit[1] over a corpus of more than 7 billion words, parsed with the Stanford Parser. Word vectors are trained using the Bag-of-Words (CBOW) architecture of (Mikolov, 2013) and 1300 vector dimensions.

This strategy, as well as six others, are included in the LEXenstein framework. Among them are generators that exploit online thesauri, dictionaries and parallel corpora.

5.1.1.3 Substitution Selection

The current state-of-the-art Substitution Selection approach is the one introduced by (Paetzold & Specia, 2016) also. It treats Substitution Selection as a ranking problem, and employs a technique called Unsupervised Boundary Ranking to address it. During training, it first generates 10 candidate substitutions for thousands of target complex words in context using the aforementioned Substitution Generation technique. It then exploits the assumption that words are irreplaceable, and creates training instances in unsupervised fashion by assigning label 1 to the target word itself and 0 to all the remaining candidates. After calculating feature values for each instance in the training set, it then learns a linear model through Stochastic Gradient Descent from it. During Substitution

[1] https://en.wikipedia.org/wiki/Word2vec

Selection, it ranks the candidates according to their distance from the boundary between positive and negative instances, then keeps the 50% of candidates which are the furthest in the positive side.

This strategy, as well as seven others, are included in the LEXenstein framework. Among them are selectors that exploit word sense disambiguators, grammar models, and word clustering techniques.

5.1.1.4 Substitution Ranking

The current state-of-the-art Substitution Ranking approach is the one introduced by (Paetzold & Specia, 2015). It uses a supervised Boundary Ranking approach. It learns a ranking model from data using a binary classification setup inferred from the ranking examples. 10 morphological, semantic and n-gram probability features selected through univariate feature selection are used. N-gram probabilities were extracted from a 5-gram language model trained over the SubIMDB corpus (Paetzold & Specia, 2016). This strategy is very flexible and suits the Simpatico project perfectly, since one can add user profile information to the feature set, and hence automatically learn how to customize simplifications based on a person's background.

This strategy, as well as five others, are included in the LEXenstein framework. Among them are selectors that exploit word frequencies, distributional semantics, and Machine Learning techniques.

5.1.2 Syntactic Simplification

Syntactic simplification approaches focus on modifying the structure of a sentence. It aims to simplify syntactic constructions that are considered complex to a given audience. For example, low literacy readers may have problems to read long sentences with several clauses or with several words before the main verb. This happens because low literacy readers can have an overload of the working memory while reading a long passage. Breaking long sentences or re-arranging its words through syntactic simplification can make the sentence more understandable.

The most popular approach for syntactic simplification research revolves around rule-based systems, which apply a mix of handcrafted and automatically inferred syntactic modifications to a sentence's structure (Siddharthan, 2004). The architecture of such systems is usually composed by three phases: analysis, transformation and regeneration.

In the analysis phase, a set of pre-processing tasks are performed. Firstly, the text is segmented into sentences, words are tagged with part-of-speech tags and syntactic annotations is also performed. Then, syntactic structures that can be simplified are identified and marked. In addition, a pronoun resolution step is also included in order to provide input for the regeneration phase. Finally, the analysis phase can also contain a complexity checker that identify whether or not a sentence should be simplified (Gasperin et al., 2009).

The transformation phase applies the simplification rules. The segmented text with the information about syntactic structures for simplification is the input for the phase that will apply the simplification rules (that can be handcrafted or learned from corpus). It is worth mentioning that the simplification rules applied during the transformation phase can be learned from corpora, although it is simpler and practical to use handcrafted rules. In order to learn simplification from corpora, large parallel corpora (original versus simplified) should be available, which is often not the case (Shardlow, 2014).

Finally, the regeneration phase is needed to guarantee that the simplified text preserves the meaning and cohesion of the original text. Therefore, a treatment of discourse connectives, sentence ordering, referring expressions, determiners and anaphoric links is performed.

Available tools

Although a full framework is available for lexical simplification, other simplification levels do not have a freely available tool that fully support the simplification process. Instead, there are some online tools that either only address parts of the simplification process (focusing mainly on the analysis phase of Siddharthan's framework), are not freely available or do not meet the requirements of SIMPATICO solutions.

For English

Tools that only focus on informing the users about the complexity of their documents are also related to the area of Readability Assessment. The most widely known readability tool in the research field is the Coh-Metrix (http://cohmetrix.com) tool that has over 100 readability indices that assess documents according to their linguistic discourse representations. Although this is a valuable work, the tool is limited to an online service with no API that could be integrated into SIMPATICO.

Besides Coh-Metrix, other proprietary online tools also provide readability assessment evaluation, such as WebpageFX (http://www.webpagefx.com/tools/read-able/), Rewordfy.com (https://rewordify.com/index.php) and the Readability Checker (http://www.thewriter.com/what-we-think/readability-checker/) of The Writer. Moreover, tools like the Readability calculator of Online-Utility.org (http://www.online-utility.org/english/readability_test_and_improve.jsp) and Readability-Score.com (https://readability-score.com/text/) besides inform the user with readability scores, they also provide information about long sentences, long words and tips for simplification. Although Readability-Score.com has an API, the access is paid and the readability scores alone do not solve the syntactic simplification problem. Therefore, we intend to implement our own readability metrics for SIMPATICO project.

iSimp (http://research.bioinformatics.udel.edu/isimp/plaintext.html) (Peng et al., 2012) is a tool developed by researches of the University of Delaware that inform text authors about complex syntactic structures. Even though iSimp does not perform simplification, it is aligned to the analysis stage of Siddharthan's architecture. Although iSimp can be accessed by external processes, it is limited for 10 sentences only.

Therefore, for SIMPATICO purposes we intend to implement our own analysis module, being independent from the availability of external tools.

For Italian

Few works related to simplification have focused on Italian, mainly dealing with readability assessment (Dell'Orletta et al., 2011; Lucisano and Piemontese, 1988; Tonelli et al., 2012) and corpus creation (Brunato et al., 2015). In particular, the work in Tonelli et al., 2012 was inspired by the Coh-Metrix readability indices developed for English, and had the goal to adapt some of these indices to Italian. Since the work was carried out by an FBK team partly involved in SIMPATICO, we plan to re-used some functionalities of this readability tool for text profiling.

As regards simplification, the only system developed so far for Italian is ERNESTA (Enhanced Readability through a Novel Event-based Simplification Tool) (Barlacchi and Tonelli, 2013), which

focused on syntactic simplification of children's stories. Since no large training data are available in this language, this system applies a rule-based approach to simplify a story into a set of simple statements, containing only relevant information to the story. We plan to re-used this tool and adapt it to sentence simplification in the administrative domain for the SIMPATICO, at least in the initial stage of the project. We further plan to experiment with data-driven lexical simplification with the Lexenstein tool (Paetzold and Specia, 2015), adapting it to Italian, and to exploit newly released corpora (Brunato et al., 2015) to test machine-learning based simplification involving also the syntactic level.

5.1.3 Semantic Simplification

Semantic simplification is rather less explored level of simplification where the aim is to simplify semantic structures such as compacting the lexicon used in text. Semantic simplification can be addressed as part of the lexical simplification step (by using simplifying the lexical choices), but it could also be considered as an independent level. Kandula et al. (2010) used a module of semantic simplification for building a tool for the simplification of health content. Such module elaborates the difficult terms (e.g. disease names) with easier terms related to the semantic type of the difficult terms (e.g. symptoms). To our knowledge, this is the only work that directly addressed semantic simplification. A related area of research is called Semantic Compression, where the lexicon is reduced whilst semantic is preserved (Ceglarek et al., 2010).

The task performed during the regeneration analysis of the syntactic simplification framework can also be viewed as a semantic simplification task, since the idea is to preserve the meaning of the original sentence whilst maintaining the semantics. In this scenario, discourse structures are analysed in order to guarantee that the simplification did not include any invalid semantic/discourse relation or introduce ambiguous semantic/discourse structures.

There are no tool freely available for this level of simplification apart from the research work mentioned in Shardlow (2014) and Siddharthan (2014).

5.1.4 Text Simplification using Statistical Models

Text simplification can also be viewed as a "translation" task where the original text is "translated" into the simplified text (Shardlow, 2014). Applications of this type use SMT frameworks (e.g. MOSES (Koehn et al., 2010)), in order to build the simplification systems and to evaluate their output. Parallel corpora of original and simplified text are needed to train SMT-based simplification systems.

Other statistical approaches have explored (quasi-)synchronous grammars for text simplification. Such approaches explore the tree-to-tree transformations between original and simplified texts (Siddharthan, 2014). Therefore, parallel corpora are also needed as for SMT approaches.

Narayan and Gardent (2014) is the state-of-the-art of statistical approaches. They used a corpus of parallel simplified and original Wikipedia articles (Zhu et al., 2010) and built the simplification systems by combining Discourse Representation Structure (Kamp, 1981) with a phrase-based MT system.

It is worth mentioning that approaches using statistical models can aim to all level of simplification.

There are no tool freely available for this level of simplification apart from the research work mentioned in Shardlow (2014) and Siddharthan (2014).

5.1.5 Text Elaboration

In Text Elaboration, the goal is to find and present descriptive information about a certain word or expression in a way that it allows for a reader to gain new insight on its meaning. There are not many examples of Text Elaboration systems available, which makes it difficult to outline the state-of-the-art for the task. Academic work such as the systems of (Devlin & Unthank, 2006) and (Azab & al., 2015) adorn the words of a text with definitions, descriptive images and synonyms in order to facilitate their comprehension. The experiments of (Rello & al., 2013) reveal that, while simplification tends to increase a document's readability, enhancement tends to improve its comprehensibility. Given the evident effectiveness of Text Elaboration, we will follow the aforementioned contributions when creating solutions for the Simpatico project.

Available tools

Tools like In Simple English of Online-Utility.org (http://www.online-utility.org/english/simple_basic_helper.jsp) and Rewordfy.com (https://rewordify.com/index.php) provide online text elaboration functionalities for English documents (highlighting complex words and providing synonyms and/or links to dictionaries). During the first stage of SIMPATICO project, we have already developed a pilot tool for text elaboration that goes beyond the simple presentation of synonyms and entries on dictionaries, giving the user other functionalities such as translation and images.

More details about all simplification levels can be found in Siddharthan (2014) and Shardlow (2014) surveys.

5.1.6 Citizen Data Vault

In the context of SIMPATICO project, the Citizen Data Vault is in charge of providing a secure repository of Personal Data of the users (citizen, company) used during their interactions with the personalized services provided by the PAs by means of the SIMPATICO Platform. In the definition and implementation of the Citizen Data Vault several technologies are going to be taken into account, mainly belonging to the state of art of Personal Data Store (PDS) technologies and solutions. According to World Economic Forum (June 2010) definition, Personal Data is defined as data (and metadata) created by and about people, encompassing:

- **Volunteered data** – created and explicitly shared by individuals, e.g., social network profiles.
- **Observed data** – captured by recording the actions of individuals, e.g., location data when using cell phones.
- **Inferred data** – data about individuals based on analysis of volunteered or observed information, e.g., credit scores.

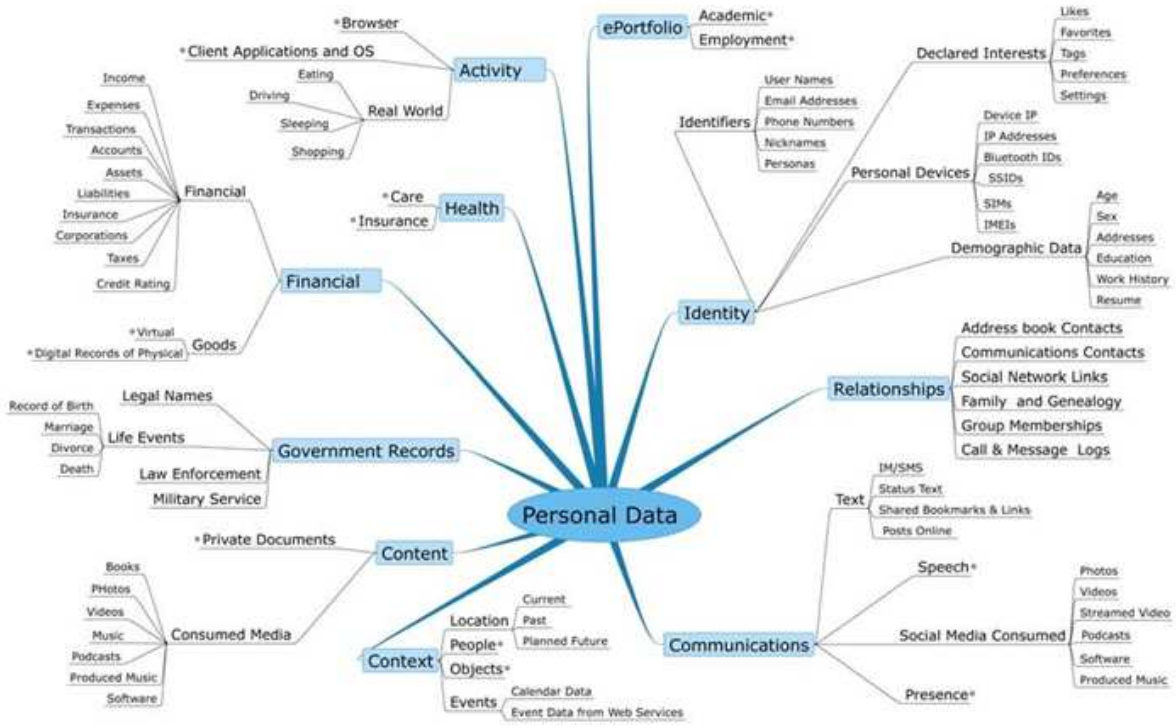


Figure 13: An excerpt of Personal data types in this diagram are drawn from the types identified in Rethinking Personal Data Pre-Read Document published by the World Economic Forum written by Marc Davis et al published in June, 2010

According to the European Commission report (Report on Personal Data Stores)a Personal Data Store (PDS) is a technology that enables individuals to gather, store, update, correct, analyse, and/or share personal data. Of particular importance is the ability to grant and withdraw consent to third parties for access to data about oneself In other definitions the term "store" is changed with the terms "Locker" or "vault".

According to this definition it is possible to identify two main parties (Figure 14) , that with the PDS itself compose the PDS ecosystem:

Data Source: It is the party that holds and provides data about each individuals. In PA sector, several governments are working with the private sector to give individuals access to a copy of data about them in a usable format which can then be stored in their personal data store and shared according to their explicit consent

Data Requester: Company and/or service that want to access data about individuals by means of the PDS

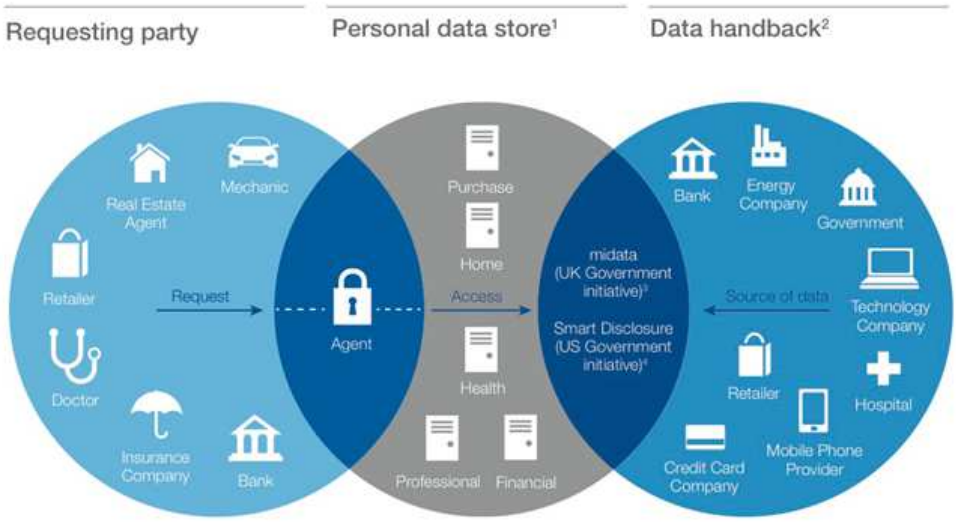


Figure 14: PDS Ecosystem (from World Economic Forum and The Boston Consulting Group, Unlocking the Value of Personal Data: From Collection to Usage, 2013)

The above report argues the importance of interoperability, envisaged as an essential feature of fully functioning PDS Ecosystem. This aspect is important in the context of SIMPATICO Platform and the related interactions among its components. Interoperability comprises aspects on data formats and data exchange protocols to facilitate and interoperable system, semantic interoperability for an easier information processing and data portability to enabling individuals to exercise their right to data portability.

In order to meet interoperability requirements Open Standards should be evaluated and selected.

As concern semantic interoperability several semantic options and tools are going to be considered: RDF, Linked Data[2], and the concept of "link contract" in the open standard XDI semantic data interchange protocol developed by the OASIS XDI Technical Committee[3]. XDI provides a standardized portable authorization format called XDI link contracts, subgraphs that express the permissions that one XDI authority (person, organization, or thing) grants to another for access to and usage of an XDI data graph. XDI link contracts enable these permissions to be expressed in a standard machine-readable format understood by any XDI endpoint.

Architectural implementations of the PDS could follow two different approaches of storing personal data:

- Cloud based: personal data resides in the cloud
- Local storage: personal data resides locally o distributed in several local devices

In the case of Local Storage the user interfaces with the API access layers through their preferred mobile or desktop device. Various protocols could be implemented for handling of data transmission, login credentials, encryption and decryption key handling, and various levels of service request

[2] https://www.w3.org/standards/semanticweb/data
[3] http://www.oasis-open.org/committees/xdi

through the PDS provider. The cloud-based infrastructure layer which consists of Software, Platform and Infrastructure as a service(SaaS,PaaS and IaaS) allows for a common access point from either side of the ecosystem (for developers and third-party services on one side, and for the user on the other side). These two approaches reveal different security issues that entail different approaches in order to balance risks and cost of implementation.

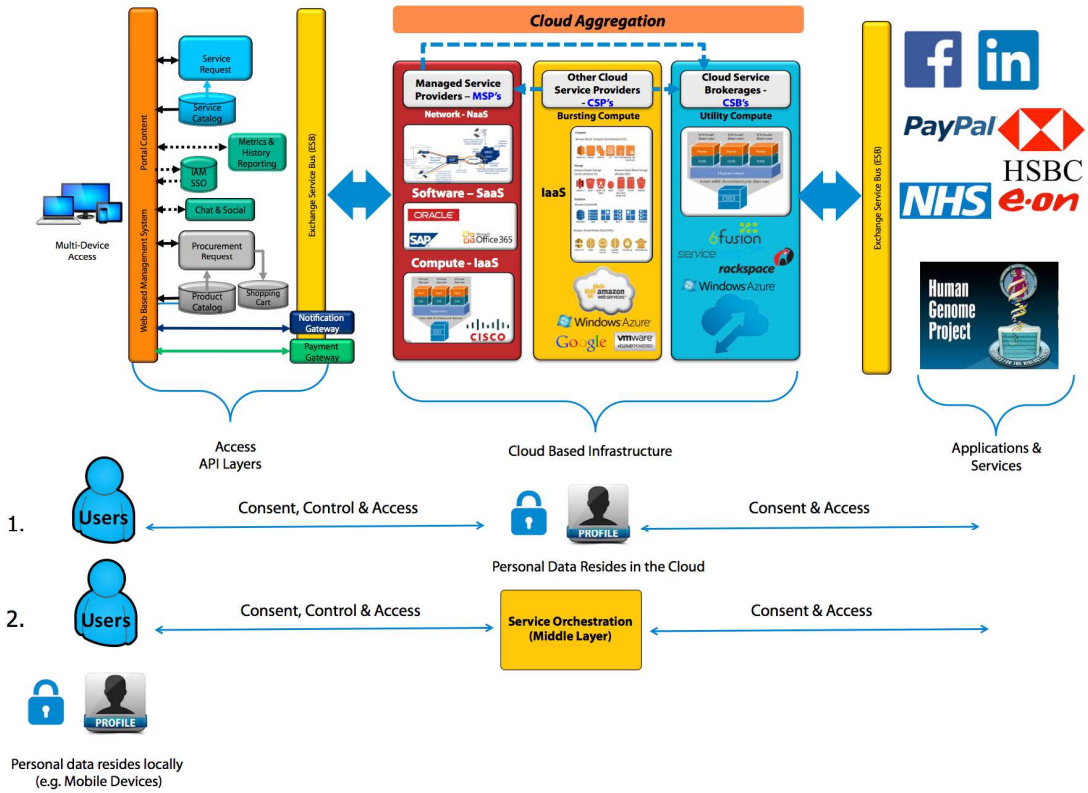


Figure 15: Two possible architectural implementations of the PDS concept, (1) cloud based, and (2) local storage. These two approaches reveal different security issues (from https://ec.europa.eu/digital-single-market/en/news/study-personal-data-stores-conducted-cambridge-university-judge-business-school)

From the above reference architecture it is possible to provide different interpretation of the concept of "distributed PDS":

- "Distributed" accordingly to Data sources. Personal Data resides in several location and provided by multiple services (for example from PAs services).
- "Distributed" accordingly to Personal Repository/Data Storage. Local Storage comprise multiple sources (mobile device, pc, net storages).
- "Distributed" accordingly to a distributed approach of authorization policies.

As concern a general list of PDS features, starting from different definitions and requirements of PDS, the range varies:

- *Data Storage*– an access point for personal information

- *Data Management*– a toolset for analyzing and understanding what data means
- *Data Sharing* – the ability to choose how to share personal information and with whom
- *Data Collection* – the ability to track and collect data
- *Verifications* – the ability to authenticate sensitive information generated by 3rd parties
- *Identity Assurance*– the ability to prove I am who I say I am
- *Privacy Management* – my info has a privacy setting determined by me, not organizations
- *Manage Permissions*– deciding the communication channels between me & my contacts
- *Express Interests & Intentions*– the ability to announce what I want to buy, do or access
- *Plan & Implement Projects* – a life management system for how I will use my info over time

In SIMPATICO context the Citizen Data Vault will not cover all the above features, but a selection according to the main scenario and according to the requirements belonging from the three pilot use cases.

Starting from the above mentioned characteristics of a Personal Data Store and related technologies, in the following a list of potential solutions are provided. These solution could influence the SIMPATICO Citizen Data Vault, or to be a means of comparison:

Cloud based

- Mydex[4] – A Personal Data Store for individuals that helps them collect, store, manage, use and share their own personal data for their own purposes.
- Personal.com [5]– protect online personal data exhaust and sell it back to advertisers

Open Source Project

- Project Higgins[6] – open source identity framework designed to integrate identity, profile and social relationship information across multiple sites, applications and devices using an extensible set of components.
- Project Danube[7] – open source project to develop an XDI-based Personal Data Store – a semantic database for personal data which is controlled by the user. Applications on top of this database include the Federated Social Web and the selective sharing of personal data with organizations.
- OpenPDS[8] - a personal metadata management framework that allows individuals to collect, store, and give fine-grained access to their metadata to third parties

[4] http://mydex.org/
[5] http://www.personal.com/
[6] http://eclipse.org/higgins/
[7] http://projectdanube.org/
[8] http://openpds.media.mit.edu/

- Enigma[9] - a decentralized cloud platform with guaranteed privacy. Private data is stored, shared and analyzed without ever being fully revealed to any party. Secure multi-party computation, empowered by the blockchain technologies.
- MyData[10] -is an effort by the Open Knowledge Finland community to define a human-centric way to manage and process personal information. The core is that individuals should be in control of their own personal data.

Particular attention will be paid to MyData Solution, especially on consent management and service registration, and to Personal.com for the approach used on metadata management and web-form auto filling.

5.2 Front-end interaction and enrichment

This state of the art has been already discussed in the (D3.1 - User Interactions Modelling and Design) and reported here for sake of clarity.

As a general strategy, in SIMPATICO we will use *DOM manipulation for Front-End interaction deployment and its enrichment using augmented browsing techniques*. This enables us to leverage on the existing legacy front-ends for the use cases in the pilot sites at the moment.

5.2.1 DOM manipulation techniques and frameworks

According to the World Wide Web Consortium (W3C)[11], the Document Object Model (DOM) is a platform- and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure and style of documents. The document can be further processed and the results of that processing can be incorporated back into the presented page. It is a mechanism for representing and interacting with HTML, XHTML or XML documents. It allows the navigation and the manipulation of documents through a programming language, which in the browser will almost always be JavaScript, which is a high-level, dynamic, untyped, and interpreted programming language.

Using DOM manipulation techniques, *augmented browsing* allows end-users to augment and personalise how they view web documents. Augmented browsing is an emerging technology that describes the experience of using a system that can automatically augment, improve or clear up the information in web pages. Using augmented browsing, site-specific browsers can access pages from a single source (site) on a computer network to typically simplify the more complex functions of a web browser by excluding the menus and toolbars.

The most popular frameworks of augmented browsing are:

AlchemyPoint[12] turns the web-browser into a web-processor, seamlessly. Web pages can finally be edited without any programming or any knowledge of HTML.

[9] http://enigma.media.mit.edu/
[10] https://github.com/HIIT/mydata-stack
[11] https://www.w3.org/DOM/
[12] http://www.alchemyapi.com/

BFilter[13] is a filtering web proxy. It was originally intended for removing banner ads only, but since then its capabilities have been greatly extended. It has a proxy that would detect ads heuristically, much like modern anti-virus software manages to detect many viruses unknown to it.

Grammarly[14] is a writing-enhancement platform developed by Grammarly, Inc. and launched in 2009. It is a proofreading and plagiarism-detection resources check for a writer's adherence to more than 250 grammar rules. It can analyse general, business, academic, technical, creative and casual writings.

Greasemonkey[15] is a well-known augmented browsing tool composed of a Mozilla Firefox extension that allows users to install scripts that make on-the-fly changes to web page content after or before the page is loaded in the browser. It is commonly used for customizing page appearance, adding new functions to web pages, fixing rendering bugs, combining data from multiple web pages, among others.

GrimeApe[16] is an HTTP proxy which enriches every web page you visit using Greasemonkey Userscripts. It enables all the existing browsers to use Greasemonkey tool. Current GrimeApe has been seen working in Konqueror, Chrome and Firefox. It fails in IE6 and it has not been tested in Opera, Safari, IE7, or IE8.

MouseHole[17] is a personal web proxy designed to be simple to script. Scripts can directly manipulate web pages altering their content and behaviour.

Muffin[18] is a World Wide Web filtering system written entirely in Java that can filter any HTTP data sent and received by the browser. It has a graphical user interface and command-line interface. It also includes several filters which can remove cookies, kill GIF animations, remove advertisements, modify arbitrary HTML tags, and remove Java applets and JavaScript.

Privoxy[19] is a web proxy with advanced filtering capabilities for enhancing privacy, manipulate web page data and HTTP headers, controlling access and removing ads. It has a flexible configuration and can be customized to suit individual needs and tastes. It has application for both stand-alone systems and multi-user networks.

Tampermonkey[20] is a browser extension and the most popular user script manager. Even though some of the supported browsers have native user script support, it will give developers much more convenience in managing their user scripts. It provides easy script installation, automatic update checks and a built-in editor.

Table 21 summarizes the main features for these tools. We must highlight that they are implemented either as browser extensions or as a proxy. If we were to choose one for SIMPATICO, most solutions are browser extensions due to their ease of development. However, they are completely dependent on the browser they are designed for, which could create maintainability and compatibility issues,

[13] http://bfilter.sourceforge.net/
[14] http://www.alexa.com/siteinfo/grammarly.com
[15] http://www.greasespot.net/
[16] http://hwi.ath.cx/twiki/bin/view/Neuralyte/WebIndex
[17] https://github.com/evaryont/mousehole
[18] http://muffin.doit.org/
[19] https://www.privoxy.org/
[20] http://tampermonkey.net/

especially with mobile browsers. On the other hand, proxies do not depend on browser developers but presents privacy related issues.

Thus, we will implement our front-end as a JavaScript library. JavaScript libraries work in almost every browser (both desktop and mobile) and provide us freedom to modify the DOM of almost any page, controlling any potential privacy issue. The only drawback is that the library must be inserted within the source code of the page to be modified.

Table 21: SoA Analysis for DOM manipulation

Framework	Open Source	Browser Extension	Main relevant features
AlchemyPoint	No	Firefox	Direct manipulation of web content and scraping of data from websites
BFilter	Yes	No	Direct manipulation of web content
Grammarly	No	Firefox and Chrome	No
Greasemonkey	Yes	Firefox	Direct manipulation of web content
Grimeape	Yes	Konqueror, Chrome and Firefox	Direct manipulation of web content
MouseHole	Yes	Firefox	Direct manipulation of web content
Muffin	Yes	No	Direct manipulation of web content
Privoxy	Yes	No	Direct manipulation of web content
Tampermonkey	No	Chrome, Opera, Safari and Firefox	Direct manipulation of web content

Since the majority of the described tools are open source, they will be taken as a basis to develop a new SIMPATICO ad-hoc tool to satisfy the project needs (i.e. mobile devices support) and the augmented browsing of the public administration e-services. A thorough discussion on the approach that will be adopted within the SIMPATICO project can be found in the Section 2.3 of deliverable (D3.1 - User Interactions Modelling and Design).

5.2.2 Other alternatives

In addition to DOM manipulation as discussed in section 4.2.1, we have also studied the possibility of using Model-Based UI Design (MBUI). This is a technique that allows user interfaces to be designed in successive modelling levels of concreteness, from a very high level Abstract UI in which only the high level intents of the interface are described to a very specified Concrete UI in which all of the implementation details have been solved. The overall methodology is to provide transformation models upon the different levels so that from the very same Abstract UI different Concrete UIs could be constructed that comply with changing contexts of use -- say a different device or, critically for SIMPATICO, a user with a different expertise level.

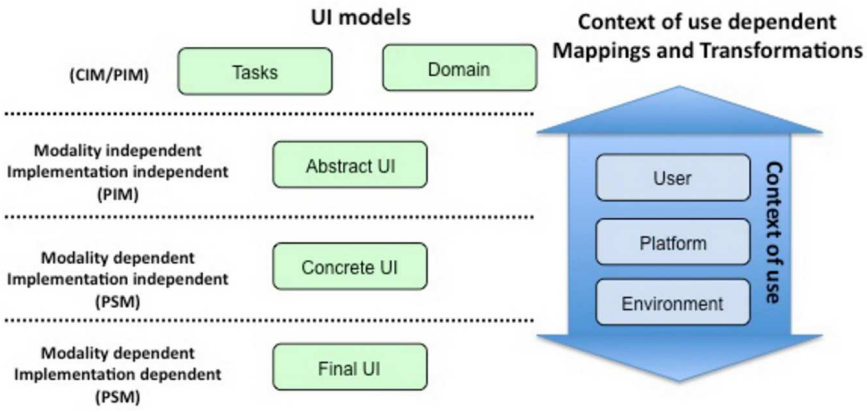


Figure 16: Model Based UI Design levels

Well-known implementations of MBUI stacks exists. Among this is MARIAXML offered by ISTI-CNR which had the participation of the W3C in its design and is based on the results of the MBUI Incubator group organized by the W3C themselves.

This approach was considered for SIMPATICO because this offers a number of options to implement the front-end. The system could be envisaged as using different 'modalities' as different presentations for the profiles of the users: citizens with full understanding of the procedures, citizens with limited knowledge that require certain simplifications, etc. These could be chosen by the context engine that would connect to SIMPATICOs Citizen Data Vault to extract this data, and transformations could be applied to the abstract description of the tasks in the procedures so that the final UIs would be achieved.

However, it was chosen not to follow this path because it would require that the legacy systems offered by the Use Cases in the project be essentially discarded and reworked from scratch, which would be very costly and greatly exceed the capabilities of the project.

5.3 Human computation

Human computation or Human-based computation (HBC) is a paradigm in where humans perform small tasks and a computer system is in charge of orchestrating their results. These tasks would require complex computer algorithms to be accomplished, but due to their nature they are easily completed by humans, e.g. to recognize an image or some manipulated and struggled text (Quinn, 2011).

The trend of using human computation as a resource in science and engineering began around 2005 with the thesis of Luis von Ahn, where he defined several contributions that have been widely used in popular internet sites (von Ahn, 2005). The most popular ones where the ESP Game, an application where users earned points tagging images, and the CAPTCHAs, distorted and difficult-to-read texts that a user must decode to access to certain web/resource. The use of these techniques enabled the image and text processing of thousands of elements that would have required years of computing power.

In some scenarios, human computation techniques might not get as many responses as the problem would require. In those cases, it is common to include in the platform gamification techniques: the paradigm of including game-like elements (e.g. badges, rankings…) in non-game contexts to increase

user engagement. This concept was described in the dissertation of Luis von Ahn, describing them as "games with a purpose". Since then, it has been used in many environments, such as collaborative web pages.

This section compiles some of the most relevant contributions in the field human computation we have found, that will serve as reference in the development of the Citizenpedia. We have divided the section in two groups: first, we describe already existing platforms, such as online web platforms or mobile applications. Then, we enumerate and summarize some contributions found in the scientific literature, i.e. papers from scientific journals and conferences.

5.3.1 Deployed applications

From 5-10 years to now, and due to technological advance of web and mobile systems, we have witnessed the creation of many human computation-based platforms. The topic and field of these platforms vary immensely. In this section, we have selected the most popular ones that could serve as reference for Citizenpedia.

The most popular human computation platform is Amazon Mechanical Turk, whose lemma is "Artificial Artificial intelligence". It is a marketplace where people post jobs and offer money for getting them done. These jobs can range from small tasks that could take seconds to a human (e.g. filling a survey form or commenting on a Tweet) to more time-costly ones (e.g. transcribing the audio of a 2 hour-long file). The amount of money a "turker" (the person who actually does the job) varies depending on the complexity of the task. A screenshot of some of the offered jobs in Mechanical Turk is shown in Figure 17.

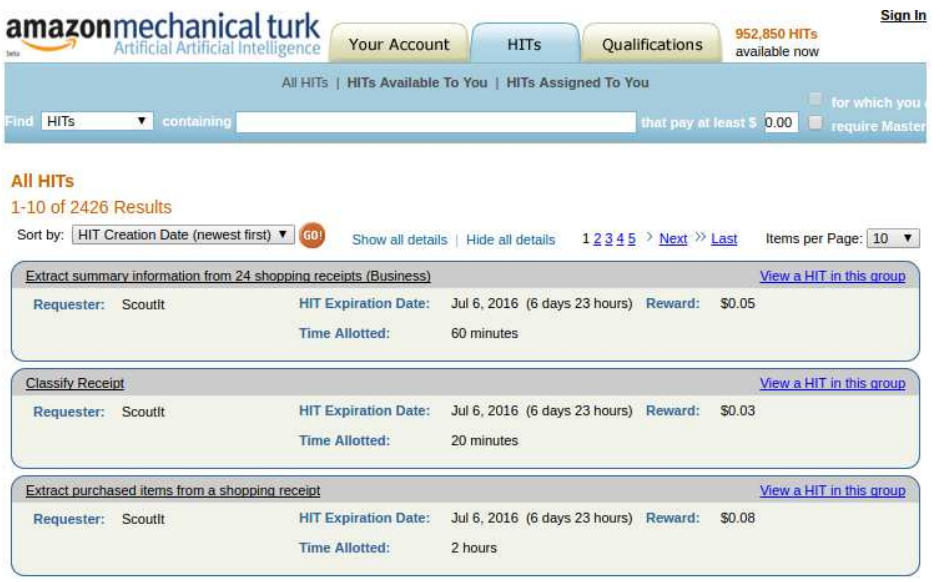


Figure 17: Screenshot of Amazon Mechanical Turk

On the mobile applications side, we have chosen Waze (Figure 18): a GPS path-guiding application for smartphones and tablets. It calculates the path to be followed by a user, but also alerts about traffic jams, accidents, etc… Other users enter these alerts in the system. In contrast to Mechanical Turk,

the reward here is a better driving experience thanks to the availability of more information on what is happening on the road. Being a free-to-download application, it gained widespread attention by the driver community and by the technological media. In 2013 it was acquired by Google.

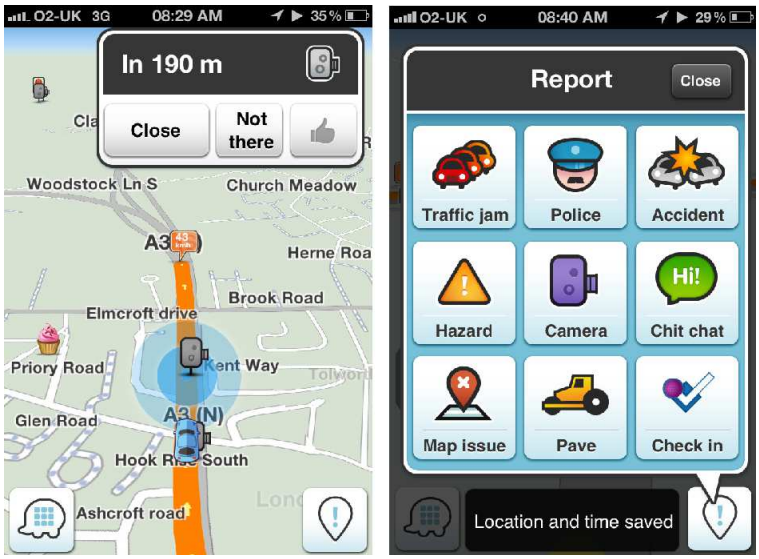


Figure 18: Screenshot of Waze

Mechanical Turk and Waze gained a large user base over the time, but in some scenarios where the owners of a platform want to increase the engagement of the people, they include gamification techniques. Examples of such are MalariaSpot and GalaxyZoo, online platforms that assign small processing tasks to humans giving reputation as reward. MalariaSpot makes its users detect Malaria parasites in images of blood smears. GalaxyZoo makes its players to classify galaxy morphologies. Both are still running and claim to have engaged more than 10.000 thousand player in their years of existence.

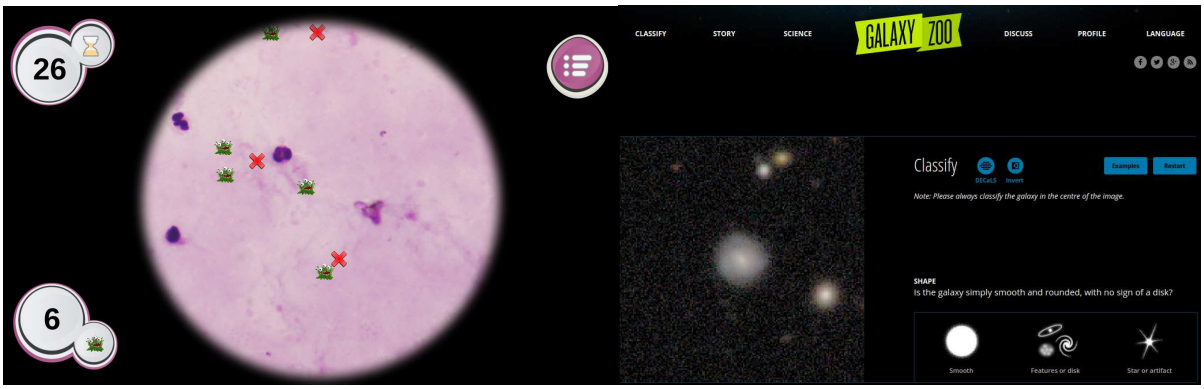


Figure 19: Screenshots of MalariaSpot and GalaxyZoo

All these platforms tackle a variety of areas, but the relevant for the SIMPATICO project is the field of public administration and e-government. In this area, the most relevant name is mySociety, an e-democracy project that offers many online democracy tools for Uk citizens. These tools are released

as open source projects, and many councils have adopted them to ease the burden between citizens and administrations. We want to highlight two that have been widely adopted:

- FixMyStreet (in Figure 20) works as a map of a city where citizens can comment on roads or paths that need mending. Each comment is transferred directly to the council, as an alert for the issue to be solved.
- WhatDoTheyKnow (in Figure 21) a public question & answer portal, where citizens post questions and requests to the public administrations. These requests are transferred to the councils, as with FixMyStreet, and civil servants are urged to answer. Both questions and answers are publicly shown, as the main claim of the platform is to push the Freedom of information.

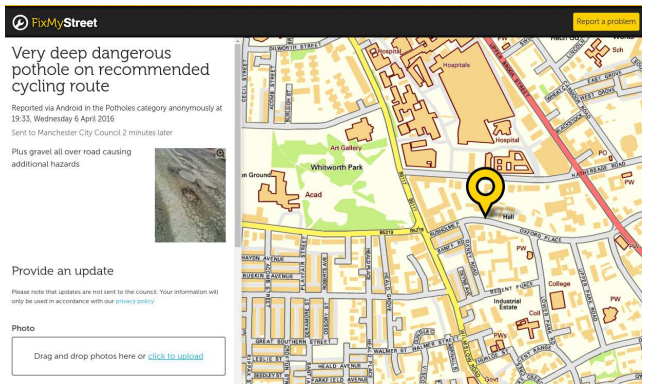


Figure 20: Screenshot of FixMyStreet in Manchester

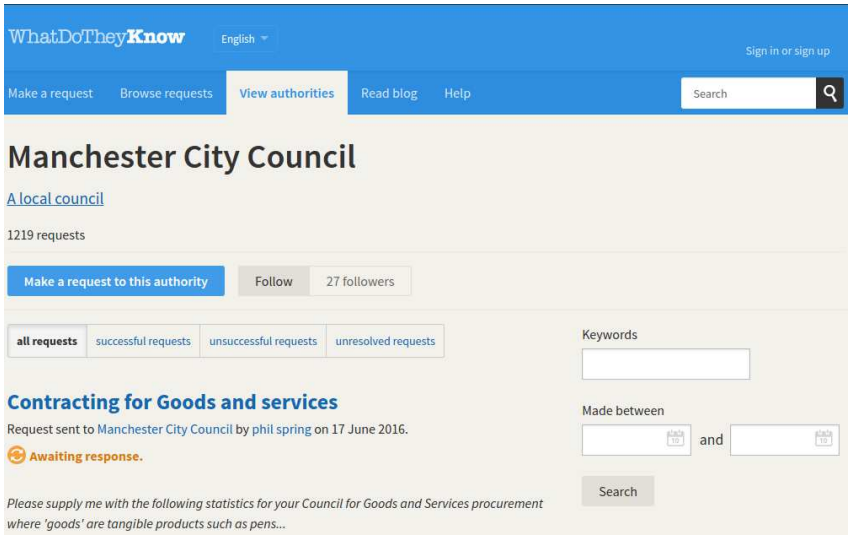


Figure 21: Screenshot of WhatDoTheyKnow for Manchester City Council

Another tool tackling e-government and e-democracy issues is LiquidFeedback, an open source web platform for proposition development and decision making. It looks similar to WhatDoTheyKnow, i.e. a question & answer platform, but the aim is slightly different: citizens make proposals to a certain political party, or raise discussions on a certain topic, and the main goal is to reach an agreement. It

promotes the Freedom of Information, as WhatDoTheyKnow does. Some examples on its use can be found in some videos, both in English[21] and Italian[22].

Figure 22: Screenshot of LiquidFeedback for Sicilia, Italy

5.3.2 Scientific contributions

The second part of the section will list and summarize some of the contributions and proposals found in the scientific literature. In contrast to the applications that have been just presented, the contributions in this section are less mature in terms of development, but present more recent ideas usually developed in universities or research centres.

As first, in (Charalabidis, 2012) authors propose a methodology that would enable public administrations to get more out of the comments that citizens post in social networks for policy making. This methodology is composed by four phases in a cycle (Listen, Analyze, Act and Receive), and for each phase the technologies that should be used are described. E.g. authors suggest using crawlers for the "Listen" phase in order to capture information and Natural Language Processing tools for the "Analyze" phase. The paper provides deep description on the methodology, but no results that could evaluate the wellness of the contribution.

The work on the previous paper has been extended in (Charalabidis, 2014) were authors present the next step over the methodology: the functional architecture of an ICT platform supporting the methodology, with some details on its technological constraints. They describe how their design was driven by several three use cases in Austria, Greece and Uk, and the steps they followed.

[21] https://www.youtube.com/watch?v=I0JuE_c-r5E
[22] https://www.youtube.com/watch?v=FVz1MXxq_QY

In (Asquer, 2014) the author presents a survey paper about gamification applied to public services. First, an extensive essay on the topic is presented, accompanied by a large list of references. Then, the author provides a discussion on how to apply gamification into public services. This discussion traverses some fields related to public services, such as education, and how the gamification would impact in them. This contribution is a good starting point to dive into the scientific literature on the topic, but staying on the theoretical concepts.

In contrast, the paper presented in (Bista, 2014) describes a practical experience on the topic: authors describe their work in introducing gamification in a online service of the Department of Human Services of the Australian Government. They provide a deep description of every phase they conducted: their motivation, the design of an ad-hoc gamification model, its implementation (including technical detail, such as its deployment), and an analysis of the usage that the online service had. As a whole is a good paper that provides extensive descriptions on their experience, and several guidelines for the ones that follow their steps.

Finally, the work in (Dargan, 2015) describes a platform that tries to promote the use of Business Process Management (BPM) practices in the public administration to improve the quality of e-services. The platform is created using the concepts of user-centered design and gamification design methodologies. Authors take the scenario of the administration of Switzerland, in particular in the Swiss Process Sharing platform, but do not put it into implementation. They use it to analyse the requirements and potential pitfalls.

5.3.3 Gamification Engine

The aim of this state of the art is to complete that one already provided in the (D4.1 - Citizenpedia Framework Specification and Architecture).

In the context of SIMPATICO project, the Gamification Engine is the component responsible of the implementation and execution of the logics associated to the rewarding mechanisms adopted to ensure user engagement. The usage of the Gamification Engine is foreseen in particular in the scope of Citizenpedia, to reward the active participation of civil servants, professionals and citizens in the collaborative development of Citizenpedia knowledge base. Further usages of the Gamification engine are however possible also in other cases, for instance to incentivize e-service users to provide feedback on the interaction.

By “gamification” we mean the use of game design elements (players, scores, challenges, prices, and so on) in non-game contexts(S. Deterding, 2011). Gamification has demonstrated potential in a variety of domains(J. Hamari, 2014), and there is a growing body of work on applying gamification as one means to persuade citizens to embrace socially beneficial collective action. One area of application is environmental sustainability, with multiple examples on energy savings(B. Cowley, 2011) (B. Orland, 2014), and sustainable mobility(S. Gabrielli, 2014)(P. Holleis, 2012)(R. Kazhamiakin A. M., 2015)(R. Kazhamiakin A. M., 2016), as well as on other salient Smart City concerns, such as tourism(A. Gordillo, 2013), or participatory governance of urban neighbor-hoods(T. Coenen, 2013).

More relevant for SIMPATICO, gamification is a also a well-known approach for effectively achieve human computation, as illustrated by the following examples:

- MalariaSpot(C. Luengo-Oroz, 2012) enabled anonymous players to detect Malaria parasites in blood based on images of blood smears. According to their results, combining 22 games results in a 99% of accuracy, and they evaluated it with 270,000 clicks on 12,000 games from 95 countries;
- GalaxyZoo(C.J. Lintott, 2008) obtained 40 million individual classifications of galaxy morphologies by 100,000 participants;
- FoldIt(F. Khatib, 2011) used gamification to involve players in collaborating and competing to create accurate protein structure models.

Some basic gamification concepts are also already adopted by query answering systems (and, in general, by systems exploiting User Generated Content): these include user status, badges and levels reached by participating actively in the community. In the case of SIMPATICO, the objective is to make it possible the usage of more general and powerful game concepts, as we will explain in the following.

For what concerns the state of the art concerning the technologies for defining and executing gamification scenarios (i.e., gamification engines, adopting the SIMPATICO terminology), we have to report that very few articles describe specific technologies and infrastructures – in the literature, gamification is usually described from a methodological point of view, without giving much emphasis to the underlying software tools. An exception is the work of Herzig et al.(P. Herzig M. A., 2012)(P. Herzig B. W., 2013), where technical aspects are discussed, and a combination of Complex Event Processing and Rule Based Systems is proposed as a good combination for the development of a Gamification Engine.

In SIMPATICO, our intention is to implement a wide variety of game concepts in combination with Citizenpedia, in order to evaluate which ones are more effective in achieving an active and durable engagement of the community. In addition to the classical game concepts already adopted by systems exploiting User Generated Content (mostly, points, badges, levels), we plan to experiment with “personalized challenges” as an instrument to incentivize users to provide specific contributions, depending both on the user profile and on the importance of the requested contribution; to understand whether the concept of “team” can better foster collaboration; to evaluate the possibility to launch “marathons” to achieve fast reactions, in particular when important evolutions in the context of Citizenpedia become relevant. Also, we want to leave open the possibility of adopting the Gamification Engine also outside Citizenpedia, to promote active participation in other areas of the platform. Finally, to effectively evaluate the effectiveness of gamification as a whole, and of the different adopted game concepts, we need flexible analytics techniques.

The requirements for the Gamification Engine are hence:

- easy integration with Citizenpedia and with other components of the SIMPATICO system;
- variety of the exploitable game concepts and flexibility in the definition of games;
- facility of analysis of the outcomes of games and of the adoption of specific gamification concepts.

By taking into account these requirements, the SIMPATICO project has selected the gamification framework developed by one of the project partners - Fondazione Bruno Kessler: this solution already offers a good level of fulfilment of the requirements; in addition, it is open source and the competences to use and extend it are available to the SIMPATICO project. The consortium will however perform regular reviews of the advances in the gamification literature, in order to ensure that the developments and experiments performed by the project are aligned with the state of the art in this area.

The architecture of the Gamification Engine of depicted in the following figure:

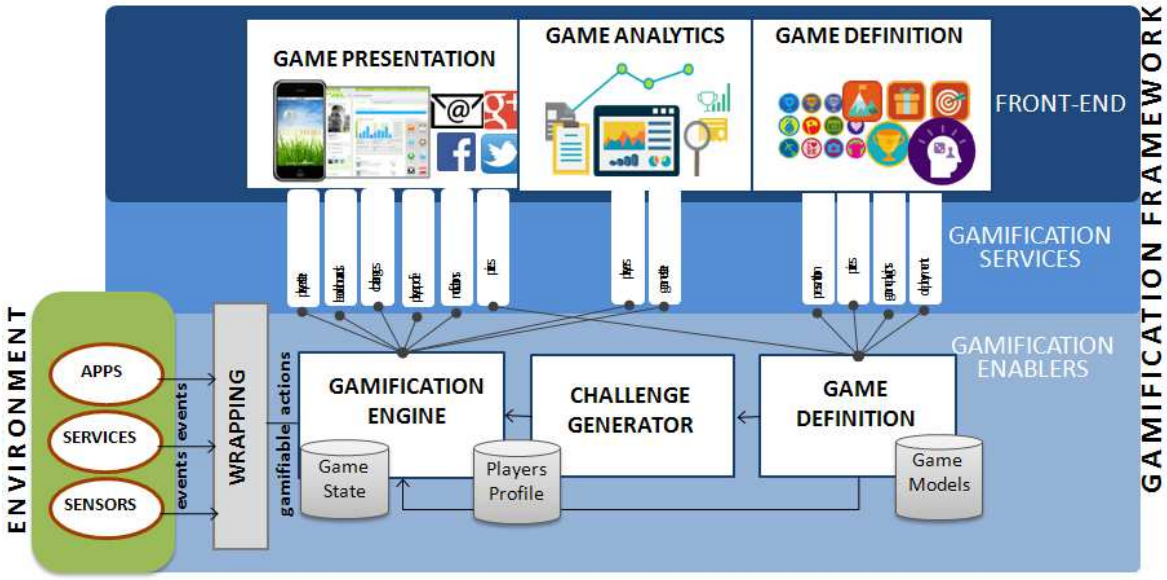


Figure 23: Gamification Engine Architecture

The architecture of the is organized in three main layers: Gamification Enablers, Gamification Services and Gamification Front-end. Gamification Enablers support for the basic functionality related to the design, deployment (the Game Definition component), and execution (the Gamification Engine component) of games, and its integration with external IT systems; it also supports advanced functionalities for the automatic generation of challenges (the Challenge Generator component), which are personalized playable units that are synthesized taking into account a player's profile, game state and game history (i.e. game results and actions, respectively). Gamification Services expose the functionality implemented by the enablers as services, which can be easily used to build new gamification components and applications that exploit the core components of the framework (e.g., services supporting the definition and deployment of games, services for accessing information about the game and player state, services supporting the configuration of notifications for communicating game results to the players).

The Gamification Front-end layer provides end-user applications to the three principal stakeholders defined above. In particular, it provides applications supporting the definition and deployment of games (for game experts), the presentation of game state (for players), and the analysis of game results (again for game experts, but also for community animators).

The framework has been designed to be open, general and extensible, with respect both to the IT affordances that it can integrate (i.e., applications, sensors, services, etc. that contribute actions of the users that advance the game), and the types of game mechanics and games that it supports. In order to achieve those objectives, from the technical point of view the gamification framework in accord with the architectural principles of service orientation. Adherence to service orientation offers a number of major advantages, including: i) the ability to integrate with open, heterogeneous and multi–ownership systems (including legacy) that are operating in a complex environment like that on e-service delivery – the requirement being that these systems expose logs on the actions performed by the users in terms of standard services (or, at least, in a way that makes possible to wrap them as services); ii) the standardization of interactions that occur between our gamification

framework and players game presentation, independently of the medium and device delivering that presentation; iii) the extensibility of the set of game concepts that can be used in specific gamification applications.

For what concerns the usage of this gamification framework in SIMPATICO, the intention is to re-use the Gamification Engine and the Game Definition enablers, as well as the Game Definition and (partially) the Game Presentation front-end components. The other components (and some of the Game Presentation functionalities) are expected to be evolved to achieve a better integration con Citizenpedia e with the SIMPATICO platform.

5.3.4 Collaborative Procedure Designer

The aim of this state of the art is to complete that one already provided in the (D4.1 - Citizenpedia Framework Specification and Architecture).

The Collaborative Procedure Designer (CPD) consists of two main components:

1. Web-based frontend;
2. Collaborative workflow management backend.

The web-based frontend is an Angular 2 application that interacts with the collaborative workflow management through REST API.

Angular 2 is a framework for building client applications in HTML and either JavaScript or a language (like Dart or TypeScript) that compiles to JavaScript. For the CPD application, TypeScript has been chosen. The framework consists of several libraries, some of which are core and some other are optional. Angular applications are written by composing HTML *templates* with Angularized markup, writing *component* classes to manage those templates, adding application logic in *services*, and boxing components and services in *modules*. The app is launched by *bootstrapping* the *root module*. The following figure depicts the main building blocks of a typical Angular 2 application:

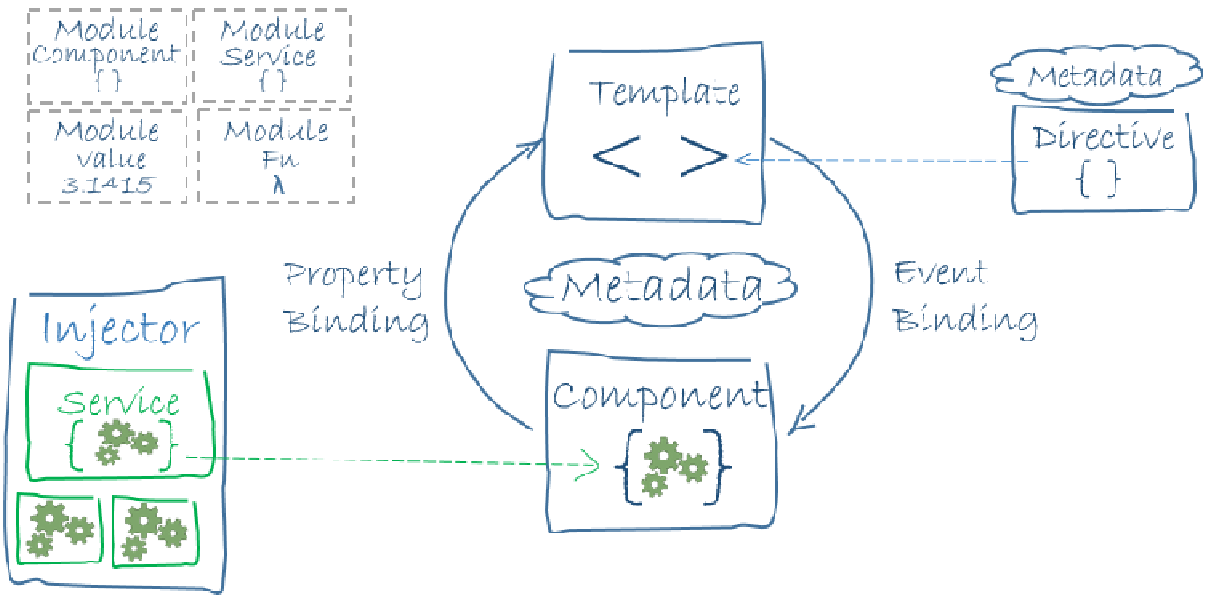


Figure 24: Architecture diagram showing the main building blocks of an Angular 2 application

We decided to opt for Angular 2 for the following reasons:

- it is a full featured framework for highly interactive web application development;
- it is a Google's well supported technology (just shifted from beta to rc);
- there is a plenty of freely usable components;
- it allows the use of TypeScript which provides more consistency compared to plain JavaScript.

With regards to the collaboration workflow management, an open source solution will be adopted: Activiti ™ BPM Platform by Alfresco®. This product will be deployed in a servlet container such as Apache Tomcat®.

6 Conclusion

The objective of this document is the definition of a clear and consolidated picture of the SIMPATICO architecture with specific focus on the definition of the interactions among the architecture's components. Having this purpose in mind, this document has formalized the integration requirements coming from the pilot scenarios, and specified the way SIMPATICO components will have to interact to each other in order for those requirements to be met.

In this regard, three different perspectives of the integration have been proposed. Two are static, while the third one addresses the integration problem from a dynamic point of view. Specifically, the presented perspectives are: an *architectural view*, which highlights direct relationships among components; a *component-centric* view (component card), which addresses the services each component is in need of and is capable of providing respectively; a *workflow view* (sequence diagrams), which put the focus on how components' integration dynamically takes place in respect to the specific goals declared in the use cases.

While in the document a general view of the SIMPATICO architecture and of the components' integration is given, the next deliverable (D5.2) will undertake a refinement process. In particular, the next integration step is geared toward the improvement of the level of the technical details of the architecture's components and the consolidation of the SIMPATICO architecture itself. Also, a more refined view for each component will be given in terms of the offered functionalities and the API to expose. Finally, an exhaustive list of project's use cases will be provided and detailed along. Once more, all use cases will be worked under a dynamic perspective by further specifying the sequence of actions each component is responsible of carrying on in order for the use cases' goals to be successfully attained.

7 References

A. Gordillo, D. G. (2013). The city as a learning gamified platform. *Frontiers in Education Conference, 2013 IEEE. IEEE* , 372-378.

Asquer, A. (2014). Not Just Videogames: Gamification and its Potential Application to Public Services. *Digital Public Administration and E-Government in Developing Nations: Policy and Practice, edited by Professor Edward F Halpin. IGI Global* .

Azab, M., & al., e. (2015). Using Word Semantics To Assist English as a Second Language Learners. *Proceedings of 2015 ACL.*

B. Cowley, J. M. (2011). Learning Principles and Interaction Design for 'Green My Place': A Massively Multiplayer Serious Game. *Entertainment Computing* .

B. Orland, N. R. (2014). Saving energy in an office environment: A serious game intervention. *Energy and Buildings, vol. 74* , 43-52.

Bista, S. K. (2014). Gamification for online communities: A case study for delivering government services. *International Journal of Cooperative Information Systems* .

C. Luengo-Oroz, M. A. (2012). Crowdsourcing malaria parasite quantification: an online game for analyzing images of infected thick blood smears. *Journal of medical Internet research* .

C.J. Lintott, K. S. (2008). Galaxy Zoo: morphologies derived from visual inspection of galaxies from the Sloan Digital Sky Survey. *Monthly Notices of the Royal Astronomical Society* , 1179-1189.

Charalabidis, Y. e. (2014). Passive crowdsourcing in government using social media. *Transforming Government: People, Process and Policy* , 283-308.

Charalabidis, Y. e. (2012). Public policy formulation through non moderated crowdsourcing in social media. *Electronic participation. Springer Berlin Heidelberg* , 156-169.

Dargan, T. a. (2015). Designing Engaging e-Government Services by Combining User-Centered Design and Gamification: A Use-Case. *Proceedings of the 15th European Conference on eGovernment 2015: ECEG 2015. Academic Conferences Limited* .

Devlin, S., & Unthank, G. (2006). Helping aphasic people process online information. *Proceedings of 2006 ACM.*

F. Khatib, S. C. (2011). Algorithm discovery by protein folding game players. *Proceedings of the National Academy of Sciences* , 18949-18953.

J. Hamari, J. K. (2014). Does Gamification Work?–A Literature Review of Empirical Studies on Gamification. *System Sciences (HICSS), 2014 47th Hawaii International Conference on. IEEE* , 3025–3034.

Mikolov, T. e. (2013). Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems.*

P. Herzig, B. W. (2013). Efficient Persistency Management in Complex Event Processing: A Hybrid Approach for Gamification Systems. *Proceedings of the 7th International Conference on Theory, Practice, and Applications of Rules on the Web, Berlin, Heidelberg* , 129-143.

P. Herzig, M. A. (2012). A Generic Platform for Enterprise Gamification. *Proc. of the 2012 Joint Working Conference on Software Architecture (WICSA) and European Conference on Software Architecture (ECSA). IEEE* .

P. Holleis, M. L. (2012). TRIPZOOM: A System to Motivate Sustainable Urban Mobility. *SMART 2012, The First International Conference on Smart Systems, Devices and Technologies* , 101-104.

Paetzold, G. H., & Specia, L. (2016). Benchmarking Lexical Simplification Systems. *Proceedings of 2016 LREC.*

Paetzold, G. H., & Specia, L. (2015). LEXenstein: A framework for lexical simplification. *Proceedings of 2015 ACL.*

Paetzold, G. H., & Specia, L. (2016). SV000gg at SemEval-2016 Task 11: Heavy Gauge Complex Word Identification with System Voting. *Proceedings of 2016 SemEval* .

Paetzold, G. H., & Specia, L. (2016). Unsupervised Lexical Simplification for Non-Native Speakers. *Proceedings of AAAI-16.*

Pistore, M. (2016). *D6.1 - Use-case Planning & Evaluation.* SIMPATICO project deliverable.

Pretel, I. (2016). *D4.1 - Citizenpedia Framework Specification and Architecture.* SIMPATICO project deliverable.

Quinn, A. J. (2011). Human computation: a survey and taxonomy of a growing field. *Proceedings of the SIGCHI conference on human factors in computing systems. ACM* .

R. Kazhamiakin, A. M. (2016). A Gamification Framework for the Long–term Engagement of Smart Citizens. *Smart Cities Conference (ISC2), 2016 IEEE Second International* .

R. Kazhamiakin, A. M. (2015). Using gamification to incentivize sustainable urban mobility. *Smart Cities Conference (ISC2), 2015 IEEE First International* , 1-6.

Rello, L., & al., e. (2013). Simplify or help?: text simplification strategies for people with dyslexia. *Proceedings of the 2013 International Cross-Disciplinary Conference on Web Accessibility.*

Report on Personal Data Stores. (n.d.). Retrieved May 2016, from https://ec.europa.eu/digital-single-market/en/news/study-personal-data-stores-conducted-cambridge-university-judge-business-school

Robertson, S. &. (2012). Mastering the requirements process: Getting requirements right. *Addison-wesley* .

S. Deterding, M. S. (2011). Gamification. Using Game-Design Elements in Non-Gaming Contexts. *Proc. of the 2011 Conference on Human Factors in Computing Systems. ACM* , 2425–2428.

S. Gabrielli, P. F. (2014). Design Challenges in Motivating Change for Sustainable Urban Mobility. *Computers in Human Behavior, vol 41* , 416–423.

Santos de la Cámara, R. (2016). *D3.1 - User Interactions Modelling and Design.* SIMPATICO project deliverable.

Siddharthan, A. (2014). A survey of research on text simplification. *International Journal of Applied Linguistics.*

T. Coenen, P. M. (2013). Zwerm: stimulating urban neighborhood self-organization through gamification. *Using ICT, Social Media and Mobile Technologies to Foster Self-Organisation in Urban and Neighbourhood Governance* .

von Ahn, L. (2005). Human Computation. *Ph.D. Dissertation, Carnegie Mellon University.* , Tech. Report CMU-CS-05-193.