



EUROPEAN COMMISSION

Research Executive Agency (REA)

Inclusive, Innovative and Reflective Societies



Simpatico

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

D5.4 – SIMPATICO Interoperability Framework and Use-Case Environments v2

Executive Summary: The aim of this document is to provide a technical overview of the installation and deployment of the software components developed by SIMPATICO for the three use-cases to be executed during the second iteration of SIMPATICO validation. The document also contains an analysis of the principles European Interoperability Framework and guidelines of their adoption by SIMPATICO.

WP: WP5 – Integration and Environment Setup

Author(s): A. Filograna (ENG), V. Savarino (ENG), R. Kazhamiakin (FBK), M. Trainotti (FBK), R. Santos de la Cámara (HIB), K. Zabaleta (DEUSTO), K. Moss (SPARTA), M. Garzoglio (TRENTO), N. López (XUNTA), M. Riggall (SCC)

Editor: A. Filograna (ENG)

Leading Partner: Engineering (ENG)

Participating Partners: FBK, DEUSTO, USFD, HIB, SPA, BENG, TRENTO, XUNTA, SCC

Version: 1.0

Status: Final

Deliverable Type: OTH: Other

Dissemination Level: PU: Public

Official Submission Date: 30/04/2018

Actual Submission Date: 30/04/2018

Version History

Version	Date	Author	Partner	Description
0.1	10/01/2018	A. Filograna	ENG	ToC proposal
0.2	19/02/2018	A. Filograna	ENG	Section 2
0.3	16/03/2018	G. Di Modica R. Kazhamiakin	BENG FBK	Section 3.2: docker adoption and installation&setup of components
0.4	20/03/2018	R.Santos V. Savarino K. Zabaleta	HIB ENG DEUSTO	Section 3.2: docker adoption and installation&setup of components
0.5	29/03/2018	C. Scarton	USFD	Section 3.2: docker adoption and installation&setup of components
0.6	04/04/2018	ALL	ALL	Section 3.1: Integration Use Cases
0.7	10/04/2018	A. Filograna	ENG	Executive Summary
0.8	12/04/2018	A. Filograna	ENG	Conclusion
0.9	19/04/2018	G. Di Modica K. Moss	BENG SPA	Internal Review
0.10	20/04/2018	A. Filograna	ENG	Final version
1.0	24/04/2018	M. Pistore, P. Forner	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This deliverable contains original unpublished work except where clearly indicated otherwise.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.

Table of contents

1	Introduction	8
1.1	SIMPATICO project	8
1.2	Structure of the deliverable	9
2	The new EIF and relation with SIMPATICO project	11
2.1	SIMPATICO EIF: performed updates.....	12
2.1.1	Subsidiarity and proportionality.....	12
2.1.2	Openness.....	12
2.1.3	Transparency	12
2.1.4	Reusability	12
2.1.5	Technological neutrality and data portability	12
2.1.6	User-centricity	13
2.1.7	Inclusion and accessibility	13
2.1.8	Security and privacy	13
2.1.9	Multilingualism	13
2.1.10	Administrative simplification.....	13
2.1.11	Preservation of information	13
2.1.12	Assessment of effectiveness and efficiency.....	13
2.2	Usability and Inclusive design definition	13
2.2.1	Identify	14
2.2.2	Analyse	14
2.2.3	Improve	14
3	Integration and deployment plan within each use-case	15
3.1	Updated integration use cases	15
3.2	Deployment of SIMPATICO solution in the second phase	21
3.2.1	Infrastructure requirements and setup.....	21
3.2.2	CDV installation and setup	22
3.2.3	TAE, WAE and AST installation and setup	23
3.2.4	UPM installation and setup	23
3.2.5	LOG, SF, and DA installation and setup	24
3.2.6	eSM installation and setup.....	25
3.2.7	Citizenpedia installation and setup	25
3.2.8	IFE and e-Service configuration.....	26
3.3	Trento use-case	28
3.4	Galicia use-case	29
3.5	Sheffield use-case	30
4	Conclusion	31
5	References.....	32



List of figures

Figure 1: SIMPATICO concept as a glance	8
Figure 2: EIF Conceptual Model (source: [2])	11

List of tables

Table 1: Integration use cases	15
Table 2 : Recommended environment of the SIMPATICO platform	21

Glossary

AAC	Authentication and Authorization Component
AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
AST	Authoring Support Tool
CDV	Citizen Data Vault
CKB	Collective Knowledge Base
CPD	Collaborative Procedure Design
DA	Data Analysis
DB	Dashboard
DB	Data Base
DOM	Document Object Model
EE	Enrichment Engine
eSM	e-Service Monitor
EIF	European Interoperability Framework
GE	Gamification Engine
GUI	Graphical User Interface
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IFE	Interactive Front-End
IL	Integration Layer
JSON	JavaScript Object Notation
LOG	Interaction Log (module)

PA	Public Administration
PDS	Personal Data Store
QAE	Question Answering Engine
RDF	Resource Description Framework
REST	Representational State Transfer
RO	Research objective
SF	Session Feedback
SME	Small and Medium Enterprise
SSO	Single Sign On
TAE	Text Adaptation Engine
UI	User Interface
UPM	User Profile Manager
WAE	Workflow Adaptation Engine
XML	EXtensible Markup Language



Executive summary

This report includes the results of project tasks T5.2 “**SIMPATICO interoperability framework**” and T5.3 “**Use-case deployment and connection with legacy systems**”. The aim of this document is to provide a technical overview on the SIMPATICO components’ installation and deployment.

Part of this work is also the study and adoption of the new European Interoperability Framework; in particular, we try to identify the connection between the **12 EIF principles** and the interoperability features offered by SIMPATICO. Following these principles, the Usability and Inclusive Design for components was defined to be adopted during the second execution phase.

The new functionalities provided by each component led to **new or updated Integration Use Cases** that aim to evaluate the integration degree of the whole platform. Also, the customized deploy of the platform for each Use Case was updated according to the new functionalities provided by each component.

To facilitate the installation and deployment of each component and the overall platform, the **dockerization** of them was performed. The description of instruction for each component have been published on the project Github account.

1 Introduction

This deliverable presents the outcomes of SIMPATICO project tasks T5.2 "*SIMPATICO interoperability framework*" and T5.3 "*Use-case deployment and connection with legacy systems*" in the scope of WP5 "*Integration and environment setup*". During the second phase of project execution, this task has worked to analyse and receive the second version of European Interoperability Framework and adapt it to SIMPATICO project, where possible. Moreover, all the activities carried out to deploy, set up and integrate the SIMPATICO platform in order to be ready for the second phase of use-cases execution were performed.

To better understand the aim and scope of this document, in this introductory section we provide an overview of the SIMPATICO project (Section 1.1) and a description of the structure of the remainder of this deliverable (Section 1.2).

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).

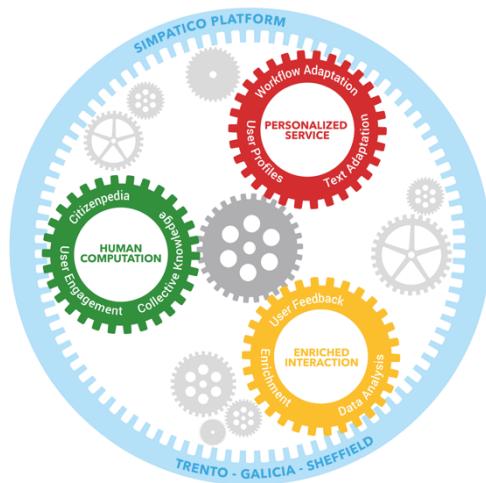


Figure 1: SIMPATICO concept as a glance

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 RO3, covering both the definition of the integration use cases to be interoperable among all platform component, and the adoption of European Interoperability Framework to be interoperable with the PA legacy systems.

1.2 Structure of the deliverable

The remainder of the deliverable is organized as follows:

Section 2 describes the new concepts of the European Interoperability Framework and the relation with the SIMPATICO project. Therefore, a definition of the usability and inclusive design was provided.



Section 3 describes the activities performed for the integration and set up of the SIMPATICO platform in the three use-cases for the second phase of use case execution.

The Conclusions are described in **Section 4**.

2 The new EIF and relation with SIMPATICO project

The European Interoperability Framework (EIF) was launched in 2010 with a first version of 25 recommendations. The new technological trends like cloud computing, open and big data have raised the need to upgrade and align the previous EIF with new requirements. These requirements come from a survey conducted among the EIF stakeholders (Public Administrations, citizens, standardisation organizations, research centres, etc.) that stated their new requirements about the interoperability issues they face with.

The new EIF has improved and increased the recommendations from 25 to 47, focusing on the ease-to-use and simplicity of their implementation. More emphasis was put on the **openness and information management, data portability, interoperability governance, and integrated service delivery**[1].

The **Digital Transformation** is in tight relationship with interoperability since the latter facilitates the communication among different PAs, citizens and businesses. This interoperability concerns all aspects of public services provided by European Public Administrations:

- **Organizational** level: EIF encourages public administrations to simplify their organisation, streamline their processes and listen to the needs of business and citizen (e.g. pledging a formal agreement about the conditions among different organizations).
- **Semantic** level: EIF calls upon public administrations to structure their data in commonly agreed formats (e.g. defining a set of common descriptions of exchanged data).
- **Technical** level: EIF promotes the sharing and reuse of common infrastructures, services and IT systems (e.g. guaranteeing a technical environment that allows a seamless data flow).
- **Legal** level: EIF proposes that EU and national legislation and policies are clear, coherent to each other and make good use of digital technologies (e.g. assuring a common vision about the regulations that guarantees the reuse of data in different areas).

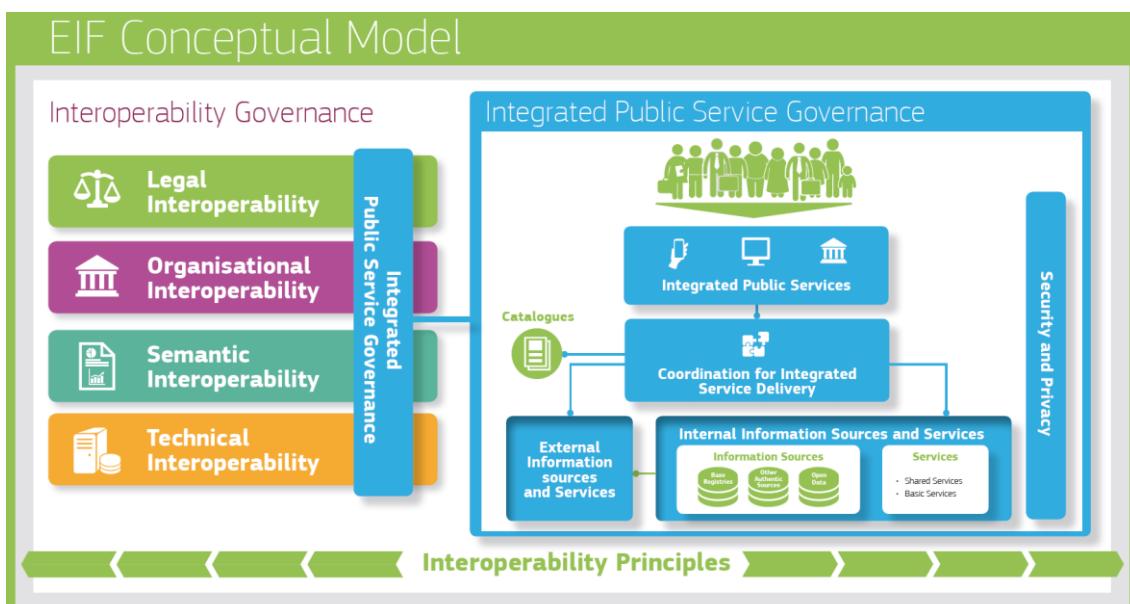


Figure 2: EIF Conceptual Model (source: [2])



The first three levels already belonged at old EIF, the Legal aspect was introduced in the new version.

In particular, the digitalisation of Public Administration at European level allows for all Member States to plan the adoption of new EIF and its related recommendations. In this way, the provided procedures and public services will automatically be digital natives and compliant with the most important EU regulation, like eIDAS [3] and GDPR [4].

2.1 SIMPATICO EIF: performed updates

The new European Interoperability Framework defines **12 principles** aiming to achieve a general behaviour about the interoperability. These principles set out the best practise to design and implement European public services.

SIMPATICO project is trying to be compliant with the principles as many as possible, at least with some recommendations provided by them. Below is the list of twelve principles and their adoption within the SIMPATICO project.

2.1.1 Subsidiarity and proportionality

This principle foresees that all the actions undertaken by EU cannot be less effective towards the citizens than those ones already defined at national level. This is out of the scope of SIMPATICO project.

2.1.2 Openness

This principle boosts the openness of data, software and specification. In SIMPATICO, all the specification and software are open and open source. In particular, the definition and documentation of APIs is made of by adopting Swagger.

2.1.3 Transparency

This principle deals with the visibility of rules, processes data and service within the Public Administration and outside guaranteeing the protection of personal data. In SIMPATICO, the Citizen Data Vault is in charge to provide and manage the personal data in compliance with the GDPR. In addition, all the processes involving an interaction with the citizens through e-services were modelled by means of the Collaborative Procedure Design.

2.1.4 Reusability

The reusability of software APIs and in general IT solution enables the interoperability. All the software developed within SIMPATICO follow this basic principle.

2.1.5 Technological neutrality and data portability

This principle is based on having public services without any duty of specific technology and able to transfer data from a technological system to another one. A basic rule of SIMPATICO is to develop software basing on no-vendor lock-in. Moreover, the CDV provides all users with the possibility to download their data in a standard format in order to transfer them wherever they want.



2.1.6 User-centrity

PA has to put the user at the centre when planning and providing public services. This is the aim of SIMPATICO, simplifying the interaction with the Public Administration, taking into account their feedback and put in place the action to redesign the public services according to the received feedback.

2.1.7 Inclusion and accessibility

The public services have to be provided to all citizens and be accessible to each kind of disadvantaged user. In addition, the new technologies should overcome the social and economic divide. In the second year of project, SIMPATICO is developing the usability and inclusive design (see section 2.2) to support this part of citizens.

2.1.8 Security and privacy

This principle guarantees the security and privacy of citizens that use the public services. SIMPATICO adopts all the technologies to ensure the security and privacy of data exchange among them and the PA.

2.1.9 Multilingualism

The service should be provided in different languages to be used by all citizens. The citizens that use a public service written in a language they do not speak, usually they do not use it. SIMPATICO is compliant with the internationalization (following the in8 standard).

2.1.10 Administrative simplification

This principle aims at helping citizens and businesses to reduce the administrative burden due to EU and national bureaucracy. This is out of the scope of SIMPATICO project.

2.1.11 Preservation of information

The goal of this principle is to assure a long-term preservation of information provided by European public services and their exchange among Member States. This is out of the scope of SIMPATICO project.

2.1.12 Assessment of effectiveness and efficiency

This principle suggests taking into account the effectiveness and efficiency of the interoperability solution put in place. This should give the value of level of flexibility, administrative burden, simplification and the balance between costs and benefit. This is out of the scope of SIMPATICO project.

2.2 Usability and Inclusive design definition

A new inclusiveness and usability methodology has been created in SIMPATICO project. The idea of creating this methodology comes not only from the EIF principles but also from the feedback received in the pilots. Stakeholders, elderly and disabled people in particular reported problems related to inclusiveness and usability.



The goal of this new methodology is the detection of the problems related to usability and inclusiveness, and finding the best way to solve them. The methodology is iterative and each iteration has the following steps.

2.2.1 Identify

The aim of this step is to identify all the issues that should be improved and prioritize them in order to detect the most critical ones. A spreadsheet has been created with all the principles written in the first column. Inclusive design principles [5] and the ten usability heuristics proposed by Nielsen [6] were used.

The duty of the evaluators is to write below each of the principles all the issues related to them and score them. The score is based on the following classification scheme.

1. Mild trouble: is not considered as an error
2. Not serious error: Must be solved but is not a priority
3. Error: Trouble that must be solved
4. Critical error: Severe problem that must be solved as a priority.

This spreadsheet has also non mandatory columns where the evaluator can write some notes or his/her contact details in order to facilitate the job of the responsible of the evaluated component.

2.2.2 Analyse

Once the identification step is finished the responsible for the component to improve must analyse one by one all the open issues sorted by priority and try to identify the best way to proceed. If the responsible has any doubt or does not understand what the problem is, they can contact with the person who wrote the issue.

2.2.3 Improve

In this last step the developers must improve the component using the guidelines obtained in the previous steps.

Once the last step is concluded the iteration also concludes and a new one must start. In this new identification step the issues identified in the previous one should be checked again.

This new methodology has been used to improve the QAE and CPD components in first instance. Based on the good results of the first iteration the methodology is being used with the rest of the components which have visual interfaces.



3 Integration and deployment plan within each use-case

This Chapter is devoted to show all the integration and deployment activities at both component and pilot level. All the integration use cases are illustrated reporting the dependencies by other components, the updated guidelines to deploy and install every component and the definition of their related adoption of Docker container. In the end, the customization in each pilot is described.

3.1 Updated integration use cases

Table 1 shows the integration use cases updated after the second round of new component functionalities collection and provided in order to test the integration of the whole platform. During the final validation of the SIMPATICO platform, one of the activities to be carried out will be the validation of these use cases. All the information will be reported in the Deliverable D5.6 [7] at M36 (January 2019).

Table 1: Integration use cases

COMPONENT	USE CASE	SYNOPSIS	COMPONENTS	DEPENDENCIES	PREREQUISITES	BEHAVIOUR	EFFECT
IFE	IFE1	Login	IFE, AAC			Open a service page, press login button. A new window where AAC authentication takes place. Upon window closure, the toolbar is updated and user name is shown	User is authenticated, toolbar unlocked, user name is shown
	IFE2	Session execution	IFE, LOG			User starts/terminates the e-service session	Start/end session events are logged and appear in LOG
	IFE3	E-service form compilation	IFE, LOG			User starts/terminates compilation of the e-service module	Start/end form events are logged and appear in LOG component
	IFE4	Access to an annotated element	IFE, LOG			User access any annotated element (TAE, QAE, CDV)	Click event is logged and appears in LOG component



	CDV1	Link Account to Service	IFE, CDV, AAC	IFE1	e-service is configured in CDV.	Click CDV button in toolbar, CDV popup is shown. Popup asks the user to create a new CDV account and link it to the service	A new account is registered in CDV. Account is linked to the current e-service
CDV	CDV2	Save user data	IFE, CDV	IFE1, CDV1	e-service is configured in CDV. the user has created a CDV account	The user fills in the e-service form and clicks on the "Save Data" button shown in CDV popup	The user data are saved in CDV
	CDV3	Read/populate user data	IFE, CDV	IFE1, CDV1, CDV2	e-service is configured in CDV. the user has already created a CDV account	During the e-service form compiling, the user can use his/her own data already stored in the CDV selecting them by drop down menu in the requested field	The user can fill in the e-service form retrieving the data in the CDV, previously saved
	TAE1	Free text simplification	IFE,TAE,LOG	IFE2		Select a sentence and click on "Free text simplification" toolbar button. A popup appears, tabs show enriched text.	Show the simplified sentence sent from TAE and ask for feedback. Phrase simplification event is registered in LOG component
TAE	TAE2	Free text simplification	IFE,TAE,LOG	IFE1		Select a phrase and click on "Free text simplification" toolbar button. A popup appears, tabs show enriched text.	Each tab shows content from TAE and asks for feedback. Sentence simplification event is registered in LOG component
	TAE3	Free word simplification	IFE,TAE,LOG	IFE1		Select a single word and click on "Free text simplification" toolbar button. A popup appears, tabs show enriched word.	Corresponding tab shows content from TAE and asks for feedback. Word simplification event is registered in LOG component
WAE	WAE1	Workflow simplification	IFE,WAE,LOG	IFE1	E-service workflow model is uploaded	Click "workflow adaptation" button. The simplified process	The profile is requested. The workflow model is downloaded



					to WAE repository, form is annotated with the model URI value	starts (the page scrolls to the first workflow block, the block is highlighted, next/prev buttons are added, the questions are issued to user). The interaction depends on the user profile obtained from UPM. The user can execute the workflow.	from the WAE repository, the engine starts the model execution. Workflow adaptation request event is registered in LOG component
QAE	QAE1	Login with AAC	QAE,AAC			User signs in using AAC	The user successfully logged in, the profile data is shown in QAE
	QAE2	See e-service questions	IFE,QAE,LOG	IFE1	e-service elements are annotated with a specific class corresponding to e-service paragraph	Click "QAE" toolbar button, the annotated page elements are highlighted. Clicking on the highlighted element shows the questions associated to the paragraph (if any) and a "add a question" link	The list of existing paragraph questions are associated to each annotated paragraph. Upon selecting a specific element, the 'Citizenpedia content request' event is registered in LOG component.
	QAE3	See specific question	IFE,QAE,LOG	IFE1,QAE2		User clicks on a specific question link. A Citizenpedia page with question details is opened in a new window.	Citizenpedia page is opened. The 'Citizenpedia question request' event is registered in the LOG component
	QAE4	Create new question	IFE,QAE,LOG	IFE1,QAE1, QAE2	e-service category is created	Within QAE2 scenario, user clicks the "add a question link". Another browser tab is opened with the precompiled Question form.	QAE is opened with the precompiled form for question creation. Once saved, the question appears in the corresponding paragraph section in IFE. The 'Citizenpedia new question' event is registered in LOG component



	QAE5	Create new answer	QAE,LOG	QAE1		Inside Citizenpedia, the user provides an answer to an existing question.	The number of answers associated to the question is incremented, the answer is saved in the DB. The 'new answer' event is registered in the LOG component
SF	SF1	SF is invoked	SF, IFE		A session is complete	The front-end side of the SF captures the appropriate event of the IFE and asks the back-end of the SF to display a feedback capture form.	The SF feedback form is presented in the front-end for the user to fill in their opinions.
	SF2	SF stores user feedback	SF, LOG	SF1	The user presses the 'Send' button in the SF UI.	The data that the user produces is compiled and stored as the current session's feedback in the LOG.	The LOG is updated with data from the last session.
	SF3	SF generates an appropriate feedback form	SF, LOG	SF1	The SF is invoked	The SF internal logic retrieves relevant data from the last session (e.g., was the text adaptation invoked at any stage) and designs an appropriate feedback form which is sent to the front-end part.	The SF front-end part receives the requested form.
CPD	CPD1	Login with AAC	CPD, AAC			User signs in using AAC	The user successfully logs in, the profile data is shown in CPD. Depending on the user role (citizen/civil servant) access to some functionality can be denied
	CPD2	Open an existing procedure diagram	CPD		The user has successfully logged in	The user opens a diagram from a list of available (previously created) administrative procedures	The user is presented with a graphical representation of the selected procedure



	CPD3	View existing questions	CPD, QAE		The user has opened an existing procedure diagram	The user selects to view the questions posted on a procedure's activity	The user gets re-directed to the QAE, where they are presented with all the questions related to that activity
	CPD4	Submit a new question	CPD, QAE		The user has opened an existing procedure diagram	The user selects to submit a new question on a procedure's activity	The user gets re-directed to the QAE, where they are provided with a text editor to write a new question related to that activity
	CPD5	Submit a feedback	CPD		The user has opened an existing procedure diagram	The user selects to submit a new feedback (suggestion, comment) on a procedure's activity	The user is presented with a pop-up editor to post a new feedback related to that activity
	CPD6	Start a collaborative process to design an administrative procedure	CPD		The Procedure owner has successfully logged in	The Procedure owner assigns participants to the collaborative process and instantiates it	A new collaborative process is instantiated and participants receive notifications
	CPD7	Execute a collaborative process	CPD		The collaborative process' participant has successfully logged in	The participant executes the task they have been assigned	The administrative procedure is published and available
DA	DA1	Data Analysis is invoked on schedule	DA		There exists some user interaction data in the LOG that can be processed.	The scheduling mechanism (e.g., cron) invokes the DA upon its required schedule (e.g., each minute, each hour). The DA executes its internal rules and data available in the LOG which hasn't been processed before is processed.	Aggregated processed data is stored in the LOG for further analysis by DA or representation in the EE



AST	AST1	The text is analyzed and scores about it are provided	AST	TAE	The TAE must be running.	The AST accepts a text as input. It gives a list of information about text as output, such as difficulty measures, simplification suggestions and statistics on the text.	The data is shown in a web interface and can be used by the civil servant.
UPM	UPM1	Demographic data is sent to be stored in the UPM database	UPM, IFE	IFE		IFE requests demographic information from the user and send the data to be stored in the UPM	A request to add the demographic data is processed by the UPM database
	UPM2	Interaction data is sent to be stored in the UPM database (given a request from the UPM)	UPM, DA	DA	There are interaction data in the LOG and such data has already been processed by the DA module	DA processes the interaction data from the LOG and selects the interaction data requested by the UPM. This would happen in batches (every 24 hours for instance)	A request to add the interaction data is processed by the UPM database
	UPM3	TAE and WAE request user profile	UPM, TAE, WAE	TAE, WAE	There are data available in the UPM database	The TAE or WAE request data to the UPM. Such data maybe demographic data only, interaction data only or a mix o both. If UPM evolves and it is capable of learning profiles, a profile class could also be send	TAE and WAE obtain the required data



3.2 Deployment of SIMPATICO solution in the second phase

To facilitate the deployment of the SIMPATICO platform component, in the 2nd phase of the project the **Docker [8] containerization platform** has been adopted. Docker allows for configuring and running the software components in isolation and without affecting the operation of other software components, and without compromising the resource exploited. Following this model, each SIMPATICO component is provided with its own Docker configuration, which defines:

- the set of Docker images to be created;
- the deployment-specific configuration of the software to run in the containers.

It is important to note that for some of the SIMPATICO components there may be more than one image provided. This is required, in particular, in case the underlying database is provided separately from the main executable, or if the component is deployed as a set of microservices.

Besides, on top of the configurations of individual components, SIMPATICO provides a container configuration to run the entire platform. This is needed (apart from simplifying the development and integration phases of the platform creation) in order to facilitate the adoption of the platform by other cities. More specifically, the configuration is achieved using the Docker "compose" framework. The instructions for the platform configuration and deployment using Docker are described here: <https://github.com/SIMPATICOProject/simpatico-platform>.

The following subsections describe both the Docker configuration and the updated installation and set-up for every SIMPATICO platform component.

For sake of clarity, the setup and deployment component description provided in the first phase is reported with the related updates, where needed.

3.2.1 Infrastructure requirements and setup

Deployment of the SIMPATICO platform amounts to installation and integration of a set of the software components that provide the required APIs or Web user interface towards citizens or civil servants.

The recommended environment for the platform installation is described in the following table:

Table 2 : Recommended environment of the SIMPATICO platform

Operating System	Linux	No specific constraints.
Execution Environment	JVM Python Node.js Apache Tomcat Elasticsearch	1.8+ 2.7+ 4.2.3+ 8+ 5.x
Database	MySQL MongoDB	5.5+ 3.4+
Storage	> 100GB	Depends on TAE configuration
RAM	> 30GB	Depends on TAE configuration
Processors	> 8CPU	Depends on TAE Configuration



The required resources may vary depending on the configuration used by the Text Adaptation Engine. More specifically, the amount of RAM, storage, and processing depends on the language(s) engaged by the specific platform instance.

For the authentication and access control, the SIMPATICO platform components rely on the SIMPATICO AAC component, which provides the support for OAuth2.0 authentication protocol and serves as an authorization server for accessing the platform resources and APIs. It also intermediates the identity provisioning tasks, allowing for integration with the city-specific or national-level authentication systems through its modular architecture. The AAC component is deployed as a Java Web application in a standard JEE container (e.g., Apache Tomcat). It provides the out-of-the-box integration of the Google and Facebook authentication systems.

Detailed description of the installation and setup instructions are presented here:

<https://github.com/SIMPATICOProject/aac>

It is strongly recommended that the access to the platform APIs and components is protected using HTTPS connection as some of the user-related information is being passed between the user fronted (IFE) and the server components (e.g., CDV).

Docker configuration

The deployment of the AAC components is available in two methods. Firstly, it can be built and executed from the source code using the corresponding run-time environment. This procedure has been also exploited in the 1st phase of the project. Secondly, AAC may be deployed with the help of Docker containerization platform. In the latter case, the AAC Docker configuration defines two elements:

- AAC executable. Specifically, the AAC executable is provided as a Java Web application deployed within JEE server (Tomcat).
- AAC database. Specifically, the MySQL database image is exploited for that purpose.

The configuration instructions are available here: <https://github.com/SIMPATICOProject/aac>

3.2.2 CDV installation and setup

Citizen Data Vault exposes the functionality of the citizen personal data repository. This functionality is exposed via API, specifically RESTful services. CDV setup requires the following environment:

- MySQL 5.7+, MongoDB 3.3+
- Java 1.8+
- Apache Tomcat 8+

The detailed installation and setup procedure is described here:

<https://github.com/SIMPATICOProject/CDV>

Please note that for the city-specific e-services to be integrated with CDV, it is necessary to populate the CDV database with the mapping of the e-service fields and the user profile data fields. This mapping should be provided via CDV API at the moment of the e-service configuration.

Docker configuration

According to its architecture and its main functionality, CDV Docker structure and development comprises 3 main images. The decoupling in three images enables the management of CDV Server



APIs, database for key security management and data repository. Besides, this decomposition ensures a good decoupling into multiple containers and makes CDV application much easier to scale horizontally and reuse containers, but at same time to guarantee the data isolation requirements needed for the deployment behind PA Legacy System (see Trento Use Case).

The configuration instructions are available here: <https://github.com/SIMPATICOProject/CDV>

3.2.3 TAE, WAE and AST installation and setup

Text Adaptation Engine and Workflow Adaptation Engine expose advanced techniques for improving the use of the e-services by citizens and the interactions with those services. These components, together with the Authoring Support Tool, are deployed and provided as a single module. The module setup consists of server-side components installation and a front-end configuration as a part of the Interactive Front-End setup. As for the server-side components, the specific procedure depends on the languages used and may vary. The setup requires the following environment:

- MongoDB 3.4+
- Java 1.8+, Python 2.7+ (for English and Spanish setup)

More detailed installation instructions may be found here:

<https://github.com/SIMPATICOProject/simpatico-adaptation-engines>

Please note that the workflow models of the e-services deployed by the city should be stored in the WAE repository using the corresponding APIs. The details about the workflow model and its usage can be found here: <https://github.com/SIMPATICOProject/simpatico-adaptation-engines/blob/master/doc/wae-model.docx>

Docker configuration

The deployment of the SIMPATICO adaptation engine components (TAE, WAE and AST) is available in two flavours. Firstly, it can be built and executed from the source code using the corresponding runtime environment. This procedure has been also exploited in the 1st phase of the project. Secondly, the engines may be deployed with the help of Docker containerization platform. In the latter case, the Docker configuration defines two elements:

- TAE/WAE/AST executable. Specifically, the executable is provided as a Java standalone application.
- TAE/WAE storage. Specifically, the MongoDB database image is exploited for that purpose.
- TAE Server for English and Spanish with MUSST tool for syntactic simplification. This module provides an extension for the basic TAE implementation and is provided as a single container with Python executables.

The configuration instructions are available here: <https://github.com/SIMPATICOProject/simpatico-adaptation-engines>

3.2.4 UPM installation and setup

The User Profile Manager module (UPM) has two functionalities:

1. Store information about demographic data and adaptation information from users of SIMPATICO platform;



2. Support other components, mainly TAE and WAE, with information about user profiles, based on the information gathered.

The engine setup consists of server-side components installation and interfaces to connect with IFE, DA, TAE and WAE. The setup requires the following environment:

- MySQL server (Version 14.14 Distribution 5.7.21) - <https://dev.mysql.com/>
- Python 2.7 (or older)
- Python libraries: mysql (<https://dev.mysql.com/doc/connector-python/en/>) -- I installed using: pip install mysql-connector-python-rf

More detailed installation instructions may be found here:

<https://github.com/SIMPATICOPROJECT/SimpaticoUPMServer>

Docker configuration

The deployment of the SIMPATICO UPM component can also be performed using the Docker containerization platform. In this case, the Docker configuration defines two elements:

- MySQL DB as a storage for the profile information
- The Python executables that provide the API for interacting with the UPM component.

The installation instructions for the Docker-based set up may be found here:

<https://github.com/SIMPATICOPROJECT/SimpaticoUPMServer>

3.2.5 LOG, SF, and DA installation and setup

The LOG, SF and DA components are packaged together and can be downloaded from the project Github at <https://github.com/SIMPATICOPROJECT/logs> where also instructions for installation are available.

Installing and configuring the components is straightforward. The dependences for the installation are:

- Java 1.8+
- J2EE Servlet Container (Tomcat 7+)
- Elastic Search (5.0+)
- Kibana (5.0+) (Optional)
- Java IDE (eclipse)
- Piwik/Matomo analytics 3.0+

Following the deployment of the software for these components, they need to be configured. This is done following simple instructions for each component mentioned in the instructional readme file README.MD.

Docker configuration

There is a Docker deployment system available. Instructions to deploy it are available at the README.MD file. The Docker is straightforward and performs the following actions:

- Installing an image of Apache Tomcat and opening port 8080
- Deploying the web application with LOG, SF and DA as a .war file.
- Apply configurations for the different modules contained in the web application.



3.2.6 eSM installation and setup

The eSM module is delivered in the project code repository at <https://github.com/SIMPATICOProject/eSM>. It is based on Node.js; for that reason, the main dependency is:

- Node.js 4.2.3 or above

Other modules may be helpful to deal with some aspects of the module (e.g., version management or debugging) and as such as marked as *Optional* in what follows.

Following the deployment of the software for these components, they need to be configured. This is done following simple instructions for each component mentioned in the instructional readme file README.MD.

Docker configuration

There is a Docker deployment system available. Instructions to deploy it are available at the README.MD file. The Docker performs the following actions:

- Installing an image of the Node.js operating environment and opening the required port 3700.
- Installing the node.js dependencies required for the eSM using npm.
- Apply configurations for the Node.js code.

3.2.7 Citizenpedia installation and setup

Question and Answer Engine (QAE) provides the user with a knowledge base, where the citizen can ask and discuss the issues related to the specific e-services. The Collaborative Process Designer (CPD) is a tool for visualizing the administrative procedures overarching the e-services provided by a city. Servicepedia is a tool in which all the information (description, related e-services, questions, etc) about an administrative procedure is summarized for a best procedure understanding together these tools form SIMPATICO Citizenpedia.

The execution environment required by the QAE includes:

- Node.js 4.2.3+
- MongoDB 3.4+

The detailed instructions for the QAE installation and setup can be found here: <https://github.com/SIMPATICOProject/citizenpedia>. Also, a Docker container version is present in the same repository, in case the application is to be executed that way.

The execution environment required by the CPD includes:

- Java Runtime Environment 1.8+
- MongoDB 3.4+

The detailed instructions for the CPD installation and setup can be found here: <https://github.com/SIMPATICOProject/CPD>.

The execution environment required by the Servicepedia only needs a web server (Apache or nginx).

The detailed instructions for the Servicepedia installation and setup can be found here: <https://github.com/SIMPATICOProject/servicepedia>.



A new git repository was created to host the whole Citizenpedia bundle. In the repository, the components' repositories are linked as sub-modules. The Citizenpedia repository can be found here: <https://github.com/SIMPATICOPROJECT/simpatico-citizenpedia>. In the repo, instructions are provided on how to build and run a docker image of the Citizenpedia.

Docker configuration

The QAE is available in two different versions: as a node/express application that can be executed in a MEAN server and Docker Compose file that will launch the multi-container file that runs everything needed to deploy a working version.

The source code for both versions can be found in the public Github repository (<https://github.com/SIMPATICOPROJECT/citizenpedia>). There, in the README file, the versions and needed tools are specified.

The CPD will be available for deployment in two methods: the first one is a regular build from the source code, which produces artifacts (executables) for a run-time environment that has to be manually set up; the second is the Docker way, which delivers a ready-to-execute container bundled with the run-time environment and the CPD executable. The container will also be pushed in the Dockerhub public registry. The 17.12.1-ce version of the tool will be used for the dockerization purpose. Instructions to run the build and run the dockerized CPD will be released here: <https://github.com/SIMPATICOPROJECT/CPD>

3.2.8 IFE and e-Service configuration

Interactive front-end (IFE) represents an extension to be added to e-service Web page in order to enable the access to the SIMPATICO functionality. This extension is implemented as a set of JavaScript files and other assets (e.g., images, CSS, etc) that provide the user with the UI tools to access text adaptation, workflow adaptation, CDV, Citizenpedia, and session feedback.

The source code of the IFE modules with the corresponding setup instructions and examples can be found here:

<https://github.com/SIMPATICOPROJECT/IFE>

The standard IFE configuration assumes that the SIMPATICO tools are made available via a fixed toolbar, where a set of buttons provides access to the specific tool functionality. The toolbar is made available to authenticated users only, hence the integration with AAC for user identification and authentication. Please note that the combination of the tools as well as the visualization of the toolbar may be customized via the simpatico-ife.js file, where it is possible to change the way the individual components are activated and structured. The appearance of the toolbar and of the individual tools may be customized via CSS changes.

The standard integration of the SIMPATICO platform with an e-service consists of the following steps:

- I. Set up and configure the SIMPATICO environment

The IFE is integrated into the e-service page via importing a set of JavaScript and CSS files required by the UI tools. It is also necessary to configure a set of global properties specific to the current e-service page.

```

<!-- SIMPATICO BEGIN -->
<link rel="stylesheet" type="text/css" href="css/simpatico.css" />

<link rel="stylesheet"
      href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css" />
<script src="https://code.jquery.com/jquery-1.12.4.js"></script>
<script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>

<script type="text/javascript">
    var simpaticoEservice = "2"; // the id corresponding to the e-service
    var simpaticoCategory = "Infanzia"; // the general category of the e-service
    var simpaticoMapping=["AventeTitolo_EMailPEC","AventeTitolo_Fax","AventeTitolo_EMail"];
    var ERROR_LABELS = {
        'block1' : 'Manca il codice fiscale',
        'block4' : 'Manca selezione Part-time / Full-time'
    }
</script>

<script src=".js/ctz-ui.js"></script>
<script src=".js/ctz-core.js"></script>
<script src=".js/cdv-ui-popup.js"></script>
<script src=".js/cdv-core-popup.js"></script>
<script src=".js/tae-core.js"></script>
<script src=".js/tae-ui.js"></script>
<script src=".js/wae-core.js"></script>
<script src=".js/wae-ui.js"></script>
<script src=".js/tae-core-popup.js"></script>
<script src=".js/tae-ui-popup.js"></script>
<script src=".js/simpatico-auth.js"></script>

<script src=".js/simpatico-ife.js"></script>
<!-- SIMPATICO END -->

```

Besides, the individual tools may require specific configuration that are achieved via HTML5 custom attributes or through a set of custom CSS classes as shown below.

II. Set up Workflow Adaptation SIMPATICO component binding

First, it is necessary to bind the form to the corresponding workflow model:

```
<form data-simpatico-workflow="http://simpatico.eu/test" ...>
```

Second, it is necessary to mark-up the definition of the workflow blocks. More specifically, each workflow block is identified by the HTML class element “simpatico-block”. The workflow block definition can be done in two ways.

- if the workflow block matches with an HTML element, the class parameter of this element must be

overloaded with the value “simpatico-block” (e.g. <table class="simpatico-block ...">).

```
<table class="simpatico-block ..." ...>
```



```
...  
</table>
```

- b) If the workflow block matches with more than one HTML elements, a new element surrounding the elements set must be added (e.g. `<div class="simpatico-block">`).

```
<div class="simpatico-block">  
  <p ...>...</p>  
  <table ...>...</table>  
  <table ...>...</table>  
</div>
```

III. Set up the Question and Answer SIMPATICO component binding

```
<table class="simpatico-question-and-answer ..." ...>  
  ...  
</table>  
  
or  
  
<div class="simpatico-question-and-answer">  
  <p ...>...</p>  
  <table ...>...</table>  
  <table ...>...</table>  
</div>
```

IV. Set up the Text Adaptation SIMPATICO component binding

```
<p class="simpatico-text-paragraph">...</p>
```

In the following three sections each Use-case describes what is the configuration and deployment activities specific for their City.

3.3 Trento use-case

In the first evaluation phase Trento pilot adopted the standard SIMPATICO integration approach for all components but the Citizen Data Vault. The e-services HTML pages have been enriched with the required JavaScript and tags in order to enable the SIMPATICO toolbar. The requirement to integrate the CDV with a different approach comes directly as a constraint from the Municipality of Trento.

The Municipality of Trento has adopted “Sportello Online”, an end-to-end solution provided by company “GLOBO srl”, specifically targeting the digitalization of modules for service provision by public administrations. Within this solution, the digital module is a composition of sections of organic information (e.g., birth data section, residence data section, real estate registry data section). The module designer explicitly maps the logic of the interaction with an information section. The integrations with legacy systems are handled via a centralized REST web service, which routes the



service request to the right data source service. Finally the solution supports module hierarchy, which guarantees the definition of a well-organized digital module library.

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

The integration uses two different strategy:

- 1) Injection of JavaScript in the different digital modules;
- 2) Invocation of REST web services to handle interaction with legacy systems.

The initial assumption was that the injection of JavaScript in the digital module could be possible only if it does not interfere with the module interaction logic. This means that the injected JavaScript can operate on the DOM 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.

From the experimentation and tests made in phase one, it was clear that the change of the dynamic element (the field the user has to fill) is safe because it does not interfere with the business logic included in the module. Basing on this, in order to enable the pre-filling of some fields with the CDV data, the integration model chosen (and confirmed for phase two) is the one based on JavaScript injection.

For the integration with the local IT systems, we deployed the CDV component within the borders of the information system of the Comune di Trento, while the other SIMPATICO tools were deployed on the project cloud infrastructure. This is mainly due to the constraints posed the fact that the CDV contains user personal data. This solution prevents voluntary or accidental access to the personal data and simplifies the integration security and authentication issues between “Sportello Telematico” and CDV.

The evaluation raised some issues and in particular which requires a slight change in the solution deployment: Little usage of the text adaptation and little usage of Query & Answer engine. In phase two, in order to address these issues, the text adaptation and citizenpedia tools will be applied to each service description pages where the features could have a higher impact.

3.4 Galicia use-case

During this second year of the SIMPATICO project, a third e-service has been added to the Galicia use.

- BS811A regarding the assessment of the degree of disability.
(original e-service: <https://sede.xunta.gal/detalle-procedemento?codtram=BS611A>)

As in the first phase the e-services have been replicated and the SIMPATICO tools have been integrated to them. In this phase all the tools used in the first phase have been maintained and two new have been added.

- Form simplification: Using this tool the users can avoid filling some fields in the electronic forms making the process faster and easier.



- Servicepedia: The Servicepedia is a new tool inside the Citizenpedia component. Thanks to it, now, the citizens and civil servants can access to all the information about an administrative procedure (Description, steps, e-service and its components description) and all the questions and answers related to them.

3.5 Sheffield use-case

In the first evaluation phase the Sheffield pilot adopted the standard SIMPATICO integration approach for all components apart from the Citizen Data Vault. The e-services HTML pages have been enriched with the required JavaScript and tags in order to enable the SIMPATICO toolbar.

During the second phase of the SIMPATICO project, the Sheffield use-case has selected a further 4 more complex e-services to integrate the SIMPATICO tools to them.

- Apply for free school meals
- Request equipment or an adaptation to a property for a disabled person
- Calculate your contributions to the cost of social care
- Apply for parenting classes.

The Sandbox environment on Sparta's server showed the integration of the SIMPATICO tools with SCC website however for the second phase Sparta will be given access to a Sandbox environment on SCC's server to ensure full integration of the SIMPATICO tools is possible.



4 Conclusion

In this document we have presented what have been the activities carried out to integrate and deploy the SIMPATICO components to be ready for the second phase of experimentation. Besides, we have adopted, where feasible, the **12 principles of the new European Interoperability Framework**.

The definition of **Usability and Inclusive Design** was stated to make all the component compliant with this guideline. It will be taken into account during the final evaluation of platform (at M36).

In the end, we have developed a **Deployment Plan** for the second execution phase and updated the **integration use cases**, according to the new functionalities, provided by the SIMPATICO components for the final execution phase, to be validated during the pre-evaluation of the whole platform.

5 References

- [1] E. Commission, “European Interoperability Framework,” [Online]. Available: https://ec.europa.eu/isa2/eif_en. [Accessed 20 02 2018].
- [2] E. Commission, “European Interoperability Framework – Implementation Strategy,” [Online]. Available: <http://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:52017DC0134&from=EN>. [Accessed 20 02 2018].
- [3] E. Commission, “eIDAS Observatory,” [Online]. Available: <https://ec.europa.eu/futurium/en/content/eidas-regulation-regulation-eu-ndeg9102014>. [Accessed 20 02 2018].
- [4] E. Commission, “The EU General Data Protection Regulation,” [Online]. Available: <https://www.eugdpr.org/>. [Accessed 20 02 2018].
- [5] “Inclusive Design Principle,” [Online]. Available: <http://inclusivedesignprinciples.org/>. [Accessed 04 03 2018].
- [6] “Usability heuristic,” [Online]. Available: <https://www.nngroup.com/articles/ten-usability-heuristics/>. [Accessed 04 03 2018].
- [7] A. F. e. all., “SIMPATICO platform validation report v2,” SIMPATICO Project, 2019.
- [8] Docker, “Docker,” [Online]. Available: <https://www.docker.com/>. [Accessed 20 03 2018].