ICT-2011-288048

EURECA

Enabling information re-Use by linking clinical Research and CAre

IP Contract Nr: 288048

Deliverable: Report on evaluation and validation of EURECA components

Due date of deliverable: (30-05-2014)
Actual submission date: (13-08-2014)

Start date of Project: 01 February 2012 Duration: 42 months

Responsible WP: FORTH

Revision: <Final >

Project co-funded by the European Commission within the Seventh Framework Programme (2007-2013)		
Dissemination level		
PU	Public	**X**
PP	Restricted to other programme participants (including the Commission Service	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (excluding the Commission Services)	

0 DOCUMENT INFO

0.1 Author

Author	Company	E-mail
Lefteris Koumakis	FORTH	koumakis@ics.forth.gr
Haridimos Kondylakis	FORTH	kondylak@ics.forth.gr
Maria Psaraki	FORTH	psaraki@ics.forth.gr
Manolis Tsiknakis	FORTH	tsiknaki@ics.forth.gr
Jasper van Leeuwen	Philips	Jasper.van.leeuwen@philips.com
Kristof De Schepper	Custodix	kristof.deschepper@custodix.com
Kerstin Rohm	FhG IBMT	kerstin.rohm@ibmt.fraunhofer.de
Cyril Krykwinski	IJB	cyril.krykwinski@bordet.be
Monique Hendriks	Philips	monique.hendriks@philips.com
Sheng Yu	UOXF	sheng.yu@oncology.ox.ac.uk
Sergio Paraíso	UPM	sparaiso@infomed.dia.fi.upm.es
Salah Ait Mokhtar	Xerox	Salah.Ait-Mokhtar@xrce.xerox.com
Norbert Graf	USAAR	Norbert.Graf@uniklinikum-saarland.de
Magdalena Góralczyk	LUH	goralczyk@iri.uni-hannover.de
Vassilina Nikoulina	Xerox	vassilina.nikoulina@xrce.xerox.com
Zhisheng Huang	VUA	huang@cs.vu.nl
Jeroen Keijser	Philips	jeroen.keijser@philips.com
Timm Kißels	FhG IAIS	timm.kissels@iais.fraunhofer.de
David Pérez	UPM	dperez@infomed.dia.fi.upm.es
Scott Marshall	Maastro	m.scott.marshall@maastro.nl
Keyur Mehta	GBG	Keyur.Mehta@germanbreastgroup.de
Alan Dahi	LUH	dahi@iri.uni-hannover.de

0.2 Documents history

Document version #	Date	Change
V0.1	15/03/2014	Starting version, template
V0.2	22/03/2014	Definition of ToC
V0.3	24/07/2014	First complete draft
V0.4	25/07/2014	Updated version (send to project internal reviewers)
Sign off	04/08/2014	Signed off version (for approval to PMT members)
V1.0	13/08/2014	Approved Version to be submitted to EU

0.3 Document data

Keywords	

Editor Address data	Name: Lefteris Koumakis Partner: FORTH Address: N. Plastira 100 Vassilika Vouton Heraklion Phone: +30 2810 391424 Fax: +30 2810 391448 E-mail: koumakis@ics.forth.gr
Delivery date	

0.4 Distribution list

Date	Issue	E-mailer

Table of Contents

1 Introduction

For quality assurance, within the EURECA project, norms have been defined in accordance with the International Organization for Standardization[1] (ISO). Specifically the Software Product Quality Requirements and evaluation (SQUARE) has been used as a reference model, shown also in Figure 1. It describes the general processes and details the activities and tasks providing their purposes, inputs, outcomes and complementary information that can be used to guide a software product quality evaluation.

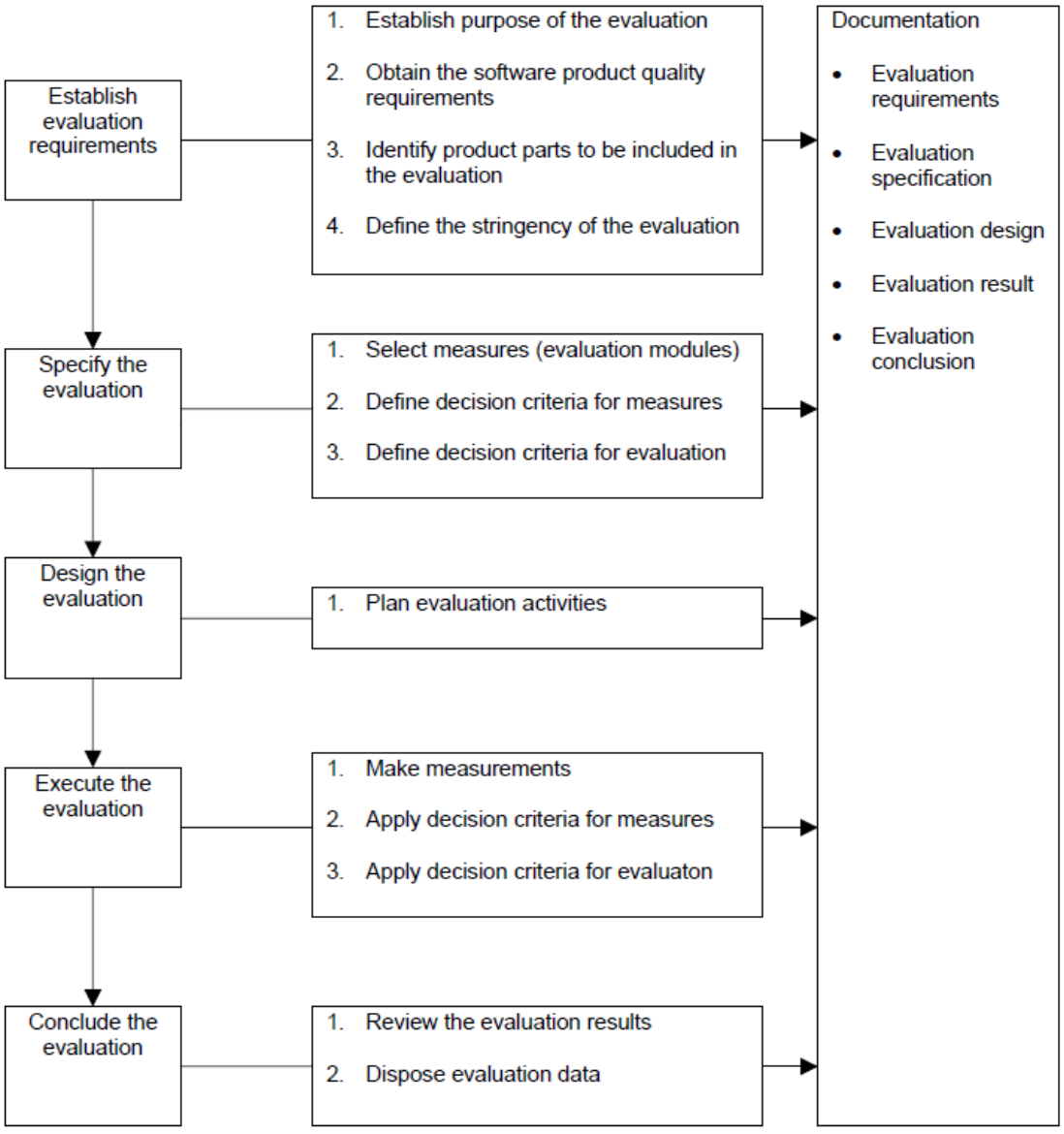


Figure 1: Software product Quality Evaluation Process reference model adapted from ISO/IEC 25040

[1] http://www.iso.org

The main blocks of the evaluation process as shown in Figure 1 are the “Establish evaluation requirements”, “Specify the Evaluation”, “Design the evaluation”, “Execute the evaluation” and “Conclude the evaluation”. Deliverable 8.1 introduced the first two blocks (“Establish evaluation requirements”, “Specify the Evaluation”) while Deliverable 8.2 delivered the third (“Design the evaluation”) and this document report the last two blocks (“Execute the evaluation” and “Conclude the evaluation”).

This is the first round of the evaluation for the EURECA components. In this deliverable the results of the evaluation and validation procedures are reported. The evaluation process is based on the ISO/IEC 25000 standard, so the relevant evaluation criteria are used. Under the rules of the ISO/IEC 25000 standard, as a result of the analysis of the evaluation activities this report covers the following for each EURECA component:

1. Deficiencies, any relevant analysis, and how each deficiency was resolved. Resolution of deficiencies may include the fact that:
    - one of the evaluation methods has provided assurance that the deficiency is not major;
    - a satisfactory “workaround” can be found to alleviate the impact of the deficiency; e.g., modification to the tool, disable or remove unneeded functionality, regenerate missing design requirements using reverse engineering;
    - the original requirement is not mandatory and the deficiency can be accepted;
    - the deficiency is acceptable provided that the use of the tool will be controlled by specific conditions or limitations;
    - additional evaluation work is required to resolve the deficiency or gaps in the evaluation;
2. Any additional evaluations performed to resolve any identified deficiencies:
    - to determine the scope or impact of a deficiency;
    - to establish confidence that there is no deficiency;
    - to verify that a workaround is technically feasible and/or suitable and acceptable;
    - to verify the correct and acceptable performance of the software once a design change or changes have been made to correct deficiency.
3. In a case where it is necessary to limit or control the use of the tool, whether the limitation:
    - interferes with the tool meeting the mandatory requirements of the application;
    - impacts on the application's design, budget, and schedule;
    - requires additional evaluation work;
    - introduces any possibility of failure in the application;
4. Any exclusions from scope of evaluation and/or restrictions on the results for each evaluation, such as:
    - this evaluation does not include a detailed review of the functionality of the tool
    - this tool is deemed to be qualified to the required integrity level provided a full evaluation of the required functionality for the product is completed successfully.
5. The integrated results of all the evaluation activities to allow an overall conclusion for the evaluation of the components to be made.

In EURECA, the guidance on the validation and evaluation recommends that activities must be conducted throughout the entire software life cycle. According to (ISO/IEC12207:2008) software life cycle processes define a common framework, with well-defined terminology, that can be referenced by the software industry and contains processes, activities, and tasks. Software evaluation is accomplished through a series of activities and tasks that are planned and executed at various stages of the software development life cycle.

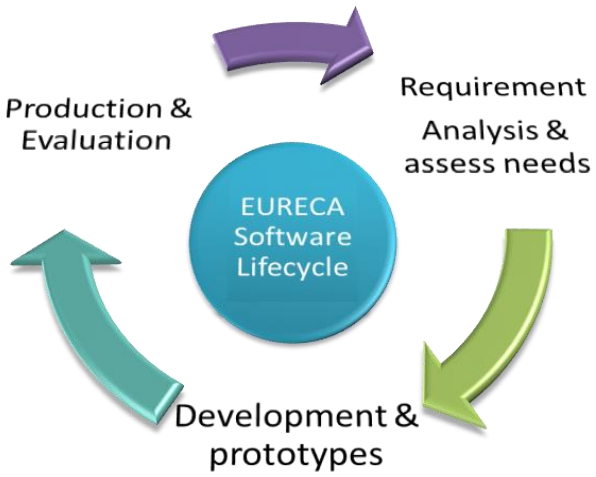


Figure 2: Software lifecycle in EURECA

The lifecycle of EURECA software development is an iterative procedure where we identify requirements and needs, develop and provide prototypes and evaluate the prototypes to assess again needs (if any) as shown in Figure 2.
The evaluation and validation procedure for the EURECA components is shown in Figure 3.

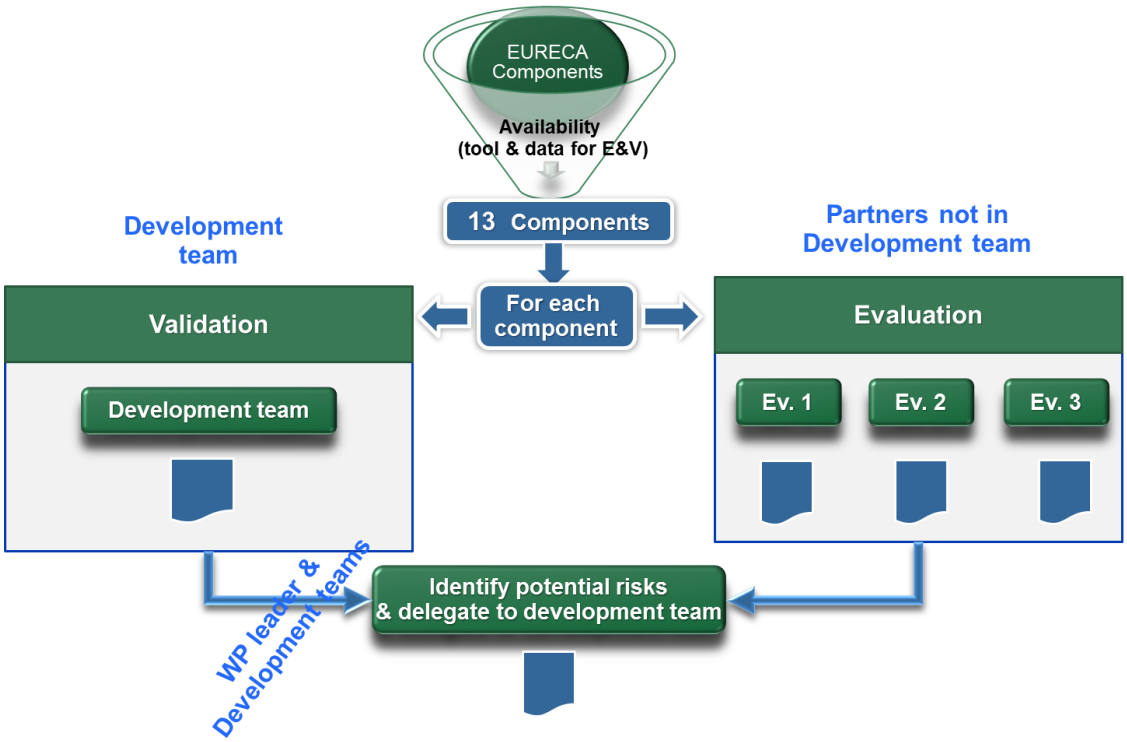


Figure 3: Evaluation and validation procedure for the EURECA components

1.1 Validation procedure

The validation procedure aims to assure that each tool will consistently produce (or not) the expected results. Developers performed the validation based on the validation templates (as described in Deliverable 8.2), that satisfy the user requirements.
Specifically, each developer of a tool has identified the initial user requirements. The validation procedure identify the specifications which conform or fail to meet user needs and intended uses. The validation results for the available EURECA components are reported in this document.

1.2 Evaluation procedure

The evaluation procedure of EURECA components checks if all the requirements, regulations and quality issues are met and if the tools were developed in a well-structured way. For each component 3 EURECA partners (independent from the developer's teams) evaluated the component using an evaluation scenario (provided by the developer) and the evaluation questionnaires, as shown in Figure 3. FORTH as the leader of WP8 evaluated all the components. We have to note also that FORTH is not technical leader (main developer) of any of the components evaluated in this iteration.
According to the ISO/IEC 25000 standard the evaluation results of the tools should:

- establish an appropriate degree of confidence that the tools are able to meet the evaluation requirements
- identify any specific deficiencies with regard to the evaluation requirements and any additional evaluations needed to determine the scope of those deficiencies
- identify any special limitations or conditions placed on the use of the tools
- identify any weaknesses or omissions in the evaluation itself and any additional evaluation that is needed

- identify any options for the use of the tools uncovered by the evaluation

End-user evaluation of the EURECA components is conducted through a number of selected scenarios covering the anticipated usage of each component. The evaluators filled in an evaluation form for each EURECA component. The evaluation forms cover all the appropriate quality characteristics from the product quality model of the ISO/IEC 25000 series and have been reported in D8.2.
At the evaluation phase different type of users, such as physicians and system developers participated. Having such a diverse target group of evaluators, the evaluation forms must be:

- simple
- accurate
- easy to understand (especially for non IT experts)
- non time consuming
- without loss of functionality/quality

For that reason the evaluation questionnaires include the crucial sub-characteristics of software quality measures into simple questions (in natural language). The evaluation form of EURECA is a list of such questions where the evaluator will answer with a degree of satisfaction with Likert scale. Likert scale is based on forced-choice questions, where a statement is made and the respondent then indicates the degree of agreement or disagreement with the statement on a 5 point scale. The Generic Evaluation Questionnaire consists of two forms:

1. The selected sub-characteristics, for the evaluation form of the EURECA scenarios and components, and its translation into a simple question for the end user (Form A).
2. We also use the System Usability Scale (SUS) for global assessment of systems usability (Form B).

SUS yields a single number representing a composite measure of the overall usability of the system being studied. Note that scores for individual items are not meaningful on their own. To calculate the SUS score we first sum the score contributions from each item. Each item's score contribution will range from 0 to 4. For questions with positive answers (specifically 1, 3, 5, 7, and 9) the score contribution is the scale position minus 1. For questions with negative answers (specifically 2,4,6,8 and 10), the contribution is 5 minus the scale position. Finally we multiply the sum of the scores by 2.5 to obtain the overall value of SU. SUS scores have a range of 0 to 100. Even though a SUS score can range from 0 to 100, it isn't a percentage. While it is technically correct that a SUS score of 70 out of 100 represents 70% of the possible maximum score, it suggests the score is at the 70th percentile. A score at this level would mean the application tested is above average. In fact, a score of 70 is closer to the average SUS score which is 68.
Section 2 reports the Forms A for all the EURECA components from the three evaluators and the final SUS score for each evaluator.
The results of the evaluation are important for supporting managerial decisions about next steps in the software development life cycle. For instance, do the requirements have to be changed or are more resources needed for the development process?

1.3 Examination of the results

Further processing of the result of the evaluation is an important step that enables us to be prepared for potential problems that can occur, and to develop a solution or plan of action for addressing those problems.
Hence, for each component, the validation and evaluation results should be examined in order to detect any issues of product quality. If the results of measurements deviate

from the expected results, then a mitigation plan, a contingency plan and a responsible person/team should be identified. The activities should not be forwarded to next stage before the resolution of the issues and the effective improvement of the product quality.

The rest of this deliverable is organized as follows: Section 2 presents the results from the evaluation and the validation of each component. For each component ideas for future development are subsequently described, or if there are flaws in the results then a risk assessment and the following actions required are presented. The deliverable concludes in Section 3 with an overview of the evaluation results.

2 Evaluation of EURECA components

2.1 Authorisation Service

2.1.1 Validation

Authorisation Service					
Category ID	**Measurement**	**Description**	**Quality measure elements**	**Measure Type**	**result**
A.1.1	Functional completeness	The system should create for each incoming AC request an AC decision based on the available policies	It will be checked that for each AC request the system generates an AC Decision	Number of decisions	100 different requests were sent to the authorization service, resulting in 100 decisions
A.1.2	Functional correctness	The system should return correct (this is, according to the policies) AC decisions based on the available policies	To measure this we need to compare precision and recall of decisions made by the PDP engine with the ones that are expected	Precision & recall	After evaluation of the previous measurement, the 100 messages that were sent, resulted in correct decisions (using an automated checker)
A1.3	Functional appropriateness	The authorization service should comply to the EURECA legal requirements	The system will be audited by legal experts	Audit outcome	This is not performed yet
A.3.2	Interoperability	PEP components should be able to send AC requests to the system and receive the AC decisions made by the system	A web service will be available to accept AC request and provide the AC decisions results to the PEP components	Web Service successful execution	A request to the authorization web service from the PEP, results in a decision

A.5.1	Maturity	The amount of faults that happen in the system during execution should be low	The number of faults while running the system will be recorded, if this number is too high corrective actions will be taken	Number of faults	After analyzing the logs of the authorization service in the EURECA development environment (23456 different incoming requests), none of the requests generated faults
A.5.2	Availability	The system should have a high availability as authorization is required in each EURECA service	The uptime of the system will be measured	Uptime of the system	After an uptime of 30 days and heavy usage, the authorization service in the EURECA development environment was still available
A.5.3	Fault tolerance	The system should keep running after malformed (syntactical, semantically) requests enter the system	A stress test will be developed, to check how the system reacts corrupt and malfunction requests	Stress test outcome	Semantically incorrect requests were generated and sent to the authorization service, this resulted in soap faults as expected, the service kept running after each fault
A.6.1	Confidentiality	Confidentiality of messages from/to the system should be guaranteed	Traffic to the system will be checked. Only HTTPS connections are valid, others are discarded.	HTTPS connections	Different Messages (10 requests) send to the authorization service, were sniffed using Wireshark. The request content of each request was not readable. Requests sent over HTTP are not executed on the authorization service

A.6.2	Integrity	The integrity of messages from/to the system should be guaranteed	Messages will be checked if they contain the required signing of body and header fields	Messages	A signed request and a non-signed request were sent to the authorization service, only the message with the signed body and header was accepted by the service, Next to this, both requests were manually checked if they contained the correct signing configuration
A.6.3	Non-repudiation	Messages send from/to the system can be proven to have taken place	It will be checked if each incoming/outgoing message is logged onto the system	Logging	100 request were sent to the authorization service, this resulted in 100 loggings recorded on the auditing service
A.6.4	Accountability	The sender of requests to the system should be traceable	SAML tokens will be checked if they contain the required identity information attributes of the sender	SAML tokens	A signed and non-signed request was sent to the authorization service, only the signed request was accepted by the authorization service (containing sender information), the other one was dropped
A.6.5	Authenticity	The identity of each sender can be proved to be the one claimed, for each request to the system	Requests to the system should always contain a valid signed SAML token, containing authentication information of the sender	Requests	A signed and non-signed request was sent to the authorization service, only the signed request was accepted by the authorization service (containing sender information), the other one was dropped
A.7.1	Modularity	New request handlers should be easily integrated/removed in the request handler pipeline	The authorization service will be tested with different configurations of handlers in the handler pipeline. It will be checked if these configurations	Configuration outcome	4 different configurations were tested. 10 different requests were send to each of these configuration, this resulted in 10 decision that gave the expected result

			work as expected.		
A.7.3	Analyzability	Failures and deficiencies in the system should be easy diagnosable	It will be checked if exceptions are well logged in the system	Logging	Semantically incorrect requests (100 requests) were sent to the authorization service, for each incorrect request, an exception was registered in the logging
A.7.5	Testability	A script will test the system frequently to check if everything is ok	automatic tests will be executed to test the system	Response to automatic tests	Each hour, 5 random requests were sent to the authorization service, 7 days long. Each request resulted in a correct decision.
A.8.3	Replaceability	The authorization service should be easily replaceable by another service based on XACML	It will be tested that the authorization service is fully compliant with XACML 3.0	Compliant test outcome	This cannot be tested at this moment

2.1.2 Evaluation

Questionnaires

Evaluator 1

FORTH

	FORM A	Rating (1 low, 5 high)				
	(Software quality characteristics)	1	2	3	4	5
Functionality	Can software perform the tasks required? i) accept AC requests ii) generate AC decisions		x			
	Is the result as expected? (Incoming AC request return correct AC decisions)		x			
	Is the system compliant with XACML (version 3.0)?					
Compatibility						
	Can the system share information/data with other Eureca components? Is the web service for remote invocation up & running? Does it provide results?			x		
Reliability	Have most of the faults in the software been eliminated over time?					
	Is the software capable of handling errors?					x
Security	Does the system provide identification access wherever is needed?					x
	Are data accessible only to authorized users?					x
	Can the system trace actions uniquely?					x
	Does the system prevent unauthorized access?					
Maintainability	Can faults be easily diagnosed?					x
	Can the software be easily modified?			x		
	Can the software continue functioning if changes are made?					
	Can the software be tested easily?		x			

Portability	Can the software easily replace other software?					
Quality in use	How accurate and complete is the software for the intended use?			x		
	Does the software improve the time or reduce resources for the intended goal?					
	Does the software satisfy the perceived achievements of pragmatic goals?			x		
	Can the software harm people in the intended contexts of use?					

System Usability Scale (SUS) score: 42.5

Evaluator 2
FhG IAIS

	FORM A	**Rating (1 low, 5 high)**					**Unknown***
	(Software quality characteristics)	1	2	3	4	5	
Functionality	Can software perform the tasks required? i) accept AC requests ii) generate AC decisions					x	
	Is the result as expected? (Incoming AC request return correct AC decisions)				x		
	Is the system compliant with XACML (version 3.0)?					x	
Compatibility							
	Can the system share information/data with other Eureca components? Is the web service for remote invocation up & running? Does it provide results?					x	
Reliability	Have most of the faults in the software been eliminated over time?			x			
	Is the software capable of handling errors?				x		
Security	Does the system provide identification access wherever is needed?					x	

* "Software as a service" and therefore not applicable or impossible to validate

	Are data accessible only to authorized users?					x	
	Can the system trace actions uniquely?						x
	Does the system prevent unauthorized access?					x	
Maintainability	Can faults be easily diagnosed?			x			
	Can the software be easily modified?						x
	Can the software continue functioning if changes are made?						x
	Can the software be tested easily?				x		
Portability	Can the software easily replace other software?			x			
Quality in use	How accurate and complete is the software for the intended use?				x		
	Does the software improve the time or reduce resources for the intended goal?				x		
	Does the software satisfy the perceived achievements of pragmatic goals?				x		
	Can the software harm people in the intended contexts of use?	x					

System Usability Scale (SUS) score: 52.5

Evaluator 3

Chg. IBMT

	FORM A	**Rating (1 low, 5 high)**					**Comments**
	(Software quality characteristics)	1	2	3	4	5	
Functionality	Can software perform the tasks required? i) accept AC requests ii) generate AC decisions					x	
	Is the result as expected? (Incoming AC request return correct AC decisions)					x	
	Is the system compliant with XACML (version 3.0)?					x	
Compatibility							
	Can the system share information/data with other Eureca components? Is the web service for remote invocation up & running? Does it provide results?					x	

Reliability	Have most of the faults in the software been eliminated over time?					x	
	Is the software capable of handling errors?					x	
Security	Does the system provide identification access wherever is needed?					x	
	Are data accessible only to authorized users?					x	
	Can the system trace actions uniquely?					x	
	Does the system prevent unauthorized access?					x	
Maintainability	Can faults be easily diagnosed?						
	Can the software be easily modified?						
	Can the software continue functioning if changes are made?						
	Can the software be tested easily?				x		as far as timestamp and the token are 100% valid
Portability	Can the software easily replace other software?						
Quality in use	How accurate and complete is the software for the intended use?					x	
	Does the software improve the time or reduce resources for the intended goal?						
	Does the software satisfy the perceived achievements of pragmatic goals?					x	
	Can the software harm people in the intended contexts of use?	x					the software can not harm any people

System Usability Scale (SUS) score: 85

Evaluation Comments

Evaluator 1

"In this web service the evaluation scenario could not be completed successfully.
The service provides a well-defined way to detect errors and to inform users with explanatory error messages, such as:

- "Exception in authorisation: Exception while parsing xacml response: Authorisation failed for :" (authorization issues)
- "Error reading XMLStreamReader." (wrong content of Security Token Service response)

However, the last error that returned the message: "The given SOAPAction http://www.custodix.com/schemas/authz/1.0/IAuthorisationSOAPWSEndpoint/evaluate does not match an operation." could not be overcome.
Moreover, the scenario requires a SAML security token that is supposed to be valid for 30 minutes. Actually it is valid for only 5 minutes, as after that time the following error message is returned: "The message has expired (WSSecurityEngine: Invalid timestamp The security semantics of the message have expired)". This is probably an issue of Security Token Service.
The steps of the evaluation scenario for the Authorisation Service have been repeated several times, entirely from the beginning, at different times, but each time the results were the same as mentioned above."

Evaluator 3

"We evaluated the Authorisation Service based on the Evaluation Scenario by the tool "SOAP UI". Within a valid username and password, a valid SAML token, and a valid (very strict) time stamp, we received a "Permit-response" as expected."

2.1.3 Conclusions

According to the results there are minor problems in functionality, compatibility and reliability. However no major actions or updates are planned for the authorisation service for the near future.

The major issues identified is the following:

"Could not complete the evaluation successfully. Error message: "The given SOAPAction http://www.custodix.com/schemas/authz/1.0/IAuthorisationSOAPWSEndpoint/evaluate does not match an operation.""

It seems that the cause is a wrong contract binding in the authorisation test client. The risk for further complications for this component is low and the test client will be investigated to identify the error.

2.2 Data Push Service (ETL)

2.2.1 Validation

Data Push Service					
Category ID	**Measurement**	**Description**	**Quality measure elements**	**Measure Type**	**Result**
A.1.1	Functional completeness	The service is able to store HL7 messages on the CDM	To measure this we need to search if the patient information sent to the Data Push Service is stored.	The number of matches	It depends on the input message. In case of informed consent message it is created an entity instance, an observation and a value for this observation. In case of more complex messages it is created an undetermined number of instances on the CDM
A.2.1	time behaviour	The system should respond in a timely manner	System 's response time will be measured	Response time	It depends on the input. Between 100-200 ms
A.2.2	resource utilisation	The system should not be resource intensive	The CPU and memory utilization will be measured in the pc where the service is executed	CPU & memory utilization	Between 1-4% of CPU and memory

A.3.2	Interoperability	The results of the system should be provided to other EURECA components (CDM and Mirth Connect) The service will accept messages in various interchange standard formats HL7 v2, HL7 v3 and HL7 IHE-based	A web service should be available to provide the matching results to other Eureca components To measure this we need to search if the patient information sent to the Data Push Service is stored.	Web Service successful execution	The web service accept HL7 messages of a given library. These results is provided to other EURECA components as SNAQL service and SAE application.
A.5.2	Availability	The system should be up & running almost always	The uptime of the system will be measured	Uptime of the system	The web service is always running where server is running. Last time it was restarted on 5 of June
A.7.1	Modularity	The service could uses different ETL tools as modules e.g Mirth Connect and Kettle	To measure this we need to search if the patient information sent to the Data Push Service is stored.	The number of matches	It depends on the input message. In case of informed consent message it is created an entity instance, an observation and a value for this observation. In case of more complex messages it is created an undetermined number of instances on the CDM
A.7.5	Testability	A script will test the system frequently to check if everything is ok	automatic tests will be executed to test the system	Response to automatic tests	Between 100-200ms per message. First message executes slower than the others.
A.8.2	Installability	The system should be easily installed by an IT-expert	The time to install the system in another machine will be tested	Time to install the system	6 hours. Necessary to install Mirth Connect and deploy a CDM
Comments:	Data Push is a service for storing HL7 messages on CDM. The main issue of this service is the lack of soap errors and it is necessary to add more HL7 template to generate more messages				

2.2.2 Evaluation

Questionnaires

Evaluator 1

FORTH

	FORM A	Rating (1 low, 5 high)				
	(Software quality characteristics)	1	2	3	4	5
Functionality	Can software perform the tasks required? i)store normalized information on CDM					x
	Is the result as expected?					x
	Can the system interact with another system? i) Eureca CDM and ii) ETL process					x
	Is the system compliant with standards? A)HL7 RIM					
Efficiency	How quickly does the system respond?					x
	Does the system utilize resources efficiently?					x
Compatibility	Can the system share resources without loss of its functionality?					x
	Can the system share information/data with other Eureca components? Is the web service for remote invocation up & running? Does it provide results?					x
Usability	Does the user comprehend how to use the system easily?				x	
	Can the user learn to use the system easily?					x
	Can the user use the system without much effort?					x
	Does the interface look good?	x				
Reliability	Have most of the faults in the software been eliminated over time?					
	Is the software capable of handling errors?					x
	Can the software resume working & restore lost data after failure?					
Security	Does the system provide identification access wherever is needed?					

	Are data accessible only to authorized users?					
	Can the system trace actions uniquely?					
	Does the system prevent unauthorized access?					
Maintainability	Can the software continue functioning if changes are made?					
	Can the software be tested easily?				x	
Portability	Can the software be moved to other environments?					x
	Can the software be installed easily?					x
	Can the software easily replace other software?				x	
Quality in use	How accurate and complete is the software for the intended use?					x
	Does the software improve the time or reduce resources for the intended goal?					
	Does the software satisfy the perceived achievements of pragmatic goals?					x
	Can the software harm people in the intended contexts of use?					

System Usability Scale (SUS) score: 85

Evaluator 2

FhG IAIS

	FORM A	**Rating (1 low, 5 high)**					**Unknown***
	(Software quality characteristics)	1	2	3	4	5	
Functionality	Can software perform the tasks required? i)store normalized information on CDM			x			
	Is the result as expected?			x			
	Can the system interact with another system? i) Eureca CDM and ii) ETL process						x
	Is the system compliant with standards? A)HL7 RIM			x			
Efficiency	How quickly does the system respond?					x	

* "Software as a service" and therefore not applicable or impossible to validate

	Does the system utilize resources efficiently?						x
Compatibility	Can the system share resources without loss of its functionality?						x
	Can the system share information/data with other Eureca components? Is the web service for remote invocation up & running? Does it provides results?			x			
Usability	Does the user comprehend how to use the system easily?				x		
	Can the user learn to use the system easily?			x			
	Can the user use the system without much effort?	x					
	Does the interface look good?			x			
Reliability	Have most of the faults in the software been eliminated over time?	x					
	Is the software capable of handling errors?	x					
	Can the software resume working & restore lost data after failure?						x
Security	Does the system provide identification access wherever is needed?						x
	Are data accessible only to authorized users?	x					
	Can the system trace actions uniquely?						x
	Does the system prevent unauthorized access?	x					
Maintainability	Can the software continue functioning if changes are made?						x
	Can the software be tested easily?			x			
Portability	Can the software be moved to other environments?						x
	Can the software be installed easily?						x
	Can the software easily replace other software?			x			

Quality in use	How accurate and complete is the software for the intended use?			x			
	Does the software improve the time or reduce resources for the intended goal?			x			
	Does the software satisfy the perceived achievements of pragmatic goals?			x			
	Can the software harm people in the intended contexts of use?	x					

System Usability Scale (SUS) score: 55

Evaluator 3

FhG IBMT

	FORM A	**Rating (1 low, 5 high)**					**Comments**
	(Software quality characteristics)	1	2	3	4	5	
Functionality	Can software perform the tasks required? i)store normalized information on CDM					x	
	Is the result as expected?					x	
	Can the system interact with another system? i) EURECA CDM and ii) ETL process						the ETL works properly, but the normalization is a manual process (until now)
	Is the system compliant with standards? A)HL7 RIM					x	
Efficiency	How quickly does the system respond?			x			
	Does the system utilize resources efficiently?						difficult to measure
Compatibility	Can the system share resources without loss of its functionality?						difficult to measure
	Can the system share information/data with other EURECA components? Is the web service for remote invocation up & running? Does it provides results?						difficult to measure
Usability	Does the user comprehend how to use the system easily?			x			

	Can the user learn to use the system easily?			x			
	Can the user use the system without much effort?			x			
	Does the interface look good?						The Web Service does not provide any GUI
Reliability	Have most of the faults in the software been eliminated over time?			x			
	Is the software capable of handling errors?	x					
	Can the software resume working & restore lost data after failure?						N/A
Security	Does the system provide identification access wherever is needed?						Security is provided by the Security layer
	Are data accessible only to authorized users?						
	Can the system trace actions uniquely?						
	Does the system prevent unauthorized access?						
Maintainability	Can the software continue functioning if changes are made?						no, new classes have to be deployed for the web service
	Can the software be tested easily?	x					no because no sufficient feedback is provided by the system
Portability	Can the software be moved to other environments?						not yet tested
	Can the software be installed easily?						not yet tested
	Can the software easily replace other software?						N/A

Quality in use	How accurate and complete is the software for the intended use?	x					currently there is manual step to be performed on the server side that makes the workflow incomplete
	Does the software improve the time or reduce resources for the intended goal?						N/A
	Does the software satisfy the perceived achievements of pragmatic goals?						N/A
	Can the software harm people in the intended contexts of use?	x					

System Usability Scale (SUS) score: 35

Evaluation Comments

Evaluator 1

"This is a web service for storing data on data warehouse. In general the web service runs smoothly without problems, in a reasonable time. Considering reliability, I cannot tell if the most of the faults in the software have been eliminated over time as I have not seen previous versions of the service. Moreover, there was not a web interface implemented, but since the web service is going to be used internally in other applications this is not a deficiency of the service."

Evaluator 3

"The DataPushService is a regular SOAP-based web service described by a WSDL file. Since there are only a very few methods provided by the service it is not so difficult to detect the functionalities of the service BUT one really has to indeed detect those functionalities because there is no proper documentation (or even in-line comments given) about them given. Also, the feedback given by the service is mediocre since no proper return values or exceptions are provided and therefore it is impossible at points to detect whether the data was pushed really pushed correctly or if an exception appears why this exceptional state occurred."

2.2.3 Conclusions

The Data Push Service deployed for the evaluation purpose is a test version based on the stable services deployed on the development and stage server. This means that the service uses a database for testing with realistic non-patient data. The other important difference with the real environment is that this service is not using the security layer. Because, this evaluation has to be focused on the functionalities of the Data Push service and the security layer is evaluated on their respective components. On the other hand, it is important to note that all the services also were deployed in HTTP secure (HTTPS) under an SSL/TLS protocol.

In Deliverable 4.4 “Initial prototype of the semantic interoperability framework” detailed documentation will be provided for better understanding of the solution. Examples will be included to facilitate the use of the framework. Regarding the error management; homogenization and error resolution are discussed for next versions.

2.3 Query Normalization Service

2.3.1 Validation

Query Normalization Service					
Category ID	**Measurement**	**Description**	**Quality measure elements**	**Measure Type**	**Results**
A.1.1	Functional completeness	The service is able to result a SPARQL template to retrieve core dataset information on CDM	To measure this we need to check if the results SPARQL template could be launched on CIM Access Service	Number of results	1 template
A.2.1	time behavior	The system should respond in a timely manner	System 's response time will be measured	Response time	Less than 100ms
A.2.2	resource utilization	The system should not be resource intensive	The CPU and memory utilization will be measured in the pc where the service is executed	CPU & memory utilization	Between 1-4% of CPU and memory
A.3.2	Interoperability	The results of the service should be provided to other EURECA components (Query Execution Service). Query Builder service will accept Core Dataset Concepts (SNOMED CT, LOINC,…etc.)	A web service should be available to provide query templates to other Eureca components	Web Service successful execution	Templates are used by SNAQL component to execute queries on Query Execution Service

A.3.2	Interoperability: Access Core Dataset Service for normalizing concepts	In order to get a template we have to retrieve terminology binding information on the Core Dataset Service	Queries should be executed to Core Dataset Service	Query responses	getUncontextualized uses getNormalForm of a given concept for resulting the corresponding templates
A.5.2	Availability	The system should be up & running almost always	The uptime of the system will be measured	Uptime of the system	The web service is always running where server is running. Last time it was restarted on 5 of June
A.7.5	Testability	A script will test the system frequently to check if everything is ok	automatic tests will be executed to test the system	Response to automatic tests	50 concepts were send to obtain the corresponding templates. Resulting in 3463 ms.
A.8.2	Installability	The system should be easily installed by an IT-expert	The time to install the system in another machine will be tested	Time to install the system	1 hour. Necessary to deploy Core Dataset Service first
Comments:	Query Normalization service returns the corresponding template for a given core dataset information. It is necessary to define a query template library for retrieving information for the concepts. The service works fast and it is integrated with security layer. Templates could be updated easily for better performance or changes in the CDM				

2.3.2 Evaluation

Questionnaires

Evaluator 1

FORTH

	FORM A	Rating (1 low, 5 high)				
	(Software quality characteristics)	1	2	3	4	5
Functionality	Can software perform the tasks required? i) return a correct SPARQL template					x
	Is the result as expected?					x
	Can the system interact with another system? i) Core Dataset Service					x
	Is the system compliant with standards?					
Efficiency	How quickly does the system respond?					x
	Does the system utilize resources efficiently?					x
Compatibility	Can the system share resources without loss of its functionality?					x
	Can the system share information/data with other Eureca components? Can que query be executed on CIM/CDM Access Service					x
Usability	Does the user comprehend how to use the system easily?				x	
	Can the user learn to use the system easily?					x
	Can the user use the system without much effort?					x
	Does the interface look good?	x				
Reliability	Have most of the faults in the software been eliminated over time?					
	Is the software capable of handling errors?					x
	Can the software resume working & restore lost data after failure?					
Security	Does the system provide identification access wherever is needed? Security access? SSL?					

	Are data accessible only to authorized users?					
	Can the system trace actions uniquely?					
	Does the system prevent unauthorized access?					
Maintainability	Can faults be easily diagnosed?					x
	Can the software continue functioning if changes are made?					
	Can the software be tested easily?					x
Portability	Can the software be moved to other environments?					x
	Can the software be installed easily?					x
	Does the software comply with portability standards?					x
	Can the software easily replace other software?				x	
Quality in use	How accurate and complete is the software for the intended use?					x
	Does the software improve the time or reduce resources for the intended goal?					
	Does the software satisfy the perceived achievements of pragmatic goals?					x
	Can the software harm people in the intended contexts of use?					

System Usability Scale (SUS) score: 85

Evaluator 2

Custodix

	FORM A	**Rating (1 low, 5 high)**				
	(Software quality characteristics)	1	2	3	4	5
Functionality	Can software perform the tasks required? i) return a correct SPARQL template					x
	Is the result as expected?					x
	Can the system interact with another system? i) Core Dataset Service				x	
	Is the system compliant with standards?				x	
Efficiency	How quickly does the system respond?				x	

	Does the system utilize resources efficiently?			x		
Compatibility	Can the system share resources without loss of its functionality?			x		
	Can the system share information/data with other Eureca components? Can query be executed on CIM/CDM Access Service				x	
Usability	Does the user comprehend how to use the system easily?				x	
	Can the user learn to use the system easily?				x	
	Can the user use the system without much effort?				x	
	Does the interface look good?			x		
Reliability	Have most of the faults in the software been eliminated over time?				x	
	Is the software capable of handling errors?		x			
	Can the software resume working & restore lost data after failure?				x	
Security	Does the system provide identification access wherever is needed? Security access? SSL?			x		
	Are data accessible only to authorized users?			x		
	Can the system trace actions uniquely?			x		
	Does the system prevent unauthorized access?			x		
Maintainability	Can faults be easily diagnosed?		x			
	Can the software continue functioning if changes are made?				x	
	Can the software be tested easily?				x	
Portability	Can the software be moved to other environments?				x	
	Can the software be installed easily?				x	
	Does the software comply with portability standards?			x		
	Can the software easily replace other software?			x		
Quality in	How accurate and complete is the software for the intended use?				x	

	Does the software improve the time or reduce resources for the intended goal?				x	
	Does the software satisfy the perceived achievements of pragmatic goals?				x	
	Can the software harm people in the intended contexts of use?				x	

System Usability Scale (SUS) score: 70

Evaluator 3

FhG IBMT

	FORM A	**Rating (1 low, 5 high)**					**Comments**
	(Software quality characteristics)	1	2	3	4	5	
Functionality	Can software perform the tasks required? i) return a correct SPARQL template				x		CDA, Core Dataset (SNOMED, LOINC, ..) knowledge is needed in order to build a valid query
	Is the result as expected?			x			
	Can the system interact with another system? i) Core Dataset Service						not possible to evaluate from our side
	Is the system compliant with standards?					x	
Efficiency	How quickly does the system respond?					x	
	Does the system utilize resources efficiently?					x	
Compatibility	Can the system share resources without loss of its functionality?						
	Can the system share information/data with other EURECA components? Can the query be executed on CIM/CDM Access Service						

Usability	Does the user comprehend how to use the system easily?		x				The usage of Query Builder Service requires knowledge in programming: WebService Client, Axis2; CDA and Core Dataset (SNOMED, LOINC, ..)
	Can the user learn to use the system easily?		x				
	Can the user use the system without much effort?		x				
	Does the interface look good?						There is no user interface available until now
Reliability	Have most of the faults in the software been eliminated over time?						we did not recognized any faults
	Is the software capable of handling errors?						we did not receive any error messages
	Can the software resume working & restore lost data after failure?	x					not needed for this service
Security	Does the system provide identification access wherever is needed? Security access? SSL?						security is not needed for this service
	Are data accessible only to authorized users?						security is not needed for this service
	Can the system trace actions uniquely?					x	
	Does the system prevent unauthorized access?						security is not needed for this service
Maintainability	Can faults be easily diagnosed?						we did not receive any error messages
	Can the software continue functioning if changes are made?						not needed for this service

	Can the software be tested easily?				x		The usage of Query Builder Service requires knowledge in programming: WebService Client, Axis2; CDA and Core Dataset (SNOMED, LOINC, ..)
Portability	Can the software be moved to other environments?	x					The Query Builder is part of the EURECA Interoperability platform
	Can the software be installed easily?						
	Does the software comply with portability standards?					x	
	Can the software easily replace other software?	x					
Quality in use	How accurate and complete is the software for the intended use?			x			CDA / HL7 v3 knowledge is needed
	Does the software improve the time or reduce resources for the intended goal?		x				
	Does the software satisfy the perceived achievements of pragmatic goals?			x			
	Can the software harm people in the intended contexts of use?	x					

System Usability Scale (SUS) score: 47.5

Evaluation Comments

Evaluator 1

“This is a web service based on SOAP that generates SPARQL queries based on the CDM. In general the web service runs smoothly without problems, in a reasonable time.
Considering reliability, I cannot tell if the most of the faults in the software have been eliminated over time as I have not seen previous versions of the service. Moreover, there was not a web interface implemented, but since the web service is going to be used internally in other applications this is not a deficiency of the service.”

Evaluator 2

"The query normalisation service returns templates for creating SPARQL queries that can be executed on the query execution service. The web service interface definition is straightforward. A user can request contextualised and uncontextualised templates. With respect to the execution time of the web service it is quick and responsive. The functionality works as expected, but is a bit complex for an unexperienced user. Exception handling should be improved, as for the moment no SOAP Faults are thrown (error message is included in the return message). This validation was performed without using the security proxy, so no evaluation was made for security."

Evaluator 3

"The Query Builder of the EURECA Interoperability platform can be used in order to build EURECA CDM conform SPARQL Queries. It is a regular SOAP-based web service described by a WSDL file. The WebService Client has to be generated and programmed manually.
The return values of the corresponding WebService function (parameter: Core Dataset concept) should be used for the building of a valid SPARQL query to the EURECA CDM. The return value consists of a "common" SPARQL query with several "optional" parameters.
It took us some time to learn how to build simple queries based on these return values, not least because there is no detailed user guide. But we were not able to generate more complex queries without personal support from UPM."

2.3.3 Conclusions

The Query Normalization Service for the evaluation purpose is a test version based on the stable services deployed on the development and stage server. The same restrictions/limitations as for Data Push Service apply here too. This means that the service uses a database for testing with realistic non-patient data. In Deliverable 4.4 "Initial prototype of the semantic interoperability framework" detailed documentation will be provided for better understanding of the solution. Examples will be included to facilitate the use of the framework. Regarding the error management; homogenization and error resolution are discussed for next versions.

2.4 Query Execution Service

2.4.1 Validation

Query Execution Service					
Category ID	**Measurement**	**Description**	**Quality measure elements**	**Measure Type**	**Results**
A.1.1	Functional completeness	The service is able to retrieve information of CDM	To measure this we need to search if data retrieved is the correct	The number of matches	It depends on the query and the dataset. If the query search for the total of patients for example, it will return 4673 results on GBG dataset
A.2.1	time behavior	The system should respond in a timely manner	System 's response time will be measured	Response time	343 ms
A.2.2	resource utilization	The system should not be resource intensive	The CPU and memory utilization will be measured in the pc where the service is executed	CPU & memory utilization	Between 1-4% of CPU and memory
A.3.2	Interoperability	The results of the system should be provided to other EURECA components. Query Execution Service will accept queries in SPARQL format.	A web service should be available to provide the SPARQL results to other Eureca components	Web Service successful execution	The SPARQL results is used on Patient Management service or SAE

A.3.2	Interoperability: Access Core Dataset Service for expanding query with hierarchical concepts	In order to retrieve information on CDM we can use Core Dataset Service to expand concepts of the given query	Core Datasets Concepts are added to the original query	Query is ok	Core dataset service expand all the core dataset concepts of the original query in less than 100ms
A.5.2	Availability	The system should be up & running almost always	The uptime of the system will be measured	Uptime of the system	The web service is always running where server is running. Last time it was restarted on 5 of June
A.7.1	Modularity	The service could use other modules instead of other internal modules such as D2R, MORPH, CDM	To measure this we need to search if data retrieved is the correct	The number of matches	It is independent from the database management system, CDM and from the SPARQL-SQL engine used (MORPH or D2R)
A.7.5	Testability	A script will test the system frequently to check if everything is ok	automatic tests will be executed to test the system	Response to automatic tests	A list of queries for TBP criteria were tested in 4973ms.
A.8.2	Installability	The system should be easily installed by an IT-expert	The time to install the system in another machine will be tested	Time to install the system	3 hours. Necessary to deploy a CDM and Core Dataset Service first
Comments :	Query Execution is the service responsible of retrieving data from CDM. It is possible to execute SPARQL queries obtained from Query Normalization Service. The main issue is the query response when it is retrieving a large resultset.				

2.4.2 Evaluation

Questionnaires

Evaluator 1

FORTH

	FORM A	Rating (1 low, 5 high)				
	(Software quality characteristics)	1	2	3	4	5
Functionality	Can software perform the tasks required? The queries are correctly executed on CDM?					x
	Is the result as expected?					x
	Can the system interact with another system? EURECA CDM and Core Dataset Service with query expansion method					x
	Is the system compliant with standards?					
Efficiency	How quickly does the system respond?					x
	Does the system utilize resources efficiently?					x
Compatibility	Can the system share resources without loss of its functionality?					x
	Can the system share information/data with other Eureca components? Is the web service for remote invocation up & running? Does it provides results? Core Dataset Service?					x
Usability	Does the user comprehend how to use the system easily?				x	
	Can the user learn to use the system easily?					x
	Can the user use the system without much effort?				x	
	Does the interface look good?	x				
Reliability	Have most of the faults in the software been eliminated over time?					
	Is the software capable of handling errors?				x	
	Can the software resume working & restore lost data after failure?					

Security	Does the system provide identification access wherever is needed? Security access? SSL?					
	Are data accessible only to authorized users?					
	Can the system trace actions uniquely?					
	Does the system prevent unauthorized access?					
Maintainability	Can faults be easily diagnosed?				x	
	Can the software be easily modified?					x
	Can the software continue functioning if changes are made?					
	Can the software be tested easily?					x
Portability	Can the software be moved to other environments?					x
	Can the software be installed easily?					x
	Does the software comply with portability standards?					x
	Can the software easily replace other software?				x	
Quality in use	How accurate and complete is the software for the intended use?					x
	Does the software improve the time or reduce resources for the intended goal?					
	Does the software satisfy the perceived achievements of pragmatic goals?					x
	Can the software harm people in the intended contexts of use?					

System Usability Scale (SUS) score: 85

Evaluator 2

UOXF

	FORM A	**Rating (1 low, 5 high)**				
	(Software quality characteristics)	1	2	3	4	5
Functionality	Can software perform the tasks required? The queries are correctly executed on CDM?					x
	Is the result as expected?				x	
	Can the system interact with another system? EURECA CDM and Core Dataset Service with query expansion method					x
	Is the system compliant with standards?				x	

Efficiency	How quickly does the system respond?					x
	Does the system utilize resources efficiently?					x
Compatibility	Can the system share resources without loss of its functionality?				x	
	Can the system share information/data with other Eureca components? Is the web service for remote invocation up & running? Does it provides results? Core Dataset Service?					x
Usability	Does the user comprehend how to use the system easily?				x	
	Can the user learn to use the system easily?				x	
	Can the user use the system without much effort?				x	
	Does the interface look good?				x	
Reliability	Have most of the faults in the software been eliminated over time?			x		
	Is the software capable of handling errors?				x	
	Can the software resume working & restore lost data after failure?			x		
Security	Does the system provide identification access wherever is needed? Security access? SSL?	x				
	Are data accessible only to authorized users?	x				
	Can the system trace actions uniquely?	x				
	Does the system prevent unauthorized access?	x				
Maintainability	Can faults be easily diagnosed?				x	
	Can the software be easily modified?					x
	Can the software continue functioning if changes are made?					
	Can the software be tested easily?					x
Portability	Can the software be moved to other environments?					x
	Can the software be installed easily?				x	
	Does the software comply with portability standards?					x
	Can the software easily replace other software?					x

Quality in use	How accurate and complete is the software for the intended use?			x		
	Does the software improve the time or reduce resources for the intended goal?					x
	Does the software satisfy the perceived achievements of pragmatic goals?					x
	Can the software harm people in the intended contexts of use?				x	

System Usability Scale (SUS) score: 67.5

Evaluator 3

Philips

	FORM A	**Rating (1 low, 5 high)**				
	(Software quality characteristics)	1	2	3	4	5
Functionality	Can software perform the tasks required? The queries are correctly executed on CDM?				x	
	Is the result as expected?			x		
	Can the system interact with another system? EURECA CDM and Core Dataset Service with query expansion method				x	
	Is the system compliant with standards?				x	
Efficiency	How quickly does the system respond?				x	
	Does the system utilize resources efficiently?				x	
Compatibility	Can the system share resources without loss of its functionality?				x	
	Can the system share information/data with other Eureca components? Is the web service for remote invocation up & running? Does it provides results? Core Dataset Service?				x	
Usability	Does the user comprehend how to use the system easily?		x			
	Can the user learn to use the system easily?			x		
	Can the user use the system without much effort?			x		
	Does the interface look good?			x		
Reliability	Have most of the faults in the software been eliminated over time?		x			

	Is the software capable of handling errors?		x			
	Can the software resume working & restore lost data after failure?					?
Security	Does the system provide identification access wherever is needed? Security access? SSL?			x		
	Are data accessible only to authorized users?			x		
	Can the system trace actions uniquely?			?		
	Does the system prevent unauthorized access?			x		
Maintainability	Can faults be easily diagnosed?			?		
	Can the software be easily modified?			?		
	Can the software continue functioning if changes are made?		x			
	Can the software be tested easily?			x		
Portability	Can the software be moved to other environments?			?		
	Can the software be installed easily?			x		
	Does the software comply with portability standards?			?		
	Can the software easily replace other software?			?		
Quality in use	How accurate and complete is the software for the intended use?			x		
	Does the software improve the time or reduce resources for the intended goal?			x		
	Does the software satisfy the perceived achievements of pragmatic goals?			?		
	Can the software harm people in the intended contexts of use?	x				

System Usability Scale (SUS) score: 47.5

Evaluation Comments

Evaluator 1

“This is a web service based on SOAP that execute CIM or CDM-based queries on the data warehouses. In general the web service runs smoothly without problems, in a reasonable time. Considering reliability, I cannot tell if the most of the faults in the software have been eliminated over time as I have not seen previous versions of the service. Moreover, there was not a web interface implemented, but since the web service is going to be used internally in other applications this is not a deficiency of the service.”

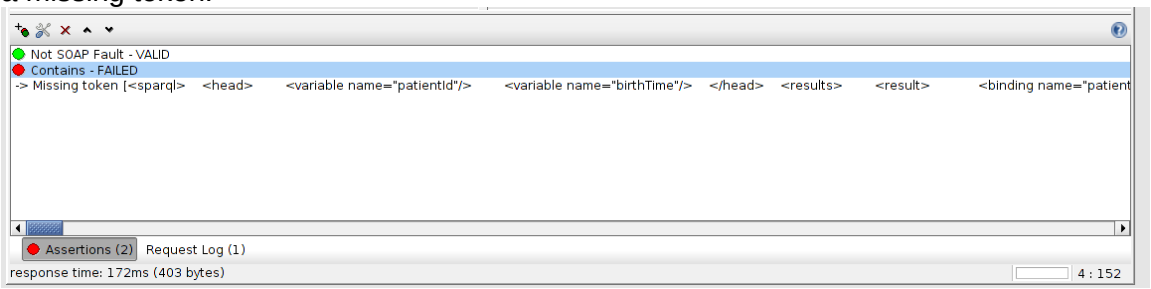
Evaluator 2

"This web service is running to accept SPARQL queries that are wrapped up in a SOAP message to retrieve clinical information according to the CDM schema. After running the tests that were provided for this web service we found the service is very responsive and efficient.
There are a few suggestions and comments about this service:
1. When running the first query to retrieve all patient information:

```
SELECT DISTINCT?patientId?birthTime WHERE {
                    ?livSubj  a hl7rim:livingSubject;
                              hl7rim:livingSubject_id?patientId;
                              hl7rim:livingSubject_birthTime?birthTime.
          }
```

The service returns an error:
<?xml version="1.0" encoding="UTF-8"?><sparql></head/><results/><!--[ERROR]java.lang.NullPointerException --></sparql>

2. When using the suggested testing toolkit soapUI, the output complains that there is a missing token:



However we have tested using a Python WSDL utility script and there was no error.

3. Assistance is needed when creating the SPARQL query that is to be sent to the service. For instance, in the following query the class code of the HL7 Act is a subclass of 'DIAG', however the code attribute of the act, which is a SNOMED-CT concept should match the class code, in other words compatible SNOMED-CT concepts such as *82711006 - Infiltrating duct carcinoma* as diagnosis concepts, but not *104846005 - Oxygen measurement (procedure).* In the current system there is no mechanism to give user any warning. It might be addressed by the Query Builder web service."

```
SELECT DISTINCT?id?code?patientId?birthTime?effectiveTime
                WHERE {
                    ?instPerson                 hl7rim:person_id?patientId.
                    ?instPerson                 hl7rim:person_code '337915000'.
                     OPTIONAL{?instPerson   hl7rim:person_birthTime?birthTime}
                    ?instPerson             hl7rim:person_role?instRole2.
                    ?instRole2                  hl7rim:role_entityId?patientId.
                    ?instRole2                  hl7rim:role_participation?instPart2.
```

```
        ?instPart2          hl7rim:participation_entityId?patientId.
        ?instPart2          hl7rim:participation_act?instAct.
        ?instAct            hl7rim:act_code?code;
                              hl7rim:act_subClassCode 'DIAG';
                              hl7rim:act_id  ?id.
        OPTIONAL {?instAct    hl7rim:act_effectiveTime?effectiveTime}
        FILTER (?code IN (isAnySubclassOf(82711006)))
}
```

2.4.3 Conclusions

The Query Execution Service deployed for the evaluation purpose is a test version based on the stable services deployed on the development and stage server. Same policy as for Data Push Service and Query Normalization Service apply here too. The service uses a database for testing with realistic non-patient data and detailed documentation for the fully functional component will be provided in the Deliverable 4.4 “Initial prototype of the semantic interoperability framework”.

2.5 Core Dataset Service

2.5.1 Validation

Core Dataset Service					
Category ID	**Measurement**	**Description**	**Quality measure elements**	**Measure Type**	**Results**
A.1.1	Functional completeness	The service is able to results related concepts of a given one and the linking on HL7 RIM of a given concept	To measure this we need to search if the results are correct and coherent	The number of matches	It is possible to obtain the unique normal form for a given concept and more information related to that one.
A.2.1	time behavior	The system should respond in a timely manner	System 's response time will be measured	Response time	Worst case: 1017 ms Best case: 86 ms
A.2.2	resource utilization	The system should not be resource intensive	The CPU and memory utilization will be measured in the pc where the service is executed	CPU & memory utilization	It is stored on memory, so it is needed 30-40% of CPU and memory on the deployment phase.
A.3.2	Interoperability	The results of the system should be provided to other EURECA components (Terminology Binding and vocabularies). Core Dataset Service stores all the information related with the medical vocabularies as SNOMED CT, LOINC, HGNC…etc.	A web service should be available to provide the medical vocabularies information to other Eureca components as CIM/CDM Access Service or Query Builder Service.	Web Service successful execution	This service is used within Query Execution Service, Query Normalization Service and Auto Complete Service.

A.5.2	Availability	The system should be up & running almost always	The uptime of the system will be measured	Uptime of the system	The web service is always running where server is running. Last time it was restarted on 5 of June
A.7.1	Modularity	The modularity of this service appears on the different modules that compounds the service, such as terminology binding, sesame repository and normal form service	To measure this we need to search if the results are correct and coherent	The number of matches	It is possible to change the version of the different ontologies used and semantic repository easily
A.7.5	Testability	A script will test the system frequently to check if everything is ok	automatic tests will be executed to test the system	Response to automatic tests	For a small test of 50 concepts; 4650 ms
A.8.2	Installability	The system should be easily installed by an IT-expert	The time to install the system in another machine will be tested	Time to install the system	2 hours
Comments:	Core Dataset service is the central medical language repository and related knowledge on the platform. Main issue is CPU and memory usage on the deployment. It is deployed on less than 4 minutes.				

2.5.2 Evaluation

Questionnaires

Evaluator 1

FORTH

	FORM A	Rating (1 low, 5 high)				
	(Software quality characteristics)	1	2	3	4	5
Functionality	Can software perform the tasks required? Expand SNOMED Concepts? The mapping on CDM of the given concept is correct?					x
	Is the result as expected?					x
	Can the system interact with another system? Terminology binding and normalized form?					x
	Is the system compliant with standards?					
Efficiency	How quickly does the system respond?					x
	Does the system utilize resources efficiently?					x
Compatibility	Can the system share resources without loss of its functionality?					x
	Can the system share information/data with other Eureca components? Is the web service for remote invocation up & running? Does it provides results?					x
Usability	Does the user comprehend how to use the system easily?				x	
	Can the user learn to use the system easily?					x
	Can the user use the system without much effort?					x
	Does the interface look good?	x				
Reliability	Have most of the faults in the software been eliminated over time?					
	Is the software capable of handling errors?					x
	Can the software resume working & restore lost data after failure?					

Security	Does the system provide identification access wherever is needed? Security access? SSL?					
	Are data accessible only to authorized users?					
	Can the system trace actions uniquely?					
	Does the system prevent unauthorized access?					
Maintainability	Can faults be easily diagnosed?					x
	Can the software be easily modified?					
	Can the software continue functioning if changes are made?					
	Can the software be tested easily?					x
Portability	Can the software be moved to other environments?					x
	Can the software be installed easily?					x
	Does the software comply with portability standards?					x
	Can the software easily replace other software?				x	
Quality in use	How accurate and complete is the software for the intended use?					x
	Does the software improve the time or reduce resources for the intended goal?					
	Does the software satisfy the perceived achievements of pragmatic goals?					x
	Can the software harm people in the intended contexts of use?					

System Usability Scale (SUS) score: 85

Evaluator 2

Custodix

	FORM A	**Rating (1 low, 5 high)**				
	(Software quality characteristics)	1	2	3	4	5
Functionality	Can software perform the tasks required? Expand SNOMED Concepts? The mapping on CDM of the given concept is correct?					x
	Is the result as expected?					x
	Can the system interact with another system? Terminology binding and normalized form?				x	
	Is the system compliant with standards?				x	

Efficiency	How quickly does the system respond?				x	
	Does the system utilize resources efficiently?			x		
Compatibility	Can the system share resources without loss of its functionality?				x	
	Can the system share information/data with other Eureca components? Is the web service for remote invocation up & running? Does it provides results?				x	
Usability	Does the user comprehend how to use the system easily?			x		
	Can the user learn to use the system easily?			x		
	Can the user use the system without much effort?			x		
	Does the interface look good?			x		
Reliability	Have most of the faults in the software been eliminated over time?				x	
	Is the software capable of handling errors?		x			
	Can the software resume working & restore lost data after failure?				x	
Security	Does the system provide identification access wherever is needed? Security access? SSL?			x		
	Are data accessible only to authorized users?			x		
	Can the system trace actions uniquely?			x		
	Does the system prevent unauthorized access?			x		
Maintainability	Can faults be easily diagnosed?				x	
	Can the software be easily modified?			x		
	Can the software continue functioning if changes are made?			x		
	Can the software be tested easily?				x	
Portability	Can the software be moved to other environments?			x		
	Can the software be installed easily?			x		
	Does the software comply with portability standards?			x		
	Can the software easily replace other software?				x	

Quality in use	How accurate and complete is the software for the intended use?				x	
	Does the software improve the time or reduce resources for the intended goal?				x	
	Does the software satisfy the perceived achievements of pragmatic goals?				x	
	Can the software harm people in the intended contexts of use?			x		

System Usability Scale (SUS) score: 57.5

Evaluator 3

FhG IAIS

	FORM A	**Rating (1 low, 5 high)**					**Unknown***
	(Software quality characteristics)	1	2	3	4	5	
Functionality	Can software perform the tasks required? Expand SNOMED Concepts? The mapping on CDM of the given concept is correct?			x			
	Is the result as expected?			x			
	Can the system interact with another system? Terminology binding and normalized form?						x
	Is the system compliant with standards?			x			
Efficiency	How quickly does the system respond?				x		
	Does the system utilize resources efficiently?						x
Compatibility	Can the system share resources without loss of its functionality?						x
	Can the system share information/data with other Eureca components? Is the web service for remote invocation up & running? Does it provides results?			x			
Usability	Does the user comprehend how to use the system easily?			x			
	Can the user learn to use the system easily?		x				
	Can the user use the system without much effort?		x				
	Does the interface look good?				x		

* "Software as a service" and therefore not applicable or impossible to validate

Reliability	Have most of the faults in the software been eliminated over time?			x			
	Is the software capable of handling errors?				x		
	Can the software resume working & restore lost data after failure?						x
Security	Does the system provide identification access wherever is needed? Security access? SSL?						x
	Are data accessible only to authorized users?	x					
	Can the system trace actions uniquely?						x
	Does the system prevent unauthorized access?	x					
Maintainability	Can faults be easily diagnosed?				x		
	Can the software be easily modified?						x
	Can the software continue functioning if changes are made?						x
	Can the software be tested easily?			x			
Portability	Can the software be moved to other environments?						x
	Can the software be installed easily?						x
	Does the software comply with portability standards?						x
	Can the software easily replace other software?			x			
Quality in use	How accurate and complete is the software for the intended use?				x		
	Does the software improve the time or reduce resources for the intended goal?			x			
	Does the software satisfy the perceived achievements of pragmatic goals?			x			
	Can the software harm people in the intended contexts of use?			x			

System Usability Scale (SUS) score: 72.5

Evaluation Comments

Evaluator 1

"Core Dataset service is responsible of inferring vocabulary knowledge on the different components of the platform. It has several methods, below are some remarks on some of them:

- expandQuery, getNextGen : they do not return a fault message (or a did not find message) if the input is wrong.

In general the web service runs smoothly without problems, in a reasonable time. Considering reliability, I cannot tell if the most of the faults in the software have been eliminated over time as I have not seen previous versions of the service. Moreover, there was not a web interface implemented, but since the web service is going to be used internally in other applications this is not a deficiency of the service."

Evaluator 2

"The core dataset service provides different methods to retrieve concepts and relationships from the core vocabularies of the EURECA project. The web service interface definition is straightforward. The execution time of the web service is quick and responsive (if no root concept is requested). The functionality works as expected, but is a bit complex for an unexperienced user. Exception handling should be improved, as for the moment no SOAP Faults are thrown (error message is included in the return message). This validation was performed without using the security proxy, so no evaluation was made for security."

2.5.3 Conclusions

The Core Dataset Service deployed for the evaluation purpose is a test version based on the stable services deployed on the development and stage server. In Deliverable 4.4 "Initial prototype of the semantic interoperability framework" detailed documentation will be provided for better understanding of the solution. Examples will be included to facilitate the use of the framework. Regarding the error management; homogenization and error resolution are discussed for next versions.

2.6 Trial Management Service

2.6.1 Validation

Trial Management Service					
Category ID	**Measurement**	**Description**	**Quality Measure Elements**	**Measure Type**	**Measurement and Discussion**
A.1.1	Functional completeness	All required trial metadata information can be stored and accessed in the trial management service	We will check that each required trial metadata element (e.g. coming from the use cases) can be managed within the service	Functionality checklist	The unit tests described below cover all required trial metadata element, resulting in 100% coverage.
A.2.1	time behavior	The system should respond in a timely manner	System 's response time will be measured	Response time	The unit tests incorporate response time measurements, see Table 1. These response times are found suitable for use in the Protocol Feasibility demonstrator.
A.3.2	interoperability	the underlying information model should follow the BRIDG standard as closely as possible	Count the number of re-used BRIDG constructs	counts	10 reused BRIDG classes versus 3 application specific classes, which confirms that the BRIDG standard is followed closely
A.3.2	interoperability	EURECA components should be able to send and receive messages to the system	A web service should be available to accept and provide messages from/to the EURECA components	Web Service successful execution	The unit tests execute the web services successfully (including authentication)
A.5.1	Maturity	The amount of faults that happen in the system	The number of faults while running the system will be	Number of faults	A limited amount of faults have been recorded (1 per 300 calls). Due to

		during execution should be low	recorded, if this number is too high corrective actions will be taken		connectivity issues with the security services, authorized users were occasionally denied access to the trial metadata repository. As corrective action, the connectivity issues are being resolved.
A.5.2	Availability	The component should have a high availability as other EURECA components depend on it.	The uptime of the system will be measured	Uptime of the system	For the last year, the trial metadata repository only had downtime (of minutes) when upgrading the component. Current uptime is 2 months. The uptime is very good.
A.7.4	Modifiability	Extensions of the underlying information model should have little impact on existing information	Number of changes to existing classes when extending the information model	counts	The trial metadata repository has been extended to cover the protocol feasibility scenario. 0 changes to existing classes have been made, therefore extensions are expected to have little impact on existing information

used test cases (unit tests):

- Retrieval of all protocol feasibility studies
- Retrieval of the service metadata
- Creation and deletion of a protocol feasibility study.

The test case includes:

- Retrieve all protocol feasibility studies
- Create a protocol feasibility study
- Retrieve all protocol feasibility studies
- Retrieve new protocol feasibility study
- Delete new protocol feasibility study
- Retrieve all protocol feasibility studies

- Creation and deletion of inclusion and exclusion criteria

The test case includes:

- Create a protocol feasibility study
- Retrieve all criteria
- Create inclusion criterion
- Retrieve all criteria
- Delete Criterion
- Retrieve all criteria
- Create exclusion criterion
- Retrieve all criteria
- Delete Criterion
- Retrieve all criteria
- Delete a protocol feasibility study

- Updating a protocol feasibility study

The test case includes:

- Create a protocol feasibility study
- Retrieve the protocol feasibility study
- Update the protocol feasibility study
- Retrieve the protocol feasibility study
- Delete the protocol feasibility study

Test case	Average time (ms)	Median time (ms)
Retrieval of all protocol feasibility studies	472.34	486
Retrieval of the service metadata	180.28	176
Creation and deletion of a protocol feasibility study.	1989.72	1989
Creation and deletion of inclusion and exclusion criteria	2282.1	2243
Updating a protocol feasibility study	894.04	886

2.6.2 Evaluation

Questionnaires

Evaluator 1

FORTH

	FORM A	Rating (1 low, 5 high)				
	(Software quality characteristics)	1	2	3	4	5
Functionality	Can software perform the tasks required?					x
	Is the result as expected?					x
	Can the system interact with another system?					x
	Is the system compliant with standards?					
Efficiency	How quickly does the system respond?					x
Compatibility	Can the system share information/data with other Eureca components?					x
Reliability	Have most of the faults in the software been eliminated over time?					x
	Is the software capable of handling errors?				x	
	Can the software resume working & restore lost data after failure?					x
Security	Does the system provide identification access wherever is needed?					x
	Are data accessible only to authorized users?					x
	Can the system trace actions uniquely?					
	Does the system prevent unauthorized access?					x
Maintainability	Can faults be easily diagnosed?				x	
	Can the software be easily modified?				x	
	Can the software continue functioning if changes are made?				x	

	Can the software be tested easily?				x	
Quality in use	How accurate and complete is the software for the intended use?					x
	Does the software satisfy the perceived achievements of pragmatic goals?				x	

System Usability Scale (SUS) score: 72.5

Evaluator 2

UPM

	FORM A	**Rating (1 low, 5 high)**					**Comments**
	(Software quality characteristics)	1	2	3	4	5	
Functionality	Can software perform the tasks required?					x	
	Is the result as expected?			x			
	Can the system interact with another system?			x			Security
	Is the system compliant with standards?				x		Bridge?
Efficiency	How quickly does the system respond?				x		
Compatibility	Can the system share information/data with other Eureca components?					x	
Reliability	Have most of the faults in the software been eliminated over time?						NA
	Is the software capable of handling errors?	x					
	Can the software resume working & restore lost data after failure?						NA
Security	Does the system provide identification access wherever is needed?					x	
	Are data accessible only to authorized users?					x	
	Can the system trace actions uniquely?						NA

	Does the system prevent unauthorized access?					x	
Maintainability	Can faults be easily diagnosed?		x				
	Can the software be easily modified?						NA
	Can the software continue functioning if changes are made?						NA
	Can the software be tested easily?	x					
Quality in use	How accurate and complete is the software for the intended use?			x			
	Does the software satisfy the perceived achievements of pragmatic goals?				x		

System Usability Scale (SUS) score: 57.5

Evaluator 3

Custodix

	FORM A	**Rating (1 low, 5 high)**				
	(Software quality characteristics)	1	2	3	4	5
Functionality	Can software perform the tasks required?					x
	Is the result as expected?					x
	Can the system interact with another system?				x	
	Is the system compliant with standards?				x	
Efficiency	How quickly does the system respond?				x	
Compatibility	Can the system share information/data with other Eureca components?				x	
Reliability	Have most of the faults in the software been eliminated over time?				x	
	Is the software capable of handling errors?				x	
	Can the software resume working & restore lost data after failure?				x	

Security	Does the system provide identification access wherever is needed?				x	
	Are data accessible only to authorized users?				x	
	Can the system trace actions uniquely?			x		
	Does the system prevent unauthorized access?				x	
Maintainability	Can faults be easily diagnosed?				x	
	Can the software be easily modified?			x		
	Can the software continue functioning if changes are made?				x	
	Can the software be tested easily?				x	
Quality in use	How accurate and complete is the software for the intended use?				x	
	Does the software satisfy the perceived achievements of pragmatic goals?				x	

System Usability Scale (SUS) score: 62.5

Evaluation Comments

Evaluator 1

“In general the web service runs smoothly without problems, in a reasonable time. Considering reliability, I cannot tell if the most of the faults in the software have been eliminated over time as I have not seen previous versions of the service.”

Evaluator 2

“The Trial Management Service is a SOAP service described by a WSDL. It is a very complex web service with all the necessary methods to view the trial metadata. This service it is implemented using Custodix security so it is necessary a specific token for the soap header, so the service is very secure. On the other hand, maybe it will be necessary to add functionalities for creating or editing trial metadata.”

Evaluator 3

“The trial management service provides a repository for trial metadata. This repository is based on the BRIDG standard. The web service interface definition is complex, caused by the set of provided method calls. The execution time of the web service is quick and responsive. The functionality works as expected. Exception handling returns clear exceptions. Authorisation and Authentication are available in the service.”

2.6.3 Conclusions

After the evaluation had been performed, one of the evaluators contacted the component developer to explain the evaluation. It turns out that the component failed to meet the minimum thresholds largely due to problems to technically access the service configuring a client to accept the ssl certificate, integrating with the EURECA security framework). Additionally, the webservice does not provide useful exception messages.
To remedy the low score, example code will be made available demonstrating how to technically access provided service. Additionally the service will be extended with more informative exception messages. Finally, views on the information model will be provided to allow creating, updating and deleting information in the repository.

2.7 Query Engine

2.7.1 Validation

Query Engine					
Category ID	**Measurement**	**Description**	**Quality Measure Elements**	**Measure Type**	**Result**
A.1.1	Functional completeness	The system should be able to execute SNAQL scripts send by end-users	It will be checked that each required function provided by the query engine generates an outcome	Functionality checklist	100 different requests were sent to the query engine service, resulting in 100 outcomes
A.1.2	Functional correctness	SNAQL scripts send to the query engine should be handled correctly and generate correct results	To measure this we need to compare precision and recall of responses made by the query engine with the ones that are expected	Precision & Recall	After evaluation of the previous measurement, the 100 messages that were sent, resulted in correct outcomes (using an automated checker)
A.3.2	interoperability	EURECA components should be able to send SNAQL scripts to the system and receive the generated results	A web service should be available to accept SNAQL scripts from other EURECA components	Web Service successful execution	A request to the query engine web service from a SOAPUI client, results in a decision

A.5.2	Availability	The system should have a high availability as the query engine (core component) will frequently receive SNAQL scripts from other EURECA components	The uptime of the system will be measured	Uptime of the system	After an uptime of 2 months and heavy usage, the query engine service in the EURECA development environment was still available
A.5.3	Fault tolerance	The system should keep running after malformed SNAQL scripts enter the system	A stress test will be developed, to check how the system reacts corrupt and malfunctioning scripts	Stress test outcome	Semantically/Syntactically incorrect requests were generated and sent to the query engine service, this resulted in soap faults as expected, the service kept running after each fault
A.6.1	Confidentiality	Confidentiality of scripts and results send to/from the system should be guaranteed	Traffic to the system will be checked. Only HTTPS connections are valid, others are discarded.	HTTPS connections	Different Messages (10 requests) send to the query engine service, were sniffed using WireShark. The request content of each request was not readable. Requests sent over HTTP are not executed on the authorisation service
A.6.2	Integrity	The integrity of scripts and results from/to the system should be guaranteed	Scripts and results will be checked if they contain the required signing of body and header fields	Messages	A signed request and a non-signed request were sent to the query engine service, only the message with the signed body and header was accepted by the service, Next to this, both requests were manually checked if they contained the correct signing configuration

A.6.3	Non-repudiation	Scripts and results send from/to the system can be proven to have taken place	It will be checked if each incoming/outgoing script/result is logged onto the system	Logging	100 request were sent to the query engine service, this resulted in 100 internal loggings
A.6.4	Accountability	The sender of scripts to the system should be traceable	SAML tokens will be checked if they contain the required identity information attributes of the sender	SAML tokens	A signed and non-signed request was sent to the query engine service, only the signed request was accepted by the query engine service (containing sender information), the other one was dropped
A.6.5	Authenticity	The identity of each sender can be proved to be the one claimed, for each request to the system	Requests to the system should always contain a valid signed SAML token, containing authentication information of the sender	Requests	A signed and non-signed request was sent to the query engine service, only the signed request was accepted by the query engine service (containing sender information), the other one was dropped
A.7.3	Analyzability	Failures and deficiencies in the system should be easy diagnosable	It will be checked if exceptions are well logged in the system	Logging	Semantically incorrect requests (100 requests) were sent to the query engine service, for each incorrect request, a exception was registered in the logging
A.7.5	Testability	A script will test the system frequently to check if everything is ok	automatic tests will be executed to test the system	Response to automatic tests	Each hour, 5 random requests were sent to the query engine service, 7 days long. Each request resulted in a correct decision.

2.7.2 Evaluation

Questionnaires

Evaluator 1

FORTH

	FORM A	Rating (1 low, 5 high)				
	(Software quality characteristics)	1	2	3	4	5
Functionality	Can software perform the tasks required? i) execute incoming SNAQL scripts		x			
	Is the result as expected?		x			
Efficiency	Does the system utilize resources efficiently?			x		
Compatibility	Can the system share information/data with other Eureca components? Is the web service for remote invocation up & running? Does it provides results?			x		
Reliability	Have most of the faults in the software been eliminated over time?					
	Is the software capable of handling errors?					x
	Can the software resume working & restore lost data after failure?					
Security	Does the system provide identification access wherever is needed?					x
	Are data accessible only to authorized users?					x
	Can the system trace actions uniquely?					x
	Does the system prevent unauthorized access?					x
Maintainability	Can faults be easily diagnosed?					x

	Can the software be tested easily?		x			
Quality in use	How accurate and complete is the software for the intended use?			x		
	Does the software improve the time or reduce resources for the intended goal?					
	Does the software satisfy the perceived achievements of pragmatic goals?			x		
	Can the software harm people in the intended contexts of use?					

System Usability Scale (SUS) score: 42.5

Evaluator 2

XEROX

	FORM A	**Rating (1 low, 5 high)**				
	(Software quality characteristics)	1	2	3	4	5
Functionality	Can software perform the tasks required? i) execute incoming SNAQL scripts					x
	Is the result as expected?					x
Efficiency	Does the system utilize resources efficiently?					
Compatibility	Can the system share information/data with other Eureca components? Is the web service for remote invocation up & running? Does it provides results?				x	
Reliability	Have most of the faults in the software been eliminated over time?					x
	Is the software capable of handling errors?					
	Can the software resume working & restore lost data after failure?					

Security	Does the system provide identification access wherever is needed?					x
	Are data accessible only to authorized users?					x
	Can the system trace actions uniquely?					
	Does the system prevent unauthorized access?					x
Maintainability	Can faults be easily diagnosed?			x		
	Can the software be tested easily?				x	
Quality in use	How accurate and complete is the software for the intended use?					
	Does the software improve the time or reduce resources for the intended goal?					
	Does the software satisfy the perceived achievements of pragmatic goals?					x
	Can the software harm people in the intended contexts of use?					x

System Usability Scale (SUS) score: 85

Evaluator 3

UPM

	FORM A	**Rating (1 low, 5 high)**					**Comments**
	(Software quality characteristics)	1	2	3	4	5	
Functionality	Can software perform the tasks required? i) execute incoming SNAQL scripts					x	
	Is the result as expected?					x	
Efficiency	Does the system utilize resources efficiently?						NA

Compatibility	Can the system share information/data with other Eureca components? Is the web service for remote invocation up & running? Does it provides results?					x	
Reliability	Have most of the faults in the software been eliminated over time?						NA
	Is the software capable of handling errors?			x			
	Can the software resume working & restore lost data after failure?						NA
Security	Does the system provide identification access wherever is needed?					x	
	Are data accessible only to authorized users?					x	
	Can the system trace actions uniquely?					x	
	Does the system prevent unauthorized access?					x	
Maintainability	Can faults be easily diagnosed?			x			only in header
	Can the software be tested easily?			x			
Quality in use	How accurate and complete is the software for the intended use?			x			
	Does the software improve the time or reduce resources for the intended goal?				x		
	Does the software satisfy the perceived achievements of pragmatic goals?					x	
	Can the software harm people in the intended contexts of use?			x			

System Usability Scale (SUS) score: 57.5

Evaluation Comments

Evaluator 1

"In this web service the evaluation scenario could not be completed successfully.
The service provides a well-defined way to detect errors and to inform users with explanatory error messages, such as:

- "Exception in authorisation: Exception while parsing xacml response: Authorisation failed for :" (authorization issues)

- “Error reading XMLStreamReader.” (wrong content of Security Token Service response)

However, the last error that returned the message: “The given SOAPAction http://www.custodix.com/schemas/authz/1.0/IAuthorisationSOAPWSEndpoint/evaluate does not match an operation.” Could not be overcome.
Moreover, the scenario requires a SAML security token that is supposed to be valid for 30 minutes. Actually it is valid for only 5 minutes, as after that time the following error message is returned: “The message has expired (WSSecurityEngine: Invalid timestamp The security semantics of the message have expired)”. This is probably an issue of Security Token Service.
The steps of the evaluation scenario for the Query Engine have been repeated several times, entirely from the beginning, at different times, but each time the results were the same as mentioned above.”

Evaluator 3

“The Query Engine Service is a SOAP service described by a WSDL The web service definition is straightforward but it is only tested the SNAQL script for retrieving all the patients of a CDM. A user can also send different SNAQL scripts to retrieve all the information of the CDM but for this purpose it is necessary to have knowledge on the SNAQL syntax and the CDM.
On the other hand, the main advantage of this service is the security. It was tested with a specific token certificate for 30 minutes with a set or permission defined by security team.”

2.7.3 Conclusions

2.8 SAE prediction

The SAE prediction tool cannot be evaluated at this time. Due to lack of integration with the EURECA framework, availability of data for the evaluation of the use cases is an issue. Such integration should be obtained through the EURECA data mining architecture reported in Deliverable D5.3, which is at this time not yet operational. Integration with the data mining architecture should also resolve issues regarding the duration of the running time of the algorithms. Furthermore, data required for evaluation of multiple use cases is at this time not yet fully available. As the service should provide prediction algorithms suitable for prediction of different SAEs, such data should be available in order to evaluate the full potential of the service.

2.9 Protocol Feasibility

2.9.1 Validation

Protocol Feasibility					
Category ID	**Measurement**	**Description**	**Quality Measure Elements**	**Measure Type**	**Result**
A1.1	Functional completeness	The system is able to check the feasibility for trial design	a list of eligibility criteria for selection and change	numbers of eligibility criteria	22
A1.2	Functional correctness	The system should be able to produce correct measure	checking on the formal models of feasibility measure	formal models	complete
A2.1	time behavior	The system should respond in a timely manner	System 's response time will be measured	Response time	average 3 seconds
A.2.2	Resource utilization	The system should not be resource intensive	The CPU and memory utilization will be measured in the pc where the matcher is executed	CPU & memory utilization	quad-core, 4GB
A.3.2	Interoperability	Data are presented as a semantic data format	Support for SPARQL queries	SPARQL queries	yes
A.4.2	Learnability	The system should be able to be learned by clinical experts	The time for a non IT expert to use the system for the first time will be measured	Time to learn the system	less than 5 minutes
A.4.3	Operability	The system should be able to be operated by clinical experts without IT help	The time for a non IT expert to use the system will be measured	Time to use the system	less than 5 minutes

A.5.2	Availability	The system should be up & running almost always	The uptime of the system will be measured	Uptime of the system	almost always
A.7.5	Testability	Evaluator need no any training to do the test	The time for an evaluator to do the test	Time for tests	less than 5 minutes
A.8.2	Installability	The system should be easily installed by an IT-expert	The time to install the system in another machine will be tested	Time to install the system	zero second(just a few clicks)

2.9.2 Evaluation

Questionnaires

Evaluator 1

FORTH

	FORM A	Rating (1 low, 5 high)				
	(Software quality characteristics)	1	2	3	4	5
Functionality	Can software perform the tasks required? Do the absolute and relative feasibility help in designing the eligibility conditions?					x
	Is the result as expected? Is the enrollment of the trial in correspondence with expected feasibility?					x
	Can the system interact with the Eureca platform?					x
	Is the system compliant with standards?					
Efficiency	How quickly does the system respond?			x		
	Does the system utilize resources efficiently?					x
Compatibility	Can the system share resources without loss of its functionality?					x
	Can the system share information/data with other Eureca components?					x
Usability	Does the user comprehend how to use the system easily?				x	
	Can the user learn to use the system easily?				x	
	Can the user use the system without much effort?				x	
	Does the interface look good?				x	
Reliability	Have most of the faults in the software been eliminated over time?					
	Is the software capable of handling errors?					
Security	Does the system provide identification access wherever is needed?					
	Are data accessible only to authorized users?					
	Can the system trace actions uniquely?					

	Does the system prevent unauthorized access?					
Maintainability						
	Can the software be easily modified?					
	Can the software be tested easily?				x	
Portability	Can the software be moved to other environments?					
	Can the software be installed easily?				x	
	Does the software comply with portability standards?					
Quality in use	How accurate and complete is the software for the intended use?				x	
	Does the software improve the time or reduce resources for the intended goal?					x
	Does the software satisfy the perceived achievements of pragmatic goals?				x	
	Can the software harm people in the intended contexts of use?					

System Usability Scale (SUS) score: 80

Evaluator 2

MASTRO

	FORM A	**Rating (1 low, 5 high)**				
	(Software quality characteristics)	1	2	3	4	5
Functionality	The set of functions covers all the specified tasks and user objectives.				x	
	The system provides the correct results with the needed degree of precision.				x	
	The functions facilitate the accomplishment of specified tasks and objectives.				x	
Efficiency	The system responds quickly.			x		
	The system utilizes resources efficiently.				x	
Compatibility	The system shares resources without loss of its functionality.				x	

	The system shares information/data with other EURECA components?				x	
Usability	The users can recognize easily whether the system is appropriate for their needs.				x	
	The users learn to use the system easily.				x	
	The users use the system without much effort.				x	
	The system protects users against making errors.					
	The user interface enables pleasing and satisfying interaction for the users.			x		
Reliability	Most of the faults in the software been eliminated over time.			N/A		
	The software is capable of handling errors.			N/A		
	The software resumes working & restores lost data after failure.			N/A		
Security	The system provides identification access wherever is needed.			N/A		
	Data are accessible only to authorized users			N/A		
	The system traces actions uniquely.			N/A		
	The system prevents unauthorized access.			N/A		
Maintainability	Faults can be easily diagnosed.			N/A		
	The system is composed of discrete independent components.				x	
	An asset can be used in more than one system, or in building other assets.				x	
	The software can be tested easily.					x
Portability	The software can be moved to other environments easily.					x
	The software can be installed easily.					x
	The software can easily replace other software.					x
Quality in use	The software is accurate and complete for the intended use.					x
	The software improves the time or reduces resources for the intended goal.					x
	The software satisfies the perceived achievements of pragmatic goals.					x
	The software cannot harm people in the intended contexts of use.					x

System Usability Scale (SUS) score: 77.5

Evaluator 3

IJB

	FORM A	Rating (1 low, 5 high)				
	(Software quality characteristics)	1	2	3	4	5
Functionality	The set of functions covers all the specified tasks and user objectives.				x	
	The system provides the correct results with the needed degree of precision.					x
	The functions facilitate the accomplishment of specified tasks and objectives.					x
Efficiency	The system responds quickly.		x			
	The system utilizes resources efficiently.			(x)[2]		
Compatibility	The system shares resources without loss of its functionality.			(x)		
	The system shares information/data with other EURECA components?			(x)		
Usability	The users can recognize easily whether the system is appropriate for their needs.				x	
	The users learn to use the system easily.			x		
	The users use the system without much effort.				x	
	The system protects users against making errors.			(x)		
	The user interface enables pleasing and satisfying interaction for the users.			x		
Reliability	Most of the faults in the software been eliminated over time.			(x)		
	The software is capable of handling errors.			(x)		
	The software resumes working & restores lost data after failure.			(x)		
Security	The system provides identification access wherever is needed.			[x][3]		
	Data are accessible only to authorized users			[x]		
	The system traces actions uniquely.			[x]		

[2] **(X):** Cannot test this functionality
[3] **[X]:** Do not apply to this tool

	The system prevents unauthorized access.			[x]		
Maintainability	Faults can be easily diagnosed.			(x)		
	The system is composed of discrete independent components.			(x)		
	An asset can be used in more than one system, or in building other assets.			(x)		
	The software can be tested easily.					x
Portability	The software can be moved to other environments easily.					x
	The software can be installed easily.					x
	The software can easily replace other software.					x
Quality in use	The software is accurate and complete for the intended use.				x	
	The software improves the time or reduces resources for the intended goal.					x
	The software satisfies the perceived achievements of pragmatic goals.					x
	The software cannot harm people in the intended contexts of use.					x

System Usability Scale (SUS) score: 80

Evaluation Comments

Evaluator 1

"This system has a well-structured depiction of criteria, results and functions.
As the system query over RDF data with various inclusive/exclusive criteria and range of values, it takes a while to respond, as it is expected. Maybe a loading icon would be useful as long as the system process data, so the user can be aware of this.
Even though the installation steps are precise and the system is relatively easy to learn, in my opinion, the initial guidance of an expert user would be very helpful, so that the user will be able to comprehend the full functionality of the system and how to use it easily."

Evaluator 3

"The *Protocol Feasibility* tool installation on Windows platform, on which we tested the application, is pretty simple. Evaluation instructions are clear and easy to follow, even if starting manipulations (running java jar in console) would certainly be too hard to launch for non-IT users.

While using the tool, all inclusion and exclusion parameters are easy to select/unselect, even if the system would eventually be eventually more complete if it

could allow defining additional inclusion/exclusion criteria, according to the criteria that would have been defined in the initial trial protocol design.
On the one hand for *Absolute feasibility* option, the first bar graph summarizing the overall feasibility rate regarding all criteria should be differentiated from the way it is presented (in green) like for each single inclusion criteria.
On the other hand, *Relative feasibility* would need some more explanations to be fully understood by users, for example by guiding them through both choice and modification of parameters/criteria more explicitly than just the proposed graphs that are not so obvious *prima facie* without having read the very clear and complete related article[4], but that do not have to be a pre-requisite for clinical users before using the tool. In such a case, training should be considered.

Response time is fine but the tool would need a progression bar while the system is running to give users some indications on the remaining time before getting the results for both *Absolute* and *Relative feasibility* options.

The *Protocol Feasibility* tool seems to be a useful application for testing the feasibility while generating hypothesis and designing a clinical trial, but the tool need more clinical users' evaluation."

2.9.3 Conclusions

No risks have been found for the Protocol Feasibility. The developers have been informed for the results of the evaluation and they will continue to improve their component.

[4] Z. Huang, F. van Harmelen, A. ten Teije, and A. Dekker, "Feasibility Estimation for Clinical Trials," in *Proceedings of the 7th International Conference on Health Informatics (HEALTHINF2014)*, 2014.

2.10 Identity Manager

2.10.1 Validation

Identity Manager					
Category ID	**Measurement**	**Description**	**Quality measure elements**	**Measure Type**	**result**
A.1.1	Functional completeness	The system should provide the identity management functionality required by the users	It will be checked that each required function is available in the identity management service	Functionality checklist	An internal checklist has been made and maintained since the start of the project. All required functionality hasbeen implemented.
A.1.2	Functional correctness	The identity management service should generate the right results for each available function call	To measure this we need to compare precision and recall of results made by the identity management service with the ones that are expected	Precision & Recall	The frontend has been tested by the whole team for functional correctness, id with a wide scope of user roles and rights, and with an overview of the functional requirements. 200 different SOAP requests related to different aspects of the functional requirements have been made with both correct and incorrect credentials. The IDM performed the correct actions to each request.
A1.3	Functional appropriateness	The identity management service should comply to the EURECA legal requirements	The system will be audited by legal experts	Audit outcome	Auditing has not been performed yet

A.5.1	Maturity	The amount of faults that happen in the system during execution should be low	The number of faults while running the system will be recorded, if this number is too high corrective actions will be taken	Number of faults	The logs of the IDM were evaluated for faults. The 16509 logged requests, both from the frontend and the web service did not contain any fault.
A.5.2	Availability	The system should have a high availability as the IDM is the prime way for new users and services to register	The uptime of the system will be measured	Uptime of the system	The IDM has been running for almost a year without unexpected incidents.
A.5.3	Fault tolerance	The system should keep running after malformed requests enter the system	A stress test will be developed, to check how the system reacts corrupt and malfunctioning requests	Stress test outcome	20 Faulty SOAP messages were sent to the IDM and the response of the IDM were measured. The system reacted appropriately without downtime.
A.6.1	Confidentiality	Confidentiality of messages from/to the system should be guaranteed	Traffic to the system will be checked. Only HTTPS connections are valid, others are discarded.	HTTPS connections	Different Messages (10 requests) sent to the authorization service, were sniffed using WireShark. The request content of each request was not readable. Requests sent over HTTP are not executed on the authorization service

A.6.2	Integrity	The integrity of messages from/to the system should be guaranteed	Messages will be checked if they contain the required signing of body and header fields	Messages	A signed request and a non-signed request were sent to the IDM. Only the message with the signed body and header was accepted by the service. Both requests were explicitly controlled to see if they contained the correct signing configuration.
A.6.3	Non-repudiation	Messages sent from/to the system can be proven to have taken place	It will be checked if each incoming/outgoing message is logged onto the system	Logging	100 request were sent to the authorisation service, this resulted in 100 loggings
A.6.4	Accountability	The sender of requests to the system should be traceable	A sender can only have access to the identity management if they provide credentials. These credentials are linked to sender attributes. The sender is traceable this way	Credentials	The credentials found in a request to the IDM should contain all necessary information to unambiguously identify the requestor. Traceability is thus guaranteed.
A.6.5	Authenticity	The identity of each sender can be proved to be the one claimed, for each request to the system	A sender will authenticate himself by providing his credentials to the identity management. It will be tested if only persons with valid credentials can access the service.	Credentials	200 different SOAP messages with both correct and incorrect credentials were sent. The resulting responses were checked for their content. All requests were found to lead to appropriate responses.

A.7.1	Modularity	The components of the identity management should be modular.	The identity management will be tested with different configurations	Configuration outcome	The IDM has been tested using 10 different configurations. The expected differences in behavior have been checked. Extra care has been taken that functionality that is not expected to change remains unmodified.
A.7.3	Analyzability	Failures and deficiencies in the system should be easy diagnosable	It will be checked if exceptions are well logged in the system	Logging	Errors and exceptions are thoroughly logged by the IDM.
A.7.5	Testability	A script will test the system frequently to check if everything is ok	automatic tests will be executed to test the system	Response to automatic tests	The IDM has a full suite of test which are automatically executed at predefined intervals.
A.8.3	Replaceability	The IDM component of the identity management service should be easily replaceable by other implementations that use the same standards	It will be tested that IDM are loosely coupled and compliant with defined standards	Compliant test outcome	Has not yet been tested

2.10.2 Evaluation

Questionnaires

Evaluator 1

FORTH

	FORM A	Rating (1 low, 5 high)				
	(Software quality characteristics)	1	2	3	4	5
Functionality	Can software perform the tasks required? i) provide user/services management ii) authenticate users/services			x		
	Is the result as expected?				x	
	Is the system compliant with standards? i) SAML 2.0 ii) WS-Trust					
Efficiency						
Compatibility						
	Can the system share information/data with other Eureca components? Is the web service for remote invocation up & running? Does it provides results?					
Usability						
Reliability	Have most of the faults in the software been eliminated over time?					
	Is the software capable of handling errors?			x		
Security	Does the system provide identification access wherever is needed?				x	
	Are data accessible only to authorized users?					x
	Can the system trace actions uniquely?					

	Does the system prevent unauthorized access?					x
Maintainability	Can faults be easily diagnosed?					
	Can the software be easily modified?					
	Can the software continue functioning if changes are made?					
	Can the software be tested easily?					x
Portability						
	Can the software easily replace other software?					
Quality in use	How accurate and complete is the software for the intended use?				x	
	Does the software improve the time or reduce resources for the intended goal?					
	Does the software satisfy the perceived achievements of pragmatic goals?				x	
	Can the software harm people in the intended contexts of use?					

System Usability Scale (SUS) score: 95

Evaluator 2

LUH

	FORM A	**Rating (1 low, 5 high)**					**Comments**
	(Software quality characteristics)	1	2	3	4	5	
Functionality	Can software perform the tasks required? i) provide user/services management ii) authenticate users/services				x		
	Is the result as expected?				x		
	Is the system compliant with standards? i) SAML 2.0 ii) WS-Trust						Out of our scope of review
Efficiency							

Compatibility							
	Can the system share information/data with other Eureca components? Is the web service for remote invocation up & running? Does it provides results?						Out of our scope of review
Usability							
Reliability	Have most of the faults in the software been eliminated over time?				x		
	Is the software capable of handling errors?					x	
Security	Does the system provide identification access wherever is needed?					x	
	Are data accessible only to authorized users?					x	
	Can the system trace actions uniquely?						Out of our scope of review
	Does the system prevent unauthorized access?						Out of our scope of review
Maintainability	Can faults be easily diagnosed?					x	
	Can the software be easily modified?						Out of our scope of review
	Can the software continue functioning if changes are made?						Out of our scope of review
	Can the software be tested easily?					x	
Portability							

	Can the software easily replace other software?						Out of our scope of review
Quality in use	How accurate and complete is the software for the intended use?					x	
	Does the software improve the time or reduce resources for the intended goal?					x	
	Does the software satisfy the perceived achievements of pragmatic goals?					x	
	Can the software harm people in the intended contexts of use?	x					

System Usability Scale (SUS) score: 90

Evaluator 3

USAAR

	FORM A	**Rating (1 low, 5 high)**					**Comments**
	(Software quality characteristics)	1	2	3	4	5	
Functionality	Can software perform the tasks required? i) provide user/services management ii) authenticate users/services					x	
	Is the result as expected?					x	
	Is the system compliant with standards? i) SAML 2.0 ii) WS-Trust					x	as far as I know and understand those standards
Efficiency							
Compatibility							
	Can the system share information/data with other Eureca components? Is the web service for remote invocation up & running? Does it provides results?						This was not tested according to the description for the evaluation of the EURECA IDM. But I suppose this will be the case, as without the IDM one cannot access EURECA components

Usability							
Reliability	Have most of the faults in the software been eliminated over time?			x			see report
	Is the software capable of handling errors?			x			see report
Security	Does the system provide identification access wherever is needed?						This was not tested according to the description for the evaluation of the EURECA IDM
	Are data accessible only to authorized users?						This was not tested according to the description for the evaluation of the EURECA IDM
	Can the system trace actions uniquely?						This was not tested according to the description for the evaluation of the EURECA IDM
	Does the system prevent unauthorized access?					x	
Maintainability	Can faults be easily diagnosed?						Cannot tell
	Can the software be easily modified?						Cannot tell
	Can the software continue functioning if changes are made?						Cannot tell
	Can the software be tested easily?					x	
Portability							
	Can the software easily replace other software?						I do not know, as the software needs to be integrated in other systems. How easy that is cannot be judged by a clinician

Quality in use	How accurate and complete is the software for the intended use?				x		
	Does the software improve the time or reduce resources for the intended goal?						
	Does the software satisfy the perceived achievements of pragmatic goals?				x		
	Can the software harm people in the intended contexts of use?						I do not think so

System Usability Scale (SUS) score: 100

Evaluation Comments

Evaluator 1

“The evaluation scenario was nearly completed. Some issues occurred regarding the emails the system should have sent. Neither the account activation email nor the password recovery email I was able to receive. Moreover, in changing your password task, the system accepted the old password as the new one.”

Evaluator 2

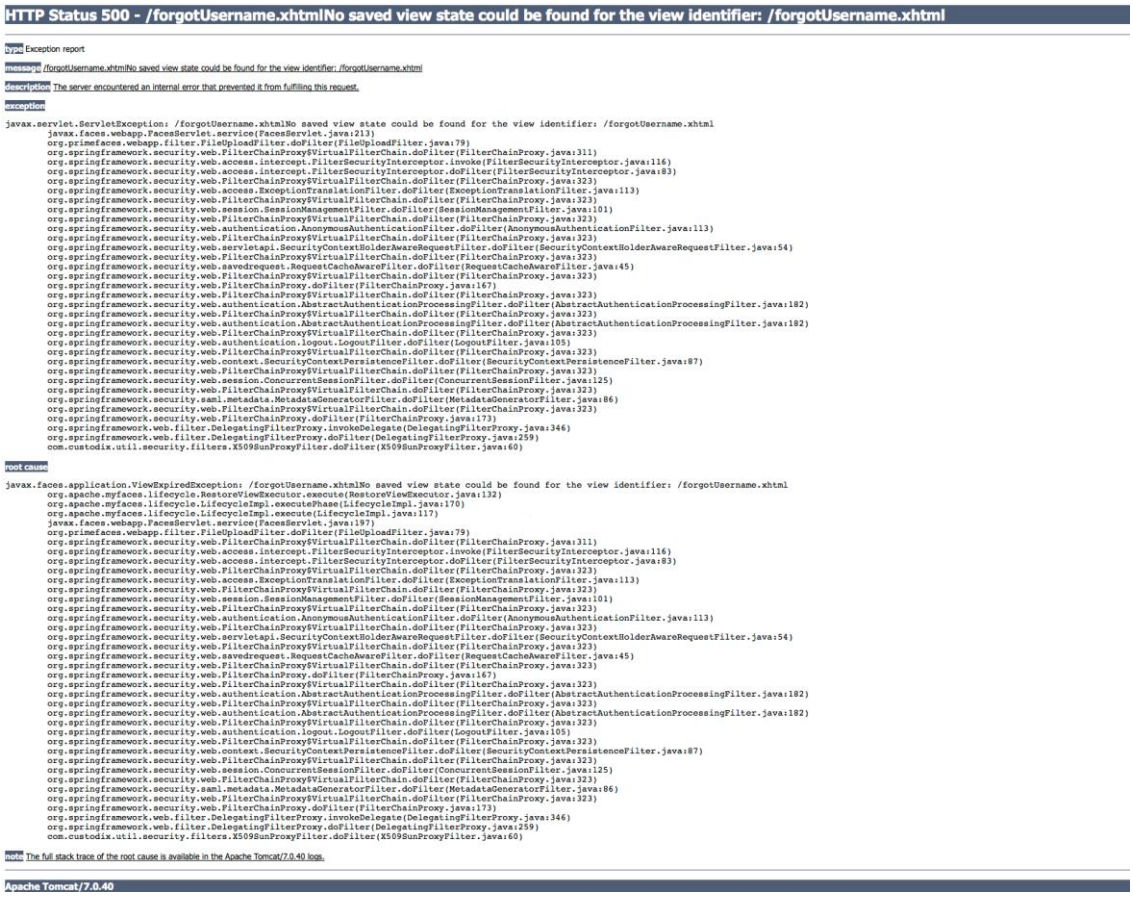
“Following a couple of suggestions regarding the interface after completing the evaluation tasks:

Sign up form: Suggest replacing "Place" for "City"
Account activation page: Suggest removing the security question, see eg http://www.theatlantic.com/technology/archive/2012/08/security-questions-the-biggest-joke-in-online-identity-verification/260835/ Two-step verification would be more secure.
User activation successful page: Can automatic forwarding to https://idm-fp7.custodix.com/idm/activationSuccess.xhtml be implemented? Currently User has to click on Ok.
Profile page Account Details: Enter current password should be entered right away and not via popup. If User uses a password manager and already swapped out the old password with the new password in the password manager, there is a risk that User might not know the old password anymore.
forgotUsername.xhtml: Message "You'll soon receive an email containing the user names linked to" is displayed twice in a row. Did not receive the email.
forgotPassword.xhtml: Does not accept security answer (it is correct)”

Evaluator 3

In general, the evaluation scenario was conducted without any major problems. Following some issues that occurred during the evaluation. Firstly, the approval for the new account took several days and was only solved by writing to the IDM administrator. The reason for the delay might be the fact, that the validation email was

automatically put to the spam folder that was not searched for. So this needs to be checked or told to people that they have to check their spam folder. Secondly, in “edit account” task, by changing the password to a new one, the old password is also accepted as a new password. Should that be the case? Or do I need to provide always a new one?
Also, in “recovery username” task everything done as described, very easy but no email was received, even after hours. Also not in the spam folder. Trying it a second time I got the following bug:



“Recover password” task done as described but, the security question was another than given to the system before. Providing the answer of the previously selected question the following page appeared:

As I could not finalize this part of the evaluation, I did not receive an email. Therefore the following steps were not possible to do.

2.10.3 Conclusions

The current deployed version of the identity manager differs from the last implemented version. Several Suggestions given by the validators where already fixed in the last implemented version and will be deployed before the end of the year.

One of the main Issues is the following
“Cannot receive neither the account activation email nor the password recovery email”
This is caused due to a firewall setting the identity manager which was unable to send out email messages to external mail services. The risk for this issue is low and concerning the mitigation actions the firewall settings will be changed to enable external email communication

Another issue was the following which was caused by a disabled configuration setting on the identity manager.

“In changing your password task, the system accepted the old password as the new one”

The configuration will be changed to enable unique passwords so the risk for this issue is low,

Besides these issues there were also a number of suggestion described bellow

- “Sign up form: Suggest replacing "Place" for "City"”
    - Comment: This will be given as feedback to the idm developers
    -
- “Account activation page: Suggest removing the security question, see eg http://www.theatlantic.com/technology/archive/2012/08/security-questions-the-biggest-joke-in-online-identity-verification/260835/ Two-step verification would be more secure”
    - Comment: The security question is not used as identity verification as mentioned in the article, we send an email to the end-user for this. The security question is used for avoiding spam to the user’s email addresses.

- “User activation successful page: Can automatic forwarding to https://idm-fp7.custodix.com/idm/activationSuccess.xhtml be implemented? Currently User has to click on Ok.”
    - Comment: This will be given as feedback to the idm developers

- “Profile page Account Details: Enter current password should be entered right away and not via popup. If User uses a password manager and already swapped out the old password with the new password in the password manager, there is a risk that User might not know the old password anymore.”
    - Comment: This will be given as feedback to the idm developers

- “forgotUsername.xhtml: Message "You'll soon receive an email containing the user names linked to" is displayed twice in a row. Did not receive the email.”
    - Comment: This will be given as feedback to the idm developers. The email issue was discussed before.

2.11 Microbiology Module

2.11.1 Validation

Microbiology Module					
Category ID	**Measurement**	**Description**	**Quality Measure Elements**	**Measure Type**	**Result**
A1 FUNCTIONAL SUITABILITY					
A.1.1.1	Functional completeness	The Microbiology Application is able to merge and display all needed Data from the linked Hospital Systems	CRFs of a patient includes correct data	CRFs of Microbiology Application	ok, but only for specific data sets. Other data sets will follow.
A.1.1.2	Functional completeness	The Microbiology Application enables statistical analyses within the merged data sets	Query Interface	Query Interface of Microbiology Application	Possible for first scenario "All first evidences of MRSA, VRE or MRGN during a time period". Other scenarios (see D6.3) will follow

A.1.1.3	Functional completeness	The linkage of wards to the Microbiology Application enables the transmission of corresponding data from the Hospital Systems	data exchange from the new linked wards possible	Microbiology Application/Push Services/Update Services	Works for import into Microbiology Application from the EURECA CDM for a specific data set. An automatic data provision from the hospital systems to the Push Services have to be implemented. The Push Service works for a specific data set. Other data sets will follow.
A.1.1.4	Functional completeness	The Microbiology Application enables the entry of missing data into the corresponding CRFs	CRFs	CRFs of the Microbiology Application	yes
A.1.2.1	Functional correctness	The Microbiology Application merges all data sets of a patient correctly	CRFs of a patient includes correct data	Microbiology Application: CRFs of a patient	yes
A.1.2.2	Functional correctness	The Query Interface shows the correct results	Query Interface	Query Interface of Microbiology Application	ok for first scenario "All first evidences of MRSA, VRE or MRGN during a time period". Other scenarios (see D6.3) will follow.
A.1.3	Functional appropriateness	The Query Interface enables sufficient statistical analyses	Query Interface	Query Interface of Microbiology Application	ok for first scenario "All first evidences of MRSA, VRE or MRGN during a time period". Other scenarios (see D6.3) will follow
A.2 PERFORMANCE EFFICIENCY					

A.2.1.1	Time Behaviour	The software should respond in a timely matter	System 's response time will be measured	Response time	Depends on the amount of available data. Huge data volumes are very cost intensive
A.2.1.2	Time Behaviour	The Services should transmit the data in a timely matter	System 's response time will be measured	Response time	Depends on the amount of available data. Huge data volumes are very cost intensive
A.2.2	Resource utilization	The system should not be resource intensive	The CPU and memory utilization will be measured in the pc where the matcher is executed	CPU & memory utilization	ok
A.2.3.1	capacity	The Push-Services should be able to transmit large volumes of data to the EURECA CDM	Push-Services	possibility/time	possible, but an import of huge data volumes is time intensive
A.2.3.2	capacity	The Upload-Services should be able to import large volumes of data into the Microbiology Application	Update-Services	possibility/time	possible, but an import of huge data volumes is time intensive
A.3 COMPATIBILITY					

A3.1	co-existence	the MSS should be installed in any hospital environment easily	MSS installation	possibility	"black box" solution of EURECA CDW is required; all data items and structures from the hospital information system needs to be mapped into the EURECA compatible format (HL7 CDA messages where all data items are linked to corresponding Core Dataset concepts)
A.3.2.1	Interoperability: Push Data from Hospital Systems into the EURECA CDM	The data from the Hospital Systems have to be stored in the EURECA CDM	Data are merged into the CDM (semantically annotated EURECA CDS; linked to corresponding patient)	EURECA ETL successful execution	ok
A.3.2.2	Interoperability: Get Data from the EURECA CDM	The microbiology data should be accessible for the Microbiology Application	Queries should be executed to the EURECA CDM	Query responses	ok
A.4 USABILITY					
A.4.1	appropriateness recognisability	Required statistical analyses are possible	Microbiology Application: Query Interface	Correctness of Query Result	Possible for first scenario "All first evidences of MRSA, VRE or MRGN during a time period". Other scenarios (see D6.3) will follow.

A.4.2	Learnability	The application should be able to be learned by clinicians	The time for a non IT expert to use the application for the first time will be measured	Time to learn the system	As far as the Microbiology Application is improved by the feedbacks of the evaluation process the system should be useable by a clinical after an introduction.
A.4.3	Operability	The application should be able to be operated by clinicians without IT help	The time for a non IT expert to use the application in his routine will be measured	Time to use the system in daily routine	As far as the Microbiology Application is improved by the feedbacks of the evaluation process the system should be useable by a clinical after an introduction.
A.4.5	User interface aesthetics	Intuitive processing of the application by clinical experts possible	intuitive processing	GUI	Yes, beside some minor facts we got positive feedback about the processing with the GUI of the Microbiology Application
A.5 RELIABILITY					
A.5.2.1	Availability	The Push Services should be up & running almost always	The uptime of the Push Services will be measured	Uptime of the system	The EURECA Services are hosted at UPM. A local installation at UdS is planned. We did not cover any uptime problems with UPM's systems until now.
A.5.2.2	Availability	The EURECA CDW should be up & running almost always	The uptime of the EURECA CDW will be measured	Uptime of the system	The EURECA Services are hosted at UPM. A local installation at UdS is planned. We did not cover any uptime problems with UPM's systems until now.
A.5.2.3	Availability	The Update-Services should be up & running almost always	The uptime of the Update Services will be measured	Uptime of the system	The EURECA Services are hosted at UPM. A local installation at UdS is planned. We did not cover any uptime problems with UPM's systems until now.

A.5.2.4	Availability	The Microbiology Application be up & running almost always	The uptime of the Microbiology Application will be measured	Uptime of the system	Until now, we did not cover any problems (e.g. restart) with the uptime of the Microbiology Application
A.5.4.1	Recoverability	Data completeness in the EURECA CDW after a system failure of the Push Services	data completeness	Push Services	no, a mechanisms for system failures have to be implemented
A.5.4.2	Recoverability	Data completeness in the Microbiology Application after a system failure of the Update Services	data completeness	Update Services	yes, we implemented a mechanism that enables the import of all available datasets from the EURECA CDW and avoids a duplicate importing of datasets.
A.6 SECURITY					
A.6.1	Confidentiality	Patient data must be pseudonymized stored in the EURECA CDW			yes
A.6.5.1	Authenticity	The Access to the Microbiology Application should only be possible to authorized persons (rights & roles management)	Log in to Microbiology Application	Microbiology Application	yes
A.6.5.2	Authenticity	The Visibility of specific data should only be possible to authorized persons (rights & roles management)	Visibility of data into the Microbiology Application	Microbiology Application	yes
A.7 MAINTAINABILITY					

A.7.4	Modifiability	The system can be extended by new parameters (e.g. new Antibiotics, new agents) easily	Extension possible	Whole Service	only possible for Microbiology Application until now
A.7.5	Testability	A script will test the system frequently to check if everything is ok	automatic tests will be executed to test the system	Response to automatic tests	we did not develop a scrip for testing
A.8 PORTABILITY					
A.8.2	Installability	The system should be easily installed by an IT-expert	The time to install the system in another machine will be tested	Time to install the system	Several instances have to be installed in order to run the whole service in a hospital: a local EURECA CDW within the Data Push Service and the Query Execution Service; ObTiMA web application (as end user application); a system that maps and transforms and finally sends the data from the clinical systems to the Data Push Service. It is hard to measure the time for the whole installation process not least, because some instances are currently not available for local an installation.

2.11.2 Evaluation

Questionnaires

Evaluator 1

FORTH

	FORM A	Rating (1 low, 5 high)				
	(Software quality characteristics)	1	2	3	4	5
Functionality	Can software perform the tasks required?					x
	i) Statistical Analyses possible (Query Interface)?					x
	ii) Display the data from the Hospital Systems into CRFs (automatically prefilled)?					
	iii) Enter missing data (Barcode, Keyboard) possible?					
	Are these results as expected?					
	Can the service interact with the EURECA CDM?					x
	i) Push-Services (data from Hospital Systems)					
	ii) Update Services (Upload data from EURECA CDM into Microbiology Application)					x
	Is the system compliant with standards?					
Efficiency	How quickly does the Microbiology Application interact?					x
	How quickly does the Microbiology Application receive required data (synchronize data form Hospital Systems)?				x	
	Does the application utilize CPU and memory efficiently?					x
Compatibility	Does the Push Services push the clinical data into the EURECA CDM successfully?					x
	Does the Update Services upload the clinical data from the CDM into the Microbiology Application successfully?					x
	Are the uploaded data in the CRFs of the Microbiology Application complete?					x
Usability	Does the user comprehend how to use the system easily?				x	
	Can the user learn to use the system easily?					x

	Can the user use the system without much effort?					x
	Does the interface look good?				x	
	Does the interface provide all required information?					x
	Is the usage of the application intuitive?					x
Reliability	Have most of the faults in the software been eliminated over time?					
	Is the software capable of handling errors?				x	
	Can the services resume working & restore lost data after failure?					
	Can the Microbiology Application resume working & restore lost data after failure?					
Security	Are data accessible only to authorized users?					x
	Does the system prevent unauthorized access?					x
	Are the pushed patient data sufficient pseudonymised?					x
	Can the data be merged to one patient even the pseudonymisation?					
	Can the Microbiology Application display the real patient data (re-pseudonymisation)?					x
Maintainability	Can the software be tested easily?				x	
Portability	Can the software be moved to other environments?					
	Can the software be installed easily?					
	Can the software easily replace other software?					
Quality in use	How accurate and complete is the software for the intended use?					
	Does the software improve the time or reduce resources for the intended goal?					x
	Does the software satisfy the perceived achievements of pragmatic goals?					x
	Can the software harm people in the intended contexts of use?					

System Usability Scale (SUS) score: 95

Evaluator 2

IJB

	FORM A	**Rating (1 low, 5 high)**				
	(Software quality characteristics)	1	2	3	4	5
Functionality	Can software perform the tasks required?					
	i) Statistical Analyses possible (Query Interface)?				x	
	ii) Display the data from the Hospital Systems into CRFs (automatically prefilled)?	x				
	iii) Enter missing data (Barcode, Keyboard) possible?			(x)		
	Are these results as expected?			(x)		
	Can the service interact with the EURECA CDM?					
	i) Push-Services (data from Hospital Systems)			(x)		
	ii) Update Services (Upload data from EURECA CDM into Microbiology Application)			(x)		
	Is the system compliant with standards?			(x)		
Efficiency	How quickly does the Microbiology Application interact?				x	
	How quickly does the Microbiology Application receive required data (synchronize data form Hospital Systems)?				x	
	Does the application utilize CPU and memory efficiently?					x
Compatibility	Does the Push Services push the clinical data into the EURECA CDM successfully?			(x)		
	Does the Update Services upload the clinical data from the CDM into the Microbiology Application successfully?					x
	Are the uploaded data in the CRFs of the Microbiology Application complete?	x				
Usability	Does the user comprehend how to use the system easily?			x		
	Can the user learn to use the system easily?				x	
	Can the user use the system without much effort?				x	
	Does the interface look good?			x		
	Does the interface provide all required information?			x		

	Is the usage of the application intuitive?			x		
Reliability	Have most of the faults in the software been eliminated over time?			(x)		
	Is the software capable of handling errors?			(x)		
	Can the services resume working & restore lost data after failure?			(x)		
	Can the Microbiology Application resume working & restore lost data after failure?			(x)		
Security	Are data accessible only to authorized users?					x
	Does the system prevent unauthorized access?					x
	Are the pushed patient data sufficient pseudonymised?					x
	Can the data be merged to one patient even the pseudonymisation?			(x)		
	Can the Microbiology Application display the real patient data (re-pseudonymisation)?			(x)		
Maintainability	Can the software be tested easily?				x	
Portability	Can the software be moved to other environments?			(x)		
	Can the software be installed easily?			(x)		
	Can the software easily replace other software?			(x)		
Quality in use	How accurate and complete is the software for the intended use?				x	
	Does the software improve the time or reduce resources for the intended goal?			(x)		
	Does the software satisfy the perceived achievements of pragmatic goals?				x	
	Can the software harm people in the intended contexts of use?	x				

System Usability Scale (SUS) score: 75

Evaluator 3

GBG

	FORM A	Rating (1 low, 5 high)				
	(Software quality characteristics)	1	2	3	4	5
Functionality	Can software perform the tasks required?				x	
	i) Statistical Analyses possible (Query Interface)?					
	ii) Display the data from the Hospital Systems into CRFs (automatically prefilled)?					
	iii) Enter missing data (Barcode, Keyboard) possible?					
	Are these results as expected?					
	Can the service interact with the EURECA CDM?					
	i) Push-Services (data from Hospital Systems)					
	ii) Update Services (Upload data from EURECA CDM into Microbiology Application)					
	Is the system compliant with standards?					
Efficiency	How quickly does the Microbiology Application interact?					
	How quickly does the Microbiology Application receive required data (synchronize data form Hospital Systems)?			x		
	Does the application utilize CPU and memory efficiently?					
Compatibility	Does the Push Services push the clinical data into the EURECA CDM successfully?			x		
	Does the Update Services upload the clinical data from the CDM into the Microbiology Application successfully?			x		
	Are the uploaded data in the CRFs of the Microbiology Application complete?					
Usability	Does the user comprehend how to use the system easily?					x
	Can the user learn to use the system easily?					x
	Can the user use the system without much effort?					x
	Does the interface look good?					x
	Does the interface provide all required information?					x
	Is the usage of the application intuitive?					x
Reliability	Have most of the faults in the software been eliminated over time?					
	Is the software capable of handling errors?					

	Can the services resume working & restore lost data after failure?					
	Can the Microbiology Application resume working & restore lost data after failure?					
Security	Are data accessible only to authorized users?					
	Does the system prevent unauthorized access?					
	Are the pushed patient data sufficient pseudonymised?					
	Can the data be merged to one patient even the pseudonymisation?					
	Can the Microbiology Application display the real patient data (re-pseudonymisation)?					
Maintainability	Can the software be tested easily?				x	
Portability	Can the software be moved to other environments?					
	Can the software be installed easily?					
	Can the software easily replace other software?					
Quality in use	How accurate and complete is the software for the intended use?					
	Does the software improve the time or reduce resources for the intended goal?					
	Does the software satisfy the perceived achievements of pragmatic goals?					
	Can the software harm people in the intended contexts of use?					

System Usability Scale (SUS) score: 92.5

Evaluator 4

USAAR

An extra evaluation has been conducted from USAAR as they were interested in the tool.

	FORM A	**Rating (1 low, 5 high)**					**Comments**
	(Software quality characteristics)	1	2	3	4	5	

Functionality	Can software perform the tasks required?				x		
	i) Statistical Analyses possible (Query Interface)?					x	
	ii) Display the data from the Hospital Systems into CRFs (automatically prefilled)?					x	
	iii) Enter missing data (Barcode, Keyboard) possible?				x		
	Are these results as expected?				x		
	Can the service interact with the EURECA CDM?					x	
	i) Push-Services (data from Hospital Systems)					x	
	ii) Update Services (Upload data from EURECA CDM into Microbiology Application)					x	
	Is the system compliant with standards?				x		which standards, I would say yes
Efficiency	How quickly does the Microbiology Application interact?				x		
	How quickly does the Microbiology Application receive required data (synchronize data form Hospital Systems)?			x			
	Does the application utilize CPU and memory efficiently?						???
Compatibility	Does the Push Services push the clinical data into the EURECA CDM successfully?					x	
	Does the Update Services upload the clinical data from the CDM into the Microbiology Application successfully?					x	
	Are the uploaded data in the CRFs of the Microbiology Application complete?					x	
Usability	Does the user comprehend how to use the system easily?				x		
	Can the user learn to use the system easily?					x	
	Can the user use the system without much effort?					x	
	Does the interface look good?					x	
	Does the interface provide all required information?				x		
	Is the usage of the application intuitive?				x		Back Button

Reliability	Have most of the faults in the software been eliminated over time?			x			
	Is the software capable of handling errors?			x			
	Can the services resume working & restore lost data after failure?						???
	Can the Microbiology Application resume working & restore lost data after failure?						???
Security	Are data accessible only to authorized users?						???
	Does the system prevent unauthorized access?					x	
	Are the pushed patient data sufficient pseudonymised?						see comment
	Can the data be merged to one patient even the pseudonymisation?						???
	Can the Microbiology Application display the real patient data (re-pseudonymisation)?			x			
Maintainability	Can the software be tested easily?					x	
Portability	Can the software be moved to other environments?						???
	Can the software be installed easily?						???
	Can the software easily replace other software?						there is no other software available for that scenario
Quality in use	How accurate and complete is the software for the intended use?			x			
	Does the software improve the time or reduce resources for the intended goal?				x		
	Does the software satisfy the perceived achievements of pragmatic goals?				x		
	Can the software harm people in the intended contexts of use?	x					

System Usability Scale (SUS) score: 80

Evaluation Comments

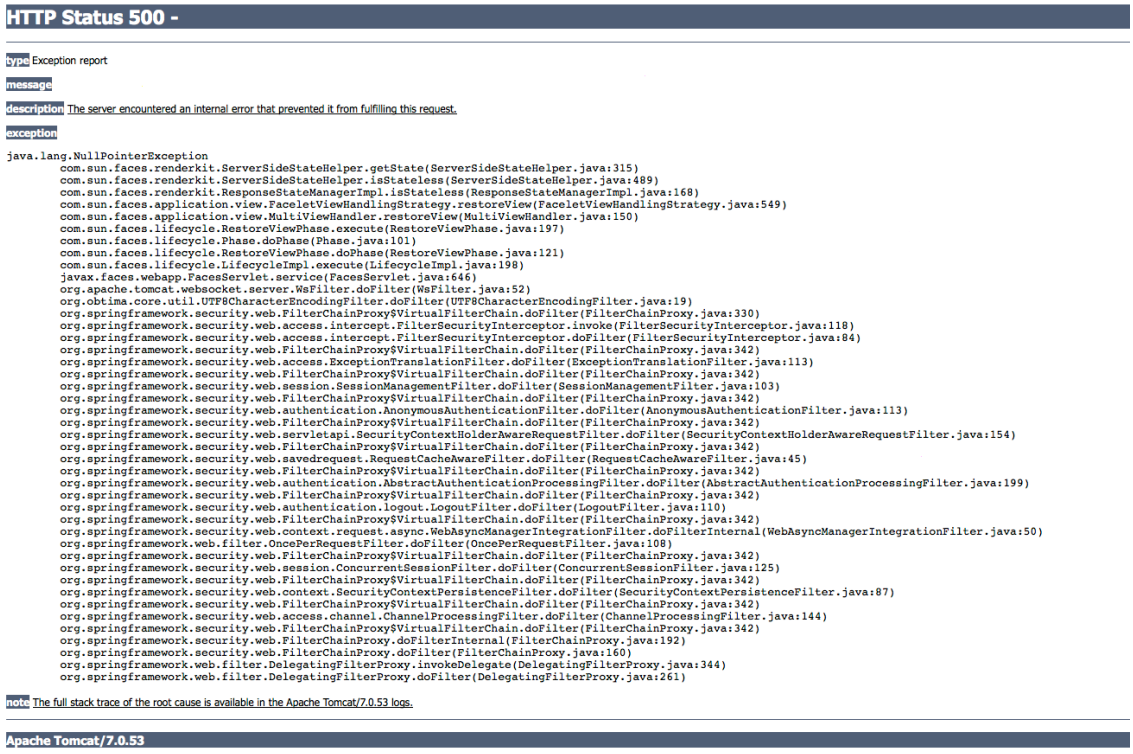
Evaluator 1

"The Microbiology Safety Service (MSS) is a service that a clinician can use to get fast knowledge and analyses about antibiotic treatments, specific infectious agents, their resistance profile and possible serious side effects.
As a suggestion for this system, firstly, larger font and icons would be preferable. Also, the back button functionality would be very useful, as the navigation to previous pages is slightly awkward with the current implementation. Finally, if the user type a wrong date, the system does not proceeds (rightly) but no message is displayed to the user as a feedback."

Evaluator 4

"In summary the scenario is now possible to run as described. There are the following comments in addition:

1. There is no button to go back. Every time one needs to select the tool and start from beginning. If I want to select resistance from different scenarios I always have to start with the selection of the microbiology scenario, instead of going back to the query interface.
2. In the query result there is the ward mentioned as 'Ward:' but no ward is given or can be seen. Why is ward displayed there? The result would be interesting by searching in different wards, that for each ward the agents, the total number and the resistance against is displayed
3. Why on patient data the pseudonym and the name are shown? There should be never the combination of both shown on one screen, as you might list this linkage on a sheet of paper. If you have allowance to see the name, then you do not need the pseudonym. If you are not allowed to see the name then only the pseudonym should be displayed. Interestingly by clicking on print barcode then the name of the patient is no longer shown. If you then use the back button of the browser then the name is displayed again. Such things should not happen in a productive version.
4. How to delete or unlink a ward from the linked wards?
5. After linkage of the ward Pediatric Oncology and Hematology a bug occurs: By clicking on name of the ward or head organization or headperson the following error occurs:



HTTP Status 500 -

type Exception report

message

description The server encountered an internal error that prevented it from fulfilling this request.

exception

```
java.lang.NullPointerException
        com.sun.faces.renderkit.ServerSideStateHelper.getState(ServerSideStateHelper.java:315)
        com.sun.faces.renderkit.ServerSideStateHelper.isStateless(ServerSideStateHelper.java:489)
        com.sun.faces.renderkit.ResponseStateManagerImpl.isStateless(ResponseStateManagerImpl.java:168)
        com.sun.faces.application.view.FaceletViewHandlingStrategy.restoreView(FaceletViewHandlingStrategy.java:549)
        com.sun.faces.application.view.MultiViewHandler.restoreView(MultiViewHandler.java:150)
        com.sun.faces.lifecycle.RestoreViewPhase.execute(RestoreViewPhase.java:197)
        com.sun.faces.lifecycle.Phase.doPhase(Phase.java:101)
        com.sun.faces.lifecycle.RestoreViewPhase.doPhase(RestoreViewPhase.java:121)
        com.sun.faces.lifecycle.LifecycleImpl.execute(LifecycleImpl.java:198)
        javax.faces.webapp.FacesServlet.service(FacesServlet.java:646)
        org.apache.tomcat.websocket.server.WsFilter.doFilter(WsFilter.java:52)
        org.obtima.core.util.UTF8CharacterEncodingFilter.doFilter(UTF8CharacterEncodingFilter.java:19)
        org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:330)
        org.springframework.security.web.access.intercept.FilterSecurityInterceptor.invoke(FilterSecurityInterceptor.java:118)
        org.springframework.security.web.access.intercept.FilterSecurityInterceptor.doFilter(FilterSecurityInterceptor.java:84)
        org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:342)
        org.springframework.security.web.access.ExceptionTranslationFilter.doFilter(ExceptionTranslationFilter.java:113)
        org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:342)
        org.springframework.security.web.session.SessionManagementFilter.doFilter(SessionManagementFilter.java:103)
        org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:342)
        org.springframework.security.web.authentication.AnonymousAuthenticationFilter.doFilter(AnonymousAuthenticationFilter.java:113)
        org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:342)
        org.springframework.security.web.servletapi.SecurityContextHolderAwareRequestFilter.doFilter(SecurityContextHolderAwareRequestFilter.java:154)
        org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:342)
        org.springframework.security.web.savedrequest.RequestCacheAwareFilter.doFilter(RequestCacheAwareFilter.java:45)
        org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:342)
        org.springframework.security.web.authentication.AbstractAuthenticationProcessingFilter.doFilter(AbstractAuthenticationProcessingFilter.java:199)
        org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:342)
        org.springframework.security.web.authentication.logout.LogoutFilter.doFilter(LogoutFilter.java:110)
        org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:342)
        org.springframework.security.web.context.request.async.WebAsyncManagerIntegrationFilter.doFilterInternal(WebAsyncManagerIntegrationFilter.java:50)
        org.springframework.web.filter.OncePerRequestFilter.doFilter(OncePerRequestFilter.java:108)
        org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:342)
        org.springframework.security.web.session.ConcurrentSessionFilter.doFilter(ConcurrentSessionFilter.java:125)
        org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:342)
        org.springframework.security.web.context.SecurityContextPersistenceFilter.doFilter(SecurityContextPersistenceFilter.java:87)
        org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:342)
        org.springframework.security.web.access.channel.ChannelProcessingFilter.doFilter(ChannelProcessingFilter.java:144)
        org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:342)
        org.springframework.security.web.FilterChainProxy.doFilterInternal(FilterChainProxy.java:192)
        org.springframework.security.web.FilterChainProxy.doFilter(FilterChainProxy.java:160)
        org.springframework.web.filter.DelegatingFilterProxy.invokeDelegate(DelegatingFilterProxy.java:344)
        org.springframework.web.filter.DelegatingFilterProxy.doFilter(DelegatingFilterProxy.java:261)
```

note The full stack trace of the root cause is available in the Apache Tomcat/7.0.53 logs.

Apache Tomcat/7.0.53

The same happens under CRF if you click ‘CREATE CRF’ or “ADD CRF FROM REPOSITORY’ The error occurs also after pressing any other link. By logout and login this is not happening.

2.11.3 Conclusions

The evaluators found the Microbiology Safety Service component functional and usable. No risks have been reported for the component. Error handling under specific conditions needs improvement. The developers have been informed for the results of the evaluation and they will continue to improve their component.

2.12 Policy Administration Point

2.12.1 Validation

Policy Administration Point					
Category ID	**Measurement**	**Description**	**Quality measure elements**	**Measure Type**	**Result**
A.1.1	Functional completeness	The system should enable CRUD operations on the XACML policies	It will be checked that each CRUD operation on the policies work	Functional report	
A.1.2	Functional correctness	Changes should result in semantically correct policies that can be evaluated by the PDP	To measure this we need to compare precision and recall of operations made by the PAP with the ones that are expected	Precision & recall	
A1.3	Functional appropriateness	The PAP should comply to the EURECA legal requirements	The system will be audited by legal experts	Audit outcome	This has not been performed yet
A.4.2	learnability	Administrators should be easily perform CRUD operations on the PAP	During validation, it will be timed how low it takes to learn the system	timing results	
A.5.1	Maturity	The amount of faults that happen in the system during execution should be low	The number of faults while running the system will be recorded, if this number is too high corrective actions will be taken	Number of faults	

A.5.3	Fault tolerance	The system should keep running after malformed (syntactical, semantical) requests enter the system	A stress test will be developed, to check how the system reacts to corrupt and malfuncting requests	Stress test outcome	20 Faulty SOAP messages were sent to the PAP and the response of the PAP were measuered. The system reacted appropriately without downtime.
A.6.1	Confidentiality	Confidentiality of messages from/to the system should be guaranteed	Traffic to the system will be checked. Only HTTPS connections are valid, others are discarded.	HTTPS connections	Different Messages (10 requests) sent to the authorisation service, were sniffed using WireShark. The request content of each request was not readable. Requests sent over HTTP are not executed on the authorisation service
A.6.2	Integrity	The integrity of messages from/to the system should be guaranteed	Messages will be checked if they contain the required signing of body and header fields	Messages	A signed request and a non-signed request were sent to the IDM. Only the message with the signed body and header was accepted by the service. Both requests were explicitly controlled to see if they contained the correct signing configuration.
A.6.3	Non-repudiation	Messages sent from/to the system can be proven to have taken place	The PAP will be connected to a central audit service which keeps track of the senders	audit log	The PAP is integrated with a central audit system which accepts reports of user actions.
A.6.4	Accountability	The sender of requests to the system should be traceable	The PAP will be connected to a central audit service which keeps track of the senders	audit log	The PAP is integrated with a central audit system which accepts reports of user actions.

A.6.5	Authenticity	The identity of each sender can be proved to be the one claimed, for each request to the system	The PAP will be connected to the identity provider, which will authenticate the user	valid credentials	100 requests with both valid and invalid credentials were sent to the PAP. The credentials were forwarded to the connected IDP which was able to correctly authenticate all users with valid credentials and reject the users with invalid credentials.
A.7.3	Analyzability	Failures and deficiencies in the system should be easy diagnosable	It will be checked if exceptions are well logged in the system	Logging	Semantically incorrect requests (100 requests) were sent to the authorisation service, for each incorrect request, an exception was registered in the logging
A.7.5	Testability	A script will test the system frequently to check if everything is ok	automatic tests will be executed to test the system	Response to automatic tests	The PAP has a full suite of test which are automatically executed at predefined intervals.
A.8.3	Replaceability	The PAP should be easily replaceable by other PAP implementations based on XACML 3,0	It will be tested that the PAP is fully compliant with XACML 3,0	Compliant test outcome	100 XACML 3,0 compliant requests of varying content were sent to the PAP and correctly processed.

2.12.2 Evaluation

LUH

The evaluation procedure could not be carried out due to a legal issue, as only Custodix and LUH can access the EURECA policies. Hence, testing the Policy Administration Point is restricted. Since Custodix is the developer of the specific component, evaluation performed only by LUH.

Questionnaires

Evaluator 1

LUH

	FORM A	**Rating (1 low, 5 high)**				
	(Software quality characteristics)	1	2	3	4	5
Functionality	Can software perform the tasks required? i) Create new policies ii) Edit existing policies iii) delete policies iv) display policies					x
	Is the result as expected? (CRUD operations on the policies work as expected)					x
	Can the system interact with a) The eureca policy store?					
	Is the system compliant with XACML v3,0?					
Efficiency						
Compatibility						
Usability	Does the user comprehend how to use the system easily?					x
	Can the user learn to use the system easily?					x
	Can the user use the system without much effort?					x
	Does the interface look good?					x

Reliability	Have most of the faults in the software been eliminated over time?					
	Is the software capable of handling errors?					x
Security	Does the system provide identification access wherever is needed?					x
	Are data accessible only to authorized users?					x
	Can the system trace actions uniquely?					
	Does the system prevent unauthorized access?					x
Maintainability	Can faults be easily diagnosed?					
	Can the software be easily modified?					
	Can the software continue functioning if changes are made?					
	Can the software be tested easily?					
Portability						
Quality in use	How accurate and complete is the software for the intended use?					
	Does the software improve the time or reduce resources for the intended goal?					
	Does the software satisfy the perceived achievements of pragmatic goals?					
	Can the software harm people in the intended contexts of use?					

System Usability Scale (SUS) score: 82.5

2.12.3 Conclusions

According to the evaluator, the component performed excellent in the applicable functional and non-functional characteristics. No risks have been found for the Policy Administration Point. The developers have been informed for the results of the evaluation.

2.13 Security Token Service

2.13.1 Validation

Security Token Service					
Category ID	**Measurement**	**Description**	**Quality Measure Elements**	**Measure Type**	**Result**
A.1.1	Functional completeness	The system should create for each incoming issue token request an token response, including a SAML token	It will be checked that for each issue token request the system generates an token response with SAML token	Number of responses	100 different messages were sent to the STS, with 100 generated responses. Each token contained a SAML token.
A.1.2	Functional correctness	The system should only return a token response with SAML token, if the credentials are correct	To measure this we need to compare precision and recall of decisions made by the STS with the ones that are expected	Precision & recall	100 different messages with both correct and incorrect credentials were sent. The resulting responses were checked for their contents (i.e. Whether or not they contain a SAML token). All requests were found to lead to appropriate responses.
A1.3	Functional appropriateness	The STS should comply to the EURECA legal requirements	The system will be audited by legal experts	Audit outcome	Auditing has not been performed yet
A.3.2	Interoperability	STS clients should be able to send issue token requests to the system and receive the token responses made by the	A web service will be available to accept issue token requests	Web Service successful execution	Multiple services (5 at the moment of evaluation) have been set up to send requests to and accept responses from the STS service successfully.

		system			
A.5.1	Maturity	The amount of faults that happen in the system during execution should be low	The number of faults while running the system will be recorded, if this number is too high corrective actions will be taken	Number of faults	The logs of the STS were evaluated for faults. None of the 113656 different requests on the currently running deployment resulted in a fault.
A.5.2	Availability	The system should have a high availability as the STS is required in most EURECA services	The uptime of the system will be measured	Uptime of the system	The STS has been running for almost a year without unexpected incidents, through extended periods of heavy traffic and times with large bursts of requests. The STS has been found to react in a timely manner.
A.5.3	Fault tolerance	The system should keep running after malformed (syntactical, semantically) requests enter the system	A stress test will be developed, to check how the system reacts corrupt and malfunction requests	Stress test outcome	20 incorrect messages were sent to the STS at different times. The system reacted appropriately without downtime.
A.6.1	Confidentiality	Confidentiality of messages from/to the system should be guaranteed	Traffic to the system will be checked. Only HTTPS connections are valid, others are discarded.	HTTPS connections	Unappropriated connections through HTTP are always rejected.

A.6.2	Integrity	The integrity of messages from/to the system should be guaranteed	Messages will be checked if they contain the required signing of body and header fields	Messages	A signed request and a non-signed request were sent to the STS. Only the message with the signed body and header was accepted by the service. Both requests were explicitly controlled to see if they contained the correct signing configuration
A.6.3	Non-repudiation	Messages send from/to the system can be proven to have taken place	The STS will be connected to a central audit service which keeps track of the senders	audit log	The STS is integrated with a central audit system which accepts reports of user actions.
A.6.4	Accountability	The sender of requests to the system should be traceable	The STS will be connected to a central audit service which keeps track of the senders	audit log	The STS is integrated with a central audit system which accepts reports of user actions.
A.6.5	Authenticity	The identity of each sender can be proved to be the one claimed, for each request to the system	Only users with valid credentials can issue tokens	valid credentials	100 different messages with both correct and incorrect credentials were sent. The resulting responses were checked for their contents (i.e. Whether or not they contain a SAML token). All requests were found to lead to appropriate responses.
A.7.3	Analyzability	Failures and deficiencies in the system should be easy diagnosable	It will be checked if exceptions are well logged in the system	Logging	Errors and exceptions are thoroughly logged by the sts.

A.7.5	Testability	A script will test the system frequently to check if everything is ok	automatic tests will be executed to test the system	Response to automatic tests	The STS has a full suite of test which are automatically executed at predefined intervals.
A.8.3	Replaceability	The STS should be easily replaceable by another STS based on WS-Trust 1,4	It will be tested that the STS is fully compliant with WS-Trust 1,4	Compliant test outcome	Has not yet been tested

2.13.2 Evaluation

Questionnaires

Evaluator 1

FORTH

	FORM A	Rating (1 low, 5 high)				
	(Software quality characteristics)	1	2	3	4	5
Functionality	Can software perform the tasks required? i) issue tokens ii) delegate tokens					x
	Is the result as expected? (When correct credentials are provided, A SAML token is issued, else an exception is thrown)					x
	Is the system compliant with SAML (version 2.0) and WS-Trust 1.4?					
Compatibility						
	Can the system share information/data with other Eureca components? Is the web service for remote invocation up & running? Does it provides results?					x
Reliability	Have most of the faults in the software been eliminated over time?					
	Is the software capable of handling errors?					x
Security	Does the system provide identification access wherever is needed?					
	Can the system trace actions uniquely?					
	Does the system prevent unauthorized access?					x
Maintainability	Can faults be easily diagnosed?					x
	Can the software be easily modified?					
	Can the software continue functioning if changes are made?					
	Can the software be tested easily?					x

Portability	Can the software easily replace other software?					
Quality in use	How accurate and complete is the software for the intended use?					x
	Does the software improve the time or reduce resources for the intended goal?					
	Does the software satisfy the perceived achievements of pragmatic goals?					x
	Can the software harm people in the intended contexts of use?					

System Usability Scale (SUS) score: 85

Evaluator 2

FORTH

	FORM A	**Rating (1 low, 5 high)**				
	(Software quality characteristics)	1	2	3	4	5
Functionality	Can software perform the tasks required? i) issue tokens ii) delegate tokens				x	
	Is the result as expected? (When correct credentials are provided, A SAML token is issued, else an exception is thrown)					x
	Is the system compliant with SAML (version 2.0) and WS-Trust 1.4?					
Compatibility						
	Can the system share information/data with other Eureca components? Is the web service for remote invocation up & running? Does it provides results?					x
Reliability	Have most of the faults in the software been eliminated over time?					
	Is the software capable of handling errors?					x

Security	Does the system provide identification access wherever is needed?					
	Can the system trace actions uniquely?					
	Does the system prevent unauthorized access?					x
Maintainability	Can faults be easily diagnosed?					x
	Can the software be easily modified?					
	Can the software continue functioning if changes are made?				x	
	Can the software be tested easily?					x
Portability	Can the software easily replace other software?					
Quality in use	How accurate and complete is the software for the intended use?					x
	Does the software improve the time or reduce resources for the intended goal?					
	Does the software satisfy the perceived achievements of pragmatic goals?					x
	Can the software harm people in the intended contexts of use?	x				

System Usability Scale (SUS) score: 90

Evaluator 3

Philips

	FORM A	**Rating (1 low, 5 high)**				
	(Software quality characteristics)	1	2	3	4	5
Functionality	Can software perform the tasks required? i) issue tokens ii) delegate tokens			x		
	Is the result as expected? (When correct credentials are provided, A SAML token is issued, else an exception is thrown)					x
	Is the system compliant with SAML (version 2.0) and WS-Trust 1.4?					x

Compatibility						
	Can the system share information/data with other Eureca components? Is the web service for remote invocation up & running? Does it provides results?					x
Reliability	Have most of the faults in the software been eliminated over time?				x	
	Is the software capable of handling errors?					x
Security	Does the system provide identification access wherever is needed?					x
	Can the system trace actions uniquely?				x	
	Does the system prevent unauthorized access?					x
Maintainability	Can faults be easily diagnosed?				x	
	Can the software be easily modified?			?		
	Can the software continue functioning if changes are made?			?		
	Can the software be tested easily?			?		
Portability	Can the software easily replace other software?			?		
Quality in use	How accurate and complete is the software for the intended use?				x	
	Does the software improve the time or reduce resources for the intended goal?			?		
	Does the software satisfy the perceived achievements of pragmatic goals?					x
	Can the software harm people in the intended contexts of use?	x				

System Usability Scale (SUS) score: 70

Evaluation Comments

Evaluator 1

"This is a web service that generates a SAML token. In general the web service runs smoothly without problems, in a reasonable time. It was tested easily since the necessary initial configuration was explained in the evaluation scenario."

2.13.3 Conclusions

The Security Token Service performed well in the evaluation process with high System Usability Scale scores. No potential risks have been found. The developers have been informed for the results of the evaluation and they will continue to improve their component.

2.14 Auto Complete Service

2.14.1 Validation

Auto Complete Service					
Category ID	**Measurement**	**Description**	**Quality measure elements**	**Measure Type**	**Results**
A.1.1	Functional completeness	The service is able to retrieve information about Core Dataset	To measure this we need to search if data retrieved is the correct	The number of matchings'	Given a free text and a number of results, it is possible to obtain the maximum results given as a parameter
A.2.1	time behavior	The system should respond in a timely manner	System 's response time will be measured	Response time	more than 200ms
A.2.2	resource utilization	The system should not be resource intensive	The CPU and memory utilization will be measured in the pc where the matcher is executed	CPU & memory utilization	Between 1-4% of CPU and memory
A.3.2	Interoperability	The results of the system should be provided to other EURECA components. Auto complete service retrieve information from CDM and Core Dataset	A web service should be available to provide the query results to external Eureca components	Web Service successful execution	Autocomplete service is used on UI applications for obtaining the code for generating queries on Query normalization service
A.5.2	Availability	The system should be up & running almost always	The uptime of the system will be measured	Uptime of the system	The web service is always running where server is running. Last time it was restarted on 5 of June

A.7.1	Modularity	Internal modules of the service could be changed by others, e.g. Sesame, clients	To measure this we need to search if data retrieved is the correct	The number of matchings'	It is possible to change the version of the different ontologies used and semantic repository easily
A.7.5	Testability	A script will test the system frequently to check if everything is ok	automatic tests will be executed to test the system	Response to automatic tests	A battery of 50 concepts are tested for obtaining the corresponding code in almost 10 seconds
A.8.2	Installability	The system should be easily installed by an IT-expert	The time to install the system in another machine will be tested	Time to install the system	1 hour. Necessary to deploy Core Dataset Service first
Comments	Autocomplete is the service responsible for obtaining the core dataset code given a free text. It is necessary to decrease the timing				

2.14.2 Evaluation

Questionnaires

Evaluator 1

FORTH

	FORM A	Rating (1 low, 5 high)				
	(Software quality characteristics)	1	2	3	4	5
Functionality	Can software perform the tasks required? Retrieve concepts from Core Dataset? Retrieve metadata of concepts stored on CDM?					x
	Is the result as expected?					x
	Can the system interact with another system? Core Dataset Service and CDM?					x
	Is the system compliant with standards?					
Efficiency	How quickly does the system respond?					x
	Does the system utilize resources efficiently?					x
Compatibility	Can the system share resources without loss of its functionality?					x
	Can the system share information/data with other Eureca components? Is the web service for remote invocation up & running? Does it provides results?					x
Usability	Does the user comprehend how to use the system easily?				x	
	Can the user learn to use the system easily?					x
	Can the user use the system without much effort?					x
Reliability	Have most of the faults in the software been eliminated over time?					
	Is the software capable of handling errors?					x
	Can the software resume working & restore lost data after failure?					

Security	Does the system provide identification access wherever is needed? Security access? SSL?					
	Are data accessible only to authorized users?					
	Can the system trace actions uniquely?					
	Does the system prevent unauthorized access?					
Maintainability	Can faults be easily diagnosed?					x
	Can the software be easily modified?					
	Can the software continue functioning if changes are made?					
	Can the software be tested easily?					x
Portability	Can the software be moved to other environments?					x
	Can the software be installed easily?					x
	Does the software comply with portability standards?					x
	Can the software easily replace other software?				x	
Quality in use	How accurate and complete is the software for the intended use?					x
	Does the software improve the time or reduce resources for the intended goal?					
	Does the software satisfy the perceived achievements of pragmatic goals?					x
	Can the software harm people in the intended contexts of use?					

System Usability Scale (SUS) score: 85

Evaluator 2

Philips

	FORM A	**Rating (1 low, 5 high)**				
	(Software quality characteristics)	1	2	3	4	5
Functionality	Can software perform the tasks required? Retrieve concepts from Core Dataset? Retrieve metadata of concepts stored on CDM?					x
	Is the result as expected?					x
	Can the system interact with another system? Core Dataset Service and CDM?					x

	Is the system compliant with standards?				?	
Efficiency	How quickly does the system respond?				x	
	Does the system utilize resources efficiently?				x	
Compatibility	Can the system share resources without loss of its functionality?			x		
	Can the system share information/data with other Eureca components? Is the web service for remote invocation up & running? Does it provides results?				x	
Usability	Does the user comprehend how to use the system easily?				x	
	Can the user learn to use the system easily?			x		
	Can the user use the system without much effort?				x	
Reliability	Have most of the faults in the software been eliminated over time?					x
	Is the software capable of handling errors?			x		
	Can the software resume working & restore lost data after failure?			?		
Security	Does the system provide identification access wherever is needed? Security access? SSL?				x	
	Are data accessible only to authorized users?				x	
	Can the system trace actions uniquely?			?		
	Does the system prevent unauthorized access?				x	
Maintainability	Can faults be easily diagnosed?				x	
	Can the software be easily modified?			x		
	Can the software continue functioning if changes are made?				x	
	Can the software be tested easily?				x	

Portability	Can the software be moved to other environments?				?	
	Can the software be installed easily?				x	
	Does the software comply with portability standards?				?	
	Can the software easily replace other software?				?	
Quality in use	How accurate and complete is the software for the intended use?					x
	Does the software improve the time or reduce resources for the intended goal?				x	
	Does the software satisfy the perceived achievements of pragmatic goals?				x	
	Can the software harm people in the intended contexts of use?	x				

System Usability Scale (SUS) score: 72.5

Evaluator 3

Custodix

	FORM A	**Rating (1 low, 5 high)**				
	(Software quality characteristics)	1	2	3	4	5
Functionality	Can software perform the tasks required? Retrieve concepts from Core Dataset? Retrieve metadata of concepts stored on CDM?					x
	Is the result as expected?					x
	Can the system interact with another system? Core Dataset Service and CDM?				x	
	Is the system compliant with standards?				x	
Efficiency	How quickly does the system respond?				x	
	Does the system utilize resources efficiently?				x	
Compatibility	Can the system share resources without loss of its functionality?			x		
	Can the system share information/data with other Eureca components? Is the web service for remote invocation up & running? Does it provides results?				x	

Usability	Does the user comprehend how to use the system easily?				x	
	Can the user learn to use the system easily?			x		
	Can the user use the system without much effort?			x		
Reliability	Have most of the faults in the software been eliminated over time?				x	
	Is the software capable of handling errors?		x			
	Can the software resume working & restore lost data after failure?				x	
Security	Does the system provide identification access wherever is needed? Security access? SSL?			x		
	Are data accessible only to authorized users?			x		
	Can the system trace actions uniquely?			x		
	Does the system prevent unauthorized access?			x		
Maintainability	Can faults be easily diagnosed?				x	
	Can the software be easily modified?			x		
	Can the software continue functioning if changes are made?			x		
	Can the software be tested easily?				x	
Portability	Can the software be moved to other environments?			x		
	Can the software be installed easily?			x		
	Does the software comply with portability standards?			x		
	Can the software easily replace other software?			x		
Quality in use	How accurate and complete is the software for the intended use?				x	
	Does the software improve the time or reduce resources for the intended goal?				x	

	Does the software satisfy the perceived achievements of pragmatic goals?				x	
	Can the software harm people in the intended contexts of use?		x			

System Usability Scale (SUS) score: 65

Evaluation Comments

Evaluator 1

"This is a web service for retrieving information related to Core Dataset concepts and the links with the CDM. In general the web service runs smoothly without problems, in a reasonable time. Considering reliability, I cannot tell if the most of the faults in the software have been eliminated over time as I have not seen previous versions of the service."

Evaluator 3

"The auto complete service provides a web service that enable user to receive medical concepts for a given search query. The web service interface definition is straightforward. The execution time of the web service is quick and responsive. The functionality works as expected, but is a bit complex for an unexperienced user. Exception handling should be improved, as for the moment no SOAP Faults are thrown (error message is included in the return message). This validation was performed without using the security proxy, so no evaluation was made for security."

2.14.3 Conclusions

The Auto Complete Service deployed for the evaluation purpose is a test version based on the stable services deployed on the development and stage server. An update of the component and more details will be reported in the Deliverable 4.4 "Initial prototype of the semantic interoperability framework.

2.15 API components

The evaluation by the users of the Natural Language Processing (NLP) components, i.e. the concept identifier and the relation extractor, is postponed to the next WP8 deliverables due to two issues: (a) the EURECA partners could only download the concept identifier and the Xerox relation extraction module in mid-January 2014, because of a delay in the finalization of the Consortium Agreement; and (b) there was a delay in the availability of patient free-text datasets, especially in the language handled by the initial NLP prototypes, English.

2.15.1 Functional suitability

Regarding functional suitability, in information extraction from free text, functional completeness and correctness are mainly measured by recall and precision respectively. Recall measures the proportion of identified items among the total number of reference items, i.e. the whole set of manually annotated elements in the reference input texts. For example, in the case of the concept identifier, recall would be the percentage of identified UMLS concept occurrences, among all the existing UMLS concept occurrences in the input texts. As for precision, it measures the percentage of correctly identified items among all extracted items. An identified item is considered to be correct if it belongs to the set of manually annotated elements in the reference text.
For evaluation of functional suitability, two possible options will be considered depending on the availability of reference data: direct evaluation and indirect evaluation.

2.15.2 Direct evaluation

Direct evaluation consists in measuring directly the completeness and correctness of the output of the NLP components and requires that reference test data, i.e. with manually checked ground truth annotations, exist or be developed for the type of annotations produced by each NLP component. In the case of the concept identifier, reference data would consist in texts where all occurrences of UMLS concepts are manually annotated with UMLS concept identifiers. In the case of relation extraction, direct evaluation requires reference data for at least one of the use cases where the relation extractor is instantiated. For example, in the trial recruitment use case, direct evaluation of the relation extraction component would require free text eligibility criteria sentences manually annotated with a formally structured representation of the criteria they express, following the proposed triple-based semantic representation of eligibility criteria (see D3.2).

2.15.3 Indirect evaluation

If reference test data for the direct evaluation of the NLP components is not available, and is costly to develop, then indirect evaluation will be conducted. Indirect evaluation consists in estimating the quality and usefulness of an NLP component by evaluating its impact on the quality of a service, tool that uses it. This can be done by evaluating the completeness and correctness of the service or tool without the use of the annotations produced by the NLP component (baseline configuration) and comparing the results with those obtained on the same data when the service/tool is run with the NLP component (NLP-based configuration).

2.15.4 Performance efficiency of the concept identifier

We ran a preliminary evaluation of the computational efficiency of the initial prototype of the Concept Identifier (see D3.1) since such evaluation does not require manually annotated test data or full integration of this component into a specific use case. Experiments have been run on a desktop with an i7 CPU, 3.07GHz, 16GB of RAM and a 256GB HDD. The dataset consisted in 100 of free text files, containing eligibility sections from randomly selected clinical trial documents for breast cancer. The dataset

size was 276K bytes; about 33K tokens (words). The measured time includes reading the files from disk and saving the output of the concept identifier to disk.

Category ID	Measurement	Measure Type	Result
A.2.1	Time Behaviour	Response time (total in ms.)	22903
		Speed (in tokens/second)	1440
		Speed (in Kbytes/second)	12
A.2.2	Resource utilization	Memory utilization (in MB)	535

Table: Preliminary results in performance efficiency of the initial prototype for concept identification

Note that a new version of the concept identifier, with new features like the possibility of integrating user-defined terminologies, is currently being finalized and will be delivered soon to the partners. We expect that the performance efficiency of the new version will decrease since the integration of the new features and functionalities requires additional computations.

3 CONCLUSIONS

This deliverable reports the results of the evaluation and validation of the software components that are finalized so far for the EURECA project. The evaluation and validation was performed for 8 web services, 1 third party tool and 4 web applications. There are 1 more web application, the SAE prediction tool, and 2 API NLP components finalized so far in EURECA project but these could not be evaluated and validated in this stage due to restrictions reported also in this deliverable.
The SAE prediction tool cannot be evaluated due to lack of integration with the EURECA framework, and availability of data for the evaluation of the use cases is an issue. Such integration will be obtained through the EURECA data mining framework reported in Deliverable D5.3, which is at this time not yet operational. Integration with the data mining framework should also resolve issues regarding the duration of the running time of the algorithms. Furthermore, data required for evaluation of multiple use cases is at this time not yet fully available. As the service should provide prediction algorithms suitable for prediction of different SAEs, such data should be available in order to evaluate the full potential of the service.
The evaluation by the users of the NLP components, i.e. the concept identifier and the relation extractor, is postponed to the next WP8 deliverables due two issues:

- the EURECA partners could download the concept identifier and the Xerox relation extraction module only in mid-January 2014, because of a delay in the finalization of the Consortium Agreement
- there is a delay in the availability of patient free-text datasets, especially in the language handled by the initial NLP prototypes, English

Therefore, the instantiation, integration and use of the NLP prototypes in EURECA use cases and services were delayed accordingly. Nevertheless, discussions have already started on how the evaluation of the NLP components will be conducted to measure functional suitability. In addition, we performed a preliminary evaluation of the performance efficiency of the first version of the concept identifier as this does not require its full integration in use cases.
Table 1 gives an overview of the evaluation results. Each component has been evaluated by three EURECA partners who did not participate in the implementation phase. The evaluation forms of EURECA are list of questions where the evaluator answered with a degree of satisfaction with Likert scale (indicates the degree of agreement or disagreement with the statement on a 5 point scale). Each question is associated to a functional or non-functional sub-characteristic according to the ISO/IEC 25000 series (first column of the table). The acceptable level for this phase has been set to 3. Table 1 reports the mean values for each sub-characteristic from each evaluator. Sub-characteristics which have been evaluated with value less than 3 are marked using red colour and these are considered to be the potential risks for the corresponding component. For the sub-characteristics in risk, the developers have been informed and reported a risk management procedure (section 2) to assure the high quality of the EURECA components. Sub-characteristics marked using orange colour are considered to be in acceptance level but the developers have been informed in order to explore and identify if further improvements can be implemented. Green is used for sub-characteristics with good rating level.
Exception in this procedure is the policy administration. The evaluation of the policy administration point component could not be carried out due to a legal issue, as only Custodix and LUH can access the EURECA policies. Since Custodix is the developer of the specific component, evaluation performed only by LUH.

Table 1: Overview of the evaluation results

Component	Security Token Service			Protocol Feasibility		
Evaluator	FORTH	FORTH	Philips	FORTH	MASTRO	IJB
Functionality	5	4.5	4.3	5	4	4.7
Efficiency	-	-	-	4	3.5	2
Compatibility	5	5	5	5	4	
Usability	-	-	-	4	3.8	3.5
Reliability	5	5	4.5			
Security	5	5	4.7			
Maintainability	5	4.7	4	4	4.3	5
Portability				4	5	5
Quality in use	5	5	4.7	4.3	5	4.8
SUS Score	85	90	70	80	77.5	80

Component	Identity Manager			Policy Administration Point	Trial Management Service		
Evaluator	FORTH	LUH	USAAR	LUH	FORTH	UPM	Custodix
Functionality	3.5	4	5	5	5	3.8	4.5
Efficiency	-	-	-	-	5	4	4
Compatibility				-	5	5	4
Usability	-	-	-	5	-	-	-
Reliability	3	4.5	3	5	4.7	1	4
Security	4.7	5	5	5	5	5	3.8
Maintainability	5	5	5	-	4	1.5	3.8
Portability				-	-	-	-
Quality in use	4	5	4	-	4.5	3.5	4
SUS Score	95	90	100	82.5	72.5	57.5	62.5

Component	Query Engine			Microbiology Module			
Evaluator	FORTH	XEROX	UPM	FORTH	IJB	GBG	USAAR
Functionality	2	5	5	5	2.5	4	4.6
Efficiency	3	4		4.7	4.3	3	3.5
Compatibility	5	5	5	5	3	3	5
Usability	-	-	-	4.7	3.3	5	4.5
Reliability	5	5	3	4	-	-	3
Security	3.3	3.5	5	5	5	-	4
Maintainability		3.5	3	4	4	4	5
Portability	-	-	-				
Quality in use	3	5	3.8	5	3	-	4
SUS Score	42.5	85	57.5	95	75	92.5	80

Component	Authorisation Service			Data Push Service (ETL)			Query Normalization Service		
Evaluator	FORTH	FhG IAIS	FhG IBMT	FORTH	FhG IAIS	FhG IBMT	FORTH	Custodix	FhG IBMT

Functionality	3	4.7	5	5	3	5	5	4.5	4
Efficiency	-	-	-	5	5	4	5	3.5	5
Compatibility	3	5	5	5	3		5	3.5	
Usability	-	-	-	3.8	2.8	3	3.8	3.8	2
Reliability	3	3.5	5	4	1	2	4	3.3	1
Security	5	5	5		1			3	5
Maintainability	5	3.5	4	4	3	1	4	3.3	4
Portability	3.5	3		4.7	3		4.8	3.5	2.3
Quality in use	3	4.3	5	5	3.5	3	5	4	3.3
SUS Score	42.5	52.5	85	85	55	35	85	70	47.5

Component	Query Execution Service			Core Dataset Service			Auto Complete Service		
Evaluator	FORTH	UOXF	Philips	FORTH	Custodix	FhG IAIS	FORTH	Philips	Custodix
Functionality	5	4.5	3.8	5	4.5	3	5	5	4.5
Efficiency	5	5	4	5	3.5	4	5	4	4
Compatibility	5	4.5	4	5	4	3	5	3.5	3.5
Usability	3.5	4	2.8	3.8	3	2.8	4.7	3.7	3.3
Reliability	4	3.3	2	4	3.3	3.5	5	4	3.3
Security		1	3		3	1		4	3
Maintainability	4.7	4.7	2.5	5	3.5	3.5	5	3.8	3.5
Portability	4.8	4.8	3	4.8	3.3	3	4.8	4	3
Quality in use	5	3.8	3	5	3.8	3.3	5	4.5	4
SUS Score	85	67.5	47.5	85	57.5	72.5	85	72.5	65

For the most of the components the evaluation and validation result showed high quality software components. In some cases however, some deficiencies where revealed, which however have low risk for the project. Nevertheless, a plan is presented for their management.
Since other EURECA components are currently under heavy development, the evaluation and the validation of all EURECA components will be finalized with D8.5.